CompatibleOne Resource Description System (CORDS)

Technical Reference V2.12

Iain James Marshall (Prologue)
Jean-Pierre Laisné (Bull)

22/11/2012

This document provides the complete reference guide to the CORDS specifications.

Introduction

This document provides an exhaustive reference for the CompatibleOne Resource Description System, CORDS, describing the component categories, attributes and actions, the relationships between the different categories and their intended use.

Disclaimer

This is a none-normative work and information provided here, concerning categories and their attributes, is subject to change as required to satisfy the evolving needs of the ACCORDS (Advanced Capabilities for CORDS) platform.

Acknowledgements

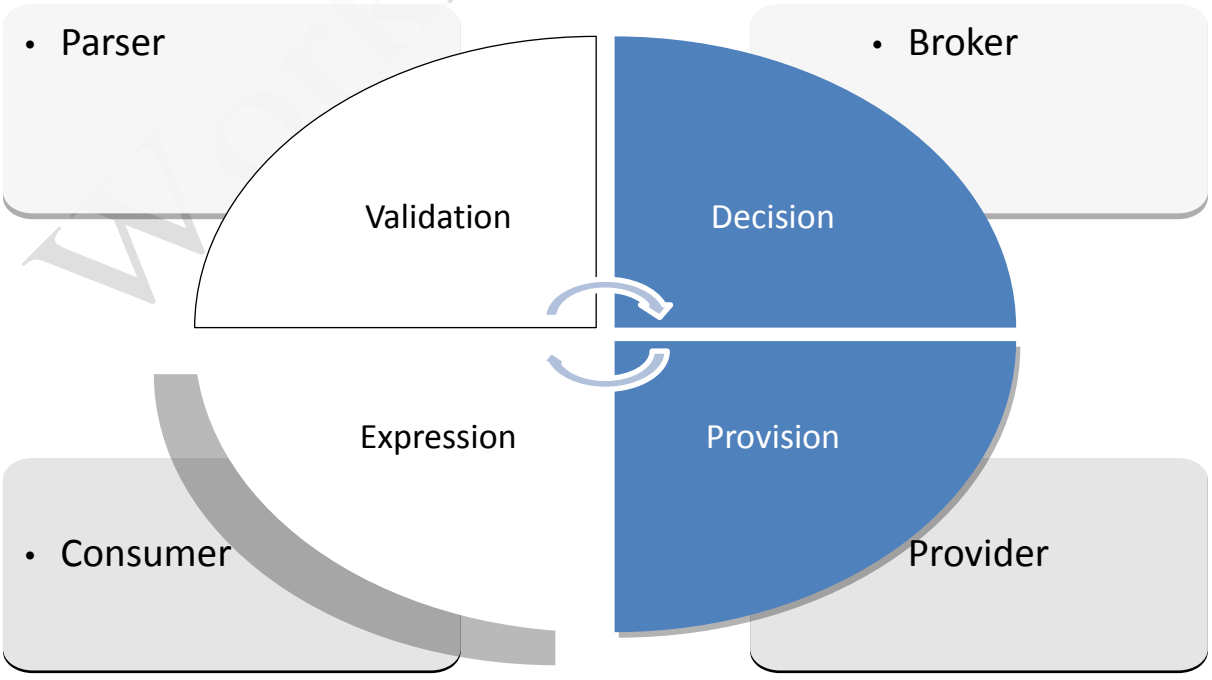
The authors would like to thank the following for their participation in the CompatibleOne project:

Marouene Mechtri for work on CONETS,
Sami Yangi, Mohamed Mohamed, and Samir Tata for their work on the Generic PAAS,
Salma Rebai for work on COPS.

Overview

The CORDS specification is an XML application and as such the machine readable form of the schema of this document type is available at the following address [1]. It should be used for the validation of manifest documents by all document production and processing tools. The schema document also describes the semantics and behaviour of the parsing operation required to be performed by the ACCORDS parser. The type definitions on which this schema is based is available at the following address [2] and is also used by the category instance bulk import schema definition [3] and the schema of the customer service agreement document type [4]. Finally the schema for provider service level agreements can be found here [14].

The CORDS model provides a framework to be used for the description of complex cloud provisioning configurations and systems and its operational cycle can be seen below:



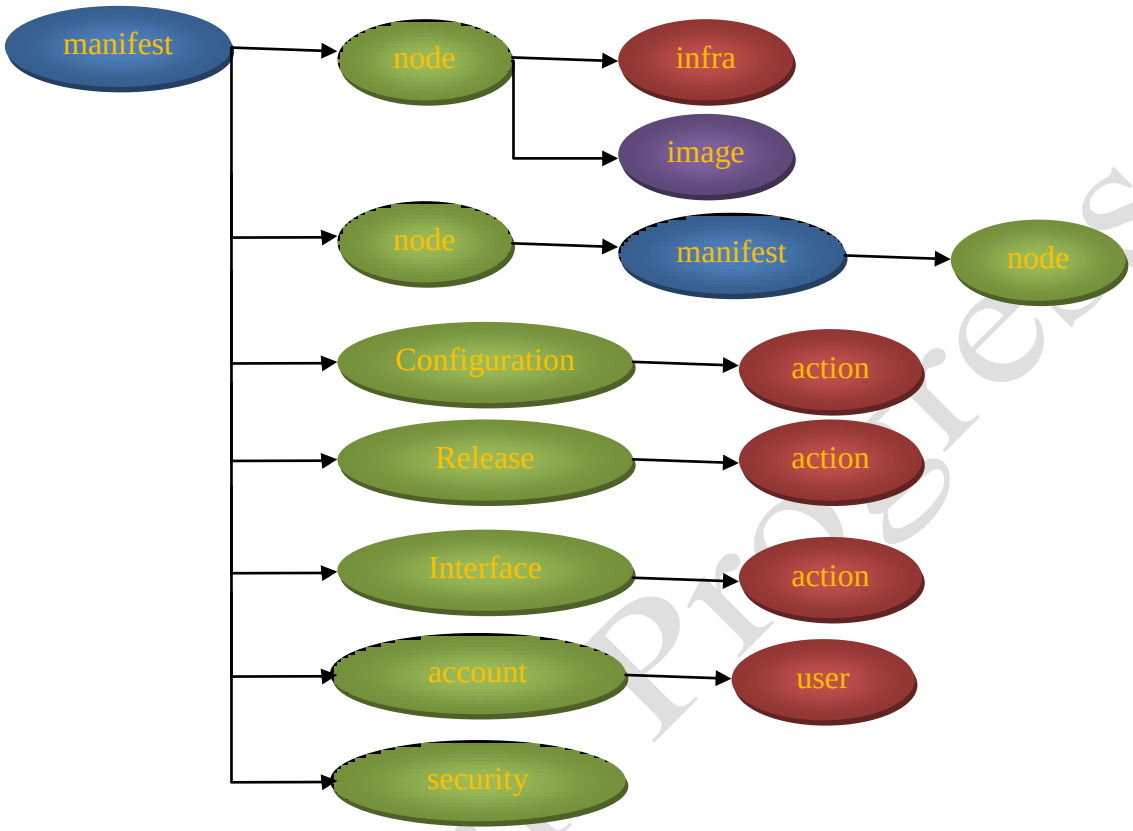
From the preceding diagram it should be understood that the consumer will express their provisioning requirements by creating a CORDS manifest document. This document will be submitted for validation by the parser leading to the creation of a provisioning plan. This plan will provide the basis for the decisions taken by the Broker coordinating the provision of resources negotiated with multiple providers for delivery of service to the user having initiated the process. These steps are not necessarily contiguous, must be performed in the indicated sequence and may be iterated as required.

Categories

All elements of CORDS are represented by an OCCI category. All categories are directly related to the standard OCCI core type and consequently inherit the attributes from this base type. These base attributes are specified, for clarity, in each attribute list of each category definition. These categories are used by the parser, broker and other components of the ACCORDS platform not only for the description and management of provisioning of service instances but also for the dynamics of the operation of the platform itself. The conversion of XML category descriptions to their OCCI category instance counterparts is performed by the ACCORDS Parser tool. For precise information concerning the operation of the ACCORDS parser please consult the corresponding document [5]. The process referred to as brokering involves the creation of a service control graph from a manifest description resulting from the Parser operation. This service control graph will be used for the management of the required provisioning of resources and through which action invocation will be performed.

Categories of the Manifest

This section of this document presents the collection of categories that may be used in an XML CORDS document for the description of a manifest in terms of provisioning and its configuration. This list of categories describes the categories that are defined in the CORDS MANIFEST XSD and their relationships can be seen in the following diagram.



Manifest

This category corresponds to the outermost element of the CORDS manifest document structure. It is used to define a particular cloud provisioning configuration comprising nodes, configuration actions, accounting and security information. The manifest is the basic work unit of the ACCORDS platform from which a provisioning plan is produced by the operation of the ACCORDS parser. The resulting plan provides the description of the system for the brokering and provisioning operations of the ACCORDS Broker.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the name of the manifest. This value will be used to refer to the manifest in complex node type attribute definitions and must have a globally unique value.

- Plan

This attribute, when present, provides the fully qualified universal unique identifier of the provisioning plan category instance that is currently to be used for the brokering and provisioning of service instances of this manifest.

- Nodes

This attribute specifies the count of nodes defined for the manifest description.

- Zone

This attribute specifies the default geographical zone value that is to be applied when processing the manifest constructions.

Elements

- Description

This element provides a textual informative description of the manifest and the nature of the configuration for which it defines a solution.

- Configuration

This element provides the configuration actions which after processing will produce the contractual configuration instructions.

- Release

This element provides the actions which are to be performed prior to the release of resources and compliments the contractual configuration instructions.

- Interface

This element provides the interface methods or actions that may be invoked for instances of the manifest.

- Account

When present, this element provides the account information, price information, invoicing details and the list of valid users that are to be granted access to resources provisioned through the manifest.

- Security

This element will describe the details and the level of security to be engaged when provisioning and operating service instances of the particular manifest.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI manifest category instance identified by the attribute values provided in the OCCI message. The DELETE operation will be propagated through to each of the node category instances linked to the particular manifest instance prior to the removal of the link. When the deletion of all linked nodes has been completed the manifest instance will be removed. In addition, since a provisioning plan instance that was created as a result of successfully parsing a manifest, depends entirely on the manifest for the description of the provisioning requirements, the DELETE operation must also remove the provisioning plan referenced by the “plan” attribute of the parent manifest instance.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

- Node

Instances of the manifest category expect a collection of links to be defined providing an ordered list of nodes describing the precise details of each of the individual manifest resource components.

Node

Instances of this category are used to represent the fundamental work units of processing contributing to the overall behaviour of a particular manifest described provisioned system. Nodes have been referred to as appliances, or cores, and may be used to represent both physical and virtual resources and systems of any kind. The first version of CORDS made provision for the description of simple nodes, described solely in terms of their infrastructure requirements (hardware) and image requirements (software). The second version of CORDS allows nodes to be described in terms of their complex, manifest defined, type in which case the infrastructure and image information will be omitted.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the text value of the name of the node. This name will be used to represent the node in configuration action expressions and must be unique within a particular manifest and may be unique within a particular ACCORDS provisioning platform.

- Provider

This attribute value may define a user preference for the provider platform type to be used for the provision of the resources defined by the node. If this value is omitted then the platform operator's default value will be used instead.

- Profile

Used in conjunction with the preceding provider attribute value, this attribute may specify the subscription account information to be used for provision of the resources defined by the node. When the platform operator's default provider is used, and when this value is omitted, then the corresponding operator's subscription account will also be used.

- Type

This attribute determines the nature or type of the node. When omitted, or set to "simple", it indicates that the node is to be described in terms of its infrastructure and image elements. Otherwise the value provided by this attribute is expected to resolve to the valid name of a valid manifest that has been processed to produce a corresponding provisioning plan. The manifest consequently describes the structure and behaviour of the node.

- Category

This attribute defines the name of the category of the preceding type name. By default this will be the "manifest" category in which case the value of the "type" attribute represents the name of a manifest. This value may be set to "PAAS_manifest" to indicate that the value of the type attribute is the name of a PAAS manifest.

- Access

This attribute, added in version two of CORDS, indicates whether the node is to be made accessible from outside of the service instance in which it has been instanced as a contract. Public nodes will need Public IP addresses to be attributed such that they will be effectively accessible to clients outside of their encapsulating service. This is also necessary for the definition of entry points for use by subsequent client nodes in derived manifest referencing through complex nodes. The value of this attribute influences the nature of the instance of contracts of both simple and complex nodes when declared of common scope.

- Scope

This attribute, introduced in version two of CORDS, allows the instantiation nature of a node contract to be influenced. The default value of "normal" indicates that the resulting node is fully encapsulated within the parent service and will be provisioned for each and every service instance of the particular manifest type. The alternative value of "common" indicates that the node instanced as a contract is to be provisioned as a unique contract and that this contract will be accessed for use by all service instances of the same manifest type, in the case of private access, and from any service instance of any manifest defining the public common node in the case of public access type.

- Common

This element, in the case of a node of "common" scope, will be used to store, and consequently provide, the universal unique identifier of the contract category instance or the service category instance that has been provisioned to satisfy the needs of private common

simple or complex nodes. This element is not required and its behaviour is not defined for nodes of “normal” scope.

Elements

- Infrastructure

This element, in the case of a “simple” typed node, will provide the universal unique identifier of the infrastructure category instance that describes the infrastructure (hardware) requirements of the node. This element is not required and its behaviour is not defined for complex manifest type defined nodes.

- Image

This element, in the case of a “simple” typed node, will provide the universal unique identifier of the image category instance that describes the image (software) requirements of the node. This element is not required and its behaviour is not defined for complex manifest type defined nodes.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI node category instance identified by the attribute values provided in the OCCI message. Since no link types are defined for this category no further processing is required.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Infrastructure

Instances of this category are referenced though universal unique identifier elements of the node category for simple node types to provide the infrastructure (hardware) description of the associated node. An infrastructure category instance is required to be provided, either explicitly in the node description or implicitly by a predefined named node, for all “simple” type nodes.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute may be used to provide a text name for the infrastructure description.

Elements

- Compute

This attribute provides the fully qualified universal unique identifier of the compute category instance providing the details of the machine or virtual machine in terms of architecture, memory size, processors and speed.

- Network

This attribute provides the fully qualified universal unique identifier of the network category instance providing the details of the network type to which the machine is to be connected.

- Storage

This attribute provides the fully qualified universal unique identifier of the storage category instance describing the storage requirements of the infrastructure component.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Compute

Instances of this category are referenced by the universal unique identifier element of infrastructure category instances. The information provided here allows the nature of the calculation hardware to be determined. A compute category instance must have been defined and attributed to an infrastructure category instance otherwise the infrastructure will not be valid.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides an optional name for identification of the particular collection of compute information.

- Hostname

This attribute provides the value of an eventual host name for the computational unit.

- Architecture

This attribute defines the nature of the computational hardware described by the compute category instance. The complete collection of values that are to be permitted here are still to be determined but the standard x86, x286, x386, x486, x586, INTEL values are amongst the types currently expected by most platform providers though an area which will be extended to cover GPU's and other HPC types of hardware.

- Speed

This attribute defines the processor speed in Giga Hertz such as 0.5G 1G 2G 3G 4G 8G etc.

- Cores

This attribute indicates the number of processors or processor cores required by the computational unit.

- Memory

This attribute defines the required memory size expected for use by the computational unit and is to be expressed in Kilobytes (K), Megabytes (M), Gigabytes (G) or Terabytes (T) such as 0.64G 1G, 2G, 3G,4G,8G,16G,32G etc.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Storage

Instances of this category are referenced by the universal unique identifier element of infrastructure category instances. The information provided here allows the nature and quantity of the physical disk space to be determined. A storage category instance must have been defined and attributed to an infrastructure category instance otherwise the infrastructure will not be valid.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides a name for the particular storage category instance.

- Size

This attribute defines the size of the storage unit and is to be expressed in Kilobytes (K), Megabytes (M), Gigabytes (G) or Terabytes (T). Example of legal values are 512M, 0.5G 1G 160G 640G, 0.1T 100T etc.

- Type

This attribute specifies the nature of the storage device and may define eventual access attributes such as read/write, USB IDE SATA etc.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Network

Instances of this category are referenced by the universal unique identifier element of infrastructure category instances. The information provided here allows the nature of the network interaction required for the particular infrastructure description to be determined. A network category instance must have been defined and attributed to an infrastructure category instance otherwise the infrastructure will not be valid.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides a name for the particular network description.

- Vlan

This attribute indicates that the network resource is a virtual local area network resource.

- Label

This attribute provides the value of the label to be used for the network resource.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

The network category expects link list items to be of the port category and providing the description of individual service ports of the required application firewall.

Port

Instances of this category are appended to both the network and package category instances and allow the description of individual service ports required by the applications of a package and by a network interface of the infrastructure. The collection of information that they provide will be used conjointly to configure the application firewall of the provisioned resources and an eventual gateway used for inter cloud communication.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides a name for the particular network description.

- Description

This attribute provides a human readable textual description of the particular service port and its intended usage for documentary purposes.

- Number

This attribute provides the numeric, integer, value of the service port in the range of 1 to 64K.

- Direction

This attribute describes the application or usage of the service port and may be one of the following

  - "NONE" no traffic will be allowed to pass and the defined service port is disabled.
  - "IN" only inbound traffic will be allowed to pass.
  - "OUT" only outbound traffic will be allowed to pass.
  - "INOUT" both inbound and outbound traffic will be permitted.

- Protocol

This attribute provides the protocol type name associated with the port and may be either "TCP" or "UDP".

- Range

This attribute provides the range of IP addresses with which the firewall record is associated to allow specific access.

- Target

This attribute provides an eventual target service port number when port redirection is to be performed in network address translation type modes of operation.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Image

Instances of this category are referenced by the universal unique identifier element of node category instances. An image category instance is required to be provided, either explicitly in the node description or implicitly by a predefined named node, for all “simple” type nodes. The information provided here allows the complete application software image to be described in an abstract manner similar to that which would have been performed by hand for the installation of the application on a raw machine. The information defined for the image category, in terms of its system and application packages, should also be of a suitable format for use as input to third party automated image preparation procedures and Dev-Ops configuration tools such as Juju, Puppet, Chef, CfEngine etc.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute allows a name to be defined for the particular image description instance.

- Agent

The value of this attribute defines the nature of the application agent embedded in the virtual machine image. The default value, provided by the “cordstypes.xsd” schema is “cosacs”. This value indicates that a standard COSACS agent, respecting Accords Platform security constraints, is available for image preparation and post configuration purposes. Alternative values of “http:cosacs” and “https:cosacs” may be specified for images which carry either the insecure or secure versions of COSACS. The value of “none” may be specified to indicate that the COSACS post configuration agent is not present or is not to be use for the particular image.

- Price

This attribute provides the fully qualified unique universal identifier of the price category instance defined for the particular image category instance. When a value has been defined for this attribute it will provide a global price that will take precedence over eventual discrete price information of component software packages and base operating systems.

- Packages

This attribute provides the count of package categories instances linked to the image category instance.

Elements

- System

This element provides the description of the base operating system that is required to be installed prior to the collection of application software products, packages and configuration patches.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI image category instance identified by the attribute values provided in the OCCI message. The removal of an application image category instance must also ensure that all unnamed package instance on the package linked list of the image instance are also removed.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

- Package

Package category instances are appended to the list of links for the image category instance. These packages, appended in sequential order provide the layers of software to be installed over the base operating system and may provide software application programs, database tables and records, configuration files for the preparation of the software image in order that it will be able to perform the operational computing task for which it is intended.

System

Instances of this category are referenced from the fully qualified universal unique identifier of image category instances and describe the underlying base operating required for the particular application image. A system category instance must have been defined and attributed to an image category instance otherwise the image will not be valid.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the name of the particular operating system type.

- Version

This attribute provides the version of the system.

- Distribution

This attribute, when appropriate, defines a particular distribution of the named and versioned system.

- Constraint

This attribute provides an eventual list of constraints which must be satisfied for correct use of the operating system.

- Requirements

This attribute provides an eventual list of requirements that must be satisfied for correct use of the operating system.

- Price

This attribute provides an eventual fully qualified universal unique identifier of the price category instance that is applicable for use of images based on the corresponding operating system. This value will be used, along with eventual price information of package components of the parent image unless a global price has been defined for the image instance.

- License

This attribute provides an eventual description of licensing conditions that are to be satisfied for use of images based on the corresponding operating system.

- Description

This attribute provides a full, human readable, textual description of the operating system.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Package

Package category instances are similar to system category instances, providing a complete application software configuration in a standard installation tool compatible format. Package category instances are appended to the list of links of a parent image category instance.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the name of the particular application package.

- Version

This attribute provides the version of the package.

- Distribution

This attribute, when appropriate, defines a particular distribution of the named and versioned package.

- Mirrors

This attribute provides a comma separated list of mirror servers from which the software package may be retrieved. Each mirror server URL must provide precise information concerning the protocol and may provide authentication credentials that are to be presented in order that authorization may be granted prior to retrieval of the software package.

- Constraint

This attribute provides an eventual list of constraints which must be satisfied for correct use of the application package.

- Requirements

This attribute provides an eventual list of requirements that must be satisfied for correct use of the application package.

- Price

This attribute provides an eventual fully qualified universal unique identifier of the price category instance that is applicable for use of images based on the corresponding operating system. This value will be used, along with eventual price information of package components of the parent image unless a global price has been defined for the image instance. Financial transactions will be generated upon successful completion of the provisioning of contracts making use of priced application software packages.

- License

This attribute provides an eventual description of licensing conditions that are to be satisfied for use of images based on the corresponding application package.

- Installation

This attribute provides the precise installation tool syntax required to be employed to ensure the recommended installation procedure is performed.

- Configuration

This attribute provides the description of the post installation syntax required to be employed to ensure that the package is correctly configured.

- Description

This attribute provides a full, human readable, textual description of the application product, package or patch.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

The package category expects link list items to be of the port category and providing the description of individual service ports that are required for correct operation of the application package and are to contribute to the complete service port description of the resulting application firewall.

Configuration

Configuration category instances are referenced via the single fully qualified universal unique identifier of manifest category instances to provide the actions and instructions to be performed for the configuration of Meta Data management and monitoring system initialization for the complete manifest described final provisioned service instance.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute allows a name to be associated with a particular configuration category instance.

- Actions

This attribute contains the count of configuration action elements linked to the configuration category instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. The removal of a configuration category instance must also ensure that all unnamed action instances on the actions linked list of the configuration instance are also removed.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

- Action

Configuration category instances require links to be of the action category only.

Action

Action category instances are appended to the link list of their parent configuration category instance and provide the details of the actions and instructions required to be performed for the configuration of service instanced provisioned for the parent manifest or plan.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute allows a name to be associated with a particular configuration action category instance.

- Type

This attribute MUST define the type of syntax used for the description of the action as provided by the expression attribute.

- Expression

This attribute provides the details of the configuration action and is expressed using the syntax indicated by the value of the type attribute of the action category. Currently only “Cordscript” is recognized for use in action expressions. For precise details concerning the syntax and possibilities of Cordscript please refer to the corresponding section of this document. As a general rule, and for convenience of use, configuration action expressions may comprise multiple configurations statements that are normally separated by the instruction terminating semi colon character.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Interface

Interface category instances are identified by a name offering a particular foreseen collection of uses of instances of the manifest. The named interface action elements define the individual interface methods providing the behaviour of an instance in terms of the collection instructions for Meta data, interconnection of nodes, monitoring activity and node method invocation.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute is required and provides the name of the particular interface category instance. The interface method will be invoked through nodal instances of the parent manifest through this name.

- Actions

This attribute contains the count of interface action elements linked to the interface category instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. The removal of a configuration category instance must also ensure that all unnamed action instances on the actions linked list of the configuration instance are also removed.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

- Action

Interface category instances require links to be of the action category only.

Release

Instances of the Release category are referenced via the single, fully qualified, universal unique identifier of the parent manifest category instances and described the actions, and resulting instructions, to be performed prior to the release of provisioned resources. This element provides the complementary behaviour to the Configuration element and can be seen as describing the destruction methods for instances of the parent manifest type.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute allows a name to be associated with a particular release category instance.

- Actions

This attribute contains the count of configuration action elements linked to the release category instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. The removal of a configuration category instance must also ensure that all unnamed action instances on the actions linked list of the configuration instance are also removed.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

- Action

Release category instances require links to be of the action category only.

Security

Instances of this category are referenced through a fully qualified universal unique identifier of manifest category instances and provide the details of security conditions and policy required to be set up and enforced for the correct operation of provisioned service instances of the parent manifest and its associated provisioning plan.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute allows a name to be associated with the particular security category instance.

- Level

This attribute defines the nature of the security measures and policy. The default value of “public” implies that no security measures or policy are required. Other values such as “private”, “protected” etc will be used to indicate the need for resolution of the policy identifier provided by the Scheme attribute.

- Scheme

When the preceding attribute value is other than “public” this attribute MUST provide a valid security scheme or policy identifier describing the measures to be established for the desired level of security.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Account

Instances of this category are referenced through a fully qualified universal unique identifier of manifest category instances and provide the details of accountancy conditions and invoicing required to be set up during operation of provisioned service instances of the parent manifest and its associated provisioning plan.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Number

This attribute defines the application software account number within the accounting system.

- Name

This attribute defines account name within the accounting system.

- Security

This attribute defines an eventual account specific fully qualified universal unique identifier of the security category instance to be associated with accountancy operations for this account.

- Legal

This attribute makes provision for the description of supplementary business and legal information to be associated with the particular account.

- Users

This attribute specifies the count of user category instances linked to an account category instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

- User

Account category instances require that linked information be of the user category.

User

Instances of the user category are appended to the link list of account category instances and define the collection of authenticated users that are to be authorized for operations pertaining to the manifest, its result provisioning plan and any eventual service instances provisioned. User category instances define the basic authentication credentials of these users and an association with an authorization credential or token.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

The value of this attribute will provide the user name component of the authentication credentials.

- Password

The value if this attribute will provide the password component of the authentication credentials.

- Role

The value of this string attribute will define the role of the user in terms of their rights and capacities within the broker platform. These roles will range from super user, administration user, vendor, purchaser and user to guest. The role will contribute to control the scope of the authorization token.

- Authorization

This attribute will contain the fully qualified universal unique identifier of the authorization category instance that is currently active for the corresponding user description.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI user category instance identified by the attribute values provided in the OCCI message. When a user category

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

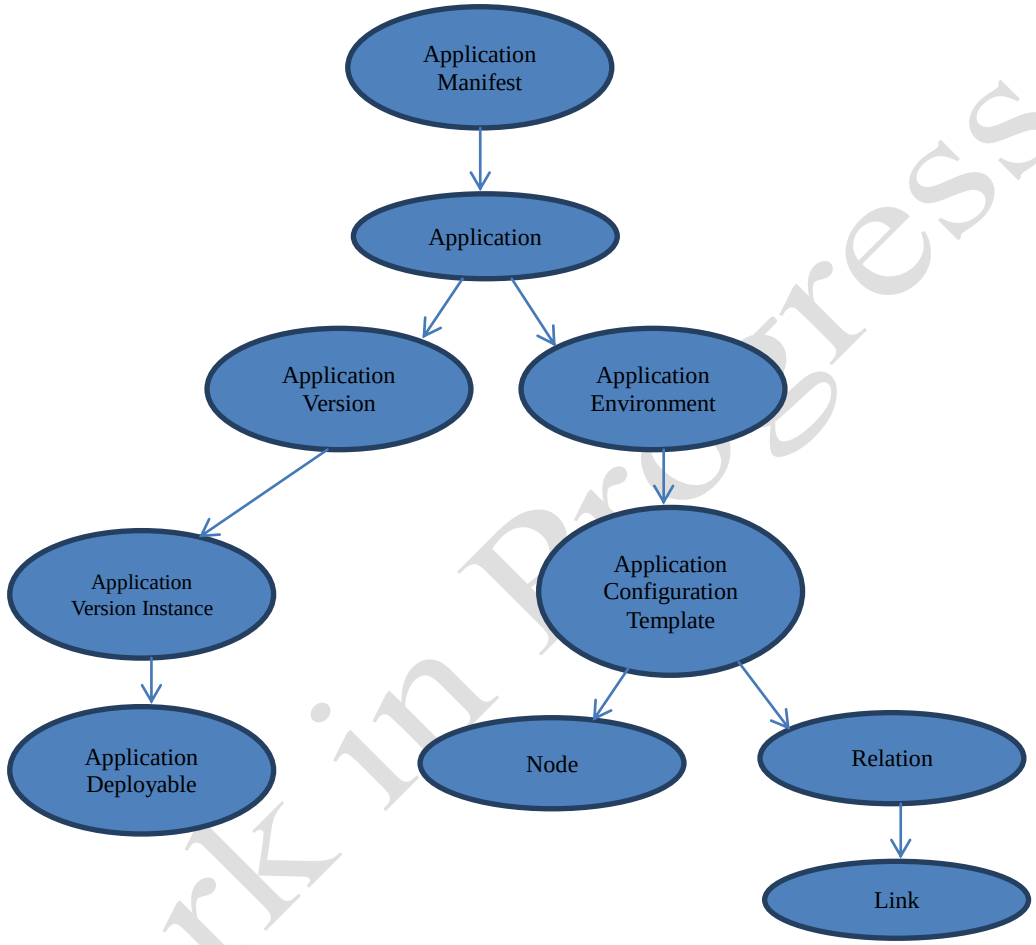
No link types are currently defined for this category.

Categories for the description of PAAS provisioning

This section of the Cords specification describes categories defined for the integration of Platform as a Service (PAAS) products and their application component provisioning within the Accords Platform. These categories are defined in the manifest schema that may be retrieved using the following URL:

http://www.compatibleone.fr/schemes/paasmanifest.xsd

The relationship between the categories describing the PAAS Provisioning Model can be seen in the following diagram:



The individual categories will now be described in terms of their attributes, elements, methods and links.

PAAS Application Manifest

This category serves an identical role to that of the IAAS Manifest and represents the ''envelope'' of the application manifest. It is used to define and describe a unique application manifest and its associated namespaces. The "paas_application_manifest" contains a unique "paas_application" element which is described in the corresponding section. The different attributes of the "paas_application_manifest" category are described below.

Attributes

The following collection of attributes is defined for this category:

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the PAAS Application Manifest and will be used to refer to the manifest in complex node definitions.

- Access

This attribute defines the access attributes of this PAAS Application Manifest and will be used to control access to the manifest in complex node definitions.

- Scope

This attribute defines the scope of the PAAS Application Manifest and will be used to control use of the manifest in complex node definitions.

- description

This attribute defines the fully qualified unique and universal identifier of the PAAS description category instance containing the human readable description of the PAAS Application Manifest.

- Paas_application

This attribute defines the fully qualified unique and universal identifier of the PAAS application defined by this PAAS Application Manifest.

- State

This attribute provides the current condition of the PAAS Application Manifest category instance.

Elements

The following sub-element is expected to be defined for the PAAS Application Version:

- PAAS Application

This element defines the application of the PAAS Manifest.

Methods

The following collection of methods is defined for this category:

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Types

No link types are currently defined for the “PAAS Application Manifest” category.

PAAS Application

This category, the PAAS application, is the core element category of the PAAS Application Manifest. It includes categories and attributes associated with the application for deployment, the application’s versions and the application’s version instances.

Attributes

The following collection of attributes is defined for this category:

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the PAAS Application Version and will be used to refer to the version from the configuration section of the parent manifest.

- Date Created

This attribute defines the creation date of the PAAS Application.

- Description

This attribute defines value of the human readable description of the PAAS Application Version.

- Paas application version

This attribute contains the unique and universal identification string of the PAAS Application Version category instance resulting from the sub element of the same name.

- Paas application environment

This attribute contains the unique and universal identification string of the PAAS Application Environment category instance resulting from the sub element of the same name.

- State

This attribute provides the current condition of the PAAS Application Version category instance.

Elements

The following two sub-elements are expected to be defined for the PAAS Application:

- PAAS Application Version

This element provides the version information for the particular PAAS Application.

- PAAS Application Environment

This element provides the description of the environment for the particular PAAS Application.

Methods

The following collection of methods is defined for this category:

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Types

No link types are currently defined for the “PAAS Application” category.

PAAS Application Version

This category provides the description of the application version for deployment. Each application can have one or more different versions deployed or to be deployed on a particular PAAS. This category allows for the selection of the required application version for the needs of the particular application

manifest.

Attributes

The following collection of attributes is defined for this category:

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the PAAS Application Version and will be used to refer to the version from the configuration section of the parent manifest.

- Label

This attribute provides the float value of the label of the application version to be deployed.

- Date_updated

This attribute provides the date at which the application version was last updated.

- Description

This attribute provides a textual description of the application version to be deployed, for information purposes only.

- Paas_application_version_instance

This attribute contains the unique and universal identification string of the PAAS Application Version Instance category instance resulting from the sub element of the same name.

- State

This attribute provides the current condition of the PAAS Application Version category instance.

Elements

The following sub-element is expected to be defined for the PAAS Application Version:

- PAAS Application Version Instance

This element provides the version instance information for the particular PAAS Application Version.

Methods

The following collection of methods is defined for this category:

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Types

No link types are currently defined for the “PAAS Application Version” category.

PAAS Application Environment

This category describes the deployment environment required by the application. This category will be handled as a separate request (in addition to the deployment of the application) for the creation of the environment in which the application will be deployed.

Attributes

The following collection of attributes is defined for this category:

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the name of the environment to be created. This value must be unique in a PAAS provider. In fact, if the environment name exists already, the environment description will be interpreted as an update query.

- Paas Application Configuration Template

This attribute defines the fully qualified unique and universal identifier of the PAAS Application Configuration Template required for the particular PAAS Environment.

- Date Created

This attribute defines the date of creation of the PAAS Environment.

- Date Updated

This attribute defines the date at which the PAAS Application Environment was last updated.

- Description

This attribute provides the human readable description of the PAAS Environment.

- State

This attribute provides the current condition of the PAAS Application Environment category instance.

Elements

The following sub-element is expected to be defined for the PAAS Application Environment:

- PAAS Application Configuration Template

This element provides the template of configuration information for the particular PAAS Application Environment.

Methods

The following collection of methods is defined for this category:

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Types

No link types are currently defined for the “PAAS Application Environment” category.

PAAS Application Version Instance

This category provides the description of the application version instance to be deployed.

Attributes

The following collection of attributes is defined for this category:

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute contains the name of the application version instance. This name must be unique in each PAAS provider.

- Uri

This attributes provides the public URI, generated by the PAAS provider, for the invocation of the application version instance once it has been deployed.

- date_instantiated

This attribute provides the creation date of the application version instance.

- Description

This attribute provides a textual description of the application version to be deployed, for information purposes only.

- Paas_application_deployable

This attribute contains the unique and universal identification string of the PAAS Application Deployable category instance resulting from the sub element of the same name.

- Default_instance

This attribute takes a Boolean: TRUE, to indicate that this application version instance is the default or FALSE otherwise.

- Provider

This attribute contains the name of the PAAS provider in which the application version instance is to be deployed. The value 'ANY' indicates that it may be deployed on any available PAAS.

- State

This attribute provides the current condition of the PAAS Application Version category instance.

Elements

The following sub-element is expected to be defined for the PAAS Application Version Instance:

- PAAS Application Deployable

This element provides information concerning the deployable PAAS Application.

Methods

The following collection of methods is defined for this category:

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Types

No link types are currently defined for the "PAAS Application Version Instance" category.

PAAS Application Configuration Template

This category describes the PAAS application configuration template to be used for the creation of the application environment. This category is mentioned in the application manifest only if is necessary to create a new configuration template.

Attributes

The following collection of attributes is defined for this category:

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the name of the environment configuration template to be created. This name must be unique in a PAAS provider and if the environment name exists already, the environment description will be interpreted as an update query.

- Date Created

This attribute defines the date of creation of the PAAS.

- Date Updated

This attribute defines the date at which the PAAS Application Environment was last updated.

- Description

This attribute provides the human readable description of the PAAS.

- State

This attribute provides the current condition of the PAAS Application Configuration Template category instance.

Elements

The following sub-elements are expected to be defined for the PAAS Application Configuration Template:

- PAAS Node
- PAAS Relation

Methods

The following collection of methods is defined for this category:

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Types

PAAS Application Configuration Template category instances may take Pass Configuration Option, PAAS Node and PAAS Relation category instances as members of their linked list.

PAAS Application Deployable

This category describes the deployable application version instance.

Attributes

The following collection of attributes is defined for this category:

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute contains the name of the application environment instance deployable.

- Content_type

This attribute provides the content type of the application environment deployable (artefacts for example).

- Location

This attribute contains the path of the directory where the deployable is located.

- Multitenancy_level

This attribute can take two possible values: 'SHAREDINSTANCE', if the application version instance can be installed in the same VM (and with a shared JVM for java applications) or 'DEDICATEDINSTANCE' otherwise.

- State

This attribute provides the current condition of the PAAS Application Deployable category instance.

Methods

The following collection of methods is defined for this category:

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all

invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Types
No link types are currently defined for the “PAAS Application Deployable” category.

PAAS Node
This category describes the unit nodes forming the environment configuration template.

Attributes
The following collection of attributes is defined for this category:

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute contains the name of the PAAS node

- Content_type

This attribute describes the type of the PAAS node. In accordance to our OCCI-PAAS extension, a PAAS node can either be a container node, a database node or a router node.

- Version

This attribute indicates the version of the PAAS node

- State

This attribute provides the current condition of the PAAS Node category instance.

Methods
The following collection of methods is defined for this category:

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Types
No link types are currently defined for the “PAAS Node” category.

PAAS Relation
This category describes the possible relations between different PAAS node elements in an environment configuration template.

Attributes
The following collection of attributes is defined for this category:

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the PAAS Relation and will be used to refer to the relation from the configuration section of the parent manifest.

- State

This attribute provides the current condition of the PAAS Relation category instance.

Methods
The following collection of methods is defined for this category:

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Types

"PAAS Relation" category instances allow definition of links to "PAAS Link" category instances.

Categories of a Service Instance

The categories described here are used for the construction, representation and management of service control graphs. A service control graph will be created by the brokering operation performed for a particular manifest of a provisioning plan.

Plan

Instances of this category are produced by the ACCORDS parser as a result of processing a CORDS manifest document. The plan instance is created for valid manifests where all constituent descriptive elements, nodes, categories have been resolved to reference a collection of category instances required to manage the precise details comprising the description of the plan. The plan provides the blue print for the production and provisioning of configurations of heterogeneous cloud resources.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the name of the manifest. This is a globally unique string of characters defining the manifest and may be used to specify this manifest as being the type description of a complex node construct.

- Manifest

This attribute provides the fully qualified universal unique instance identifier of the manifest instance describing the provisioning and resources.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI plan category instance identified by the attribute values provided in the OCCI message. The collection of service instances linked to the plan instance must also be deleted when the parent provisioning plan instance is deleted. The removal of a provisioning plan must also ensure that the reference to the particular plan instance is removed from its parent manifest description.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- INSTANCE

This action is to be invoked through a POST message providing a valid universal unique plan instance identifier and will initiate the processing of the corresponding plan by the ACCORDS Broker for the production of a new service instance of the plan. The resulting service will be added as a link element to the instance of the parent plan. Upon completion of the service instance a “START” action will be invoked through the resulting service instance identifier for the provisioning of the required resources.

Link Type

- Service

The plan instance expects link of type service. Each service link will point to an instance of the plan that was created by processing the plan, by the ACCORDS Broker, in response to an instance action being posted to the plan category instance.

Service

This category describes the service instances provisioned by the ACCORDS broker in response to instance actions invoked for a particular plan instance. Service instances regroup the collection of contracts required to satisfy the provisioning described by the nodes of the defining manifest. Service instances will be appended to the list of links of their parent plan instance.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the text string defining the name of the instance and corresponds to the name of the parent provisioning plan.

- Account

This attribute provides the fully qualified universal unique identifier of the account category instance for which the service instance was provisioned. This value is used for the generation of invoice and other accountancy events for the particular service account.

- Price

This attribute may provide the fully qualified universal unique identifier of the price category instance associated with this particular service.

- Contracts

This attribute provides the count of contract category instances that have been provisioned for the service instance.

- Session

This attribute provides the fully qualified universal unique identifier of an eventual monitoring session that has been initiated for the surveillance of the parent service instance.

- Manifest

This attribute provides the fully qualified universal unique identifier of the parent manifest category instance. This information is available through the plan and *vica verca* but it is included here for performance reasons.

- Initiation

This is the time stamp or indication of the time from which the provisioning of the service is deemed possible. Attempts at starting the provisioning of the resources prior to this will fail.

- Expiration

This is the time stamp or indication of the time after which the provisioning of the service is deemed no longer possible. Attempts at starting the provisioning of the resources after this will fail.

- Plan

This attribute provides the fully qualified universal unique identifier of the parent provisioning plan instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI service category instance identified by the attribute values provided in the OCCI message. The service instance must first be stopped, exactly for the invocation of the “stop” action, in order to ensure that no orphan provisioning instances remain after the removal of the service instance. The collection of contract instances linked to the service instance must also be deleted for the removal of their parent service category instance to be complete.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, providing a valid service instance identifier will start the provisioning and engage the actual resources described and required to fulfill the goals of the manifest described service. As a result of this action the provisioned system will be active, configured and operational. Monitoring, if defined will have been initiated and if price elements have been defined then transaction events will be generated through the accountancy channels for the corresponding customer and vendor accounts.

- STOP

This action, when invoked through a POST message, providing a valid service instance identifier will stop the provisioned system and disengage the resources that have been mobilized. Monitoring will be terminated and financial termination transaction events will be generated as required.

.

- SAVE

This action, when invoked through a POST message, providing a valid service instance identifier of an active, started service instance, will cause the current image of each the component contracts to be saved, or backed up, to create a save point from where the service instance may eventually be rolled back or restarted.

Link Type

- Contract

The service instance defines links to the collection of contract instances that were created to satisfy the requirements and constraints described by the nodes of the parent manifest.

Contract

Instances of the contract category are used for the management of the generic resources to be engaged as required to satisfy the needs of provisioned or instanced services. Contracts are said to be "negotiated" with provider platforms resulting in the creation of a corresponding provider specific contract, or provision. Contracts are appended to the link lists of their parent service instances and provide a generic intermediary interface component between the high level service and the low level provider specific provisioning system.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the textual name of the contract instance and corresponds to the node name of the defining manifest of the parent service.

- Node

This attribute provides the fully qualified universal unique identifier of the node instance describing the contract.

- Provider

This attribute provides the fully qualified universal unique identifier of the provider specific contract instance managing the engaged or provisioned resources.

- Profile

This attribute defines the subscription profile or account associated with the provider specific instance identifier.

- Placement

The value of this attribute will provide the qualified universal unique identifier of the placement category instance that was create for the management of placement solution and corresponding provider quotas.

- Firewall

This attribute provides the fully qualified universal unique identifier of the firewall category instance associated with this service contract.

- Price

This attribute, when present, provides the fully qualified universal unique identifier of the price category instance associated with this particular contract.

- Scope

This attribute provides the scope information of the contract category instance. This is taken from the original node description and may be either “normal” or “common”.

- Access

This attribute provides the access right for this contract as defined by its original node description as either “public” or “private”.

- Type

This attribute provides the fully qualified universal unique identifier of the provisioning plan that provides the manifest description of the complex contract instance.

- Common

When the scope attribute value indicates common, this attribute will provide the fully qualified universal unique identifier of the common instance to which the contract refers. Depending on the type of the contract, simple or complex, this common instance may be a service category instance identifier or a contract category instance identifier. When this “common” contract instance handling is active the following “normal” contract management attributes are not required and their presence and behaviour is undefined.

- Service

When the preceding attribute indicates a manifest described complex node, this attribute will contain the fully qualified universal unique identifier of the service category instance of the complex provisioning of this contract. Subsequent simple node management attributes will not be required and their presence and behaviour is undefined.

- Reference

This attribute contains a copy of the provider specific reference or identifier for the provisioned resources.

- Hostname

This attribute provides the host name or address through which the provisioned resources are accessible.

- Administration Password

This attribute contains the value of the administration password of the allocated resource.

- Work Load

This attribute contains the value identification of the work load image of the underlying provider platform specific contract.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI contract category instance identified by the attribute values provided in the OCCI message. Removal of a contract category instance requires that all provisioning controlled by through a provider specific contract instance first be stopped and removed, then the linked collection of configuration instruction instances must also be removed.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, providing a valid contract instance identifier will start the provisioning and engage the actual resources described and required to fulfill the goals of the node described contract. As a result of this action the provisioned resources will be active, configured and operational. Information pertaining to the provisioning of the contract will be extracted from the associated instruction set prior to launching the provisioning. Information resulting from the operation will be used to update the configuration instructions upon completion. Monitoring, if defined will have been initiated and if price elements have been defined then transaction events will be generated through the accountancy channels for the corresponding customer and vendor accounts.

- STOP

This action, when invoked through a POST message, providing a valid contract instance identifier will stop the provisioned contract and disengage the resources that have been mobilized. Monitoring will be terminated and financial termination transaction events will be generated as required.

- SAVE

This action, when invoked through a POST message, providing a valid contract instance identifier of an active, started contract instance, will cause the current image of the contract to be saved, or backed up, to create a save point from where the contract instance may eventually be rolled back or restarted.

Link Type

- Instruction

Instances of the instruction category are appended to the list of links for instances of the contract category. These instructions result from the processing of configuration actions detected as either targeting or sourcing the specific contract instance. These instructions may relate to Meta Data configuration operations or may be used for the activation of monitoring conditions for the particular contract instance.

Instruction

Instances of the instruction category are appended to the link list of contract instances in the final phase of service instance production during the brokering and provisioning operation of the ACCORDS Broker. Each instruction category instance corresponds to a single configuration action statement resulting from the Cordscript parsing of the configuration action expressions. The different attributes of this category instance type establish the relationship between the logical or generic contract and the concrete provider specific contract to which the instruction applies allowing the automatic configuration mechanisms of the ACCORDS platform to perform the required configuration of the different contract instances in terms of their Meta Data requirements and Monitoring Activity.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Target

This attribute provides the fully qualified universal unique identifier of the generic target contract instance of the configuration or monitoring statement. This value results from the interpretation of the contract/node names used to prefix the action or method of the statement.

- Prefix

This attribute provides the full textual form of the Target Contract/Node prefix which was resolved to provide the preceding Target attribute value.

- Type

This attribute specifies the type or nature of the instruction. The only value that is currently supported for the type property is “method”. This requires the name of the method to be provided by the corresponding property value.

- Method

This attribute value provides the name of the method for which the instruction category instance has been created to represent. Currently two distinct method names are possible, “configure” and “monitor” representing the two different action types that are defined by the current version of Cordscript.

- Source

This attribute provides the fully qualified universal unique identifier of an eventual generic source contract instance of the configuration or monitoring statement. This value results from the interpretation of the contract/node names used to prefix the property or probes names in the statement.

- Symbol

This attribute provides the full textual form of the Source Contract/Node prefix which was resolved to provide the preceding Source attribute value.

- Property

This attribute provides the textual name of the property or probe referred to by the Source attribute value.

- Value

This attribute provides the current value of the property in Meta Data type configuration instructions. This value, as named by Symbol, is made available for configuration of a Target Contract having been retrieved as a result of the provisioning of the Source Contract.

- Provision

This attribute provides the fully qualified universal unique identifier of the provider category instance responsible for the management of the underlying resource provisioning. It will be used by the provider procci for the recovery of configuration instructions that are to contribute to the Meta Data collection required for personalization of the contract instance image and behaviour.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. These category instances should only be deleted by their parent contract instance since they are critical for the correct operation of the meta data transfer procedure during provisioning of resources.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Firewall

Instances of the firewall category are used for the management of the collection of service ports to be configured allowing network access to the application services offered by the provisioned resources of the contract. The port descriptions are appended as links to the firewall category instances and are collected in order from the list of application package service port descriptions and the collection of network interface service port descriptions.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the textual name of the contract instance and corresponds to the node name of the defining manifest of the parent service.

- Node

This attribute provides the fully qualified universal unique identifier of the node instance describing the contract.

- Provider

This attribute provides the fully qualified universal unique identifier of the provider specific contract instance managing the engaged or provisioned resources.

- Profile

This attribute defines the subscription profile or account associated with the provider specific instance identifier.

- Price

This attribute, when present, provides the fully qualified universal unique identifier of the price category instance associated with this particular contract.

- Ports

This attribute provides the count of the associated service port category instances.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI contract category instance identified by the attribute values provided in the OCCI message. Removal of a contract category instance requires that all provisioning controlled by through a provider specific contract instance first be stopped and removed, then the linked collection of configuration instruction instances must also be removed.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, providing a valid firewall instance identifier will start the configuration operation such that the resulting firewall of the provisioned resources described by the contract identifier will be ready and operationally to allo access to and from the services of the application image.

- STOP

This action, when invoked through a POST message, providing a valid firewall instance identifier will reset the application firewall such that only the COSACS interface is possible for future control of the instance.

Link Type

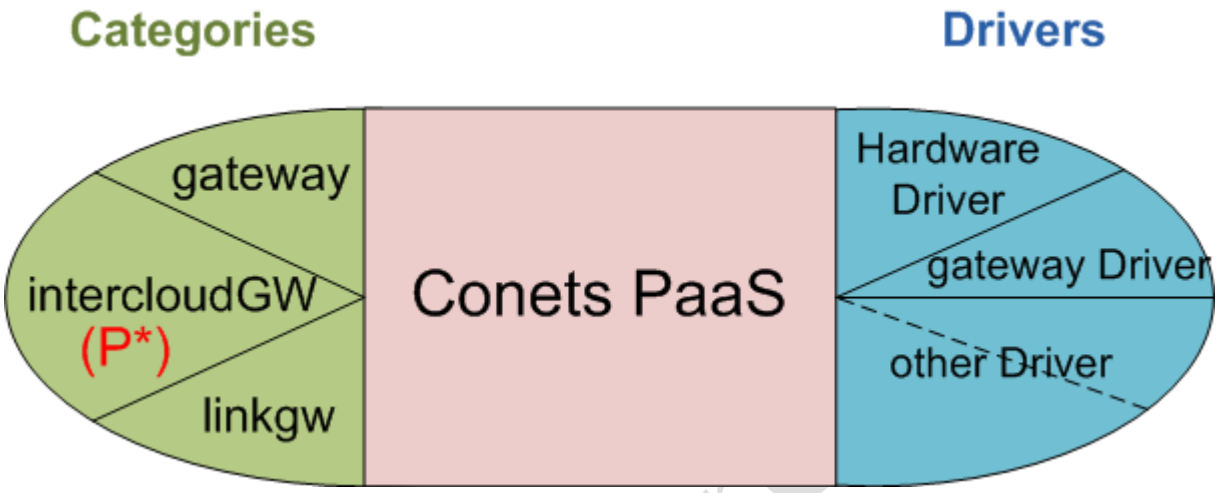
- Port

Instances of the port category are appended to the list of links for instances of the firewall category. These service ports result from the processing of network element of the

infrastructure description and the collection of package elements of the application image description.

Gateway

This category is used as the provisioning type for nodes which represent virtual machines that are to be used as gateways between different cloud provisioning systems and endpoints. The following diagram shows the way these categories contribute to the provisioning PROCCI.



Attributes

- ID

  This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

  This attribute provides an optional name for identification of the gateway category instance.

- Publicaddr

  This attribute will contain the public IP address of the virtual machine acting as the gateway.

- Privateaddr

  This attribute will contain the private IP address of the virtual machine acting as gateway.

- Ethername

  This attribute will contain the name of the network interface (ex. eth0, eth1...) that will represent and provide the connection with other gateways.

- IntercloudGW

This attribute provides the fully qualified universal unique identifier of the node category instance from which the necessary instantiation information, in terms of infrastructure and image, was retrieved.

- Contract

  This attribute provides the fully qualified universal unique identifier of the contract of the instantiated virtual machine acting as a gateway

- Provider

  This attribute will contain the name of the provider type through which provisioning of the virtual machine will acting as a gateway will be performed.

- Connection

  This attribute provides the number of active connections that have been configured through this particular gateway category instance.

- Account

  This attribute provides the fully qualified universal unique identifier of the account category instance for which the gateway was provisioned.

- State

  This attribute value defines the status of the gateway category instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Any provisioning that is currently been engaged and controlled by the particular category instance must be stopped and released prior to completion of this operation in order to ensure no orphan provisioning remains.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, providing a valid contract instance identifier will start the provisioning and engage the actual resources described and required to fulfil the goals of the gateway node described contract. As a result of this action the provisioned resources will be active, configured and operational. Information pertaining to the provisioning of the contract will be extracted from the associated instruction set prior to launching the provisioning. Information resulting from the operation will be used to update the configuration instructions upon completion.

- STOP

This action, when invoked through a POST message, providing a valid contract instance identifier will stop the provisioned contract and disengage the resources that have been mobilized.

Link Type

No link types are currently defined for this category.

LinkGW

The LinkGW category is used to represent the interconnecting link between gateways.

Attributes

- Id

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides an optional name for identification of the LinkGW instance.

- IntercloudGW

This attribute provides the fully qualified universal unique identifier of the node category instance in which the necessary information (image, network...) to instantiate the gateway was specified.

- Account

This attribute provides the fully qualified universal unique identifier of the account category instance for which the gateway was provisioned.

- Gwsrc

This attribute provides the fully qualified universal unique identifier of the gateway category instance. This attribute represent the gateway source of the link that will be established between gateways.

- Gwdst: This attribute provides the fully qualified universal unique identifier of the gateway category instance. This attribute represent the gateway destination of the link that will be established between gateways.
- Tunnelproto

  This attribute will contain the tunnel protocol (ex. GRE “generic routing encapsulation”..) that will be used to encapsulate traffic in the link between gateways.
- Addressgresrc

  This attribute will contain the address of the GRE source.
- Addressgredst

  This attribute will contain the address of the GRE destination.
- Prefix

  This attribute will contain the address prefix of the GRE source.
- Authenticationkey

  This attribute will contain the authentication key that will be used to establish the ipsec tunnel between gateways
- Endpointsrc

  This attribute provides the fully qualified universal unique identifier of the node category instance that will be attached to the gateway. This endpoint source will be connected to the gateway source.
- Endpointdst

  This attribute provides the fully qualified universal unique identifier of the node category instance that will be attached to the gateway. This endpoint destination will be connected to the gateway destination.
- State

  This attribute value defines the status of the LinkGW instance. When the link is configured between gateways, the value of this attribute will be set to 1.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI contract category instance identified by the attribute values provided in the OCCI message. Removal of a contract category instance requires that all provisioning controlled by through a provider specific contract instance first be stopped and removed, then the linked collection of configuration instruction instances must also be removed.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, providing a valid LinkGW instance identifier will start the configuration of the gateway link.

- STOP

This action, when invoked through a POST message, providing a valid LinkGW instance identifier will reset the configuration of the gateway link.

Link Type

No link types are currently defined for this category.

Categories for Platform Operation

The categories described in this section are used for the management of the provisioning system and the instances of service created for provisioning of resources as described by the provisioning plans and manifests.

Operator

Instances of this category are used not only to describe the platform operator but also the different operators of service provider platforms operating through the particular ACCORDS platform instance. The concept of operator, and the legal entity that is thereby represented, is important not only in the different aspects relating to the management of Service Level Agreements but also to the Invoicing, Billing and Accounting operations. At the brokering level, Service Level Agreements are considered to have been negotiated between the primary ACCORDS platform operator and each of the subordinate service provider platform operators. These agreements are necessary in order to establish the basic commercial conditions, quotas and pricing, on which the placement algorithms will base their decisions during the brokering cycle and in particular the choice of placement for the provisioning of a particular component contract of a service instance.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the legal name of the operator.

- Description

This attribute provides a long textual description of the operator.

- Account

This attribute contains the unique and universal identifier of the account category instance associated with this operator providing the legal and accountancy details for the operator of the platform.

- Security

This attribute contains the unique and universal identifier of the security category instance associated with this operator defining the default security preferences and policy.

- Pricelist

This attribute contains the unique and universal identifier of the price list category instance associated with this operator and consequently gives access to the collection of priced category instances published by the operator.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI operator category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Publication

Instances of this category are used by the service provider components of the ACCORDS platform for the publication of their service categories. Each category instance represents a particular category of service as published by a service provider component either of the base ACCORDS platform or as made available by operators of provisioning platforms of IaaS, PAAS or SaaS types. Each individual service provider component may publish any number of categories which may or may not be priced commodities.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- What

This attribute contains the name of the category for which the publication category instance has been created. Many different category instances of the same name are possible and allow different providers and operators to publish competitive services of differing quality and price.

- Where

This attribute provides a subcategory specification that allows filtering and selection of publications. This is currently a constant value of “marketplace”.

- Why

This attribute provides the reason for the publication and as such contains the URL identity of the service provider that published the category. This value is to be used to contact the service provider for provision of this particular category of service.

- Zone

This attribute provides a localization selection criteria allowing for the organization of categories by geographical zone, physical location or as other abstract collections of categories, providers and operators.

- Price

For priced categories this attribute will provide the fully qualified universal unique identifier of the price category instance relating to this published category.

- Rating

For opinion rated categories this attribute will provide the fully qualified universal unique identifier of the rating category instance relating to this published category.

- Operator

This attribute will provide the fully qualified universal unique identifier of the operator category instance responsible for the service provider that published the particular category.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Publication category instances should only be deleted by the service provider module that required publication of the particular category of service.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Placement

Instances of this category are created by the placement engine and provide the broker with assistance in selecting appropriate service providers for the creation of provider specific contracts for individual contract nodes of a service instance. The algorithms controlling this Placement operation will make use of the property values provided in the Placement Creation request and in particular the placement region, provider nature and eventual financial or technical constraints.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- State

The value of this attribute reflects the status of the placement record as follows:

- 0: the CHOOSE action is required for the selection of a placement solution and reservation of resources quota.
- 1: the selection of a placement solution has been performed and is pending provisioning.
- 2: the provisioning of the resources has been performed and consequently the corresponding quota has been consumed.

- Name

This attribute contains an eventual naming token for the particular placement category instance.

- Algorithm

This optional attribute value provides the fully qualified universal unique identifier of an explicit algorithm category instance that is to be used to control the placement operation. If this value is omitted then the default placement algorithm as defined by the platform operator will be used.

- Node

This attribute provides the fully qualified universal unique identifier of the node category instance that describes the infrastructure and application images required to be provisioned for this node. The node description is to be used by the placement engine to ensure that the selected provider is capable of delivering the level of service required to fulfill the needs of the particular node.

- Solution

This attribute provides the output value from the placement operation and will contain the published identity of the service provider that was selected by the placement engine as being the most suitable for the provisioning of the node.

- Price

This optional attribute provides the fully qualified universal unique identifier of the price category instance that is to be used as the reference as the financial constraint to be respected by the chosen solution for this particular placement act. If this value is omitted then placement will not consider financial criteria unless directed to do so by an explicit algorithm category instance identifier or as directed by the platform operator's default preferences.

- Zone

This optional attribute provides the geographic region or zone description criteria to be respected by the chosen solution for this particular placement operation. If this value is omitted then placement will not consider geographical or zone criteria unless directed to do so by an explicit algorithm category instance identifier or as directed by the platform operator's default preferences.

- Opinion

This optional attribute provides an indication that the provided opinion related criteria are to be used to assist in the selection of a for this particular placement operation. If this value is omitted then placement will not consider opinion criteria unless directed to do so by an explicit algorithm category instance identifier or as directed by the platform operator's default preferences.

- Security

The value of this attribute may provide required security constraints to be taken into consideration during the selection of the resulting placement solution.

- Provider

This attribute provides the type of provisioning service required. This will be the category name of the required provider type and may be any one of the currently supported provider types: openstack, opennebula, windowsazure, proactive or slap grid. If this value is omitted it should default to either "any" or the default value established by the platform operator. It is expected that multiple solutions for each provider type be available to allow the placement engine to select the most appropriate for the provisioning requirements of a particular node.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Placement categories provide the cache of placement information. It is not recommended that these instances be removed unless a complete reset of the placement engine knowledge base is absolutely required.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- CHOOSE

The CHOOSE action will be invoked by the broker for the selection of a provider as the solution for the required placement. The calculation of the placement operation will be governed by the values of the algorithm, provider, and node attributes and will make use of the accompanying attribute values as required for the selection of an appropriate solution. Reservation of provisioning elements taken from the provider quota, as required for the placement solution, will be performed during the operation

- CONSUME

The CONSUME action will be invoked by the generic contract manager when a provisioning operation, in response to a start action, has been successfully completed resulting in the deployment of the resources described by the contract with the provider indicated by the placement solution. The placement manager will convert the quota reservation information to quota consumption status.

- RESTORE

The RESTORE action will be invoked by the generic contract manager when a de-provisioning operation, in response to a stop action, has been successfully completed resulting in the liberation of the resources described by the contract with the provider indicated by the placement solution. The placement manager will convert the quota consumed to quota reserved status.

- RELEASE

The RELEASE action will be invoked by the generic contract manager when the corresponding generic contract is deleted in order that the placement manager may release the reservation of resources required for the associated provisioning.

Link Type

Quantity category instances will be appended to the linked list of placement category instances representing the quota values consumed or attributed to that particular placement operation.

Quantity

Instances of this category are created by the placement engine in order to manage the individual quantities of each quota as they are processed during the placement operations. Quantity category

instances are linked to their controlling Placement category instance and reference the corresponding Quota category instance of the service level agreement of the provider selected as the solution for the placement of the contract.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- State

The value of this attribute reflects the current status of the quantity record as follows:

- 0: the state of the quantity record is undefined.
- 1: the “value” attribute of this quantity instance has been added to the “reserved” attribute of the referenced quota instance.
- 2: the “value” attribute of this quantity instance has been added to the “consumed” attribute of the referenced quota instance.

- Quota

This attribute provides the fully qualified universal unique identifier of the quota category instance that will be affected by actions performed on this quantity instance.

- Value

This attribute provides the actual value that will be used to affect the referenced quota category instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Placement categories provide the cache of placement information. It is not recommended that these instances be removed unless a complete reset of the placement engine knowledge base is absolutely required.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- RESERVE

The RESERVE action will be invoked by the placement engine upon completion of a placement choice action for each of the quantity category instances linked to that placement. The action will increase the value of the reserved quantity attribute of the referenced quota category instance by the value provided by the “value” attribute of the “quantity” instance.

- CONSUME

The CONSUME action will be invoked by the placement engine for each quantity category instance of a placement category instance on reception of a “consume” event by that placement category instance. The action will increase the value of the consumed quantity attribute of the referenced quota category instance by the value provided by the “value” attribute of the “quantity” instance.

- RELEASE

The RELEASE action will be invoked by the placement engine for each quantity category instance of a placement category instance on reception of a “release” event by that placement category instance. The action will decrease the value of the consumed quantity attribute of the referenced quota category instance by the value provided by the “value” attribute of the “quantity” instance.

- RESTORE

The RESTORE action will be invoked by the placement engine for each quantity category instance of a placement category instance on reception of a “consume” event by that placement category instance. The action will decrease the value of the reserved quantity attribute of the referenced quota category instance by the value provided by the “value” attribute of the “quantity” instance.

Link Type

No link types are currently defined for the quantity category.

Quota

Instances of this category are created by the platform operator or by the negotiation software associated with third party operator’s platforms and informs the placement engine of provisioning quotas made available for use by the broker. Quotas provide the pools of resources selected for use by the placement algorithms and constitute the descriptions of the basic raw materials of the platform.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute contains an eventual naming token for the particular quota category instance.

- Description

This optional attribute value provides a human readable description of the quota and eventual legal and commercial conditions required for its use.

- Operator

This attribute provides the fully qualified universal unique identifier of the operator category instance that describes the platform operator for which the quota applies.

- Provision

This attribute provides the fully qualified universal unique identifier of the operator platform specific service provider for which the quota applies.

- Price

This optional attribute provides the fully qualified universal unique identifier of the price category instance that is to be used as the reference as the financial constraint associated with elements described by this quota.

- Zone

This optional attribute provides the geographic region or zone description criteria to be associated with provisioning in this quota.

- Offered

This attribute provides the upper limit of the resources described and made available by the particular quota category instance.

- Reserved

This attribute keeps track of the resources selected and reserved during placement operations taken from this particular quota.

- Consumed

This attribute keeps track of the resources actually provisioned during provider provisioning operations and taken from this particular quota.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute

values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for the placement category.

Algorithm

Instances of this category are created by the platform operator or by the negotiation software associated with third party operator's platforms and informs the placement engine of the operational details, constraints and criteria required for the control of the placement operation. A placement will be performed using a particular algorithm. The "First Come First Served" and "Round Robin" algorithms are to be provided by default for use by the placement engine. The algorithms descriptions made available through this category are intended to allow extension of the scope and intelligence of placement.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute contains the naming token for the particular algorithm category instance.

- Description

This optional attribute value provides a human readable description of the placement algorithm and eventual legal and commercial conditions required for its use.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute

values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for the algorithm category.

Schedule

Instances of this category are created by the parser and broker engines and provide assistance in the scheduling of heavy workloads such as image production and service deployment. The scheduler will prioritize jobs that have been submitted for launch in an attempt to ensure smooth operation of the platform and that a consistent quality of service may be achieved where contractually required.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Reference

This attribute contains an eventual naming token provided for identification of and reference to the reason for the dispatched work load.

- Account

This attribute provides the fully qualified universal unique identifier of the account for which the scheduling operation is to be performed. The account nature in terms of quality of service may be used to prioritize workload.

- Target

This attribute provides the URL of the targeted workload and the operation to be performed. This will always be an OCCI Action Request comprising a fully qualified category instance identifier and an Action value pair.

- Status

This attribute allows the current status of the scheduled workload to be inspected. IT will be set initially to PENDING. When the operation has been launched it will be set to SCHEDULED and when the operation has completed it will be set to SUCCESS. Failure during the operation will be reported as FAILURE with the associated termination code.

- Priority

This attribute will be used by the scheduler algorithms for the prioritization of workload for scheduling.

- Scheduled

This attribute provides the date and time stamp at which the workload was submitted to the scheduler.

- Expected

This attribute provides the date and time stamp at which the workload is currently expected to be submitted for processing. The initial value of this attribute will be calculated as the average of durations of tasks of a similar nature. The value may evolve with respect to the arrival of higher priority workload.

- Started

This attribute provides the date and time stamp at which the workload was submitted for processing.

- Completed

This attribute provides the date and time stamp at which the workload processing was completed

- Duration

This attribute provides the difference between the time at which the operation was launched and the time at which it was completed. It will be used by future scheduled tasks to attempt to predict the duration of operations of this nature.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Placement categories provide the cache of placement information. It is not recommended that these instances be removed unless a complete reset of the placement engine knowledge base is absolutely required.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for the placement category.

Provider

Instances of this category are created by the different platform PROCCI in order to publish their particular category of provision such that the placement engine may be aware of their offer of service. Quota information will be localised through the provider service level agreement defined for this operator and will be made available for use by the placement engine for deployment of resources through the corresponding PROCCI interface.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute contains an eventual naming token for the provider category instance.

- Operator

This attribute contains the unique and universal identifier of the operator category instance of the operator publishing this provider category information. The value of this attribute will be used during start-up of the PROCCI to localise the account of the operator and subsequent provider service level agreement for the retrieval of the quota definitions.

- Profile

This attribute defines the subscription account or provider profile information to be used for access to this particular operator's provision of service.

- Price

This attribute provides the fully qualified unique universal identifier of the price category instance defined for the particular provider category instance. When a value has been defined for this attribute it will provide a global price that may be used by the placement engine to determine the most cost effective operator for the particular category for provision of service.

- Category

This attribute defines the particular name of the category that is available, through the publisher, for the provisioning of the provider specific resources.

- Identity

This attribute provides the URL access address through which the category management interface may be reached for the provisioning of service contracts through this interface.

- Zone

The value of this attribute defines the zone or location of the provisioning or resources by the provider. This may be the localisation in a building or a geographical location as required.

- Opinion

The value of this attribute defines the current value of the opinion or rating of the provider.

- Security

The value of this attribute defines le level or conditions of security offered by the provider.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI provider category instance identified by the attribute values provided in the OCCI message. These category instances should only be removed by the service provider module that created them for the representation of their particular operator's category of service.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

Instances of the “quota” category are expected as the link types defined for this category.

Authorization

Instances of the authorization category are created for a particular user authentication credential and are used to represent the user and their rights within the ACCORDS platform. User category instances reference these authorization category instances as their current authorization.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- User

The value of this attribute will provide the user name component of the authentication credentials.

- Pass

The value if this attribute will provide the password component of the authentication credentials.

- Account

This attribute will contain the fully qualified universal unique identifier of the account category instance that parents the authenticated user.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Application

Instances of the application category are created for a production of software application images that are to be associated with a particular image category instance. The image category instance, through its “system” and “package” components, provides the logical description of the required work load. The “application” category instance represents the physical realisation, ready for deployment, of the required solution comprising base operating system and installed packages.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Image

The value of this attribute contains the unique and universal identification string of the image category instance providing the description of the required software image.

- Provider

The value if this attribute provides the nature of the required provider platform type.

- Url

The value of this attribute is taken from the workload of the contract once the application has been built and will be used to identify the resulting application image resources for management by the VM manager category of EZVM.

- Status

This attribute will contain the current status of the application build process and be one of the following:

1. Pending

   The application image build procedure has not yet been engaged.

2. Provisioning

   The provisioning of the infrastructure required for the application build operation is underway.

3. Installation

   The application package installation procedure is underway.

4. Snapshot

   The final application image snapshot or save operation is currently underway.

5. Ready

   The application image has been successfully created.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- BUILD

The BUILD method will launch the application build procedure comprising node and contract preparation, provisioning, package installation, image snapshot and release of resources. The value of the status attribute will reflect the current phase of the build operation.

Link Type

No link types are currently defined for this category.

Financial Categories

This collection of categories is used by the financial transaction management system of the ACCORDS platform. The creation of category instances, for which a price has been defined by the platform operator, will give rise to the creation of prices transactions. Certain particular categories also define eventual price information which will also result in transactions being created when the resources that they represent are deployed or engaged. Transactions will be created for a particular account through which they may eventually be collected by Invoicing and other Accountancy tools as required to satisfy the commercial needs operation of the platform.

Price

Instances of this category are created by platform operators in order to define both price and costing information for use in a category by category, service by service or an operation oriented manner. The unique universal identifier may be associated with a category for publication allowing eventual clients of the service to perform cost oriented placement. Transactional events will be generated for the brokering of such items for which a price has been specified and accepted.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the textual name category, service, operation or otherwise to which this price category instance is to be applied.

- Operator

This attribute will provide the fully qualified universal unique identifier of the operator category instance responsible for the price category instance.

- Fixed

This attribute provides a fixed portion of the price.

- Rate

This attribute provides a periodic variable portion used to calculate the price.

- Units

This attribute provides the type of the units involved.

- Currency

This attribute provides the currency type for which price has been defined.

- Period

This attribute provides details pertaining to the periodicity of the price information.

- Description

This attribute provides a human readable textual description of the price definition and describes its application and use.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI price category instance identified by the attribute values provided in the OCCI message. This operation will not be authorized other than for supervisor ability enabled user accounts.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Transaction

Requests for the creation of instances of the transaction category are generated for the creation of category instances for which a price has been specified. Transactions are also created by provider specific contract service modules, upon invocation of their category actions start, save and stop, when a price has been specified for their particular provisioning category type. Transactions are generated for a particular account, the universal unique instance identifier of which must accompany the transaction creation request.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Account

This attribute contains the unique and universal identification string of the account category instance for which this transaction has been generated.

- Price

This attribute provides the unique and universal identification string of the price category instance describing the financial details of the transaction.

- Source

The value of this attribute provides the name of category for which the transaction was created.

- Description

This attribute provides information describing the nature or reason for the transaction. Currently the following transaction types are defined:

1. Instance Creation

   This is indicated by the description value of “method=create”

2. Start Provisioned Resources

   When the “start” action of a provider specific contract is invoked a transaction will be generated with the description value being set to “action=start”

3. Save Provisioned Instance

   When the “save” action of a provider specific contract is invoked a transaction will be generated with the description value being set to “action=save”

4. Stop Provisioned Resources

   When the “stop” action of a provider specific contract is invoked a transaction will be generated with the description value being set to “action=stop”

- State

This attribute specifies the state of the transaction. When created a transaction may be processed by addition to the transaction list of an invoice created for a customer account. This attribute will be set to zero allowing a transaction to be processed and to one to show that it has been processed for a closed invoice.

- Reference

This attribute will contain the fully qualified universal unique identifier of the category instance for which the transaction has been generated and subsequently identifies the subject of the transaction.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- CLOSE

The CLOSE action will be invoked by a parent invoice category instance when the invoice is required to terminate the transaction list in response to the reception of a CLOSE action.

Link Type

No link types are currently defined for this category.

Invoice

Instances of the invoice category are created for a specific customer account category instance. The fully qualified universal unique account instance identifier MUST accompany the request for creation of an invoice and will be used to select the collection of transactions pending invoicing for the particular account. An HTML rendering of the invoice will be generated during the invoice creation operation, the name of which will be stored to the document attribute. This document may be retrieved via an OCCI category instance GET request that specifies acceptance of HTML rendering content type.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Number

This attribute may specify a request defined alternative invoice number for identification of the invoice by an external CRM and other accountancy packages.

- Account

This attribute contains the unique and universal identification string of the account category instance for which this invoice has been produced.

- Date

This attribute provides the standard second value time stamp at which the invoice was created.

- Document

The universal and unique identifier of the human readable legal document produced for this invoice. Details of all cost items are included providing identification of the involved resources. This is currently an HTML document that may be retrieved for inspection through the OCCI HTML rendering interface.

- Total

This attribute will contain the total cumulated price of all transactions that were selected for processing by the invoice processing operation.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message. The method will instigate processing of all transactions pending invoicing for the account defined by the account instance identifier.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of an invoice category instance does not require removal of the collection of transaction contributing to the invoice since it may be necessary to regenerate an invoice for a collection of transactions.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message. As for the POST method, this method will also perform transaction processing for an existing invoice allowing transactions to be re-evaluated and the invoice totals to be calculated.

Actions

- PROCESS

The PROCESS action allows the collection of transaction pending for the Account of the invoice to be processed and updated as the content of the invoice. This action will fail if the CLOSE action has been invoked.

- CLOSE

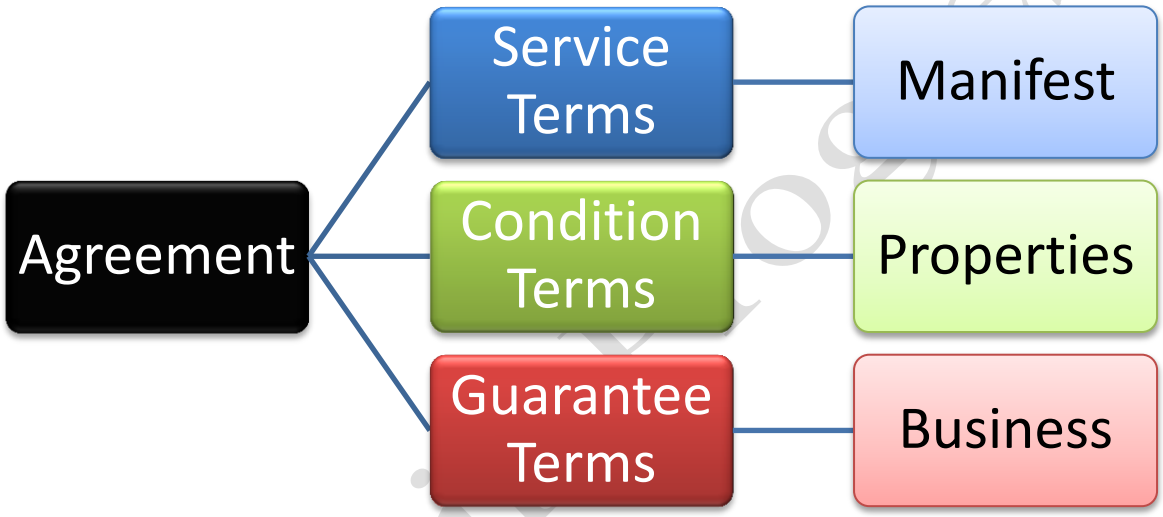
The CLOSE action terminates the invoice and protects it from subsequent modification through the PROCESS action. The collection of transactions will be transformed to the "invoiced" status by invocation of the corresponding action for each transaction category instance.

Link Type

No link types are currently defined for this category but it would be convenient, but not strictly necessary, that the list of transaction instances processed for a particular invoice be linked to the invoice instance. In this way it would be easier to recognize the list of already processed transactions during recalculation of an invoice in response to reception of a PUT method request.

Service Level Agreement Categories

The CORDS specification has been extended to provide the definition of the collection of categories required for the management of provisioning through service level agreements (SLA). This work concerning the SLA is based on the work performed by the Open Grid Forum and published in the form of the WS-Agreement Standard. The WS-Agreement specification was intended for use for the description of service level agreements for Web Services. The CORDS specification aims at providing a service level agreement description and management model for Cloud Services and requires a certain degree of poetic license in order to be able to attain these goals. The different constituent elements of an Agreement, as described by the WS-Agreement specification, have consequently been adapted for discrete and distributed operation as OCCI category instances. An XML document schema [4] has been prepared to allow processing of these documents by the standard ACCORDS document parser. The following diagram gives an overview of the structure of an Agreement.



Each of the categories involved in the construction and management of these agreements will now be described. They are currently implemented in the ACCORDS Platform component module known as SLAM.

Agreement

Instances of this category are used to represent service level agreements not only between the broker platform operator and their customers but also between the operators and their service providers. The agreement will also define the precise nature of the service in terms of a CORDS manifest and will describe the conditions expected to be satisfied by the provision of service. Being a support for dynamic control of deployment conditions the agreement makes provision for the description of the guarantees and objectives expected for respected for the fulfilment of the service conditions.

Attributes

- ID

  This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the agreement type.

- Description

This attribute provides a human readable description of the agreement.

- Initiator

This attribute is to provide the name of the account of the initiator of the agreement. It will be resolved to the unique universal identifier of the corresponding account during post processing of the SLA document during parsing.

- Responder

This attribute is to provide the name of the account of the responder of the agreement. It will be resolved to the unique universal identifier of the corresponding account during post processing of the SLA document during parsing.

- Service provider

This attribute will indicate which of the two parties is to be considered the provider of the service. The value of this attribute must be either “initiator” or “responder”.

- Expiration

This attribute gives an indication of the eventual date and time at which the agreement will expire.

- Template

This attribute is to provide the name of the template SLA from which the terms and conditions of this guarantee are based. It will be resolved to the unique universal identifier of the corresponding agreement during post processing of the SLA document during parsing.

- Terms

This attribute gives the number of terms comprising the agreement. This will normally be three, the service description, the conditions and the guarantee.

- State

This attribute value defines the status of the agreement and may represent offered, agreed and revoked.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

The "***terms***" category instance is defined as the link type for instances of this category.

Terms

Instances of this category are used as a container to define and regroup the three important and distinct sub-sections comprising the body of the cloud service agreement. Each section, service description, service conditions and service guarantees, will be represented by an individual *terms* category instance that encapsulates the corresponding component *term* category instances each providing the relevant detail of its own particular section.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the terms section.

- Type

This attribute defines the type of the particular "terms" category instance and MUST be specified as either "service", "conditions" or "guarantees". The nature of the "term" category instances that it may contain will be conditioned by the value of this attribute.

- Description

This attribute provides a human readable description of the section of the agreement.

- Terms

This attribute gives the number of “term” category instances comprising this particular “terms” section of the agreement.

- State

This attribute value defines the status of the terms instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

The “***term***” category instance is defined as the link type for this category.

Term

Instances of this category are used to provide the precise details of the particular “*terms”* category section of the agreement to which they are attached. The nature of their content will depend upon the value of the “type” attribute of their parent “terms” category instance.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the agreement type.

- Description

This attribute provides a human readable description of the agreement.

- State

This attribute value defines the status of the agreement and may represent offered, agreed and revoked.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

The "***term***" category instance may take "variable", "guarantee", "provider" or "manifest" category instances as its link type as defined by the optional section in the schema for this category. The nature of the permitted link type is dictated by the value of the "type" attribute of the parent "terms" category instance. When the body of a service description term is provided by a "manifest" category instance then the document is said to be a customer service level agreement. When the body of a service description term is provided by a "provider" category instance then the document is said to be a provider service level agreement.

Variable

Instances of this category are used to define the elements of service level agreements that allow description of the static and dynamic conditions to be ensured for the validity of the cloud service agreement. They may be used directly in the "service conditions" or indirectly in the "guarantee" terms of the "service guarantees". Instances, of the "variable" category, are used to identify properties of the service, as described in the "service description" section, that are often referred to as key performance indicators or KPIs. Threshold values may be specified for "variable" category

instances that are used to define the dynamic conditions to be guaranteed by the cloud service agreement.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the agreement type.

- Property

This attribute provides the identification of a property, a KPI, a variable, metric or quantity constituting an element of the conditions to be respected. The property will be interpreted differently depending on whether the *variable* is used in a conditions section or within a guarantee section. The former case represents constant values that influence the provisioning whilst the later involves dynamic quantities that are to be evaluated in real time.

The following list of properties is current available for the description of the placement conditions described in the static conditions section as required for the control of placement and provisioning:

- `cords.placement.algorithm`

The value of this property will be used to identify the placement algorithm to be used by the placement engine for the placement of the resources to be provisioned. Currently only the value of “default” is defined.

- `cords.placement.zone`

The value specified for this property will be used for the selection of the most appropriate placement in geographical or other location terms. The structure of the value is dependent upon the nature of operation of the platform.

- `cords.placement.operator`

The value of this property describes the commercial name of a platform operator for use by the placement engine for the selection of the appropriate placement.

- `cords.placement.security`

The value of this property may provide security constraints to be fulfilled by the selected placement.

- `cords.placement.energy`

The value of this property may provide energetic constraints to be fulfilled by the selected placement.

- `cords.placement.opinion`

The value of this property may provide rating or performance constraints to be fulfilled by the selected placement.

- `cords.placement.price`

The value of this property may provide financial constraints to be fulfilled by the selected placement.

- Condition

This attribute is to provide the nature of the compare operation required to be performed with respecting to the value for the particular property. This must be a standard compare operation taken from the following list:

1. Equals: EQ
2. Greater than: GR
3. Less than: LS
4. Not Equal: NE
5. Greater or Equal: GE
6. Less than or Equal: LE

- Value

This attribute provides the threshold value for the corresponding property and its use will depend upon the value of the accompanying “condition” attribute.

- State

This attribute value defines the status of the variable.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No links types are currently defined for this category.

Guarantee

Instances of this category are used to represent elements comprising the guarantee terms section of the agreement that the obligated party of service has accepted to respect in order to satisfy the conditions of service. Business values will be associated with each guarantee term describing penalties and rewards that may be incurred as a result of the fluctuations of the property compared to the expressed objectives.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the guarantee.

- Description

This attribute provides a human readable description of the guarantee.

- Obligated

This attribute indicates the obligated party as either the "*initiator"* or the "*responder"*.

- Scope

This attribute defines the scope of the guarantee in terms of the service and subsequent component elements.

- Importance

This attribute value describes the level of importance that is attached to the particular guarantee element. This is a relative value compared to other guarantees comprising the same service level agreement.

- Variable

This attribute provides the fully qualified universal unique identifier of the variable category instance that defines the conditions and objectives that are to be guaranteed. The expression will be monitored and controlled and penalties and rewards will occur accordingly.

- Values

This attribute gives the number of business value elements comprising the guarantee. These *business* category instances will describe the rewards and penalties to be applied and incurred as dictated by the fulfilment of the objects described by the preceding *variable* category instance.

- State

This attribute value defines the status of the guarantee.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

Instances of the “business” category are the only link types defined for this category.

Business

Instances of this category are to be used to represent business conditions that may be incurred by the obligated party as a result of conditions arising during the surveillance of the property and its objects described by a particular guarantee element of a service level agreement. Business value elements may be defined as either penalties or reward elements. Penalties will be incurred when the variable property objective evaluates to false. The reward will be applied when and whilst the variable property of the parent guarantee evaluates to true.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the business value element.

- Description

This attribute provides a human readable description of the business value..

- Nature

This attribute defines the nature of the business value as being either a “reward” or a “penalty”. The reward will be activated when and whilst the required objective is attained. The penalty will be incurred when it fails to be met.

- Expression

This attribute provides a Cordscript expression that will be processed to produce the appropriate reward or penalty action.

- Type

This attribute indicates the nature of the expression and would currently be set to “Cordscript”.

- State

This attribute value defines the status of the business value.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for the “business” category.

Penalty

Instances of this category are created as required by the control category instances performing surveillance of the monitoring data stream. Out of band data detected by control category instances, for which penalty business values have defined, will give rise to the periodic production of penalty instances.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute defines the name of the penalty value element.

- Account

This attribute provides the fully qualified universal unique identifier of the account category instance for which the penalty has been incurred.

- Agreement

This attribute provides the fully qualified universal unique identifier of the agreement category instance governing the service contract for which the penalty has been incurred.

- Contract

This attribute provides the fully qualified universal unique identifier of the contract category instance for which the penalty has occurred.

- Control

This attribute provides the fully qualified universal unique identifier of the control category instance by which the penalty was issued.

- Data

This attribute provides the data from the monitoring packet detected by the control to be out of band with respect to the guarantees objectives.

- Sequence

This attribute provides the sequence number of the monitoring data packet from which the data was extracted.

- Timestamp

This attribute provides the time stamp at which the penalty was incurred.

- State

This attribute defines the state of the penalty category instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Removal of an account instance requires the removal of all invoice instances and associated transaction created for the particular account before completion of this operation.

- PUT

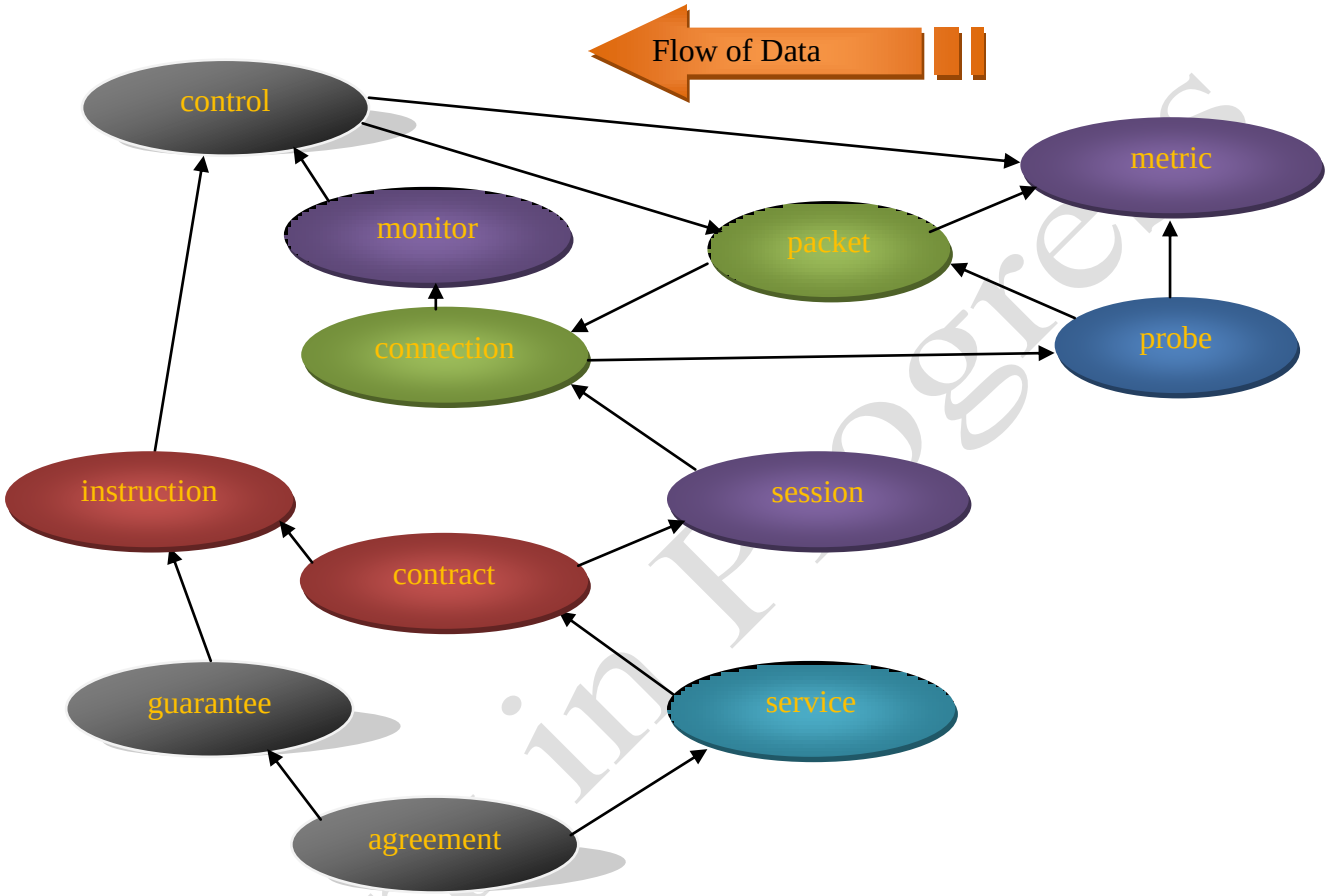
The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for the “penalty” category.

Monitoring Categories

This collection of categories is used by the monitoring and audit system components of the accords platform for the description and control of monitoring activity. This is required not only to meet the technical needs of the platform and the legal and commercial requirements of the operator but also to fulfil the monitoring required by the products of the platform as described in the configuration actions of the user manifest. During the deployment phase of a particular occurrence of service instance brokering a monitoring session will be created for the management of monitoring, if instructions of this nature are detected. Session creation will require the creation of the monitoring connections, each of which will be used to connect a collection of monitoring probes to the required monitoring consumer.



The preceding diagram demonstrates the relationships between the categories of the monitoring system. These categories, their attributes, methods and actions are described below. The monitoring system is fully operational in the form as depicted above.

Session

Instances of the session category are created for a specific contract category instance. The fully qualified universal unique contract instance identifier MUST accompany the request for creation of a monitoring session and will be used to select the collection of monitoring instructions describing the details of the monitoring activity required by the contract.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute may specify a name for the monitoring session.

- Contract

This attribute contains the unique and universal identification string of the conteact category instance for which this session has been created.

- Date

This attribute provides the standard second value time stamp at which the session was started.

- Connections

The total number of connection elements created and appended to the monitoring session.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message. The method will instigate processing of all monitoring instructions described for the service resulting in the creation of the required collection of monitoring streams.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of a session category instance requires removal of the collection of streams that were created for the service initiating the monitoring session.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message for a valid session category instance, will start the monitoring operation such that the data sample packets will be created at the monitoring consumer by the collection of probes and connections described for the system.

- STOP

This action, when invoked through a POST message for a valid session category instance, will suspend the flow of monitoring data from the probes to the connections.

Link Type

Instances of the connection category will be appended to the link list of a monitoring session as they are created between consumer and probe agent endpoints during the preparation of the monitoring system operation for a particular contract.

Connection

Instances of this category will be created during the start-up of a monitoring session and will be appended to the linked list of a monitoring session category instance of the parent's contract category instance. Connections are created for the coordination of the transport of information from a monitoring agent's probes to the defined consumer endpoint. Connections are created at the consumer endpoint by the monitoring agent and will be used to identify the destination for the packetized probe data for delivery to the consumer.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute may specify a name for the monitoring connection.

- Consumer

This attribute contains the unique and universal identification string of the consumer category instance for which this connection has been created.

- Session

This attribute provides the universal unique identifier of the session category instance for which the connection has been created.

- Date

This attribute provides the standard second value time stamp at which the connection was established.

- Monitor

This attribute provides the fully qualified universal identifier of the monitor category instance that will be used for monitoring data packet consumption in place of the default packet discard mechanism. This will be used to perform inspection verification and eventual post processing operations required to satisfy conditions imposed by an eventual service level agreement.

- Process

This attribute indicates the state of the connection packet consumption operation which may be active or inactive. When this process is inactive packet management information will accumulate both at the consumer and the probe end points. When this is active monitoring information corresponding to the connection will be processed as described by the metric and the preceding control attribute. Monitoring information packets and their management data will be discarded as indicated by the result of the operation control category instance.

- Probes

The total number of probe elements created and appended to the monitoring stream.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message. The method will activate and connect delivery of monitoring data from the probes of a monitoring agent to the indicated consumer.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of a stream category instance requires that the collection of probes also be removed since data transport to the consumer is no longer possible.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message for a valid connection category instance, will start the monitoring operation for its linked list of monitoring probes.

- STOP

This action, when invoked through a POST message for a valid connection category instance, will suspend the flow of monitoring data from its linked list of probes.

Link Type

Instances of the probe category will be appended to the connection category instance linked list as they are created by the monitoring session manager for the delivery of monitoring information to the consumer.

Monitor

Instances of this category will be created during the start-up of a monitoring session for a particular connection category instance for the management and control of service level objectives. The *monitor* category instance identifier will be passed during creation to the *connection* category and will be used in place of the standard packet discard mechanisms for controlled consumption of monitoring packet data.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute may specify a name for the monitor category instance.

- Consumer

This attribute contains the unique and universal identification string of the consumer category instance for which this monitor has been created.

- Session

This attribute provides the universal unique identifier of the session category instance for which the monitor has been created.

- Date

This attribute provides the standard second value time stamp at which the connection was established.

- Controls

This attribute provides the count of control category instances managed by the monitor category instance.

Process

This attribute indicates the state of the monitor packet consumption operation which may be active or inactive. When this process is inactive packet management information will accumulate both at the consumer and the probe end points. When this is active monitoring information corresponding to the connection will be processed as described by the metrics and the controls. Monitoring information packets and their management data will be discarded as indicated by the result of the operation control category instance.

- Status

This attribute indicates the current status of the monitor category instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message. The method will activate and connect delivery of monitoring data from the probes of a monitoring agent to the indicated consumer.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of a stream category instance requires that the collection of probes also be removed since data transport to the consumer is no longer possible.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message for a valid monitor category instance, will start the monitoring operation for its linked list of service level objective controls.

- STOP

This action, when invoked through a POST message for a valid monitor category instance, will suspend the flow of monitoring data from its linked list of service level objective controls.

Link Type

Instances of the control category will be appended to the monitor category instance linked list as they are created by the monitoring session manager for the management of service level objectives.

Control

This category describes the monitoring data packet surveillance entities of the Accords Platform. Instances of this category are created during configuration and interface processing upon completion of the service instance graph when brokering under service level agreement confines. Control category instances will be created and for each guarantee term of the service level agreement along with their corresponding monitoring instruction from which they will be referenced. Each control instance will be appended to the monitor category instance created for monitoring connections

during resource provisioning. Controls will be activated by their parent monitor and will process the monitoring data packets produced by their corresponding probe category instance. The comparison of data with objectives may result in the triggering of reward and or penalties as defined by the associated business value descriptions.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the name of a particular control category instance of monitoring packet.

- Contract

This attribute provides the fully qualified unique and universal identifier of the contract category instance for which the control is operating.

- Agreement

This attribute provides the fully qualified unique and universal identifier of the service level agreement governing the activity of the control.

- Property

This attribute provides the name of the property under surveillance.

- Condition

This attribute provides the nature of the compare operation performed by the control between the data received in the monitoring data packet and the targeted objective value of the control. This attribute may take one of the following case insensitive values:

  - Equal or EQ
  - Not Equal or NE
  - Greater or GR
  - Less or LS
  - Not Less or GE
  - Not Greater or LE

- Objective

This attribute provides the objective limit that will be used in conjunction with the preceding condition attribute to detect out of band monitoring data.

- Reference

This attribute provides the fully qualified unique and universal identifier of the guarantee term requiring this control and allowing access to the list of business values defined for the guarantee.

- Importance

This attribute reflects the relative importance of the control.

- Scope

This attribute provides the scope of the control as defined by its initiating guarantee.

- Obligated

This attribute indicates the obligated party as either the initiator or the responder.

- Session

This attribute provides the fully qualified unique and universal identifier of the session category instance on behalf of which the control is operating.

- Account

This attribute provides the fully qualified unique and universal identifier of the account category instance of the service consumer.

- Monitor

This attribute provides the fully qualified unique and universal identifier of the parent monitor category instance.

- Connection

This attribute provides the universal unique identifier of the connection category instance to which the control's parent monitor is attached.

- Metric

This attribute provides the fully qualified unique and universal identifier of the metric category instance describing the discrete operation of the control instance's monitoring probe.

- Probe

This attribute provides the universal unique identifier of the probe category instance from which data packets for this control were sent.

- Process

This attribute provides the process ID of the process spawned by the category instance for background surveillance of the data packet stream. This will be created during the start operation.

- State

This attribute defines the state of the control as either 0: idle, or 1: active.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of a consumer category instance requires that the collection of streams also be removed since data transport to the consumer is no longer possible.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Packet

Instances of this category are created by monitoring probes during the delivery of data for a monitoring session for a contract instance. Each packet category instance will be associated with a connection category instance for the identification of the source of monitoring information.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the name of a particular instance of monitoring packet. This attribute will be used for the selection of the different items of monitoring data delivered by different probes to the same consumer.

- Metric

This attribute provides the fully qualified unique and universal identifier of the metric category instance describing the discrete operation of a monitoring probe.

- Probe

This attribute provides the universal unique identifier of the probe category instance from which the data packet was sent.

- Sequence

This attribute indicates the sequence number of the packet of data sent by the probe. The sequence number is initialised to zero when a probe is first created.

- Connection

This attribute provides the universal unique identifier of the connection category instance for which the data packet is intended.

- Data

This attribute provides the values of the monitoring data as sampled and delivered by the probe.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of a consumer category instance requires that the collection of streams also be removed since data transport to the consumer is no longer possible.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, will activate the surveillance of monitoring data delivered by a monitoring probe.

- STOP

This action, when invoked through a POST message, will suspend the surveillance of monitoring data packets.

Link Type

Instances of the penalty category will be appended to the linked list of the control category instance as they are uncured for the obligated account as a result of out of band data being detected in the monitoring data packet stream.

Probe

Instances of the probe category are created during the creation of a monitoring session for a contract instance. Each probe category instance will be appended to the linked list of a connection category instance and will manage the data flow for a particular metric over that connection to the consumer.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the name of the particular probe category instance and may be used to identify information packets within the stream of monitoring data.

- Period

This attribute indicates the periodicity of the collection of data from the particular monitoring probe. This will be described using combination of a quantity and a metric identifier.

- Expression

This attribute describes the low level monitoring operation to be performed at the periodicity described above.

- Connection

This attribute provides the fully qualified universal unique identifier of the connection category instance over which data packets are intended to be delivered.

- Metric

This attribute provides the fully qualified universal and unique identifier of the metric category instance that describes the nature of the data as to be collected by the particular monitoring probe.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of a probe category instance requires its removal from the parent monitoring streams since data transport to the consumer is no longer possible.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, will activate the collection and delivery of monitoring data by the probe. The delivery of monitoring data will be performed by the creation of packet category instances on the monitoring consumer endpoint.

- STOP

This action, when invoked through a POST message, will suspend the collection and delivery of monitoring data by the probe.

Link Type

Instances of the packet category will be appended to the probe category instance linked list as they are created for the delivery of monitoring information to the consumer.

Metric

Instances of the metric category are used to define the nature and periodicity of the data collected by monitoring probes for delivery to a monitoring consumer. These category instances provide a shared description of the operation and its data format for both producer and consumer.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute provides the name of the particular metric category instance and may be used to identify data format of packets within the stream of monitoring data.

- Period

This attributes the frequency of collection of the information and is expressed in seconds.

- Samples

This attribute controls the maximum number of individual periodic samples that are to be regrouped in one single packet.

- Expression

This attribute defines the data collection expression to be used by the probe for the retrieval of the monitoring data. This may be a script or other system command to be launched for execution. The standard output produced by a single execution of the expression will be used to represent one sample of monitoring data.

- Units

This attribute provides the units in which the collected data will be described. This may be in MHz, MB/S, MB/mS, GB/mS etc whereby upper case factors will be used to represent:

- T: Tera
- G: Giga
- M: Mega
- K : Kilo

Lower case factors will be used to represent:

- d: deci
- c: centi
- m: milli
- µ : micro
- n : nano
- p : pico

The usual SI units name abbreviations are to be used to represent their measures where:

- M: Meters
- S or “: Seconds
- Min or ‘: Minutes
- H:Hours

- Day, Week, Mon, Year
- Deg or °: Degrees
- G: Grams
- Hz: Hertz
- I: Ampères
- V: Volts
- W : Watts
- Q : Coulombs
- P : Pascales
- J : Joules
- Cal : Calories
- N : Newtons
- $: Dollars
- € : Euros
- £ : Pounds
- Etc

The usual combinations are to be accepted and expected such that:

- MHz represents Mega Hertz
- µW represents micro Watts
- KW/H represents Kilo Watts per Hour
- $/Mon represents Dollars per Month

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of a probe category instance requires its removal from the parent monitoring streams since data transport to the consumer is no longer possible.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Alert

Instances of the alert category are created to represent abnormal conditions encountered during the operation of the ACCORDS platform. The processing of alerts by the monitoring control system and will allow corrective action to be taken for the resolution of the problem.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute allows a name to be specified for a particular alert category and may be used to identify the source of the alert in subsequent processing operations.

- Source

This attribute indicates the source of the alert.

- Subject

This attribute provides the subject of the alert and will be used to assist the resolution of the problem initiated by the source of the alert.

- Created

This attribute provides the date and time stamp at which the alert was created.

- Resolved

This attribute provides the data and time stamp at which the alert was resolved or closed.

- Status

This attribute indicates the current status of the alert and will be used in statistical analysis of the monitoring alert management system. Legal values for this field are: ACTIVE, SOLVED, and CLOSED.

- Nature

This attribute indicates the severity of the alert and may take one of the following values: INFORMATION, WARNING, MILD, SEVERE and CRITICAL.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of a probe category instance requires its removal from the parent monitoring streams since data transport to the consumer is no longer possible.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Event

Instances of the event category are created to represent normal request and response actions performed during the operation of the ACCORDS platform. The processing of events by the monitoring control system involves the storage of the information proved in the event for use by external trace management and statistical analysis tools.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Description

This attribute allows a human readable textual description of the event to be specified and associated with the event. This may be used to document statistics drawn from the event management system by external tools.

- Source

This attribute indicates the agent identity of the module or component source of the event.

- Nature

This attribute provides the nature of the event and may be one of the following values:

Message
Events of this type provide information about the operation of the system.
Warning
These events provide warnings of suspect conditions though no formal error is detected.
Failure
These alerts provide an indication of a serious failure of the operational platform of one of its components.
Transaction
These alerts provide financial tracing information for transaction logging tools.

- Created

This attribute provides the date and time stamp at which the event was created.

- Status

This attribute indicates the current status of the event and will be used in statistical analysis of the monitoring event management system.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Link Type

No link types are currently defined for this category.

Image Configuration Categories

The success of an automatic provisioning system, as required to fulfil the needs of cloud brokering, depends largely on the ease with which application images may be produced, configured and personalised during the operation of this system and needs to be a fully automated procedure. This section describes the interface categories made available for the management this operation and are located within the component known as CompatibleOne Software Appliance Configuration Services (COSACS). This component is to be added or embedded in the application image base operating system images and will provide the means for control of and communication with the image during its boot operation. This may be used during the image production and configuration operation or during the Meta Data configuration operations required for personalisation during deployment. For more information concerning the description of these operations please refer to the Cordscript section of this document.

Metadata

Instances of the metadata category are created for each configuration instruction targeting a particular provider specific contract category instance. Each metadata instance will represent a single environment variable expression that is required for the communication of one particular piece of personalisation information such as the IP address of a partner node within a complex multi node provisioned system.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute must be provided to specify the name of the environment variable that will be produced under the control of the metadata category Instance.

- Value

This optional attribute may be present to provide a value to be affected to the resulting environment variable.

- Status

This attribute is updated when the metadata has been consumed by an invocation of the “cosacs:start” method. The particular metadata instance will no longer be scheduled for inclusion for subsequent “cosacs:start” script category operations.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of an invoice category instance does not require removal of the collection of transaction contributing to the invoice since it may be necessary to regenerate an invoice for a collection of transactions.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message. As for the POST method, this method will also perform transaction processing for an existing invoice allowing transactions to be re-evaluated and the invoice totals to be calculated.

Link Type

No link types are currently defined for this category.

Script

Instances of the script category are created for each command invocation instruction targeting a particular provider specific contract category instance. Each script instance will represent a single command invocation expression that is required for the activation of a particular feature of the deployed application image. These categories may be used by image production tools for the preparation of the application images.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute may provide a name for identification of the command invocation operation. One specific name, “cosacs:start”, is of particular importance, and plays an important role in the overall procedure.

- Syntax

For all script instances other than the special case of “cosacs:start”, this attribute value must provide the complete command line expression to be launched on the remote provisioned machine instance.

- Nature

This attribute provides the invocation nature required for this particular command invocation operation. When specified as “system” or “command” it will indicate a synchronous foreground operation. When specified as “fork” or “process” it will indicate an asynchronous background operation.

- Timestamp

This attribute is updated when the script element has been launched and reflects the starting time of the resulting command invocation.

- Status

This attribute is updated when the command for invocation has been performed by an invocation of the “cosacs:start” method. The particular script instance will no longer be scheduled for invocation by subsequent “cosacs:start” operations.

- Result

This attribute value provides the result of the invocation of a”cosacs:start” operation and may return the exit code of a synchronous foreground operation or the process number of an asynchronous background process.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message. This method will be intercepted for further processing when the script instance name attribute evaluates to the “cosacs:start” value. This will give rise to the generation of a script file in the default temporary directory with a “sh” extension. The file will be generated to contain the collection of pending metadata instances and the collection of pending script instances, in that order. The resulting file will be then run as indicated by the nature attribute of the controlling script instance. The result value of the instance will return:

  - the exit value of the script execution for “system” invocation.
  - the process id of the background process for “fork” type invocation.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI invoice category instance identified by the attribute values provided in the OCCI message. Removal of an invoice category instance does not require removal of the collection of transaction contributing to the invoice since it may be necessary to regenerate an invoice for a collection of transactions.

- PUT

  The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message. As for the POST method, this method will also perform transaction processing for an existing invoice allowing transactions to be re-evaluated and the invoice totals to be calculated.

Link Type

No link types are currently defined for this category.

Categories of Provisioning

The contract category instance management system makes use of provider specific provisioning categories for the definition and handling of provisioning and deployment of resources. The following non exhaustive collection of categories presents the current categories of this type, each representing their own particular type of provisioning system.

OpenStack

Instances of this category are created to provide the provisioning interface between the generic contract management PROCCI and the OpenStack provisioning platform. These instances are references through their fully qualified universal unique instance identifier that will be stored in the corresponding contract category instance of the parent service instance. The category defines a basic collection of providing the values needed to manage the provisioning of compute and image resources on the Nova platform and in particular the subscription account profile identifier, the flavor identifier and the image identifier. For further information concerning the OpenStack REST API please consult the corresponding platform documentation [5].

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute allows a name to be associated with the openstack provisioning category instance for identification purposes.

- Node

This attribute value provides the fully qualified unique universal identifier of the node category instance which describes the infrastructure and application image required to be provisioned by this “OpenStack” provider specific contract. The node information will be retrieved and will be used to establish the IMAGE and IMAGE identifiers using the FLAVOR and IMAGE lists of the target provisioning platform.

- Flavor

This attribute value defines the OpenStack IMAGE reference or identifier which describes the infrastructure in terms of processor, memory and disk storage.

- Image

This attribute defines the OpenStack Image reference or identifier which identifiers the actual virtual machine image to be deployed on the provisioned infrastructure.

- Price

This attribute contains the unique and universal identifier of the price category instance that defines the price or cost of provisioning operations of this provider type.

- Account

This attribute contains the unique and universal identifier of the price category instance that is to be used for the production of transactions when the preceding price attribute has been defined.

- Profile

This attribute provides the OpenStack subscription account identifier providing the authorization for the provisioning operations.

- Agent

The value of this attribute indicates the nature of the application post configuration agent that is embedded in the virtual machine image of the provisioned contract.

- Number

This attribute contains the server number of the provisioned resource and will be used for all subsequent server oriented operations after the initial creation.

- Root Password

This attribute contains the administration password as attributed by the OpenStack platform when the server resource was first provisioned. It must be retained since it is impossible to find the value by any other means.

- Reference

This is the universal unique reference of the compute resource on the OpenStack platform.

- Public Address

This attribute will contain the public IP address of the compute resource if one has been attributed.

- Private Address

This attribute will contain the private IP address of the compute resource if one has been attributed.

- Host Name

This attribute will contain either the public IP address or the private IP address in that sequential order.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI openstack category instance identified by the attribute values provided in the OCCI message. Any provisioning that is currently been engaged and controlled by the particular category instance must be stopped and released prior to completion of this operation in order to ensure no orphan provisioning remains.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, providing a valid provider contract instance identifier will start the provisioning and engage the actual resources described and required to fulfil the goals of the node described contract. This operation will be performed through the OpenStack NOVA REST API. As a result of this action the provisioned resources will be active, configured and operational. Information pertaining to the provisioning of the contract will be extracted from the associated instruction set prior to launching the provisioning. Information resulting from the operation will be used to update the configuration instructions upon completion. Monitoring, if defined will have been initiated and if price elements have been defined then transaction events will be generated through the accountancy channels for the corresponding customer and vendor accounts.

- STOP

This action, when invoked through a POST message, providing a valid contract instance identifier will stop the provisioned contract and disengage the resources that have been mobilized. Monitoring will be terminated and financial termination transaction events will be generated as required.

- SAVE

This action, when invoked through a POST message, providing a valid contract instance identifier of an active, started contract instance, will cause the current image of the contract to be saved, or backed up, to create a save point from where the contract instance may eventually be rolled back or restarted.

Link Type

No link types are currently defined for this category.

OpenNebula

Instances of this category are created to provide the provisioning interface between the generic contract management PROCCI and the OpenNebula provisioning platform. These instances are references through their fully qualified universal unique instance identifier that will be stored in the

corresponding contract category instance of the parent service instance. The category defines a basic collection of providing the values needed to manage the provisioning of compute and image resources on the ONE platform and in particular the subscription account profile identifier, the compute identifier and storage image identifier. For further information concerning the OpenNebula REST API please consult the corresponding platform documentation [6].

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute allows a name to be associated with the opennebula provisioning category instance for identification purposes.

- Number

This attribute contains the server number of the provisioned resource and will be used for all subsequent server oriented operations after the initial resource creation.

- Node

This attribute value provides the fully qualified unique universal identifier of the node category instance which describes the infrastructure and application image required to be provisioned by this “opennebula” provider specific contract. The node information will be retrieved and will be used to establish the compute and storage identifiers using the compute and storage lists of the target provisioning platform.

- Agent

The value of this attribute indicates the nature of the application post configuration agent that is embedded in the virtual machine image of the provisioned contract.

- Flavor

This attribute value describes the nature of the compute resource.

- Image

This attribute value identifies the storage item providing the application image to be used to start the compute resource.

- Price

This attribute contains the unique and universal identifier of the price category instance that defines the price or cost of provisioning operations of this provider type.

- Account

This attribute contains the unique and universal identifier of the price category instance that is to be used for the production of transactions when the preceding price attribute has been defined.

- Public Address

This attribute will contain the public IP address attributed to the newly created compute instance provisioned for this contract.

- Private Address

This attribute will contain the public IP address attributed to the newly created compute instance provisioned for this contract.

- Public Network

This attribute provides information describing the public network resource to be used for the creation of the compute resource. This is an optional element and is only required for interconnection with other external nodes.

- Private Network

This attribute provides the description of the private network resource to be used for the creation of the compute resource.

- Profile

This attribute value provides the OpenNebula account identifier for authorization of subsequent compute operations.

- Host Name

This attribute will contain either the public IP address or the private IP address in that sequential order.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Any provisioning that is currently been engaged and

controlled by the particular category instance must be stopped and released prior to completion of this operation in order to ensure no orphan provisioning remains.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, providing a valid contract instance identifier will start the provisioning and engage the actual resources described and required to fulfill the goals of the node described contract. This operation will be performed using the OpenNebula OCCI REST API. As a result of this action the provisioned resources will be active, configured and operational. Information pertaining to the provisioning of the contract will be extracted from the associated instruction set prior to launching the provisioning. Information resulting from the operation will be used to update the configuration instructions upon completion. Monitoring, if defined will have been initiated and if price elements have been defined then transaction events will be generated through the accountancy channels for the corresponding customer and vendor accounts.

- STOP

This action, when invoked through a POST message, providing a valid contract instance identifier will stop the provisioned contract and disengage the resources that have been mobilized. Monitoring will be terminated and financial termination transaction events will be generated as required.

- SAVE

This action, when invoked through a POST message, providing a valid contract instance identifier of an active, started contract instance, will cause the current image of the contract to be saved, or backed up, to create a save point from where the contract instance may eventually be rolled back or restarted.

Link Type

No link types are currently defined for this category.

Proactive

Instances of this category are created to provide the provisioning interface between the generic contract management PROCCI and the ProActive provisioning platform. These instances are references through their fully qualified universal unique instance identifier that will be stored in the corresponding contract category instance of the parent service instance. The category defines a basic collection providing the values needed to manage the provisioning of compute and image resources on the ProActive platform and in particular the subscription account profile identifier, the selection scripts and job characteristics. For further information concerning the ProActive REST API please consult the corresponding platform documentation [8]. This provisioning category is currently being specified and is not yet complete.

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute allows a name to be associated with the proactive category instance for identification purposes.

- Agent

The value of this attribute indicates the nature of the application post configuration agent that is embedded in the virtual machine image of the provisioned contract.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Any provisioning that is currently been engaged and controlled by the particular category instance must be stopped and released prior to completion of this operation in order to ensure no orphan provisioning remains.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, providing a valid contract instance identifier will start the provisioning and engage the actual resources described and required to satisfy the objectives of the node described contract. This operation will be performed using the ProActive REST API. As a result of this action the provisioned resources will be

active, configured and operational. Information pertaining to the provisioning of the contract will be extracted from the associated instruction set prior to launching the provisioning. Information resulting from the operation will be used to update the configuration instructions upon completion. Monitoring, if defined will have been initiated and if price elements have been defined then transaction events will be generated through the accountancy channels for the corresponding customer and vendor accounts.

- STOP

This action, when invoked through a POST message, providing a valid contract instance identifier will stop the provisioned contract and disengage the resources that have been mobilized. Monitoring will be terminated and financial termination transaction events will be generated as required.

- SAVE

This action, when invoked through a POST message, providing a valid contract instance identifier of an active, started contract instance, will cause the current image of the contract to be saved, or backed up, to create a save point from where the contract instance may eventually be rolled back or restarted.

Link Type

No link types are currently defined for this category.

Paas

Instances of this category are created to provide the provisioning interface between the generic contract management PROCCI and generic PAAS provisioning platform interfaces. These instances are references through their fully qualified universal unique instance identifier that will be stored in the corresponding contract category instance of the parent service instance. The category defines a basic collection providing the values needed to manage the provisioning of PAAS components in terms of applications, environments, configurations, versions, instances and also the subscription account profile identifier. For further information concerning the PAAS REST API please consult the corresponding platform documentation [10].

Attributes

- ID

This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

This attribute allows a name to be associated with the PAAS category instance for identification purposes.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute

values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI category instance identified by the attribute values provided in the OCCI message. Any provisioning that is currently been engaged and controlled by the particular category instance must be stopped and released prior to completion of this operation in order to ensure no orphan provisioning remains.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, providing a valid contract instance identifier will start the provisioning and engage the actual resources described and required to fulfil the goals of the node described contract. This operation will be performed using the PAAS REST API. As a result of this action the provisioned resources will be active, configured and operational. Information pertaining to the provisioning of the contract will be extracted from the associated instruction set prior to launching the provisioning. Information resulting from the operation will be used to update the configuration instructions upon completion. Monitoring, if defined will have been initiated and if price elements have been defined then transaction events will be generated through the accountancy channels for the corresponding customer and vendor accounts.

- STOP

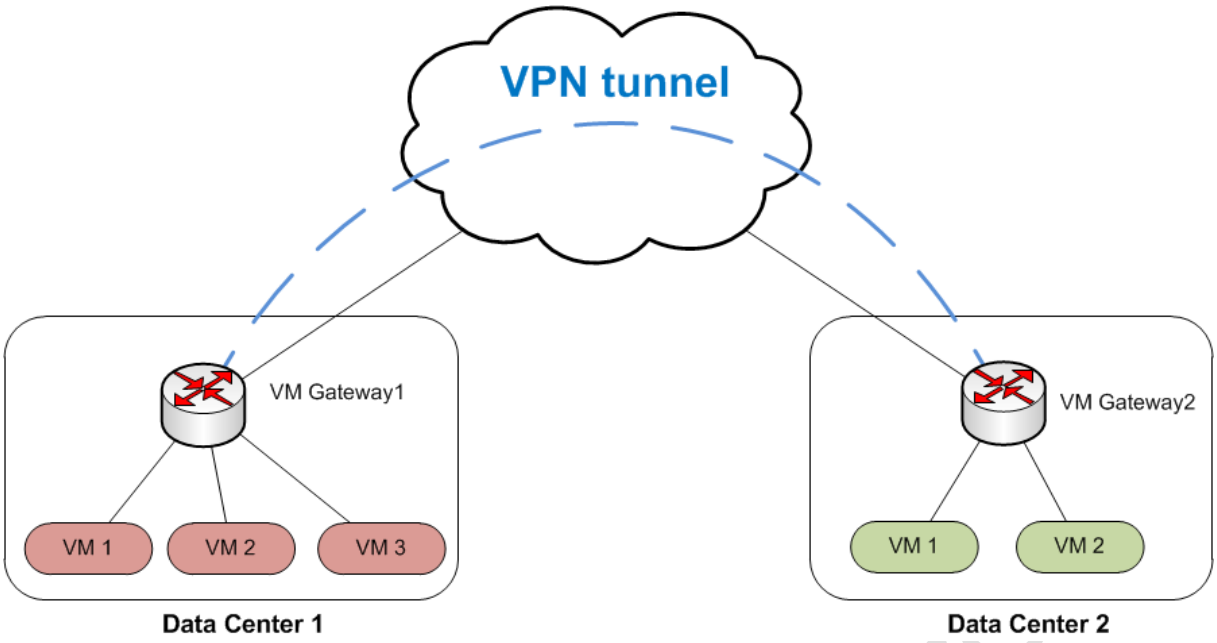
This action, when invoked through a POST message, providing a valid contract instance identifier will stop the provisioned contract and disengage the resources that have been mobilized. Monitoring will be terminated and financial termination transaction events will be generated as required.

Link Type

No link types are currently defined for this category.

IntercloudGW

This category is a provisioning category and is used to specify the name of the provider that will be used to host the manifest described node for an Inter Cloud Gateway. The following diagram shows the relationship between this category and the support categories for the management of the gateway and links.



Attributes

- Id

  This attribute contains the unique and universal identification string of a particular instance of this category. UUID are complex multi field text strings used to reference precisely one instance of a particular category in the CompatibleOne Brokering and Provisioning system.

- Name

  This attribute provides an optional name for identification of the intercloudGW instance.

- Node

  This attribute value provides the fully qualified unique universal identifier of the node category instance which describes the infrastructure and application image that is required to be provisioned by this “intercloudGW” provider specific contract. The node information will be retrieved and will be used to establish the flavor and image identifiers of the VM acting as gateway.

- Account

  This attribute provides the fully qualified universal unique identifier of the account category instance for which the gateway was provisioned.

- Price

  This attribute provides the fully qualified unique universal identifier of the price category instance defined for the particular image category instance. When a value has been defined for this attribute it will provide a global price that will take precedence over eventual discrete price information of component software packages and base operating systems.

- State

This attribute value defines the status of the IntercloudGW instance.

Methods

- POST

Reception of HTTP POST messages by the OCCI Server providing for this category should result in the creation of a new instance of the category being created using the attribute values provided in the OCCI message. The unique universal instance identifier is to be returned in the HTTP Response message.

- GET

The GET method allows retrieval of either the list of category instances identifiers matching the attribute values provided in the OCCI message or if a valid universal unique ID is provided the corresponding instance record will be returned if found to exist.

- DELETE

This method allows deletion of the OCCI contract category instance identified by the attribute values provided in the OCCI message. Removal of a contract category instance requires that all provisioning controlled by through a provider specific contract instance first be stopped and removed, then the linked collection of configuration instruction instances must also be removed.

- PUT

The PUT method allows update of the OCCI category instance identified by the value of the provided universal unique instance identifier using the attribute values provided in the message.

Actions

- START

This action, when invoked through a POST message, providing a valid IntercloudGW instance identifier will start the configuration of the gateway link.

- STOP

This action, when invoked through a POST message, providing a valid IntercloudGW instance identifier will reset the configuration of the gateway link.

Link Type

No link types are currently defined for this category.

Cordscript

This section of the CORDS reference manual describes the syntax of Cordscript, a simple scripting language that is to be used for the description of configuration action expressions. The Cordscript syntax currently provides for the needs of Meta Data handling, Node linkage and monitoring instructions.

Cordscript syntactical elements

Cordscript statements are composed using the following syntactical elements:

Whitespace Characters

Any amount of standard white space characters, SPACE, TABULATION, LINEFEED, CARRIAGE RETURN and FORMFEED may be used and are silently discarded outside of quoted string literals.

Naming Tokens

Naming Tokens may be composed of any number of contiguous, standard, token legal characters. The set of token legal characters is defined as:

- The alphabetical characters from a to z and from A to Z
- The numeric digits from 0 to 9
- The underscore character.

The naming token must start with an alphabetical character or the underscore character.

Punctuation

The SEMI COLON character is defined as the statement termination character.

Operators

The following operators are defined:

The Member Operator

This is the PERIOD character and infers that the entity represented by the token to the right of the operator is a member of the entity represented to the left of the operator where:

A . B

Should be understood to represent the relationship where

B is a member of A

The Parameter Operator

The parameter operator is represented by an opening LEFT BRACE and a closing RIGHT BRACE surrounding a constant literal expression or a member expression.

Constant Literals

Constant literals may be composed of a single or double quoted sequence of characters of any length, or a regular sequence of numeric legal characters. The set of numeric legal characters is:

- Either of the SIGN characters PLUS or MINUS
- The numeric digits 0 to 9
- The decimal separator PERIOD

The numeric string may commence with either sign character or a digit. The decimal separator may be of multiple occurrences and must be both preceded and followed by numeric digits.

Node Access Path

A node access path is used to reference or access methods or properties of service contract nodes and are composed of a sequence of naming tokens separated by member operator characters.

Statements

Cordscript is a statement oriented language and, since each configuration action expression may combine multiple statements, punctuation is required in order to separate the individual statements.

Configuration Statements

Several different types of configuration statement are offered by the current version of Cordscript.

Affectation

The first type of configuration statement allows the affectation of a constant value to named Meta Data property.

Syntax:

```
NodeAccessPath . PropertyName = Constant Value;
```

Examples:

```
Node.category = “test category”;

Node.timeout = 200;
```

Node Linkage

The second type of configuration statement involves the transfer of a Meta Data item produced by a resource provisioning operation to be used by another resource provisioning operation. This is referred to as Node Linkage. Node linkage is performed using the CONFIGURE method which takes a single parameter value.

Syntax:

```
NodeAccessPath . CONFIGURE (  NodeAccessPath . PropertyName );
```

Examples:

```
Parser.Configure( Publisher.HostName );
```

This node linkage statement should be understood to mean that the Contract Node called parser is to be configured to receive the value of the Host Name property of the Contract Node called Publisher. Contract Nodes sourcing property values in this way must have been provisioned prior to execution of the statement for provisioning of the Contract Node of the CONFIGURE method.

Node Linkage Properties

The Node Linkage expressions allow access to and circulation of the following property values:

Hostname

This property returns the public hostname or IP address of a provisioned contract and allows for interconnection of contract nodes.

Contract

This property returns the fully qualified universal unique contract identifier allowing recovery

of the details of a contract in terms of its provision and its node description.

Rerference
This property returns the reference attributed to the provisioned contract by the provider.

Password
This property returns an eventual password attributed to the contract by the provider.

Node Command Invocation
This type of configuration statement requires that the CompatibleOne Software Appliance Configuration Services have been installed in the application image to be deployed and involves the invocation of a script or command line utility within the application image of the thus provisioned resource during the Meta Data transfer phase. This is referred to as Node Command Invocation and is performed using either the SYSTEM or the FORK method. These two methods both take a single string literal parameter value describing the complete command line expression required to be invoked. In the first case the command will be invoked within the same process as the COSACS configuration interface. In the second case a background process will be spawned allowing the immediate return of the interface process.

Syntax:

NodeAccessPath . SYSTEM (“Command Line”);

Or

NodeAccessPath . COMMAND (“Command Line”);

NodeAccessPath . FORK (“Command Line”);

Or

NodeAccessPath . PROCESS (“Command Line”);

The terms “command Line” represent any legal command line syntax that may be performed by the shell command interpreter of the target platform.

Examples:

Parser.System(“co-start”);

Parser.Command(“co-start”);

In these examples the script named “co-start” would be launched directly in the foreground. The invocation process will be suspended until the operation has completed and the exit status of the shell will be returned to the caller.

Parser.Fork(“run-procci”);

Parser.Process(“run-procci”);

In these cases the script named “run-procci” will be launched as a background process and the invocation operation will return the process ID as the result to the caller. In this way multiple parallel operations may be launched without stalling the configuration procedure.

Node Process Destruction Invocation

This type of statement is generally used from within the “release” section of a “manifest” and also requires that the CompatibleOne Software Appliance Configuration Services (COSACS) have been installed in the deployed application image. The statement targets the executing process within the provisioned system as described by the single parameter and will execute a “killall -15” command requesting their graceful termination.

Syntax:

NodeAccessPath . KILL (“ProcessName”);

Examples:

Parser.Kill(“osprocci”);

In this example, taken from the accords platform manifest for the OpenStack Procci, a software termination signal will be sent to the provider specific procci requesting its termination. Accords Platform components will subsequently be able to release their publications prior to completing their shutdown procedure. Due to the inherent delay often encountered in the propagation of signals within a Linux or UNIX based system it is strongly advised to include a temporisation instruction after a termination instruction of this kind in order to allow activity to subside prior to the subsequent release of provisioned resources.

Node Interface Invocation

This type of statement allows invocation of an interface method defined by the manifest description of a complex node. Invocation of these methods will cause all actions, and consequently instructions, defined for the interface to be performed. These instructions may provide Meta Data, invoke commands, influence monitoring or perform any other of the different action types that are defined by Cordscript.

Syntax:

NodeAccessPath . InterfaceName (Parameters);

The term “Node Access Path” must resolve to a valid complex node defined by a manifest and the term “Interface Name” must identify an interface method defined for that manifest. Parameters may be provided in the invocation expression.

Examples:

<manifest name=”test”>

<node name=”one”/>

<node name=”two”/>

<interface name=”example”>

<action name=”useone” expression=”one.command(“start”);”/>

</interface>

</manifest>

<manifest name="user">

<node name="three" type="test"/>

<configuration>

<action name="four" expression="three.useone();"/>

</configuration>

</manifest>

In this example we see that the construction or configuration action of the manifest "user" will make use of the interface method "useone" defined in the manifest type description of the complex node "three".

Node Action Invocation

This group of Cordscript methods allows for the invocation of predefined actions for the service, contract and provisioning OCCI category instances for which they have been defined.

Start

This method allows invocation of the "start" action for the provisioned contract of the indicated node.

Syntax:

NodeAccessPath.START(Parameterlist);

Stop

This method allows invocation of the "stop" action for the provisioned contract of the indicated node.

Syntax:

NodeAccessPath.STOP(Parameterlist);

Save

This method allows invocation of the "save" action for the provisioned contract of the indicated node.

Syntax:

NodeAccessPath.SAVE(Parameterlist);

Snapshot

This method allows invocation of the "snapshot" action for the provisioned contract of the indicated node.

Syntax:

NodeAccessPath.SNAPSHOT(Parameterlist);

Monitoring Statements

Monitoring statements are provided to allow the topology of the monitoring system and its requirements to be expressed and subsequently activated for particular resource provisioning

operations of a manifest. Monitoring information, once activated, flows in a periodic manner, from the producer process to the consumer process. The MONITOR method is used to connect the information source, as defined by the NodeAccessPath of the target contract node, to the information consumer defined by the consumer name. The Metric property name extension defines the nature of the monitoring operation in terms of its periodicity, sampling and syntax.

Syntax:

NodeAccessPath. MONITOR (Consumer . Metric [, Control Instance]);

Examples:

Xwiki.Monitor (coobas.Transactions);

This example should be understood to express the requirement that the monitoring information described by the metric category instance named Transactions and produced by the Xwiki Contract Node is to be delivered to the monitoring information consumer of the COOBAS Contract Node. For further information concerning the operation of the monitoring system and the categories involved please refer to the section describing these categories.

The optional "[Control Instance]" parameter allows a control category instance to be specified for use by the monitoring instruction for the surveillance of the monitoring data packet stream. The fully qualified unique and universal control category instance identifier is to be provided to this effect.

Attribute Statements

The ACCORDS Platform Parser has been extended to make use of the XSD for the validation and control of the parsing operation. In order to allow important operations to be described in the manifest and to be performed during parsing, attribute values detected to contain Cordscript statements will be presented to the Cordscript parser for processing and invocation of the resulting instructions.

A group of new methods has been added to the standard object oriented syntax of Cordscript providing to the needs of post processing during document parsing.

New

This method allows a new OCCI category instance to be created using the name of the attribute as the category name and the list of parameters as the named property values of the newly created instance. The resulting fully qualified category instance identifier will be saved as the value of the attribute.

Syntax:

Attributename.NEW(Parameterlist);

Example

This construction is required for use in the standard CORDS manifest schema for the processing of the following category attributes:

1) The APPLICATION attribute of IMAGE category descriptions.

   This instruction will create a new application category instance, the instance identifier of which will be returned as the value of the APPLICATION attribute of the parent IMAGE element.

The following input manifest node description:

```
<image name="example"/>
```

Will be processed by the Cordscript instructions of the manifest schema node element attribute default value:

```
<xsd:attribute
      name="application"
      type="xsd:string"
      default="Cordscript: application.new(id.value,provider.value);"/>
```

This operation, in this case of usage, takes two parameters. The first is the value of the ID attribute of the parent IMAGE element. The second is the value of the PROVIDER attribute of the ancestor NODE element of the parent IMAGE element. The value of this PROVIDER attribute is said to cascade through to the scope of the APPLICATION attribute expression environment. The result of processing will produce the following output:

```
<image
      name="example"
      application='http://host/application/uuid' />
```

The term http://host/manifest/uuid represents the, fully qualified category instance identifier of the newly created APPLICATION category instance.

Resolve

This method allows resolution of an OCCI category instance identifier using an OCCI property identifier and the required search value. This will return the instance identifier of the first matching category instance. The first parameter must provide a valid category and attribute access path combination. The category name component will be used as the target category for the search operation. The resulting fully qualified category instance identifier will be saved as the value of the attribute.

Syntax:

Attributename.resolve(category.attribute,attribute.value);

Example

This construction is required for use in the standard CORDS manifest schema for the processing of complex node's type attributes. Here the input value of the type attribute may provide a manifest name. The manifest name must be resolved to retrieve the manifest category instance identifier which will be the output value of the attribute. An example will help to clarify this.

The following input manifest node description:

```
<node name="example" type="mysql"/>
```

Will be processed by the Cordscript instructions of the manifest schema node element attribute default value:

```
<xsd:attribute
      name="Cordscript"
      type="xsd:string"
      default="type.resolve(manifest.name,type.value);"/>
```

This operation will produce the following output:

```
<node
```

```
      name="example"
      type='http://host/manifest/uuid' />
```

The value of the TYPE attribute, http://host/manifest/uuid, is used to represent the fully qualified category instance identifier of the MYSQL category instance.

Delete

This method allows deletion of the OCCI instance, defined by the category instance identifier, provided by the value of the named attribute.

Syntax:

Attributename.DELETE();

Example

This construction is not currently employed in the standard CORDS manifest but allows the contrary of the above described NEW instruction. The will result will be the deletion of the category instance identified by the targeted attribute name and value. The attribute value will be reset to the empty string.

Build

This method allows invocation of the "build" action for the OCCI category instance defined by the category instance identifier provided by the value of the named attribute.

Syntax:

Attributename.BUILD(Parameterlist);

Example

This construction is used in the standard CORDS manifest, in conjunction with the NEW instruction, for the invocation of the BUILD action of a newly created APPLICATION category instance.

The IMAGE element description:

```
<image name="example"/>
```

Will be processed by the Cordscript instructions of the manifest schema node element attribute default value:

```
<xsd:attribute
      name="application"
      type="xsd:string"
      default="Cordscript: application.build();"/>
```

Here the BUILD action will be invoked for the APPLICATION category instance identified by the current value of the APPLICATION attribute of the parent IMAGE element.

Instance

This method allows invocation of the "instance" action for the OCCI category instance defined by the category instance identifier provided by the value of the named attribute.

Syntax:

Attributename.INSTANCE(Parameterlist);

Example

This construction is used in the standard CORDS manifest, in conjunction with the NEW instruction, for the invocation of the BUILD action of a newly created APPLICATION category instance.

The IMAGE element description:

```
<image name="example"/>
```

Will be processed by the Cordscript instructions of the manifest schema node element attribute default value:

```
<xsd:attribute
      name="application"
      type="xsd:string"
      default="Cordscript: application.build();"/>
```

Here the BUILD action will be invoked for the APPLICATION category instance identified by the current value of the APPLICATION attribute of the parent IMAGE element.

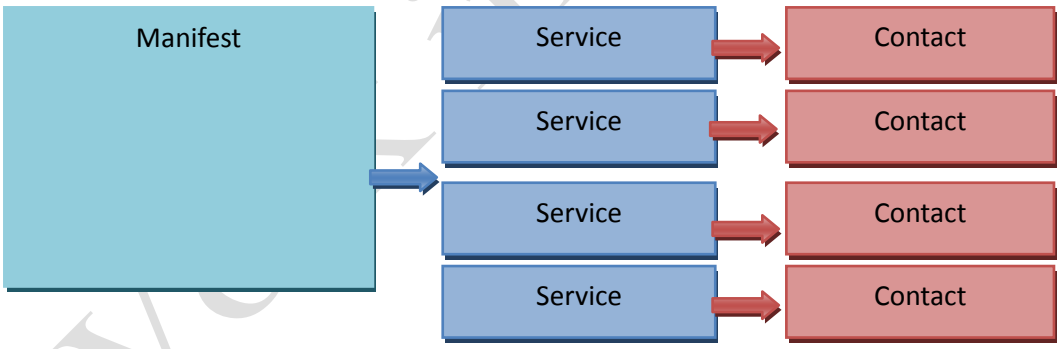
Manifest Node Types

Here we shall show the different node types that are now possible with the version two of CORDS. These different node types are possible by using the various combinations of the different values of the attributes "type", "access" and "scope". Each of the different node types is suitable for a particular purpose and allows for the flexible and extensible description of complex infrastructure provisioning configurations.

Private Normal Simple Node

This is the basic, default, node type. The node gives rise to a provisioned contract that is accessible through the parent service. The details of the node are to be provided by its infrastructure and image elements or retrieved from a previous node definition by its explicit node name attribute.

```
<manifest name= 'prnosi'>
<node name='prnosi' access='private' scope='normal' type='simple'/>
</manifest>
```

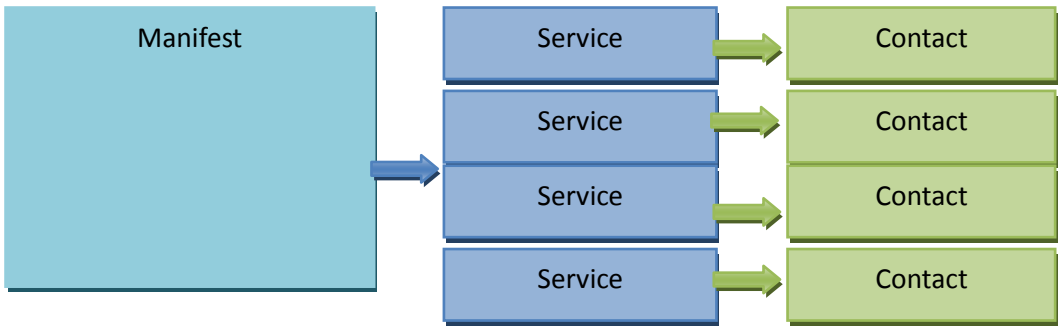


As can be seen in this diagram, the contract, described by its infrastructure and image components, is wholly contained within the service instance which is itself a member of its manifest's provisioning plan.

Public Normal Simple Node

This is similar to the preceding node description except that, being of public access, the node declares to be accessible, that is usable, through complex node expressions of this particular manifest. The details of the node are to be provided by its infrastructure and image elements or retrieved by its explicit node name.

```
<manifest name= 'punosi'>
<node name='punosi' access='public' scope='normal' type='simple'/>
</manifest>
```

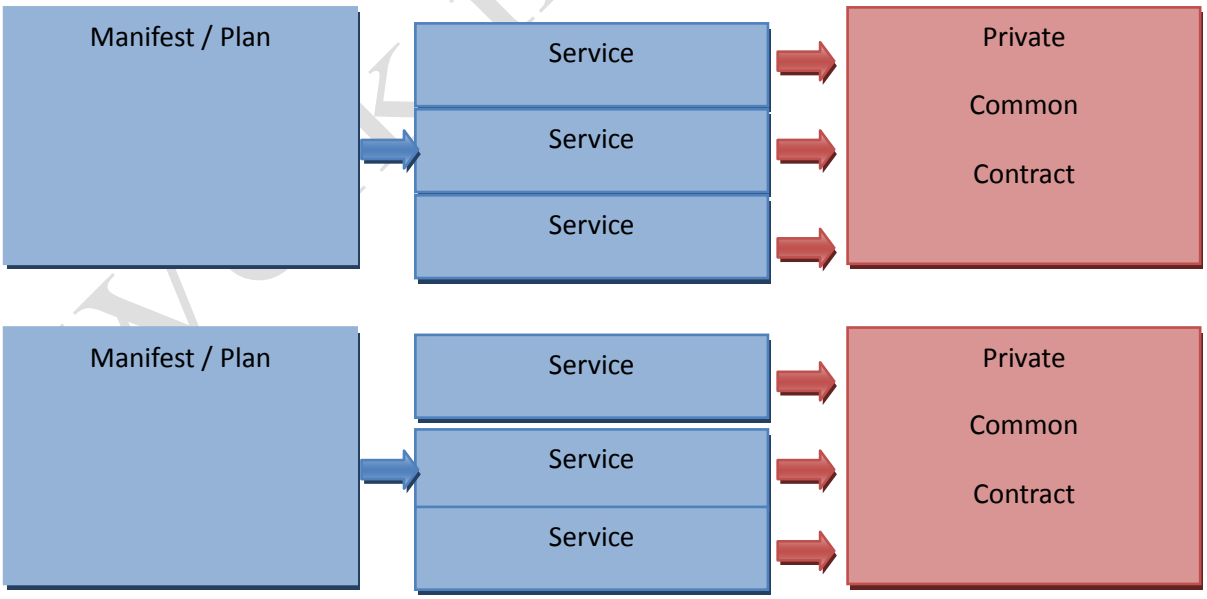


Here it should be understood that the difference between the public and private node types lies in the fact that the contract negotiated for the node is of public access and would consequently be accessible from outside of the confines of its parent service. This would be achieved by the attribution of a public IP address, for example, or by whatever means are required to enable access at a public level. Certain cases may require the configuration of the confining firewall to provide the required access to this type of provisioned resource.

Private Common Simple Node

This node definition declares a simple appliance node, described by its infrastructure and image elements, that is to be shared between all provisioned service instances of the parent manifest and provisioning plan. The provisioning of the node will occur when the first instance of the parent manifest or plan is started. The provisioned resources will be shared between all active instances and will be released when the last instance of the parent manifest is stopped.

```
<manifest name='prcosi'>
<node name='prcosi' access='private' scope='common' type='simple'/>
</manifest>
```



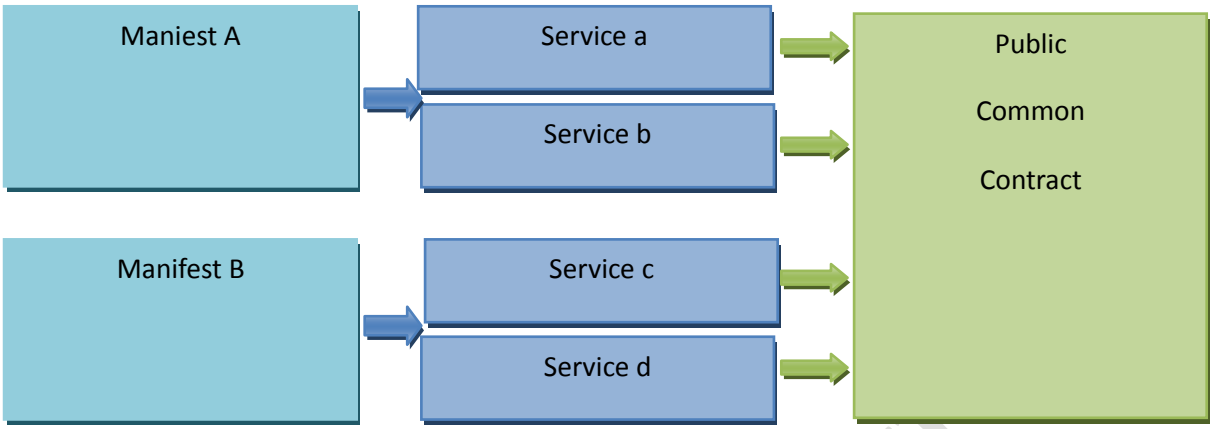
The above diagram shows several service instances of a particular manifest and provisioning plan sharing their private common contract.

Public Common Simple Node

This node definition is similar to the preceding type. The provisioned resources described by this node will be available for use by any service instance of a manifest which declares a public common

node of the same **name**. The resources will be provisioned when the first service instance referencing the public resources is started and will be shared amongst all instances until the final instance is stopped.

```
<manifest name='pucosi'>
<node name='pucosi' access='public' scope='common' type='simple'/>
</manifest>
```



Private Normal Complex Node

This complex typed node description uses a manifest description to provide the details of the resources required for the node. The node may consequently represent a complete sub-system of provisioned resources. The configuration will be instanced for each service instance of the parent manifest.

```
<manifest name='prnoco'>
<node name='prnoco' access='private' scope='normal' type='punosi'/>
</manifest>
```

Public Normal Complex Node

This node definition is similar to the preceding case except that its details are exposed for use from outside of the confines of the manifest described service instance. This declaration exposes and consequently makes available the public components of its nested service for use from outside of service instances of its defining parent manifest or provisioning plan. Consequently the public nodes of the nested service instance will be available for linkage to other nodes in the parent manifest.

```
<manifest name= 'punoco'>
<node name='punoco' access='public' scope='normal' type='punosi'/>
</manifest>
```

Private Common Complex Node

This node defines a complete manifest described system that may be shared and accessed from all service instances of the same parent manifest or provisioning plan. As for the simple common node definitions the nested and shared service instance will be started when the first client service instance is started and will remain active till the last client service instance has been stopped.

```
<manifest name='prcoco'>
<node name='prcoco' access='private' scope='common' type='punosi'/>
</manifest>
```

Public Common Complex Node

This node definition is similar to the preceding description except that the complex manifest described system is instanced for use from all service instances that declare access to the same named, public, common, complex node description.

```
<manifest name='pucoco'>
```

```
<node name='pucoco' access='public' scope='common' type='punosi'/>
</manifest>
```

Examples

This section of this document will show the way in which these categories are to be used in manifest description documents that will be submitted for processing by the ACCORDS parser for the production of a fully qualified provisioning plan. Several different kinds of manifest are possible describing very different provisioning requirements and this series of examples attempts to present the differences in a logical incremental fashion.

CompatibleOne POC XWIKI

This example presents the manifest of the first CompatibleOne Proof of Concept involving the partner XWIKI. This manifest describes a simple configuration comprising two appliance nodes. The first appliance node describes a public internet web application server platform. The second appliance node describes a back end SQL Database engine. Configuration instructions require that the database engine be interconnected for use by the application server. The simplest form of this manifest can be seen below:

```
<?xml version="1.0" encoding="UTF8"?>
<manifest name="pocxwiki" xmlns=http://www.compatibleone.fr/schemes/manifest.xsd>
<node name=”Sqldatabase”/>
<node name=”Xwiki”/>
<configuration>
<action type=”cordscript” expression=”Xwiki.configure(Sqldatabase.hostname);”/>
</configuration>
</manifest>
```

In the above example two named node elements can be seen. This is the simplest form of node definition and relies upon the type named node types having been previously defined within the collection of OCCI Category Managers comprising the parser Support section of the ACCORDS platform. The expression attribute of the configuration action indicates that the application server node, named “Xwiki”, is to be configured to use the host address of the database engine, named “Sqldatabase”.

The node elements above could be expanded, to provide and explicit description of their infrastructure (hardware) and their image (software) needs, as follows:

```
<node name=”Sqldatabase”>
<infrastructure name=”database”>
<compute name=”FastDualCore” cores=”2” speed=”3GHz” architecture=”x686” memory=”8GB”/>
<storage name=”LargeDisk” size=”1TB” type=”SATA”/>
<network name=”EtherNet” vlan=”true” label=”database”/>
</infrastructure>
<image name=”MySQL”>
<system name=”debian squeeze”/>
<package name=”MySql”/>
<package name=”XwikiData”/>
<package name=”XwikiDataPatch”/>
</image>
</node>
```

In the above fragment of manifest description we can see the definition of the node in terms of its infrastructure and image requirements. The infrastructure is defined in terms of its computational, storage and network needs and is explicit about the different parameters of each of these categories. The application image is expressed as being a Linux Debian Squeeze operating system on which has been installed a MySQL database package followed by the Xwiki Database Package and finally a collection of configuration patches required to tailor the standard software to produce the required database engine configuration.

```
<node name=”Xwiki”>
<infrastructure name=”LAMPServer”>
<compute name=”FastQuadCore” cores=”4” speed=”3GHz” architecture=”x686” memory=”32GB”/>
<storage name=”MediumDisk” size=”160GB” type=”SATA”/>
<network name=”EtherNet” vlan=”true” label=”database”/>
</infrastructure>
```

```
<image name=”Xwiki”>
<system name=”debian squeeze”/>
<package name=”Apache”/>
<package name=”Php” version=”5”/>
<package name=”TomCat”/>
<package name=”XwikiApplication”/>
<package name=”XwikiCodePatch”/>
</image>
</node>
```

In the second fragment of manifest description we can see the definition of the second node in terms of its infrastructure and image requirements. The infrastructure is defined in terms of its computational, storage and network needs and is again very explicit about the exact values of the different parameters of each of these categories. The application image is expressed as being a Linux Debian Squeeze operating system on which has been installed an Apache Web Server package followed by an explicit version of PHP, a Tom Cat Server package, the Xwiki Application Code Base and finally a collection of configuration patches required to tailor the standard software to produce the required web application server configuration.

A third fragment of manifest description follows and shows the security and account information to be associated with the nodes and their configuration.

```
<security name=”public” level=”public”/>
<account name=”poc1”>
<user name=”fabio”/>
<user name=”Jamie”/>
<user name=”stephane”/>
<user name=”brian”/>
<user name=”jean”/>
</account>
```

These manifest fragments can now be integrated to replace the simple node descriptions of the initial manifest. A description element is added to document the different aspects for an overall understanding of the purpose of the manifest and all together this gives us a complete manifest as can be seen below:

```
<?xml version="1.0" encoding="UTF8"?>
<manifest name="pocxwiki" xmlns=http://www.compatibleone.fr/schemes/manifest.xsd>

<description>This manifest describes the CompatibleOne XWIKI Proof of Concept, comprising
two appliance nodes, a web application server front end connected to a backe end sql database
engine and whilst allowing public access to service instances of the manifest restricts
administration access to the list of persons described in the account section.</description>

<node name=”Sqldatabase”>
<infrastructure name=”database”>
<compute name=”FastDualCore” cores=”2” speed=”3GHz” architecture=”x686” memory=”8GB”/>
<storage name=”LargeDisk” size=”1TB” type=”SATA”/>
<network name=”EtherNet” vlan=”true” label=”database”/>
</infrastructure>
<image name=”MySQL”>
<system name=”debian squeeze”/>
<package name=”MySql”/>
<package name=”XwikiData”/>
<package name=”XwikiDataPatch”/>
</image>
</node>

<node name=”Xwiki”>
<infrastructure name=”LAMPServer”>
<compute name=”FastQuadCore” cores=”4” speed=”3GHz” architecture=”x686” memory=”32GB”/>
<storage name=”MediumDisk” size=”160GB” type=”SATA”/>
<network name=”EtherNet” vlan=”true” label=”database”/>
</infrastructure>
<image name=”Xwiki”>
<system name=”debian squeeze”/>
<package name=”Apache”/>
<package name=”Php” version=”5”/>
<package name=”TomCat”/>
<package name=”XwikiApplication”/>
<package name=”XwikiCodePatch”/>
</image>
</node>
```

```
<configuration>
<action type=”cordscript” expression=”Xwiki.configure(Sqldatabase.hostname);”/>
</configuration>

<security name=”public” level=”public”/>
<account name=”poc1”>
<user name=”fabio”/>
<user name=”Jamie”/>
<user name=”stephane”/>
<user name=”brian”/>
<user name=”jean”/>
</account>

</manifest>
```

The Basic ACCORDS Platform

This second example presents a study of the ACCORDS platform architecture as a CORDS manifest. This might at first seem to be a chicken and egg situation but in fact is a very useful exercise since it allows us to see in a very succinct fashion the way in which the different components of the system are interconnected and also the reasons for a particular order when they first start up.

In each of the code snippets of this example the detailed description of the infrastructure and application images of the different nodes will be omitted and considered to have been previously defined within the distributed knowledge base managed by the collection of OCCI Categories comprising the Parser Support Services.

Starting with a named manifest element and a documentary description block:

```
<?xml version="1.0" encoding="UTF8"?>
<manifest name="accords" xmlns=http://www.compatibleone.fr/schemes/manifest.xsd>

<description>This manifest describes the CompatibleOne ACCORDS provisioning platform and the
collection of service components of which it is composed.</description>
```

We can now proceed by adding the different service provider node descriptions starting with the root of the ACCORDS platform: the publisher.

```
<node name=”publisher” access=”public” scope=”common”/>
```

The publisher is defined as a public common node and as such will be unique across the entire provisioning platform to be used by all subsequent components of the ACCORDS manifest and eventually any other manifest which describes the node with the same scope and access. We can now add the security service provider component known as COSS. This component will make use of the publisher for the publication of the collection of categories composing the security services interface.

```
<node name=”coss” access=”public” scope=”common”/>
```

The security services component, COSS, is also defined as a public common node to ensure that it is also unique across the entire provisioning platform. As for the publisher it will be used by all subsequent components of the ACCORDS manifest and eventually components of any other manifest descriptions. Now that security is in place we can add the monitoring services component, COMONS. The monitoring services will not only make use of the publisher for category publication but will also make use of the security services to ensure authentication and authorization is performed prior to activation of any monitoring activity.

```
<node name=”comons” access=”public” scope=”common”/>
```

As for the preceding service providers, the monitoring service component COMONS is also a public common service provider. Monitoring services will be required by all subsequent service providers

during the normal course of their operation for monitoring and logging of standard platform activity. We can now define the other components of the system:

```
<node name=”conets” access=”public” scope=”normal”/>
<node name=”cops” access=”public” scope=”normal”/>
<node name=”parser” access=”public” scope=”normal”/>
<node name=”broker” access=”public” scope=”normal”/>
<node name=”procci” access=”public” scope=”normal”/>
```

These service providers do not need to be common since they may easily be of multiple instanced to allow and cater for elasticity of the platform and its operational services. We now have the basic description of the ACCORDS platform and can proceed with configuration of the different nodes so that they can interoperate correctly. We first open the configuration section and connect the security services to make use of the publisher:

```
<configuration name=”accords”>

<action name=”coss” type=”cordscript” expression=”coss.configure(publisher.hostname);”/>
```

Next we must add configuration for the monitoring services to provide access to the publisher. Access to subsequent security services will be made possible by consultation of the publisher in order to localise the category manager for the corresponding service categories.

```
<action name=”comons” type=”cordscript” expression=”comons.configure(publisher.hostname);”/>
```

Now we can configure the remaining service providers to make use of the publication.

```
<action name=”cops” type=”cordscript”
        expression=”cops.configure(publisher.hostname);”/>
<action name=”conets” type=”cordscript”
        expression=”conets.configure(publisher.hostname);”/>
<action name=”parser” type=”cordscript”
        expression=”parser.configure(publisher.hostname);”/>
<action name=”broker” type=”cordscript”
        expression=”broker.configure(publisher.hostname);”/>
<action name=”procci” type=”cordscript”
        expression=”procci.configure(publisher.hostname);”/>
```

And finally we close the configuration section, add security and account information and then close the manifest description:

```
</configuration>
<security name=”public” level=”public”/>
<account name=”public”/>
</manifest>
```

This concludes the description of the basic ACCORDS platform but makes no provision for any form of provisioning service. This is intentional in order to show the way in which the basic manifest can be extended to add multiple and specific provisioning platform services to compliment the standard platform.

ACCORDS Platform with Provisioning

In this example we shall make use of the manifest described in the previous example and build a new manifest with multiple and heterogeneous provisioning services. These extended ACCORDS service providers will offer resource provisioning over the standard ACCORDS platform for an OpenStack Nova platform and an OpenNebula provisioning platform.

Starting with a named manifest element and a documentary description block:

```
<?xml version="1.0" encoding="UTF8"?>
<manifest name=" accords_provisioning "
xmlns=http://www.compatibleone.fr/schemes/manifest.xsd>
```

```
<description>This manifest describes an operation provisioning system built on the basic
accords platform.</description>
```

We can now proceed by adding the reference to the basic ACCORDS platform using a complex node definition and specifying the "accords" manifest name as the value of the type attribute.

```
<node name="accords" access="public" scope="common" type="accords"/>
```

We can now add the two provisioning procci service providers, one for OpenStack, the other for OpenNebula:

```
<node name="osprocci" access="public" scope="common"/>
<node name="onprocci" access="public" scope="common"/>
```

All that remains to perform the required configuration such that the two provisioner interfaces are connected to make use of the underlying publication, security and monitoring services:

```
<configuration name="accords_provisioning">
<action name="osprocci" type="cordscript"
       expression="osprocci.configure(accords.publisher.hostname);"/>
<action name="onprocci" type="cordscript"
        expression="onprocci.configure(accords.publisher.hostname);"/>
</configuration>
```

Addition of the security information and the account description clause, followed by closure of the manifest definition completes this example of use of complex node definitions:

```
<security name="public" level="public"/>
<account name="public"/>
</manifest>
```

In the above example it should be noted the use of the complex node name as a prefix to access the host name properties of the publication, security and monitoring service provider components of the basic accords platform. This compound access path technique may only be performed if the public access has been defined to components of the complex node definitions involved.

Load balanced Provisioning

One of the fundamental features of a cloud platform is that of elasticity. The approach adopted by the Accords Platform for management of elasticity involves the use of load balancing nodes. In the following manifest example we consider a service node requiring elastic scalability. The subject of the manifest will be accompanied by a special node that provides the required functionality and will perform contract oriented provisioning as required to satisfy the required availability of service.

```
<?xml version="1.0" encoding="UTF8"?>
<manifest name="elastic_example" xmlns=http://www.compatibleone.fr/schemes/manifest.xsd>
<node name="http:load-balancer" scope=normal access=public type=simple/>
<node name="elastic-service" scope=normal type=simple/>
<configuration>
<action expression="http:load-balancer.system('wget http://www.compatibleone.fr/accords-
platform/system/run-cool');/>
<action expression="http:load-balancer.set(elastic_ceiling,5);"/>
<action expression="http:load-balancer.configure(elastic-service.contract);"/>
<action expression="http:load-balancer.fork('bash run-cool {port}');"/>
</configuration>
<security name="public" level="public"/>
<account name="public"/>
</manifest>
```

The detail of the two nodes has been omitted, for clarity of the example, since the importance lies in the configuration section. The first action expression retrieves the current version of the COOL software installer from the depot. The third action expression configures the "http:load-balancer" node providing the contract identifier of the service package that is required to be load balanced by the system. This contract identifier allows the load-balancer to retrieve the node description and consequently permits subsequent provisioning to be performed as described by the configuration of

the load balancing node, an example of which can be seen in the second action expression which sets the upper ceiling of the degree of elasticity to be achieved.

It is very important to remember that load balancing is not a generic process and load balancers specific to each communication protocol type are required. A generic HTTP load balancing component is however provided in the Accords Platform package in the form of the "COOL" component. This component is made available with a primary aim of providing load balancing services for the operation of the components of the Accords Platform itself. The description of the platform and provisioning, as described in the examples above, would be extended such that each component would in fact be a complex node described by its own individual manifest. Each manifest would define three nodes:

- The common publisher node
- The service component
- The load balancer.

These nodes would be interconnected as shown in both the platform and load balancing examples whereby the service package would be configured to use the common publisher and the load balancer would be configured to use the service package. The public address of the load balancer would be used to define the public identity of the service component. In this way all requests for category management service would be received and redirected by the load balancer allowing for total scalability of the resulting platform as required to meet both real and fluctuating operational requirements.

A PAAS System Manifest

This example shows, without going into the precise details of PAAS specific node definitions, how a PAAS may be launched on an ACCORDS platform and the way in which it makes itself and its services available to the host platform. As in the preceding example we need to start the manifest with the description of access to the ACCORDS platform through the complex Accords Publisher node:

```
<?xml version="1.0" encoding="UTF8"?>
<manifest name="mypaas" xmlns=http://www.compatibleone.fr/schemes/manifest.xsd>

<description>This manifest describes an example of how a PAAS can offer its services through
the ACCORDS platform.</description>

<node name=”accords:publisher” access=”public” scope=”common” type=”accords-publisher”/>
```

With the addition of a PAAS service package node to perform the required service:

```
<node name=”mypaas” access=”public” scope=”common”/>
```

We can now add the PAAS specific interface node description which will connect to the ACCORDS platform publisher and act as a service provider PROCCI interface for the PAAS platform:

```
<node name=”paasprocci” access=”public” scope=”normal”/>
```

And the configuration instructions that connect the PAAS interface not only to the ACCORDS platform components but also to the PAAS service package:

```
<configuration>
<action name=”paasprocci” type=”cordscript”
        expression=”
              paasprocci.configure(mypaas.hostname);
              paasprocci.configure(accords:publisher.publisher.hostname);
              ”/>
</configuration>
```

In the preceding configuration section attention is drawn to the connection of the “paasprocci” node to make use of the “mypaas” node through its host name or IP address. In this way the “paasprocci” node can publish its offer of service through the “accords” publisher, respecting security and monitoring policies in order to receive requests for service from the clients of the host platform.

Addition of the security information and the account description clause, followed by closure of the manifest definition completes this example of use of complex node definitions:

```
<security name=”public” level=”public”/>
<account name=”public”/>
</manifest>
```

PAAS Product Description Manifests

The collection categories defined in the preceding section entitled “Categories for PAAS Provisioning” are to be used to define PAAS manifests to be used as the descriptive type of subsequent complex PAAS nodes in standard manifests. In this example we describe such an application manifest for the provisioning of PAAS service through the generic COPAAS and PAASPROCCI components of the Accords Platform.

```
<paas_manifest name=”paas_example” xmlns=http://www.compatibleone.fr/schemes/copaas.xsd>
        <paas_application name=”paas:example”>
                <paas_verson name=”paas:example”>
                        <paas_version_instance name=”example” uri=”urn:etc”>
                                <paas_deployable name=”example” content_type=”container”/>
                        </paas_version_instance>
                </paas_version>
                <paas_environment name=”example” provider=”CF” uri=”urn:etc”/>
        </paas_application>
</paas_manifest>
```

An XML document containing this description may be parsed using the standard Accords Parser for introduction of the application definition into the PAAS description management system provided by the COPAAS component of the Accords Platform. Different PAAS platforms and products may require the development of platform specific manifests and PROCCI interface components. The underlying platform tools, the parser, broker and contract management system will afford seamless integration to these future concepts by the simple adjunction of the appropriate schema definition, its category support service and its deployment PROCCI interface.

PAAS Product Deployment Manifests

To complement the preceding example, where we show the way in which a PAAS system can integrate with the ACCORDS platform to receive requests for service through the publisher of the platform, here we shall see the way in which PAAS products can be designed and made available through the same host platform.

As for the previous examples we start with a complex node description but this time it is the specific PAAS manifest that is used to describe the type:

```
<?xml version="1.0" encoding="UTF8"?>
<manifest name="mypaas_product" xmlns=http://www.compatibleone.fr/schemes/manifest.xsd>

<description>This manifest describes an example of how a PAAS platform can formulate manifests
to describe service products made available through the ACCORDS platform.</description>

<node name=”platform” access=”public” scope=”common” type=”mypaas”/>
```

With the addition of a second node we specify an explicit provider type as that of the provisioning type published by the PAAS platform PROCCI interface, for example in this case “mypaas”:

```
<node name=”one” access=”public” scope=”normal” category=”paas_manifest” type=”paas_example”
provider=”mypaas”/>

<configuration/>
<security name=”public” level=”public”/>
<account name=”mypaasuser”/>
</manifest>
```

No particular configuration is required. The manifest is completed as usual with the security and account information and final closure element. Manifests of this kind may be delivered with the PAAS platform application images and may be processed during start-up of the platform to make available the provisioning plans resulting from parsing of the manifest collection by the ACCORDS platform parser. In this way the PAAS platform is able to publish its offers of service for use by client processes and customers of the host platform.

PAAS Resource Deployment Manifests

An Accords Platform “enabled” PAAS platform would ideally be able to make use of the hosting platform for the brokering of resources as required for satisfaction of the received requests for the provision of its own added value PAAS services. A PAAS platform would deliver application instance services to customers. These application services, in the case of an Accords Platform enabled PAAS, would be delivered using infrastructure provisioned by the Accords Platform. To this effect the PAAS would describe these resources using standard manifests that would be parsed and made available during the start-up of the platform. An Accords enabled PAAS would include integrated components allowing communication with the parent broker respecting the security, authentication, authorization and monitoring requirements in effect on the host.

Inter-cloud Gateway Provisioning Manifest

This example shows the way in which the CONETS PROCCI may be used in application manifests for the interconnection of nodes through an inter-cloud gateway. This type of manifest requires the adjunction of a node description that specifies the “intercloudGW” category as the provider type.

```
<node name="conets:service" provider="intercloudGW">
      <infrastructure name="conets:model"/>
      <image name="conets:model"/>
</node>
```

When used in conjunction with the standard POC1V1 manifest nodes describing the MYSSQL database and the XWIKI application servers:

```
<node name="mysql" type="simple" access="public" scope="normal"
                  provider="openstack">
      <infrastructure name="mysql">
            <compute name="mysql" architecture="x86_64" cores="1"
                                                memory="1G" speed="1G"/>
            <storage name="mysql" size="10G"/>
            <network name="mysql_network" label="ethernet" vlan="100M"/>
      </infrastructure>
      <image name="mysql">
            <system name="ubuntu_image"/>
            <package name="mysql1"/>
      </image>
</node>
<node name="xwiki" type="simple" access="private" scope="normal"
                  provider="opennebula" >
      <infrastructure name="xwiki">
            <compute name="xwiki" architecture="x86_64" cores="1"
                                                memory="1G" speed="1G"/>
            <storage name="xwiki" size="10G"/>
            <network name="xwiki_network" label="ethernet" vlan="100M"/>
      </infrastructure>
      <image name="xwiki">
            <system name="ubuntu_image"/>
```

```
            </image>
        </node>
```

And with the appropriate configuration section performing the interconnection of the application nodes through the gateway node:

```
        <configuration name="conets_xwiki">
            <action expression="conets:service.configure(mysql.contract);"/>
            <action expression="conets:service.configure(xwiki.contract);"/>
            <action expression="conets_xwiki.configure(mysql.hostname);"/>
            <action expression="conets_xwiki.fork('bash /root/xwiki.sh');"/>
        </configuration>
```

They addition of the customary packaging instructions gives a complete and operational manifest as follows:

```
<?xml version="1.0" encoding="UTF8"?>
<manifest name="conets_xwiki" xmlns="http://www.compatibleone.fr/schemes/manifest.xsd">
        <node name="mysql" type="simple" access="public" scope="normal"
                        provider="openstack">
                <infrastructure name="mysql">
                        <compute name="mysql" architecture="x86_64" cores="1"
                                                        memory="1G" speed="1G"/>
                        <storage name="mysql" size="10G"/>
                        <network name="mysql_network" label="ethernet" vlan="100M"/>
                </infrastructure>
                <image name="mysql">
                        <system name="ubuntu_image"/>
                        <package name="mysql1"/>
                </image>
        </node>
        <node name="xwiki" type="simple" access="private" scope="normal"
                        provider="opennebula" >
                <infrastructure name="xwiki">
                        <compute name="xwiki" architecture="x86_64" cores="1"
                                                        memory="1G" speed="1G"/>
                        <storage name="xwiki" size="10G"/>
                        <network name="xwiki_network" label="ethernet" vlan="100M"/>
                </infrastructure>
                <image name="xwiki">
                        <system name="ubuntu_image"/>
                </image>
        </node>
        <node name="conets:service" provider="intercloudGW">
                <infrastructure name="conets:model"/>
                <image name="conets:model"/>
        </node>
        <configuration name="conets_xwiki">
                <action expression="conets:service.configure(mysql.contract);"/>
                <action expression="conets:service.configure(xwiki.contract);"/>
                <action expression="conets_xwiki.configure(mysql.hostname);"/>
                <action expression="conets_xwiki.fork('bash /root/xwiki.sh');"/>
        </configuration>
        <interface name="conets_xwiki"/>
        <account name="test"/>
        <security name="conets_xwiki"/>
</manifest>
```

The parsing and brokering of this manifest, and the resulting provisioning plan, will result in the provisioning of four virtual machines. Two of these will be deployed within an OpenStack environment, the MYSQL database server node and the gateway node for the OpenStack cloud. The other two will be deployed within an OpenNebula environment, the XWIKI application server node and the gateway node for the OpenNebula cloud. The gateway nodes will be attributed public IP Addresses and will be connected to the local IP addresses of their corresponding workload machines.

Subsequent brokering of the same provisioning plan would result in the provisioning of only two virtual machines representing the database node and the application server node. The two gateway nodes would be reused for the interconnection of the newly provisioned database and application

nodes. In this way public IP consumption is greatly reduced and security between clouds can clearly be enforced

A Customer Service Level Agreement

The preceding series of examples demonstrate the way in which the different technical descriptions of resources and provisioning can be accomplished using CORDS. In this example we shall now see how a service level agreement can be used to control the provisioning operation of such a manifest described system. Service Level Agreements are contractual documents that are to have been negotiated between the two involved parties. In this case the customer, the initiator of the agreement, will assume the role of service consumer with the Accords Platform operator, the responder, in the role of service provider. The agreement may also stipulate a period of validity in terms of the initiation and termination date and time. The following code snippet shows the agreement clause between an example customer and the accords platform operator.

```
<agreement name=”example”
        xmlns=http://www.compatibleone.fr/schemes/slam.xsd
        initiator=”example” responder=”accords”
        provider=”responder”
        initiation=”now” expiration=”never”
        template=”none”>
</agreement>
```

The initiator and responder attributes defined here provide the names of the accounts of the corresponding parties. These values will be processed during the parsing operation in order to resolve the fully qualified unique and universal identifiers of the corresponding account instances within the financial management services of the Accords Platform.
The details of the agreement are expressed in three particular sections of the document known as service description, service conditions and service guarantees. Each of these sections will be composed of a single terms element regrouping the collection of term elements that are necessary for the description of their content. The following code snippet shows an example service description terms element referencing the standard XWIKI manifest:

```
<terms name=”:service” type=”services”>
        <term name=”:service”>
                <manifest name=”xwiki”/>
        </term>
</terms>
```

In the above example the manifest element names a manifest category instance that has already been parsed and introduced into the Accords Platform system. The full manifest description could also be expressed in detail at this point of the service level agreement but it is better practice to separate the two document descriptions as shown above.

The service conditions section contains information that will assist the placement engine for the selection of the appropriate technical and commercial provisioning platform interface. The information provided by these term elements takes the form of variable elements defining a property name and a value.

```
<terms name=”:conditions” type=”conditions”>
        <term>
                <variable property=”occi.placement.algorithm” value=”zone”/>
                <variable property=”occi.placement.zone” value=”Europe”/>
                <variable property=”occi.placement.provider” value=”openstack”/>
        </term>
</terms>
```

The first of the three variable expressions defines the algorithm to be used by the placement engine as being zone oriented. The second provides the definition of the value of the zone or region required for the placement of the resources and finally the third variable expression indicates the default technical provisioning type to be used for node descriptions that do not impose any particular provisioning technology type.

The following snippet of code shows how guarantees can be defined for the control of dynamic operating conditions:

```
<terms name=":guarantees" type="guarantees">
	<term>
		<guarantee name=":g1">
			<variable property="cpu:usage" condition="greater" value="50%"/>
			<business type="penalty" expression="cordscript:alert();"/>
		</guarantee>
		<guarantee name=":g2">
			<variable property="ram:available" condition="less" value="50%"/>
			<business type="penalty" expression="cordscript:new alert();"/>
		</guarantee>
	</term>
</terms>
```

Here we can see two guarantee expressions. The first will raise an alert when the control of monitoring information detects that overall CPU activity has exceeded 50%. The second expression will raise an alert when available RAM drops below 50%. In both cases a penalty category instance will be created identifying the incident for use by an encapsulating business process.

These snippets of code are to be combined together and would produce the complete customer service level agreement document that can be seen below:

```
<agreement name="example"
	xmlns=http://www.compatibleone.fr/schemes/slam.xsd
	initiator="example" responder="accords"
	provider="responder"
	initiation="now" expiration="never"
	template="none">
<terms name=":service" type="services">
	<term name=":service">
		<manifest name="xwiki"/>
	</term>
</terms>
<terms name=":conditions" type="conditions">
	<term>
		<variable property="occi.placement.algorithm" value="zone"/>
		<variable property="occi.placement.zone" value="Europe"/>
		<variable property="occi.placement.provider" value="openstack"/>
	</term>
</terms>
<terms name=":guarantees" type="guarantees">
	<term>
		<guarantee name=":g1">
			<variable property="cpu:usage" condition="greater" value="50%"/>
			<business type="penalty" expression="cordscript:alert();"/>
		</guarantee>
		<guarantee name=":g2">
			<variable property="ram:available" condition="less" value="50%"/>
			<business type="penalty" expression="cordscript:new alert();"/>
		</guarantee>
	</term>
</terms>
</agreement>
```

This type of document is to be parsed using the standard Accords Platform Parser tools and may subsequently be used to control the deployment of the service described by the manifest.

A Provider Service Level Agreement

The counterpart to the customer service level agreement is the provider service level agreement and represents the contract established between the accords platform operator and a particular service provisioning operator. The service level agreement will be used to describe and control the provisioning of service through the corresponding provider specific PROCCI interface.
In this case the initiator of the agreement will be the provider, or operator of the provisioning service, and the responder will be the operator of the Accords Platform. The provider of service will consequently be the initiator. The duration of the agreement and eventual template description may also be specified. This can be seen in the following code sample:

```
<agreement name=”example”
        xmlns=http://www.compatibleone.fr/schemes/slam.xsd
        initiator=”example” responder=”accords”
        provider=”initiator”
        initiation=”now” expiration=”never”
        template=”none”>
</agreement>
```

The service description section is different to the preceding customer service level agreement in that it is not based on a manifest description of a technical configuration. Instead it is based on a provider description document defining the nature of the provider and the various quota records that are made available to the Accords Platform for use by the placement engine for the provisioning of contracts and service. The following code snippet shows an example of this:

```
<terms name=”:service” type=”services”>
        <term name=”:s1”>
                <provider
                        name=”example”
                        xmlns=”http://www.compatibleone.fr/schemes/provision.xsd”>
                        <quota property=”occi.compute.cores” value=”50”/>
                        <quota property=”occi.compute.memory” value=”500”/>
                        <quota property=”occi.storage.size” value=”5000” granularity=”10”/>
                        <quota property=”accords.vm.count” value=”50”/>
                        <quota property=”accords.address.count” value=”50”/>
                </provider>
        </term>
</terms>
```

In the above service description we can see five quota definitions describing the provisioning components that are made available by the provider to the Accords Platform. These quota definitions are the number of virtual machine instances, of public IP addresses, of virtual CPU cores, the total quantity of RAM and the total available disk space. The resources thus described are to be used as required by the placement engine in any combinations respecting the specified or default granularity of each quota type.
The inline inclusion of the provider description, shown here, is preferred here due to the one to one relationship between a provider SLA and its provider quota descriptions. These elements may also be provided in separate documents and parsed individually as shown for the example of the customer service level agreement in the preceding example.

The service conditions section of a provider service level agreement is currently of no consequence but will eventually be used to describe the various geographical and other placement alternatives exposed by a particular provider.

The service guarantees section will soon be developed to allow business values to be attached to the various quota definitions and consequently describing the rewards and penalties to be incurred during dynamic operational conditions as a result of the respect or violation of these deemed important terms of the agreement.

The resulting compounded example of this provider service level agreement is shown in the code below:

```
 <agreement name=”example”
        xmlns=http://www.compatibleone.fr/schemes/slam.xsd
        initiator=”example” responder=”accords”
        provider=”initiator”
        initiation=”now” expiration=”never”
        template=”none”>
<terms name=”:service” type=”services”>
        <term name=”:s1”>
                <provider
                        name=”example”
                        xmlns=”http://www.compatibleone.fr/schemes/provision.xsd”>
                        <quota property=”occi.compute.cores” value=”50”/>
                        <quota property=”occi.compute.memory” value=”500”/>
                        <quota property=”occi.storage.size” value=”5000” granularity=”10”/>
                        <quota property=”accords.vm.count” value=”50”/>
                        <quota property=”accords.address.count” value=”50”/>
```

```
            </provider>
        </term>
</terms>
</agreement>
```

The parsing of a provider specific service level agreement must be performed prior to the activation of the provider PROCCI interface to which it applies. During start-up, a provider PROCCI will localise the account of the operator, identified by the value of the appropriate configuration value, in order to locate the service level agreement currently in effect for this operator's account. This is currently a relaxed feature but will become a mandatory requirement for operation of a PROCCI interface by the next version of the Accords Platform.

References

[1] CORDS Manifest Schema XSD Specification

http://www.compatibleone.fr/occi/publisher/manifest.xsd

[2] CORDS Manifest Types Schema XSD Specification

http://www.compatibleone.fr/occi/publisher/cordstypes.xsd

[3] CORDS Import Schema XSD Specification

http://www.compatibleone.fr/occi/publisher/cords.xsd

[4] CORDS SLA Schema XSD Specification

http://www.compatibleone.fr/occi/publisher/slam.xsd

[5] Accords Parser Documentation

http://www.compatibleone.fr/occi/publisher/AccordsParserV1.02.pdf

[6] OCCI Core Model Specification

http://www.ogf.org/documents/GFD.183.pdf

[7] OCCI Cloud Infrastructure Specification

http://www.ogf.org/documents/GFD.184.pdf

[8] OCCI HTTP Binding Specification

http://www.ogf.org/documents/GFD.185.pdf

[9] OpenStack Nova REST API

http://docs.openstack.org/api/openstack-compute/1.1/content/index.html

[10] OpenNebula OCCI REST API

http://opennebula.org/documentation:archives:rel2.2:occiug

[11] Windows Azure REST API

http://msdn.microsoft.com/en-us/library/windowsazure/ee460799.aspx

[12] PROACTIVE REST API

http://proactive.inria.fr/index.php?page=release_notes&action=displaymin&productid=14&majmin=1.0

[13] SLAPOS REST API

http://packages.python.org/slapos.core/rest.html

[14] PROVIDER SLA SCHEMA

http://www.compatibleone.fr/schemes/provision.xsd

Document Change Log

Version 2.12

- Addition of the Gateway, IntercloudGW and LinkGW category descriptions.
- Addition of Inter-Cloud Gateway example.
- Addition of the Close and Process actions to the invoice category.
- Addition of the Close action to the transaction category.
- Detailed example of a Customer Service Level Agreement
- Detailed example of a Provider Service Level Agreement.
- Addition of the “quantity” category description used in the placement operation in conjunction with quota records defined for providers through their service level agreements.
- Addition of the description of the PAAS Categories.
- Addition of the PAAS provisioning category.
- Addition of the “category” attribute to the “node” category. This is to be used in conjunction with the “type” and “provider” attributes to allow the generic description of complex nodes.
- Addition of the initiation and expiration attribute to the service category to allow temporary constraints to be applied and controlled.

Version 2.11

- The section describing the service level agreement categories has been moved to just before monitoring categories in order to facilitate comprehension.
- Several editorial changes have been performed as a result of the proof reading of the preceding version.
- Added texts and explanations for the cloud service agreement categories.
- Addition of the link to the Accords Parser documentation.

Version 2.10

- Addition of the description of the *monitor*, *control* and *penalty* categories for service level objective surveillance.
- Completion of the description the service level agreement *guarantee* and *business* categories that provide the source of implicit monitoring conditions and subsequent business level guarantees, rewards and penalties.
- Addition of the “control” and “process” attributes to the monitoring “connection” category. These attributes describe and manage the behavior of the monitoring packet consumption process.

- Addition of the “agent” attribute to the “openstack”, “opennebula”, “windowsazure” and “proactive” provisioning categories.
- Description of the placement category life cycle management actions; CHOOSE, CONSUME, RESTORE and RELEASE.
- Addition of the “placement” attribute to the “contract” category allowing management of the placement quota during the contract life cycle.
- Addition of the “agent” attribute to the “image” category for fine control over COSACS or other application image interfaces.
- Major rework of the monitoring section to reflect the changes now implemented in the CORDS model and the ACCORDS platform.
- Extension of the business category to handle supervision of monitoring through service level agreements.
- Addition of the “release” category element for the description of the destruction methods of manifests.
- Addition of the Load Balancing, Scalability and Elasticity example manifest.
- Addition of the Cordscript “Kill” instruction type allowing for orderly software termination.
- Addition of the list of properties for Node Linkage expressions in the Cordscript reference section.
- Addition of the Node.Contract element to the Cordscript.

Version 2.09

- Addition of the section describing the complete collection of categories for service level agreements and their management.
- Addition of the “workload” attribute to the “contract” category.
- Addition of the “url” attribute to the “application” category for use by the “vm” category.
- Addition of “start” and “stop” actions to the “probe” category.
- Addition of the attributes “zone”, “opinion” and “security” to the “provider” category.

Version 2.08

- Addition of “*Node Action Invocation*” to the Cordscript documentation.
- Addition of “*Attribute Statements*” to the Cordscript documentation.
- The default XSD schema document is now the “manifest.xsd”.
- Addition of the ROLE attribute to the “user” category.
- Addition of the “algorithm” category.
- Addition of the “agreement” category.

Version 2.07

- Addition of this section to the document in order to provide a detailed description of the modifications and changes made to the different versions.
- Addition of the “job” category for the management of workload scheduling by the CompatibleOne Scheduling Services.
- Addition of the “application” category for the management of application images and their production.

Version 2.06

- Addition of the new categories required to cater to the needs encountered in work under way on the Network and Placement services. These new categories are:
  - ❖ Quota
  - ❖ Port
  - ❖ Firewall
- Addition of the new port category links to the network and package categories for the management of the description of their interface.