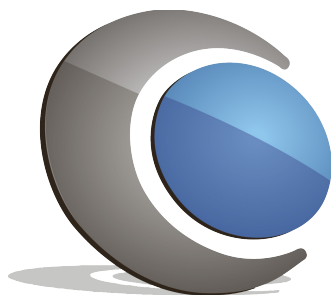


CompatibleOne Open Source Cloud Broker Architecture Overview



WHITE PAPER

October 2012

Table of Contents

Abstract	2
Background	2
Disclaimer	2
Introduction	2
Section A: CompatibleOne: Open Standards and Open Source	3
Section B: CompatibleOne Functional Cycle	3
Section C: CompatibleOne Logical Model	6
Section D: CompatibleOne Description Model	7
Section E: CompatibleOne Communication Architecture	8
Section F: Next Steps	10

List of Exhibits

Exhibit 1: Advanced Capabilities for CORDS: Architecture v2.12	4
Exhibit 2: CompatibleOne Resource Description System: Logical View	6
Exhibit 3: CORDS Service Control Graph	8
Exhibit 4: CompatibleOne ACCORDS (Communication Architecture)	9
Exhibit 5: CompatibleOne ACCORDS (OpenStack or OpenNebula Provisioning)	9



CompatibleOne

Open Source Cloud Broker

Architecture Overview

Abstract

In this white paper, we discuss the CompatibleOne open source cloud broker. CompatibleOne is an open source collaborative project based on open standards. The CompatibleOne approach is based on the open cloud computing interface OCCI. This white paper introduces the four-step functional manifest-to-service provisioning cycle of the CompatibleOne broker and provides an overview of its logical model and core concepts and its communication architecture.

Background

The white paper is based on a presentation given by Iain James MARSHALL of Prologue SA, at OW2con'11 on 24 Nov 2011. The video capture of this presentation includes a live demonstration and is available at: <http://compatibleone.org/bin/view/Videos/>.

CompatibleOne originated as a collaborative project endorsed by the Systematic Competitive Cluster and supported financially by the Paris Region. The partners in the CompatibleOne project include ActiveEon, Bull, City Passenger, Enovance, Eureva, Mandriva, Nexedi, Nuxeo, Prologue, Xwiki, Inria, Institut Telecom and OW2.

Disclaimer

The purpose of this white paper is to provide a technical overview of the CompatibleOne Open Source Broker architecture with specific attention given to the positioning of new concepts. This paper does not provide an exhaustive discussion of the CompatibleOne Open Source Broker and is not a competitive analysis between the CompatibleOne Open Source Broker and other cloud computing interoperability mechanisms.

Introduction

CompatibleOne was launched as a collaborative project to perform research work and to come up with ideas addressing the need for interoperability in the field of Cloud Computing. The project quickly evolved until it converged in developing a cloud computing broker as defined by Gartner¹, i.e. a core cloud computing mechanism providing Intermediation, Aggregation and Arbitration of cloud services. Moreover, the CompatibleOne platform is aligned with the Cloud Computing Reference Architecture² of the National Institute of Standards and Technology (NIST, U.S. Department of Commerce).

This white paper presents an overview of CompatibleOne the first open source cloud computing broker. It is blueprint for Developers, Operators and DevOps. In CompatibleOne, models and documents are shared by developers and operators, customers and vendors. CompatibleOne provides support for interoperability, portability and reversibility.

¹Gartner, Defining Cloud Service Brokerage: Taking Intermediation to the Next Level, by Daryl C. Plummer, Benoit J. Lheureux, Frances Karamouzis, 8 October 2010, G00206187

²http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505

Section A: CompatibleOne: Open Standards and Open Source

CompatibleOne is an open source project. OW2, an independent open source organization, is one of the project's partners. The work performed is published under open source license on the OW2 forge.

The CompatibleOne project is based upon the strong belief that open source and the adoption of open standards enable open innovation and foster an open cloud computing ecosystem.

For this reason, CompatibleOne is driven by open standards and open source. The project partners have identified a series of open standards that currently provide the backdrop of the project. These include the following standards:

- Open Cloud Computing Interface (OCCI)* from Open Grid Forum (OGF)
- Cloud Data Management Interface (CDMI) from Storage Networking Industry Association (SNIA)

*A special mention goes to OCCI, whose simplicity and powerful potential grabbed the attention of all project partners.

The CompatibleOne project partners are now working to find out how to leverage these standards and specifications in building a master plan.

Section B: CompatibleOne Functional Cycle

CompatibleOne is comprised of a model and an execution platform. The model, CompatibleOne Resource Description System (CORDS), is an object oriented language for the description of cloud applications, services and resources. The execution platform, Advanced Capabilities for CORDS (ACCORDS), is a cloud application provisioning and deployment control system.

Exhibit 1 below provides a high-level overview of the CompatibleOne ACCORDS platform architecture in four quadrants. These quadrants represent the four steps of the functional cycle.

Step One: Handling the user's requirements

The first step deals with the user interactivity. Here users of the platform specify their requirements in terms of the CORDS Manifest, describing the infrastructure they require. It is a service manifest document which describes in detail the services to be delivered, the technical and economical criteria, and specifications and constraints that are to be taken into consideration. The outcome of step one is the CORDS Manifest.

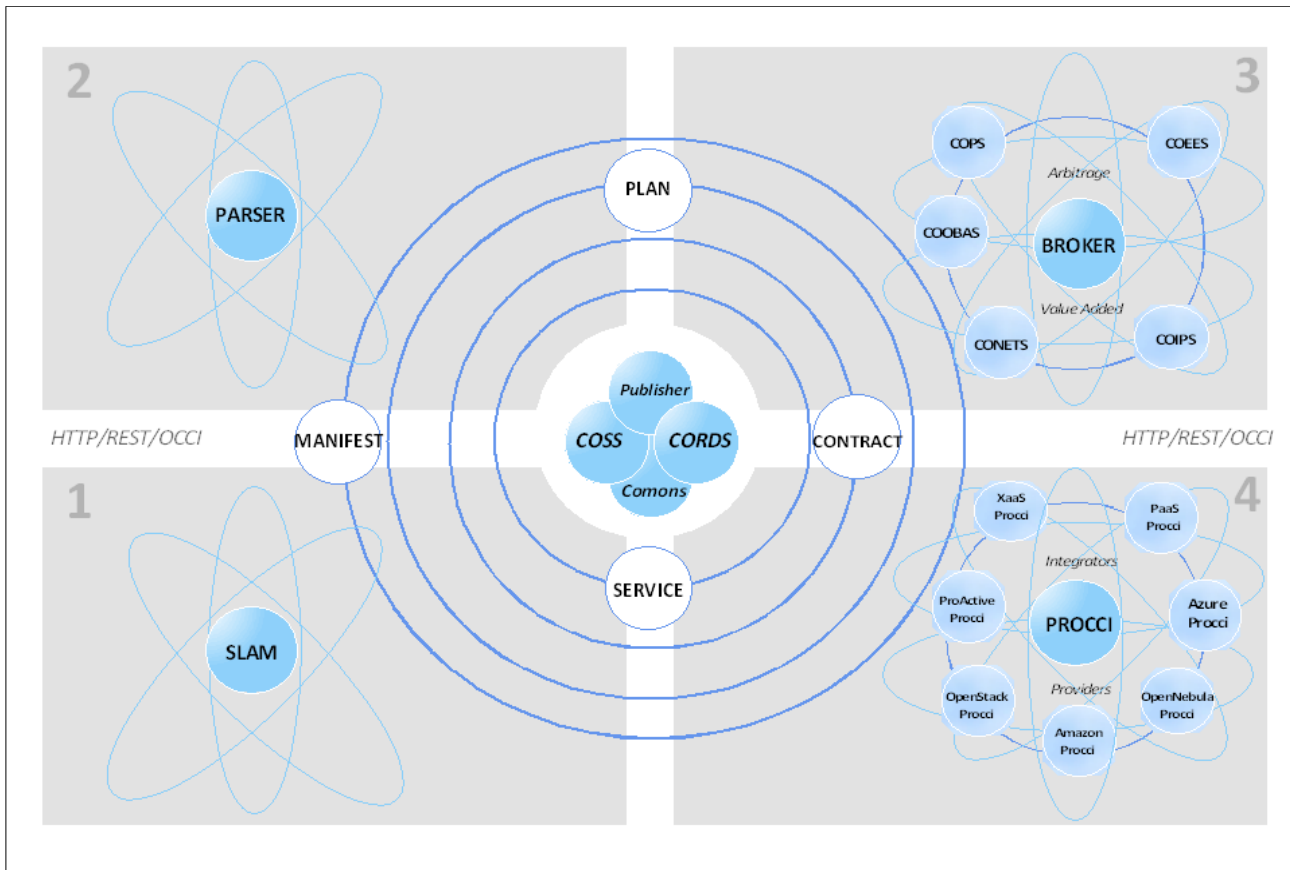


Exhibit 1: Advanced Capabilities for CORDS: Architecture v2.12

Step Two: Validation and provisioning plan

In the second step, the CORDS Manifest is transferred to the ACCORDS Parser which is responsible for parsing of the CORDS Manifest and validating the XML syntax, the schema conformance and the infrastructure feasibility.

For feasibility, the ACCORDS Parser will analyze the document and localize service providers capable of understanding and satisfying the terms described within the manifest. If this process completes successfully, the XML description will be transformed into a fully qualified and validated provisioning plan. The provisioning plan is like a blueprint, containing information required to achieve the infrastructure described by the user or initiator within the manifest.

Step Three: Execution of the provisioning plan

Once the provisioning plan has been validated, it is transferred to the ACCORDS Broker which is responsible for the actual provisioning operation. The ACCORDS Broker is connected to a collection of services. The services listed here in the third quadrant are not exhaustive. The lists indicates service components which have been foreseen and are being developed within the CompatibleOne project group.

They include the following services:

- COPS (CompatibleOne Placement Service)
- COEES (CompatibleOne Energy Efficiency Service)
- COOBAS (CompatibleOne Ordering Billing and Accounting Service)
- CONETS (CompatibleOne Network Service)

- COIPS (CompatibleOne Image Production Service)

The above services assist the ACCORDS Broker in the provisioning operations required when using the plan. This operation requires the creation of a control graph for the management of the different components involved in and resulting from the provisioning process.

Step Four: Delivering the cloud services

In the fourth step, guided by the provisioning control graph, the ACCORDS Broker negotiates the availability of resources with the providers through the ACCORDS PROCCI. The ACCORDS PROCCI enables the ACCORDS Broker to negotiate the placement and provisioning of the resources required to fulfill the needs expressed within the CORDS Manifest and qualified within the plan. This will be delivered to the user in the form of the Service. Communication with the different providers is performed through client/server interfaces described using standard OCCl specifications and implemented on the CompatibleOne side within the ACCORDS PROCCI and on the provider side within a provider-specific Procci component. CompatibleOne is currently capable of provisioning on both OpenNebula and OpenStack platforms. Other Procci components being developed include SlapOS, ProActive, Amazon and Windows Azure.*

*It is important to note that the proxy defined by CompatibleOne is an OCCl proxy because the entire architecture has been built on standard, loosely coupled, OCCl REST API interfaces. Consequently every single component communicates with the other components through these REST OCCl interfaces.

Communication Support Facilities

Interestingly, CompatibleOne operates without service bus. Many options with service buses were being evaluated until it became apparent that service buses were not necessary and were redundant. However, the following three communication support facilities provide support to all interactions:

- Knowledge Base: A collection of information maintained within the system. The elements stored in the Knowledge Base are the CORDS Manifests, the Plans, the Contracts and the Services. A unique and universal identifier is attributed to each element in the Knowledge Base and may be used to reference the element for use in subsequent plans.
- CompatibleOne Security Services (COSS): A highly security conscious system where everything is performed using Transport Layer Security (TLS). Each component in the system is required to obtain and present its own identity. Each component must also be authenticated and authorized to use the system.
- Publisher: Provides publication services, or repository services, for the connection between the components.

All interactions within ACCORDS (Advanced Capabilities for CORDS) are performed in conjunction with the Publication Service offered by the Publisher, in compliance with COSS (CompatibleOne Security Service) and using OCCl (Open Cloud Computing Interface) over HTTP (Hyper Text Transfer Protocol). In the end, this collection of protocols and specifications constitutes the cloud middleware offered by CompatibleOne.

Section C: CompatibleOne Logical Model

The CompatibleOne approach is based on the open cloud computing interface OCCI, which provides a simple approach for the description of almost anything. The OCCI OGF Group defines an Entity that is derived to give Resources, Links and Mixins. Below that, the entity defines three basic elements: Network, Compute and Storage to which have been added StorageLink and NetworkLink. The entire CORDS (CompatibleOne Resource Description System) is compliant with this model.

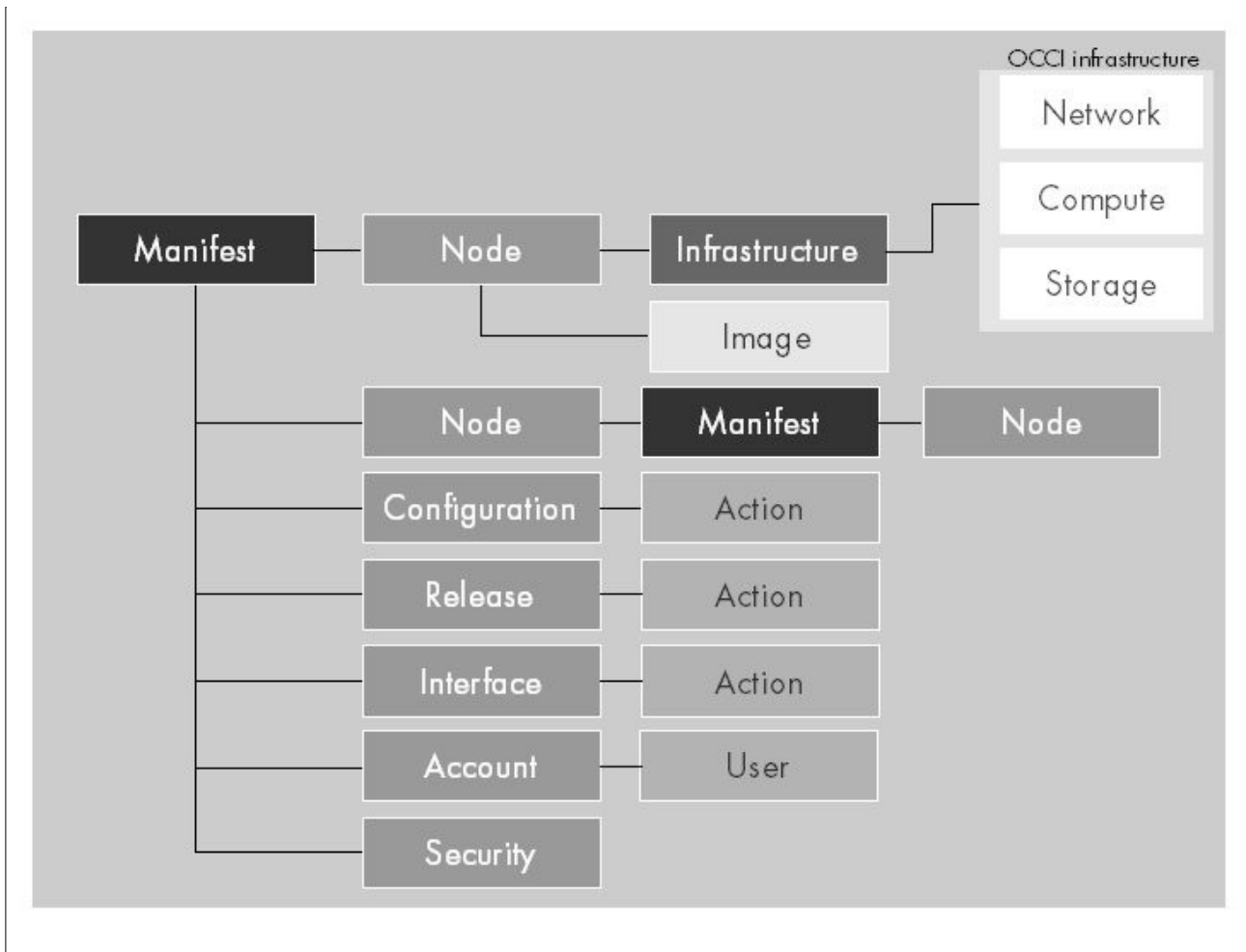


Exhibit 2: CompatibleOne Resource Description System: Logical View

In Exhibit 2, which shows the logical view of the model, each of the boxes represents a Category and the Category is a resource. CompatibleOne's rationale is completely compliant with an OCCl interface and an OCCl description understanding mechanism. In the OCCl protocol, there is a discovery capability that allows any client to discover the categories of services these OCCl servers are capable of delivering. The illustration shows the logical descriptive view of how the CORDS Manifest is organized internally to give rise to the Plan.

A CORDS Manifest is described in terms of Nodes. The Node is defined as a unit, which is described in terms of its Infrastructure and Application Image. Its Infrastructure provides its Network, Compute and Storage requirements and is

described using the standard OCCI Category elements. The Image is described in terms of its base operating system and the collection of required packages that are installed to achieve the desired application functionality.

A CORDS Manifest can be composed of as many Nodes as required to provide the right functionality. Each Node may suggest a particular type of provider. By type of provider, a Node may require specific services available only within, for instance, an OpenNebula platform, or an OpenStack platform or a ProActive platform with a section script.

If required, the Node may state its needs explicitly. It may however suggest that “any will do”, in which case the overall configuration of the platform will be allowed to apply its preferences. For a Node there is also the notion of Configuration and Monitoring actions. These actions describe the way a Node will co-operate with other Nodes within the Manifest. They also describe the requirements and the constraints which are placed upon it in this operating scenario.

Section D: CompatibleOne Description Model

The processing of a CORDS Manifest gives rise to a Plan. The Plan is produced and corresponds to a particular CORDS Manifest to which is attached a pricing scheme. The Plan, when it is processed by the ACCORDS Broker, will give rise to a Service. Several links to a Service may be produced from a single Plan. Each Service comprises the collection of Contracts required to represent each of the individual Nodes expressed in the CORDS Manifest. Attached to the Contracts are the collections of instructions required for its Configuration and to establish its conditions for Monitoring. The Contract describes the user subscription account for the provider on which a Service is to be deployed.

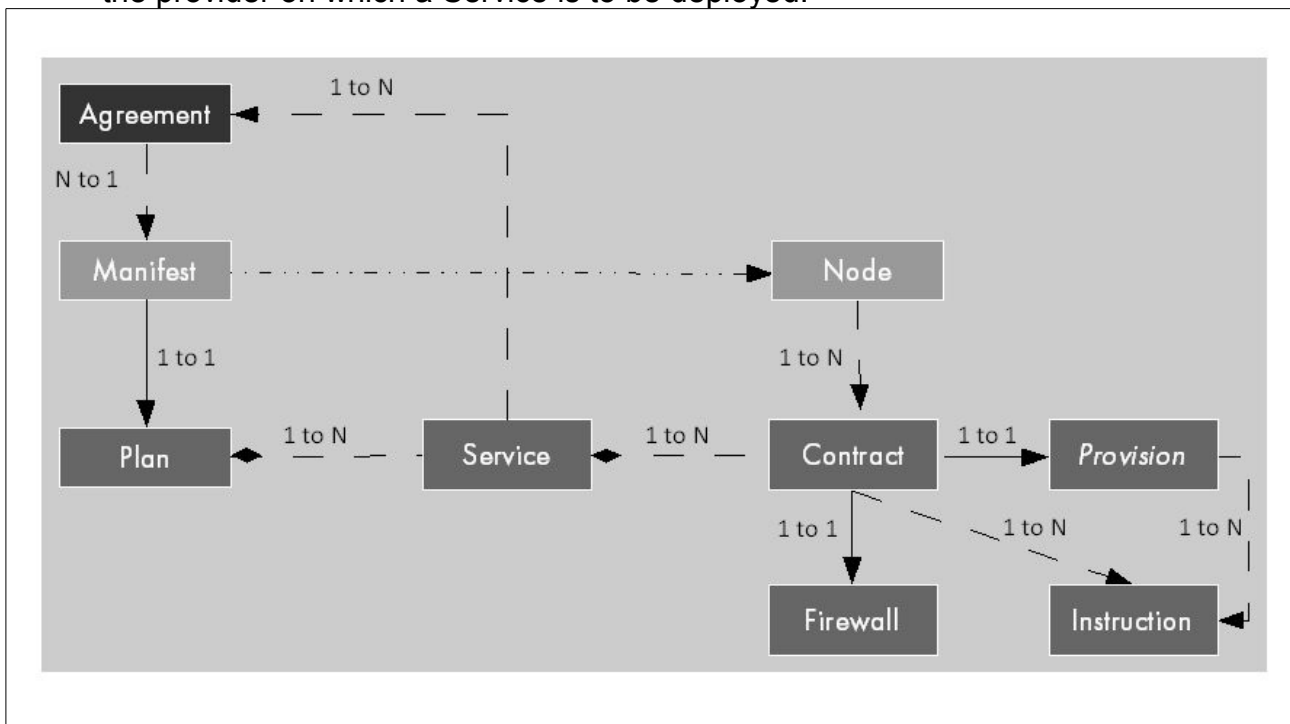


Exhibit 3: CORDS Service Control Graph

Section E: CompatibleOne Communication Architecture

Exhibit 4 shows the CompatibleOne communication architecture. The user submits a CORDS Manifest to the ACCORDS Parser and ACCORDS Broker. The ACCORDS Broker engages the service consumers which in turn discover the service providers through the central ACCORDS Publisher.

The service consumers communicate using their REST OCCl interface with individual providers to achieve the desired results. The monitoring agent under the control of a monitoring collector, or consumer, collects the various audit and monitoring data in order to ensure that the provisioning is correctly performed and correctly operated in compliance with the terms described in the CORDS Manifest or the SLA.

Exhibit 5 shows the schematic for the OpenStack Nova or OpenNebula provisioning. The user instantiates a CORDS Plan via the ACCORDS Broker. For OpenStack, the ACCORDS Broker communicates with the ACCORDS Publisher and the proxy to look for an OpenStack provider. It performs the required provisioning through the REST interface. It happens in the same way with OpenNebula. The OpenNebula provisioning system is perfectly symmetrical. The only difference is the change in the nature of the final proxy, i.e. the proxy component that is responsible for handling the communication with a particular platform.

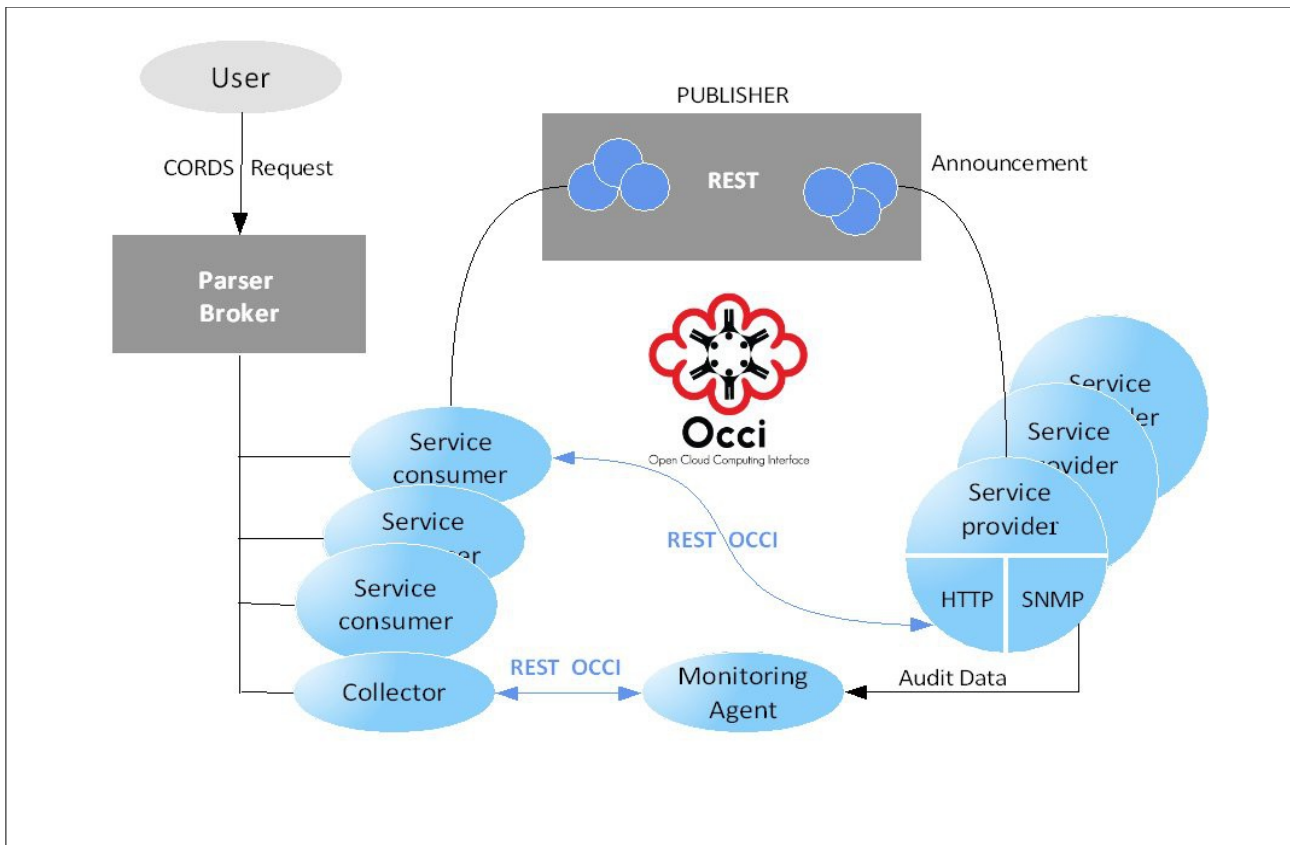


Exhibit 4: CompatibleOne ACCORDS Communication Architecture

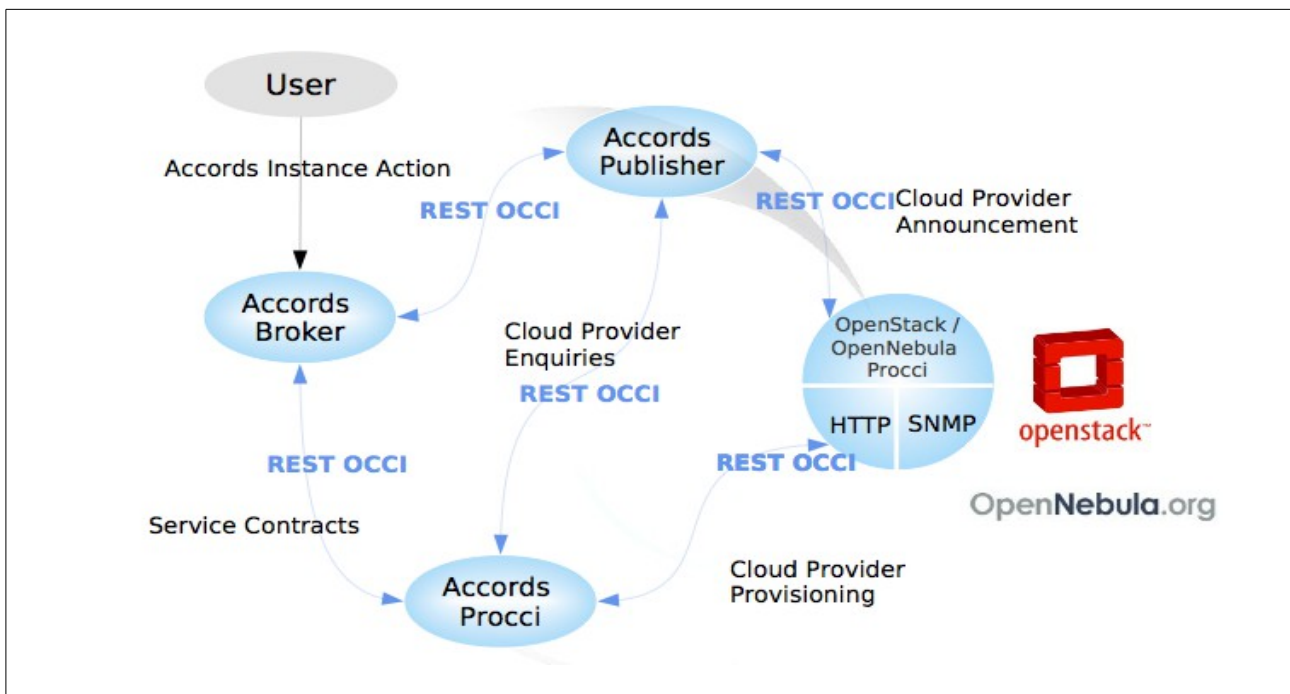


Exhibit 5: CompatibleOne ACCORDS for OpenStack or OpenNebula Provisioning

Section F: Next Steps

Having achieved proof of concept, the CompatibleOne project partners will soon announce automatic provisioning and automatic configuring based on SLA. The control of diverse Cloud Computing resources will be the next step. This will integrate the new components of the brokering system in terms of monitoring, when the blueprint and primary work has been performed. Meanwhile, the coding phase is in full swing. Security has already been defined and coded. Accountancy is currently being discussed in detail. These discussions will elaborate the financial channels that will guide and control the system in order to provide intelligent provisioning in terms of financial, energetic and legal constraints and conditions.

The CompatibleOne open source broker will further be validated with three use cases:

- Extensions for use of PaaS Services (providing Elasticity, Scalability and Storage)
- Cloud Automation – whereby the manifest will be used to provide automated instancing on demand by business processes
- Provision of HPC resources and 3D visualization (a highly interesting proof of concept whereby Graphic Processor units will be called to operate within high performance grids and provide heavy duty calculation units for 3D visualization for gaming platform delivery).

For more information

To learn more about the CompatibleOne open source cloud broker, please visit:

<http://www.compatibleone.org>

To join the conversation on our developer community, you are welcome to subscribe to:

http://mail.ow2.org/wws/subscribe/compatible-dev_contrib

The video capture of the CompatibleOne Open Source Cloud Broker Quick Demonstration, upon which this white paper is based, includes a live demonstration and is available at:

<http://compatibleone.org/bin/view/Videos/>