EGI-InSPIRE

EGI Federated Cloud Blueprint v1

EU MILESTONE: MS520

Document identifier:	**EGI-InSPIRE-MS520-FINAL**
Date:	**07/06/2013**
Activity:	**SA2**
Lead Partner:	**OeRC**
Document Status:	**FINAL**
Dissemination Level:	**PUBLIC**
Document Link:	**https://documents.egi.eu/document/1773**

Abstract

This milestone document provides background information on the activities on the EGI Federated Cloud Task Force over the past 18 months and details of the integration work carried out over the past 6-month development cycle. This includes how the three classes of actors within the task force, the users, resource providers and technology providers have contributed to the various activities. The document also has a section on the method by which new participants can join the federated cloud, what is required and expected of them.

I. COPYRIGHT NOTICE

II. DELIVERY SLIP

	Name	Partner/Activity	Date
From	David Wallom, Michel Drescher	OeRC/EGI.eu, SA2	07/06/2013
Reviewed by	**Moderator:** Gergely Sipos **Reviewers:** David Blundell Andy Edmonds	EGI.eu 100%IT ZHAW	0706/2013 06/06/2013 06/06/2013
Approved by	**AMB & PMB**		7/6/2013

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
1	03/06/2013	First draft, with contributions from Task Force Members	Michel Drescher, EGI.eu
2	03/06/2013	Second draft, ready for eternal review	David Wallom, OerC Michel Drescher, EGI.eu
3	06/06/2013	Third draft reflecting reviewer comments	Michel Drescher, EGI.eu
4	07/06/2013	Final version incorporating PMB review	Michel Drescher, EGI.eu

IV. APPLICATION AREA

This document is a formal deliverable for the European Commission, applicable to all members of the EGI-InSPIRE project, beneficiaries and Joint Research Unit members, as well as its collaborating projects.

V. DOCUMENT AMENDMENT PROCEDURE

Amendments, comments and suggestions should be sent to the authors. The procedures documented in the EGI-InSPIRE "Document Management Procedure" will be followed: https://wiki.egi.eu/wiki/Procedures

VI. TERMINOLOGY

A complete project glossary is provided at the following page: http://www.egi.eu/about/glossary/.

VII. PROJECT SUMMARY

To support science and innovation, a lasting operational model for e-Science is needed − both for coordinating the infrastructure and for delivering integrated services that cross national borders.

The EGI-InSPIRE project will support the transition from a project-based system to a sustainable pan-European e-Infrastructure, by supporting ‘grids’ of high-performance computing (HPC) and high-throughput computing (HTC) resources. EGI-InSPIRE will also be ideally placed to integrate new Distributed Computing Infrastructures (DCIs) such as clouds, supercomputing networks and desktop grids, to benefit user communities within the European Research Area.

EGI-InSPIRE will collect user requirements and provide support for the current and potential new user communities, for example within the ESFRI projects. Additional support will also be given to the current heavy users of the infrastructure, such as high energy physics, computational chemistry and life sciences, as they move their critical services and tools from a centralised support model to one driven by their own individual communities.

The objectives of the project are:

1. The continued operation and expansion of today’s production infrastructure by transitioning to a governance model and operational infrastructure that can be increasingly sustained outside of specific project funding.
2. The continued support of researchers within Europe and their international collaborators that are using the current production infrastructure.
3. The support for current heavy users of the infrastructure in earth science, astronomy and astrophysics, fusion, computational chemistry and materials science technology, life sciences and high energy physics as they move to sustainable support models for their own communities.
4. Interfaces that expand access to new user communities including new potential heavy users of the infrastructure from the ESFRI projects.
5. Mechanisms to integrate existing infrastructure providers in Europe and around the world into the production infrastructure, so as to provide transparent access to all authorised users.
6. Establish processes and procedures to allow the integration of new DCI technologies (e.g. clouds, volunteer desktop grids) and heterogeneous resources (e.g. HTC and HPC) into a seamless production infrastructure as they mature and demonstrate value to the EGI community.

The EGI community is a federation of independent national and community resource providers, whose resources support specific research communities and international collaborators both within Europe and worldwide. EGI.eu, coordinator of EGI-InSPIRE, brings together partner institutions established within the community to provide a set of essential human and technical services that enable secure integrated access to distributed resources on behalf of the community.

The production infrastructure supports Virtual Research Communities (VRCs) − structured international user communities − that are grouped into specific research domains. VRCs are formally represented within EGI at both a technical and strategic level.

VIII. EXECUTIVE SUMMARY

This milestone document summarises the activities of the EGI Federated Cloud Task Force over the past 18 months. The Task Force consists of a steering panel, and multiple representatives of the three main stakeholders in this task force: Resource Providers, Technology Providers, and User Communities. Each member of the Task Force is expected to actively contribute as their individual commitment allows, ranging from best effort to partially funded effort coming from a variety of sources.

The activities in the task force contribute to and are aligned with a general architecture revision in EGI towards a platform-oriented architecture. The EGI Cloud Infrastructure Platform is the architectural incarnation of the activities in the EGI Federated Clouds Task Force, where Resource Providers are free in their choice of Cloud Management Frameworks (in alignment with the Cloud Computing paradigms) for as long as they are abiding by the federation's requirements. This model of *abstract Cloud Management Framework subsystems* allows architecting a scalable Cloud Infrastructure Platform without entangling its stakeholders in too many dependencies.

The federation requirements mainly consider services exposed to the consuming research communities, and the details of the back-end integration with the EGI Core Infrastructure Platform. Supporting the use cases of the Federated Cloud consumers, three core management interfaces are provided: VM Management using OCCI, Data Management using CDMI, and Information Discovery using LDAP and GLUE2. Backend integration includes Federated AAI using Grid Certificates, Accounting using EGI's accounting infrastructure and an extension of the OGF UR specification, Monitoring based on EGI's SAM infrastructure and service registration in EGI's central service registry.

The work of the EGI Federated Cloud Task Force and definition of the EGI Cloud Infrastructure Platform is driven by 10 scenarios. This document provides summaries of the integration activities pertinent to Cloud Management Frameworks that are deployed in the Task Force's test bed (in no particular order: OpenNebula, OpenStack, WNoDeS and Synnefo) to meet the defined scenarios. This allows any Cloud resource provider that is interested in integrating with the EGI Cloud Infrastructure Platform to either deploy one of the Cloud Management Framework that have already been integrated, or to undertake the work necessary to integrate their existing deployment into the testbed.

TABLE OF CONTENTS

1 INTRODUCTION

The current high throughput model of grid computing has proven to be extremely powerful for a small number of different communities. These communities have thrived in the current grid environment but the uptake of e-infrastructure by other communities has been limited. EGI has therefore strategically decided to investigate how it could broaden the uptake of its infrastructure to support other research communities and application design models, that would not only be able to take advantage of the existing functionality and investment already made in EGI's Core Infrastructure, but also support different research communities and their applications on the current production infrastructure than it was previously able to.

The utilisation of virtualization and Infrastructure as a Service (IaaS) cloud computing was a clear candidate to enable this transformation. It was also clear that with a number of different open source technologies already in use across a number of different resource providers, that it would not be possible to mandate a single software stack. Instead, following on from a number of different activities already on-going in Europe including SIENA[1], an approach that required the utilisation of open standards where available and, where not, methods that have broad acceptance in the e-infrastructure community were essential.

The Task Force as originally configured had an 18-month mandate starting from September 2011, which was subdivided into 3 succinct six-month blocks:

1) Setup - Identify resource and technology providers and draft the model

2) Consolidation - Engage exemplar user communities and start configuration of test-bed

3) Integration - Establish test-bed fully with early adopter user communities and document

Overall goals for the activity were to:

- Write a blueprint document[2] for EGI Resource Providers that wish to securely federate[3] and share their virtualised environments as part of the EGI production infrastructure;
- Deploy a test bed[4] to evaluate the integration of virtualised resources within the existing EGI production infrastructure for monitoring[5], accounting[6] and information services[7];
- Investigate and catalogue the requirements[8] for community facing services based on or deployed through virtualised resources;
- Provide feedback[9] to relevant technology providers on their implementations and any changes needed for deployment into the production infrastructure;
- Identify and work with user communities[10] willing to be early adopters of the test bed infrastructure to help prioritise its future development;
- Identify issues[11] that need to be addressed by other areas of EGI (e.g. policy, operations, support & dissemination).

[1] http://www.sienainitiative.eu

[2] https://wiki.egi.eu/wiki/Fedcloud-tf:Blueprint

[3] https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:_Federated_AAI

[4] https://wiki.egi.eu/wiki/Fedcloud-tf:Testbed

[5] https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Scenario5

[6] https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Scenario4

[7] https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Scenario3

[8] https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:_Outreach#Requirements

[9] https://wiki.egi.eu/wiki/Fedcloud-tf:Blueprint:Solutions_Intentory

[10] https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:_Outreach

[11] https://wiki.egi.eu/wiki/Fedcloud-tf:Blueprint:Security_and_Policy

2 FEDERATION MODEL

The federation of IaaS Cloud resources in EGI is built upon the extensive autonomy of Resource Providers in terms of ownership of exposed resources. The current federation model in EGI for exposing Grid resources requires Resource Providers to deploy and operate a specific set of Grid Middleware components before they could be integrated into EGI's production infrastructure. In contrast, the federation model for distributed IaaS Cloud resources allows a lightweight aggregation of local Cloud resources into the EGI Cloud Infrastructure Platform (CLIP). At the heart of the federation are the locally deployed Cloud Management stacks. In compliance with the Cloud computing model, the EGI CLIP does not mandate deploying any particular or specific Cloud Management stack; it is the responsibility of the Resource Providers to research, identify and deploy the solution that fits best their individual needs for as long as the offered services implement the required interfaces and domain languages. These interfaces and domain languages, and the interoperability of their implementation with other solutions are the focus of the federation.

Consequently, the EGI CLIP is modelled around the concept of an *abstract* Cloud Management stack subsystem that is integrated with components of the EGI Core Infrastructure Platform (see Figure 1).

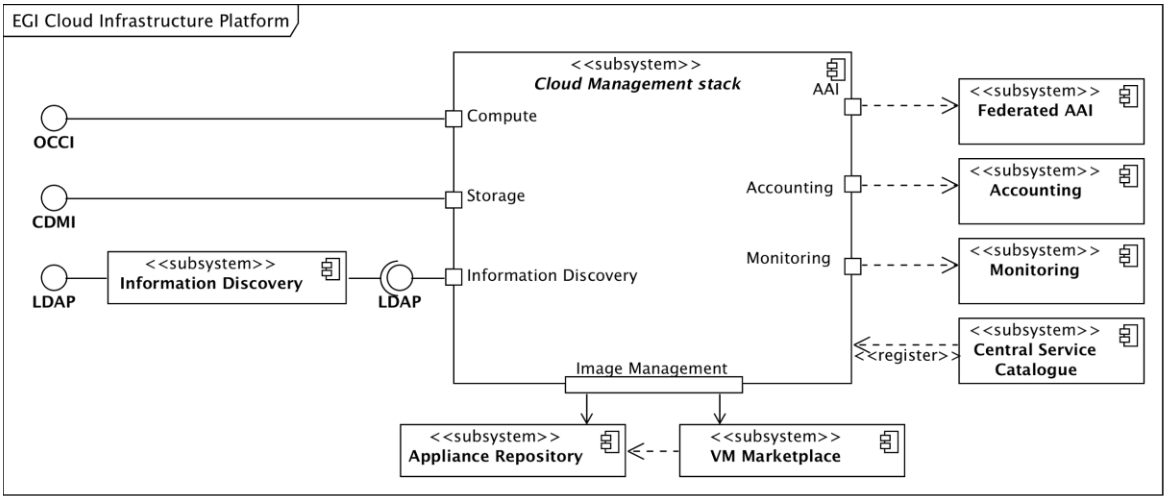

Figure 1: Architecture of the EGI Cloud Infrastructure Platform

This architecture allows EGI to define the CLIP as a relatively thin layer of federation and interoperability around local deployments and integrations of Cloud Management stacks.

This architecture defines interaction ports with a number of services from the EGI Core Infrastructure Platform, and the EGI Collaboration Platform. At the same time, it defines the required external interfaces and corresponding interaction ports. All these ports will have to be realised by local Cloud Management stack deployments.

The main interaction points of Resource Providers must take care of:

- Integrate with the EGI Core Authentication, Authorisation and Identity (AAI) systems
- Integrate with the EGI Core Accounting system
- Integrate with the EGI Core Monitoring system
- Provide a standardised Cloud Computing management interface (OCCI)
- Provide a standardised Cloud Storage interface (CDMI)
- Provide a standardised interface to an Information Service

Additionally, by means of using the Appliance Repository and the VM Marketplace from the EGI Collaboration Platform the EGI Cloud Infrastructure Platform is providing VM image sharing and re-use across EGI Research Communities.

At the time of writing the Cloud federation platform is maintaining its own, separate Information discovery system. Even though it is using the GLUE2 schema, some extensions and tweaks are not compatible with the canonical GLUE 2 specification. Hence Cloud Resource providers maintain local LDAP endpoints (usually deployed as a resource BDII) aggregated into a Cloud Platform Information Discovery service, which in turn allows access to the data using LDAP v3.

Figure 2 provides an overview of the current realisations of the abstract Cloud Management stack subsystem in the EGI Cloud federation. It illustrates that each existing realisation inherits the obligation to implement the interaction points from the generalised parent Cloud Management stack. At the same time, the EGI Federated Clouds Task (funded through the EGI-InSPIRE project) gives Resource Providers a platform to share their implementation solutions for a commonly deployed specific Cloud Management stack (e.g. OpenNebula and OpenStack). Section 5 is dedicated to the documentation of the steps necessary to integrate a local deployment of a given Cloud Management stack into the EGI Cloud federation.

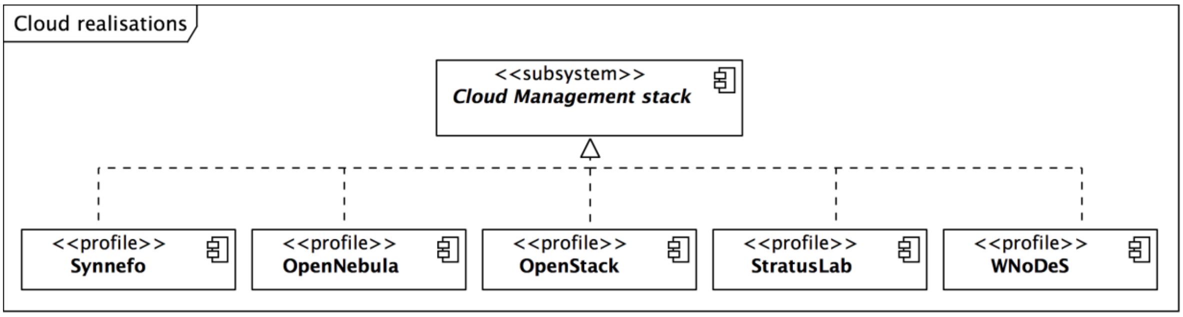

Figure 2: Current realisations of the abstract Cloud Management stack component

Through this collaboration, Resource Providers gradually develop and mature deployment and configuration profiles around common Cloud Management stacks as illustrated in Figure 2. This blueprint document is the first iteration of the resource provider oriented documentation.

3 MANAGEMENT INTERFACES

To federate a cloud system there are several functions for which a common interface must be defined. These are each described below and overall provide the definition of the method by which a 'user' of the service would be able to interact.

3.1 VM management interface: OCCI

The Open Cloud Computing Interface (OCCI) is a RESTful Protocol and API designed to facilitate interoperable access to, and query of, cloud-based resources across multiple resource providers and heterogeneous environments. The formal specification is maintained and actively worked on by OGF's OCCI-WG, for details see http://occi-wg.org/. The intended deployment is depicted in Figure 3.

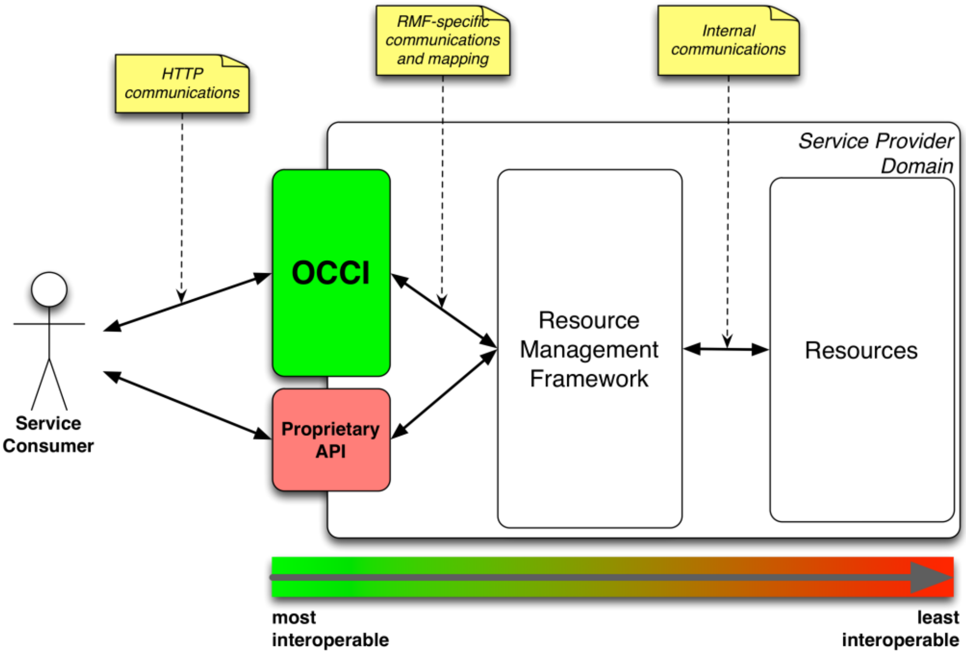

Figure 3: Deployment of OCCI in a provider's infrastructure

OCCI's specification consists of three basic elements, each covered in a separate specification document:

OCCI Core describes the formal definition of the OCCI Core Model [R 1]. **OCCI HTTP Rendering** defines how to interact with the OCCI Core Model using the RESTful OCCI API [R 2]. The document defines how the OCCI Core Model can be communicated and thus serialised using the HTTP protocol. **OCCI Infrastructure** contains the definition of the OCCI Infrastructure extension for the IaaS domain [R 3]. The document defines additional resource types, their attributes and the actions that can be taken on each resource type. Detailed description of the abovementioned elements of the specification is outside the scope of this document. A simplified description is as follows.

OCCI Core defines base types **Resource**, **Link**, **Action** and **Mixin**. Resource represents all OCCI objects that can be manipulated and used in any conceivable way. In general, it represents provider's resources such as images (Storage Resource), networks (Network Resource), virtual machines (Compute Resource) or available services. Link represents a base association between two Resource instances; it indicates a generic connection between a *source* and a *target*. The most common real-world examples are Network Interface and Storage Link connecting Storage and Network Resource to a Compute Resource. Action defines an operation that may be invoked, tied to a specific Resource instance or a collection of Resource instances. In general, Action is designed to perform complex high-level operations changing the state of the chosen Resource such as virtual machine reboot or migration. The concept of mixins is used to facilitate extensibility and provide a way to define provider-specific features.

In the Federated Cloud environment, OCCI is deployed as a variety of platform-specific implementations. An ongoing EGI-InSPIRE mini-project[12] aims to provide a common implementation to further improve interoperability.

3.2 Data management interface: CDMI

The SNIA Cloud Data Management Interface (CDMI) defines a RESTful open standard for operations on storage objects. Semantically the interface is very close to AWS S3 and MS Azure Blob, but is more open and flexible for implementation.

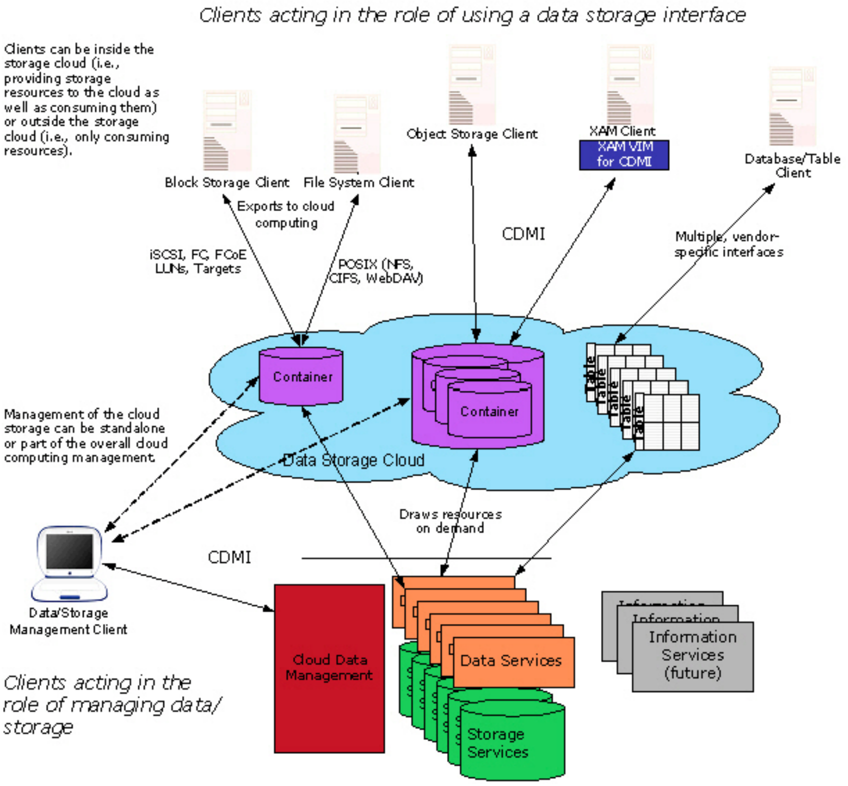
Clients acting in the role of using a data storage interface
Clients can be inside the storage cloud (i.e., providing storage resources to the cloud as well as consuming them) or outside the storage cloud (i.e., only consuming resources).
Block Storage Client
File System Client
Object Storage Client
XAM Client
XAM VIM for CDMI
Database/Table Client
Exports to cloud computing
iSCSI, FC, FCoE LUNs, Targets
POSIX (NFS, CIFS, WebDAV)
CDMI
Multiple, vendor-specific interfaces
Container
Container
Table
Table
Table
Table
Table
Data Storage Cloud
Management of the cloud storage can be standalone or part of the overall cloud computing management.
Draws resources on demand
CDMI
Data/Storage Management Client
Clients acting in the role of managing data/storage
Cloud Data Management
Data Services
Storage Services
Information
Information
Information Services (future)

Figure 4: Cloud storage reference model (courtesy to SNIAcloud.com)

Figure 4 shows the conceptual model of a cloud storage system. CDMI offers clients a way for operating both on a storage management system and single data items. The exact level of support depends on the concrete implementation and is exposed to the client as part of the protocol.

The design of the protocol is aimed both at flexibility and efficiency. Certain heavyweight operations, e.g. blob download, can be performed also with a pure HTTP client to make use of the existing

[12] *TSA4.4 Providing OCCI support for arbitrary Cloud Management Frameworks*

ecosystem of tools. CDMI is built around the concept of Objects, which vary in supported operations and metadata schema. Each Object has an ID, which is unique across all CDMI deployments.

3.2.1 CDMI Objects

There are 4 objects most relevant in the context of EGI's Federated Cloud:

- **Data object**: Abstraction for a file with rich metadata.
- **Container: A**bstraction for a folder. Export to non-HTTP protocols is performed on the container level. Container might have other containers inside of them.
- **Capability:** Exposes information about a feature set of a certain object.
- **Domain:** Deployment specific information.

3.2.2 Detection of capabilities

CDMI supports partial implementation of the standards by defining optional features and parameters. In order to discover what functionality is supported by a specific implementation, CDMI client can issue a GET request to a fixed url: /cdmi_capabilities.

3.2.3 Export protocol

Attachment of the storage items to a VM can often be performed more efficiently using protocols like NFS or iSCSI. CDMI supports exposing of this information via container metadata. A client can make use of this information to attach a storage item to a VM over an OCCI protocol.

More information about the CDMI standard can be found at http://cdmi.sniacloud.com/. An ongoing EGI-InSPIRE mini-project[13] aims to provide an implementation of CDMI, which integrates with OCCI-based infrastructure and supports use-cases needed in a Federated Cloud.

3.3 Virtual Organisation Management & AAI: VOMS

Within the task force, it has been decided to generally allow VOMS proxy certificates of VOs 'fedcloud.egi.eu' and 'ops', the latter mainly for Nagios monitoring. There are modules available for each cloud middleware that have been developed by TF members. Configuring these modules into a provider's cloud installation will allow members of these VOs to access the cloud.

Figure 5 shows the main components involved. The user retrieves a VOMS attribute certificate from the VOMS server of the desired VO and thus creates a local VOMS proxy certificate. The VOMS proxy certificate is use in subsequent calls to the OCCI endpoints of OpenNebula or OpenStack using the rOCCI client tool.

The rOCCI client directly talks to OpenNebula endpoints, which map the certificate and VO information to local users. Local users need to have been created in advance, which is triggered by regular synchronizations of the OpenNebula installation with Perun. Perun manages the VOs that participate in the FCTF testbed.

[13] *TSA4.5 CDMI Support in Cloud Management Frameworks*

In order to access an OpenStack OCCI endpoint, the rOCCI client needs to retrieve a Keystone token from OpenStack Keystone first. The retrieval is transparent to the user and automated in the workflow of accessing the OpenStack OCCI endpoint. It is triggered by the OCCI endpoint rejecting invalid requests and sending back an HTTP header referencing the Keystone URL for authentication. Users are generated on the fly in Keystone, it does not need regular synchronization with the VO Management server Perun (see below).

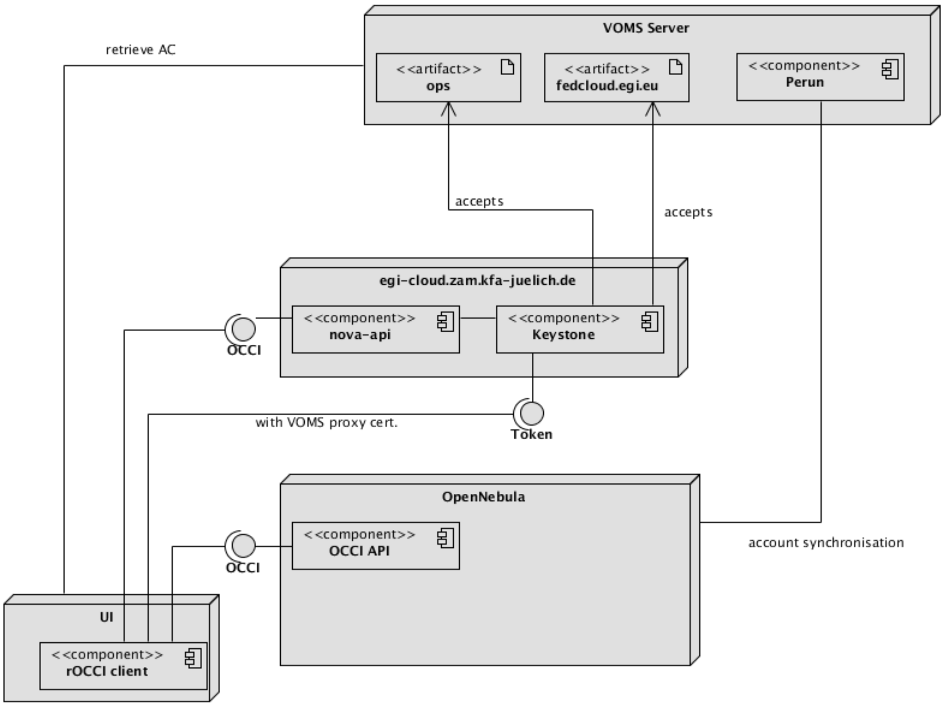

Figure 5: Model of the Federated Cloud authentication architecture

Generic information about how to configure VOMS support for;

- OpenStack Keystone can be found at http://keystone-voms.readthedocs.org/en/latest/. Information specific to FCTF is located at https://wiki.egi.eu/wiki/Federated_AAI_Configuration#OpenStack.
- OpenNebula, the information can be found here: https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:_Federated_AAI:OpenNebula.
- Stratuslab provides multiple authentication mechanisms at once. They are documented here: http://stratuslab.eu//documentation/2012/10/07/docs-syadmin-auth.html.

Since all of these different technology providers have developed their own systems then the functionality provided by the different services and methodology by which they use VOMS credentials etc. are slightly different.

3.4 VM Image management

In a distributed, federated Cloud infrastructure, users will often face the situation of efficiently managing and distributing their VM Images across multiple Cloud resource providers. The VM Image management subsystem provides the user with an interface into the EGI Cloud federation to notify supporting resource providers of the existence of a new or updated VM Image. Sites then examine the provided information, and pending their decision pool the new or updated VM Image locally for instantiation.

This concept introduces a number of capabilities into the EGI Cloud Infrastructure Platform:

- **VM Image lifecycle management** – Apply best practices of Software Lifecycle Management at scale across EGI
- **Automated VM Image distribution** – Publish VM images (or updates to existing images) once, while they are automatically distributed to the Cloud resource providers that support the publishing research community with Cloud resources.
- **Asynchronous distribution mechanism** – Publishing images and pooling these locally are intrinsically decoupled, allowing federated Resource Providers to apply local, specific processes transparently before VM images are available for local instantiation, for example:
- **Provider-specific VM image endorsement policies** – Not all federated Cloud resource providers will be able to enforce strict perimeter protection in their Cloud infrastructure as risk management to contain potential security incidents related to VM images and instances. Sites may implement a specific VM Image inspection and assessment policy prior to pooling the image for immediate instantiation.

Two command-line tools provide the principal functionality of this subsystem; “vmcaster” to publish VM image lists and “vmcatcher” to subscribe to changes to these lists, respectively.

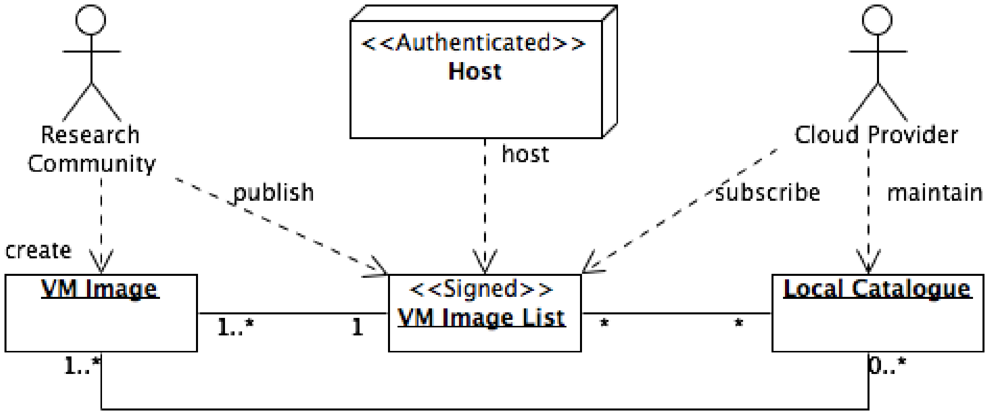

Figure 6: Main components and actors of the VM Image management subsystem

Research Communities ultimately create and update VM Images (or delegate this functionality). The Images themselves are stored in Appliance repositories that are provided and managed elsewhere, typically by the Research Community itself.

A representative of the Research Community then generates a VM Image list (or updates an existing one) and publishes it on an authenticated host, which is typically using a host certificate signed by a CA included in the EGI Trust Anchor profile.

Federated Clouds Resource Provider then subscribe to changes in VM Image lists by regularly downloading the list from the authenticated host, and comparing it against local copies. New and updated VM Images are downloaded from the appliance repository referenced in the VM Image list into a local staging cache and, where required, made available for further examination and assessment.

Ultimately, Cloud resource Providers will make VM Images available for immediate instantiation by the Research Community.

4 EGI CORE SERVICES FOR CLOUD

Alongside implementations of cloud specific interfaces it is necessary to enable the connection of these new service types with the core EGI services of Accounting, Monitoring and Service Discovery.

4.1 Information discovery: BDII

Users and service managers need tools to retrieve information about the whole infrastructure and filter the returned data to select relevant subsets of the infrastructure that fulfil their requirements. To achieve this target the information about the services in the infrastructure must be structured in a uniform schema and published by a common set of services usable both by automatic tools and human users.

The current standard deployed in EGI for the implementation of the common information system is the Berkeley Database Information Index (BDII). It is software based on a LDAP server, and it is deployed in a hierarchical structure, distributed over the whole infrastructure. The information system is structured in three levels: the grid or cloud services publish their information (e.g. specific capabilities, total and available capacity or user community supported by the service) using an OGF recommended standard format, GLUE2. The current methodology for the publishing of dynamic service data within the EGI the federated activity utilises the same configuration of BDII services as is currently deployed in EGI: The information published by the services is collected by a Site-BDII, a service deployed in almost every site in EGI. The Site-BDIIs are queried by the Top-BDIIs - a national or regional located level of the hierarchy, which contain the information of all the site services available in the infrastructure and their services. NGIs usually provide an authoritative instance of Top-BDII, but every Top-BDII, if properly configured, should contain the same set of information.

Users and tools can use the Top-BDII to look for the services that provide the capabilities and the resources to run their activities. A typical example of Top-BDII query is retrieving the list of services that support a specific user community or VO.

4.1.1 Technical implementation of the federated cloud information system

Currently the Federated Cloud information system is built starting from the resource provider level. Every resource provider is required to deploy a LDAP server publishing the information about their services structured used the GLUE schema. The best technical choice is to go for OpenLDAP, which is available in almost all the *nix machines in the world. On top of that, OpenLDAP is the server used by the gLite BDIIs, therefore it would be easy to use the same configuration files set-up used for the GRIS (Grid Resource Information Service) or the GIIS (Grid Index Information Service).

Cloud services are not yet implementing information providers, therefore the information are published directly by a site-level information provider, comparable to a site-bdii in the structure of the information published. This solution is considered acceptable as the number of cloud services deployed by a single resource provider are usually not as many as the services in grid sites.

The LDIF file to be loaded in the local LDAP server is generated by a prototype custom script, and the information published is the following:

- Cloud computing resources
- Service endpoint
- Capabilities provided by the service, such as: virtual machine management or snapshot taking. The labels that identify the capabilities are agreed within the taskforce.
- Interface, the type of interface – e.g. webservice or webportal – and the interface name and version, for example OCCI 1.2.0

- User authentication and authorization profiles supported by the service, e.g. X.509 certificates
- Virtual machines images made available by the cloud provider
- Operating system, and other environment configuration details
- Maximum number of cores – and physical memory – allocable in a single virtual machine

Clearly, multiple virtual machine types can be associated to a single cloud service. There are no limitations in the number of services published by a resource centre either. Currently the information published is modelled using the latest GLUE2.0 schema definition. An extension of the GLUE schema is under development to address the specific requirements of Cloud resources and add information about storage and network services.

The Federated Cloud taskforce deploys a central Top-BDII that automatically pulls the information from the local LDAP servers of the resource providers. This service can be used as a single entry point to query for all the resource centres supported by the test-bed, by users or other automatic tools.

Currently the central federated cloud Top-BDII is reachable at this address: ldap://test03.egi.cesga.es:2170

An example of a possible query is:

```
ldapsearch -x -H ldap://test03.egi.cesga.es:2170 -b o=glue
'(objectClass=GLUE2Endpoint)' | perl -p00e 's/\r?\n //g' | grep -E
'GLUE2EndpointURL|GLUE2EndpointInterfaceName|GLUE2EndpointInterfaceV
ersion|dn\:' | awk '{printf("%s%s", $0, (NR%4 ? " === " : "\n"))}' |
awk '{print ""$2" "$5" "$8" "$11}' | awk -F "GLUE2DomainID=" '{print
$2}' | awk -F "," '{print $1 " "$3}' | awk '{print $1" "$4" "$3"
"$5}' | sort
```

This query provides the interface names exposed by all the resource providers, the version of the implemented interface, and the endpoint that can be used to contact the service through the specific interface.

From a technical standpoint the resource centre LDAP server must answer to the port 2170, in order to be automatically polled by the Top-BDII

4.2 Central service registry: GOCDB

EGI's central service catalogue is used to catalogue the static information of the production infrastructure topology. The service is provided using the GOCDB tool that is developed and deployed within EGI. To allow Resource Providers to expose Cloud resources to the production infrastructure, a number of new service types were added to GODCB:

- eu.egi.cloud.accounting
- eu.egi.cloud.information.bdii
- eu.egi.cloud.storage-management.cdmi
- eu.egi.cloud.vm-management.occi
- eu.egi.cloud.vm-metadata.marketplace

Until such time as EGI is integrating federated Cloud resources into production, all registered Cloud resources are maintained in test-bed mode.

4.3 Monitoring: SAM

The SAM (Service Availability Monitoring) system is a framework consisting of:

- Nagios monitoring system (https://www.nagios.org),

- Custom databases for topology, probes description and storing results of tests
- web interface MyWLCG/MyEGI (https://tomtools.cern.ch/confluence/display/SAM/MyWLCG)

Probes to check functionality and availability of services must be provided by service developers. More information on SAM can be found at https://wiki.egi.eu/wiki/SAM. The current set of probes used for monitoring cloud resources consists of:

- OCCI probe: Creates an instance of a given image by using OCCI and checks its status
- BDII probe: Basic LDAP check tries to connects to cloud BDII
- Accounting probe: Checks if the cloud resource is publishing data to Accounting repository
- TCP checks: Basic TCP checks used for CDMI services.

A central SAM instance specific to the activities of the task force has been deployed for monitoring fedcloud resources (https://cloudmon.egi.eu/nagios). The available probes are in flux and as such once finalized these will be included into official SAM release. Adding probes to official SAM will follow procedure “Adding new probes to SAM” (https://wiki.egi.eu/wiki/PROC07).

4.4 Accounting

To account for resource usage across the resource providers the following have been defined:

- The particular elements or values to be accounted for;
- Mechanisms for gathering and publishing accounting data to a central accounting repository;
- How accounting data will be displayed by the EGI Accounting Portal.

The EGI Federated Cloud Task Usage Record is based on the OGF Usage Record format [R 4], and extends it where necessary[14]. It defines the data elements, which resource providers should send to the central Cloud Accounting repository. These elements are as follows:

Key	Value	Description	Mandatory
VMUUID	string	Virtual Machine's Universally Unique IDentifier	Yes
SiteName	string	Sitename, e.g. GOCDB Sitename	Yes
MachineName	string	VM Id	
LocalUserId	string	Local username	
LocalGroupId	string	Local groupname	
GlobalUserName	string	User's X509 DN	
FQAN	string	User's VOMS attributes	
Status	string	Completion status - started, completed, suspended	
StartTime	int	Must be set if Status = Started (epoch time)	
EndTime	int	Must be set if Status = completed (epoch time)	
SuspendDuration	int	Set when Status = suspended (seconds)	
WallDuration	int	Wallclock - actual time used (seconds)	
CpuDuration	int	CPU time consumed (seconds)	
CpuCount	int	Number of CPUs allocated	
NetworkType	string	Description	
NetworkInbound	int	GB received	
NetworkOutbound	int	GB sent	
Memory	int	Memory allocated to the VM (MB)	
Disk	int	Disk allocated to the VM (GB)	

[14] Once the format stabilizes and has proven its use, it will be submitted to the OGF for standardization.

StorageRecordId	string	Link to associated storage record	
ImageId	string	Image ID	
CloudType	string	e.g. OpenNebula, Openstack	

Scripts have been provided for OpenNebula and Openstack implementations to retrieve the accounting data required in this format ready to be sent to the Cloud Accounting Repository. These scripts are available from:

- OpenNebula – https://github.com/EGI-FCTF/opennebula-cloudacc
- Openstack – https://github.com/EGI-FCTF/osssm
- StratusLab – identical to OpenNebula
- WNoDeS – internal WNoDeS component

The APEL SSM (Secure STOMP Messenger) package is provided by STFC for resource providers to send their messages to the central accounting repository. It is written in Python and uses the STOMP protocol, the messages contain cloud accounting records as defined above in the following format (example data added) where %% is the record delimiter:

```
APEL-cloud-message: v0.2
VMUUID: https://cloud.cesga.es:3202/compute/47f74797-e9c9-46d7-b28d-
5f87209239eb 2013-02-25 17:37:27+00:00
SiteName: CESGA
MachineName: one-2421
LocalUserId: 19
LocalGroupId: 101
GlobalUserName: NULL
FQAN: NULL
Status: completed
StartTime: 1361813847
EndTime: 1361813870
SuspendDuration: NULL
WallDuration: NULL
CpuDuration: NULL
CpuCount: 1
NetworkType: NULL
NetworkInbound: 0
NetworkOutbound: 0
Memory: 1000
Disk: NULL
StorageRecordId: NULL
ImageId: NULL
CloudType: OpenNebula
%%
...another cloud record...
%%
...
%%
```

The OpenNebula and Openstack scripts produce the messages to be sent using the SSM package in the correct format.

The SSM package can be downloaded from http://apel.github.io/apel/

Detail about configuring SSM and publishing records may be found here:

https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Scenario4#Publishing_Records

SSM utilizes the network of EGI message brokers and is run on both the Cloud Accounting server at STFC and on a client at the Resource Provider site. The SSM running on the Cloud Accounting server receives any messages sent from the Resource Provider SSMs and they are stored in an “incoming” file system.

A Record loader package also runs on the Cloud Accounting server and checks the received messages and inserts the records contained in the message into the MySQL database.

A Cloud Accounting Summary Usage Record has also been defined and the Summaries created on a daily basis from all the accounting records received from the Resource Providers is sent to the EGI Accounting Portal. The EGI Accounting Portal also runs SSM to receive these summaries and the Record loader package to load them in a MySQL database storing the cloud accounting summaries.

The EGI Accounting Portal provides a web page displaying different views of the Cloud Accounting data received from the Resource Providers: http://accounting.egi.eu/cloud.php

4.5 Image metadata publishing & repository

The task force uses the appliance repository and marketplace developed by StratusLab[15] as repositories for images and their metadata. IaaS providers endorse images that are suitable for their infrastructure by signing their metadata and uploading them on the marketplace (https://marketplace.egi.eu) and make the image available either to EGI appliance repository (https://appliance-repo.egi.eu) or their local appliance repository. The user is then able to browse the metadata for suitable images to instantiate in one of the federated IaaS.

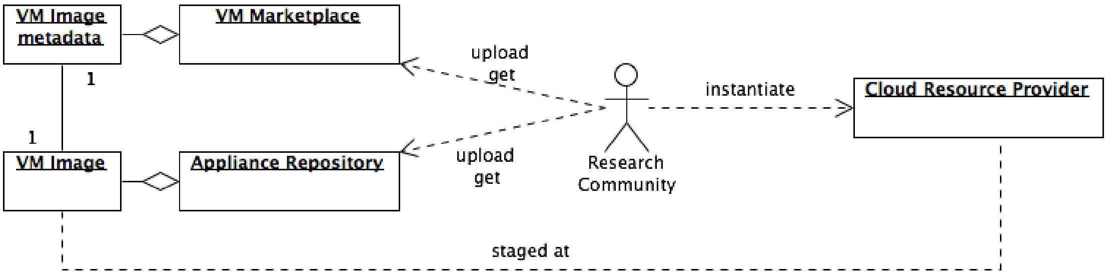

Figure 7: Using the VM Marketplace and an Appliance repository to instantiate a VM at a federated Cloud Resource Provider.

[15] http://www.stratuslab.eu

5 FEDERATING CLOUD RESOURCES TO EGI

The model of federation chosen in the EGI is one where all resources available to the user are equal. Therefore this is therefore suitable for resource consumers who do not have their own cloud infrastructure available to them. Within this section we detail the concepts behind the method of federation chosen and how those technology providers within the task force have been able to adapt or integrate their technologies. We therefore have for each technology provider a section which details work done for integration and any specific configuration a deployer of this technology needs to do to connect to EGI. We also describe the technology premise that the activity itself started with.

During the task force stage of the activity membership of the federated cloud activity was obtained through approach to the activity chair and attendance at the weekly group meeting. As we move towards a production infrastructure we must move beyond this to a moderated and measured basis by which resource providers are able to claim membership of the cloud. As such we aim to build upon the different procedures already in place as well as the standards by which different interfaces to resources are supported.

This will include a certification process[16] by which official membership of the cloud is allowed and through the dedication of resources to the activity. This will be measured through the monitoring process previously described and passing of tests, which will enable a certification of the resources a provider gives. We make no distinction as to the resource access model in terms of free at the point of use, charge at the point of use, bulk buy or other models of financial reconciliation.

In the following sections the underlying functions that the EGI Federated Cloud supports are described along with for each available technology the level of integration and any necessary changes from the default version of that technology.

5.1 Overview of requirements scenarios

The initial plan for federation of cloud resources within EGI was based on 6 different functional requirements that a user or community may have with regard to cloud technologies. Though dealt with separately we envisaged that the scenarios in some cases would build upon each other. These scenarios have expanded in number since original formation to include 4 further scenarios.

5.1.1 Scenario 1: VM Management

```
"I want to start a single existing VM image on a remote
cloud."
```

This scenario describes the details of managing the operation of a specific single VM image. It intentionally ignores any other cloud type functionality including data and information management. The key aspect here is the use of virtualization to separate consumer from provider with the focus of this scenario lying on VM management operations.

5.1.2 Scenario 2: Managing my own data

This scenario extends scenario 1 by adding the following statements:

```
"I want to start a VM instance from an image that I have created."
"I want to associate my running VM with a data set in the Cloud."
"I want to take snapshots of my running VM for restart purposes"
```

[16] https://wiki.egi.eu/wiki/PROC09_Resource_Centre_Registration_and_Certification

This scenario extends the usage of a federated Cloud deployment by mixing in the capability to configure remote or (Cloud provider-) local data storage for use while the VM is executing.

Additional use cases that fall under the same scenario are:

- Using custom VM images created/administered by someone external to the Cloud provider:
  - If not provided by other means, some VM image upload/download facilities are required;
  - Storage facilities for VM images;
- Support for local, and remote storage locations to be configured for the VM.
- Taking a snapshot of a running VM

5.1.3 Scenario 3: Integrating multiple resource providers

This scenario includes scenario 2, plus the following statements:

> “I want to choose on which resource provider I want to start my single VM.”
> “I need to know about the VMM capabilities the provider offers.”

This is the first scenario where the concept of multiple different cloud providers is introduced. The statements indicate that a user (or user group) must be able to decide with which resource provider (or a group) he or she would want to engage in business with. To allow this the scenario deals with information publication and dissemination across resource providers as follows:

- Information must be conveyed in a comparable manner (preferably through an open standard)
- Information must be publicly available
- Information must be human-readable, as well as accessible for automated queries (i.e. through an API)

5.1.4 Scenario 4: Accounting across Resource Providers

This scenario includes scenario 3, plus the following statements:

> “My usage across different resource providers needs to be recorded and reported to multiple aggregators.”

This scenario deals with how to account for resource usage. Building on the well-understood accounting of resource currently within EGI addition questions within the scenario are;

- What actually are resources that may be consumed, and thus be accounted for? (Not to forget billing for commercial providers!)
- Once identified, what is the accounting unit for such resources?
- What is the metering interval/frequency? Is this identical across providers, or must this be provided as part of the information available in Scenario 3?
- At which level of detail should be accounting data collected?
- Where should the accounting data be stored? And who shall have access to it (on which detail?)

5.1.5 Scenario 5: Reliability/Availability of Resource Providers

To build a production infrastructure users must have confidence in the availability of resource sufficient to operate their tasks. This scenario includes scenario 4, plus the following statements:

> “Information relating to the reliability/availability and current status of the remote virtualised resource needs to be available to me.”

This scenario deals with information about Resource Provider availability, which may influence a user's choice in selecting Resource Providers to engage with for further business. Also, a hypothetical Cloud federation may also put certain constraints on its members in terms of minimum availability and reliability in order to remain member of the federation. The following questions and issues need resolution:

- What are the exact semantics for availability and reliability?
- Which services are under monitoring for availability?
- How and where is this information collected and published?

5.1.6 Scenario 6: VM/Resource state change notification

> “When the status of the [VM] instance I am running changes (or will change) I want to be told about it.”

This scenario supports the concept that any change in state of a resource or instance that a user or community are using should result in them being told about it.

- Reactive feedback about events in the past must be given.
- Proactive notification of *planned* changes must be provided, too.
- What is the format of notification?
- What are machine-readable requirements for notification to facilitate automation and user-managed reliability?

5.1.7 Scenario 7: AA across Resource Providers

> “I want to use my existing identity, and not re-apply for new credentials to use the service.”

In common with many other activities across the research space the federated cloud should make use of federated identity. This will normally allow for a person to assert their identity based upon their employer when within the academic space or some other trusted identity provider. This may utilise online or token based technology and as such we would not desire to build our own but rather adopt a well-supported technology from elsewhere when available.

5.1.8 Scenario 8: VM images across Resource Providers

> “I want to use a single VM image across multiple different infrastructure providers”

This scenario deals with the requirement that the management of a user’s VMs should be as simple as possible and when they have created an instance that they may wish to deploy widely across multiple providers then this should occur from a single catalogue. This scenario deals with the following issues:

- Provide a mechanism so that a user can upload transparently his own image to the test bed, with a unique global ID.
- Provide a common place to add an endorsement to a pertinent VM so that the resource providers can trust it.

5.1.9 Scenario 9: Brokering

> "I want my VM instance to run on a resource that is suitable based on a set of policies or requirements rather than my choosing directly which resource will run it"

A user must be able to easily and quickly decide which resource they wish to use and as such there must be a cloud brokering service. The goal is for a user to have a choice between a unified, abstracted view of the cloud test bed as a whole and the opportunity to target specific providers for their needs. As a consequence, this scenario is concerned with both brokers and management interface clients.

5.1.10 Scenario 10: Contextualisation

> "When I deploy a VM instance on a resource I must be able to give it configuration information for customisation of the default template. This can only happen when it is up and running"

Users must be able to configure automatically VM instances once they have been deployed on resources. Since they may be deployed on multiple resource providers this must take place automatically. There are a number of different possibilities for this type of configuration that the scenario explores. This will also allow resource providers to add any specific requirements on configuration they give on user communities.

5.2 OpenNebula

A new Resource Provider using OpenNebula or OpenNebula-based CMF has to take the following steps to join the EGI Cloud Federation. There is only one prerequisite and that is fully functional OpenNebula installation capable of deploying, sustaining and shutting down virtual machines. There are no requirements for the underlying architecture. Resource Providers in question may choose the virtualization platform, network and storage configuration according to their preferences and needs. It is highly recommended to install OpenNebula v3.8.x where x denotes the latest security update and coordinate any future upgrades with the task force to avoid infrastructure fragmentation. Resource providers installing OpenNebula from scratch should follow its step-by-step installation and configuration guides available online[17].

The actual integration with the EGI Cloud Federation consists of the following steps:

1. Additional OpenNebula configuration
2. rOCCI-server installation and configuration
3. Integration with user management service -- Perun
4. Integration with accounting service -- APEL
5. Integration with image management service -- vmcaster/vmcatcher
6. Integration with information system -- LDAP/BDII
7. Registration of deployed services in GOCDB

Each of the above-mentioned steps is a requirement for every Resource Provider wishing to join the EGI Cloud Federation. Resource Providers are welcome to deploy and offer additional services such as object storage (CDMI) but this is not a requirement at this time. Detailed description of the listed steps is as follows.

Additional OpenNebula configuration

[17] http://opennebula.org/documentation:archives:rel3.8

Integration with EGI Cloud Federation requires the use of X.509 authentication mechanism in communication with OpenNebula. Resource Providers are encouraged to follow the step-by-step configuration guide provided by OpenNebula developers available online[18]. There is no need to change authentication driver for the *oneadmin* user or create any user accounts manually at this time.

rOCCI-server installation and configuration

The EGI Cloud Federation uses OCCI as its VM management protocol. It is necessary to install a fully compliant OCCI 1.1 server on top of RP's existing OpenNebula installation. OpenNebula's OCCI implementation is *not* compliant with the OCCI 1.1 specification. This functionality is provided by the rOCCI-server project. Detailed installation and configuration instructions are available online in the FedCloud wiki[19].

Integration with Perun

The current rOCCI-server implementation doesn't handle user management and identity propagation hence an integration with a third-party service is necessary. The Perun management server developed and maintained by CESNET is used to provide user management capabilities for OpenNebula Resource Providers[20]. It uses locally installed scripts (fully under the control of the Resource Provider in question) to propagate changes in the user pool to all registered Resource Providers. They are required to install and configure (if need be) these scripts and report back to EGI Cloud Federation for registration in Perun. Installation and configuration details are available online in the EGI-FCTF repository on GitHub[21].

Integration with APEL

One of the required integration points is accounting. The EGI Cloud Federation employs APEL framework with extended accounting records. Every Resource Provider is required to install the APEL SSM client and OpenNebula accounting script. As with the previous cases, installation and configuration details are available online on GitHub and in the FCTF wiki[22,].

Integration with vmcaster/vmcatcher

Resource Providers are required to integrate their OpenNebula with an image management service used within the federation. As with the previous cases, installation and configuration details are available online in the FCTF wiki[23]. This service ensures that all images are trusted and up-to-date for all Resource Providers across the federation.

Integration with TopBDII

Details about services offered by the Resource Provider in question are advertised to the rest of the EGI Cloud Federation using an LDAP server -- BDII. Resource Providers are encouraged to follow instructions available online in the FCTF wiki[24].

Registration in GOCDB

The procedure for registration of a resource provider within GOCDB is as per other types of resources within the EGI infrastructure[25].

[18] http://opennebula.org/documentation:archives:rel3.8:x509_auth

[19] https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:_Federated_AAI:OpenNebula

[20] http://perun.metacentrum.cz/web/

[21] https://github.com/EGI-FCTF/fctf-perun

[22] https://github.com/EGI-FCTF/opennebula-cloudacc

[23] https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Scenario8:Configuration#VMcatcher

[24] https://wiki.egi.eu/wiki/Fedclouds_BDII_instructions

5.3 OpenStack

This section describes steps necessary for new Resource Provider (RP) using Openstack middleware to join EGI Cloud Federation. It is strongly recommended using the last Openstack version. Specifically, the VOMS-enable authentication will require Grizzly version of Keystone. The installation and configuration of OpenStack are available at[26].

The actual integration with the EGI Cloud Federation consists of the following steps:

a) VOMS-enable Keystone installation and configuration
b) OCCI installation and configuration
c) Integration with accounting service APEL
d) Integration with vmcaster/vmcatcher
e) Integration with information system LDAP/BDII
f) Registration of deployed services in GOCDB

Each of the above-mentioned steps is a requirement for every Resource Provider wishing to join the EGI Cloud Federation. Resource Providers are welcome to deploy and offer additional services such as object storage (CDMI) but this is not a requirement at this time. Detailed description of the listed steps is as follows.

a) VOMS-enable Keystone installation and configuration

The installation and configuration of VOMS-enable Keystone is available at[27]. That will enable X.509 authentication mechanism and allows users with valid VOMS proxy certificate to log in. The actual VO for EGI Cloud Federation fedcloud.egi.eu should be enabled in the configuration. There is an option for automatically creating new users for trusted VO on the fly.

b) OCCI installation and configuration

The steps of installation and configuration of OCCI is available at [28]. The installation and configuration should be done on the machine with Nova server. The OCCI implementation is not perfect; occasionally Nova server needs to be restarted for refreshing OCCI configuration (especially when new images are added).

c) Integration with accounting service APEL

Like RP with OpenNebula, the client for accounting service APEL must be installed and configured. The details of installation and configuration of APEL for Openstack is available at[29,30].

d) Integration with vmcaster/vmcatcher

Resource Providers are required to integrate their Openstack with an image management service used within the federation. Installation and configuration details are available online in the FCTF wiki[31]. This service ensures that all images are trusted and up-to-date for all Resource Providers across the federation. In addition to vmcaster/vmcatcher, glancepush-vmcatcher[32] uses vmcatcher's event handler to signal glancepush that a new image was updated in vmcatcher's cache and glancepush will check and publish images from vmcatcher cache to glance service in Openstack.

[25] https://wiki.egi.eu/wiki/GOCDB/Input_System_User_Documentation
[26] http://docs.openstack.org/install/
[27] http://keystone-voms.readthedocs.org/en/latest/index.html
[28] https://github.com/stackforge/occi-os
[29] https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Scenario4
[30] https://github.com/EGI-FCTF/osssm/wiki
[31] https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Scenario8:Configuration#VMcatcher
[32] https://github.com/EGI-FCTF/glancepush

e) Integration with information system LDAP/BDII

Integration with BDII for RP with Openstack is identical as in the OpenNebula case. The instructions are available online in the FCTF wiki[33].

5.4 StratusLab

A StratusLab cloud based production release offers computing, storage, and networking services as well as the Marketplace, a high-level service to facilitate sharing of machine images. Actually StratusLab integrates and uses OpenNebula as the Virtual Machine Manager. In order to technically integrate a StratusLab Cloud with the EGI Cloud Federation, the following steps should be implemented:

Authentication: X509/VOMS authentication is done by creating users in the OpenNebula VMM service with the X509 driver. This driver should be enabled in the OpenNebula configuration file.

As the VOMS validation is passing through Apache2, grid_site and then rOCCI-server. Apache2 and grid_site should be properly configured. In our case, install gridsite packages, then load gridsite module in apache configuration file.

Compute: Nothing to do, rOCCI-server is well integrated with OpneNebula, and StratusLab actually integrates OpenNebula as it's Virtual Machine Manager.

Network: StratusLab networking service permits the allocation of "public", "private" and "local" network. When creating VM templates one of these networks should be specified. Public IP are visible from outside and inside the cloud. Local IP are visible from the Cloud (VMs running in the Cloud), access external services through NAT, this type of IP addresses could be useful for MPI jobs.

Private IP are visible only fron the host where they are running, go to outside via NAT.

Storage: StratusLab developed it's own storage service solution based on a disk approach. One of its functionality is to cache machine images, making deployment of VM very fast.

This service is well integrated with the other StratusLab services, and doesn't need any additional configuration for rOCCI-server.

Marketplace: We are using StratusLab Marketplace to instantiate VM in the StratusLab Cloud. The StratusLab Marketplace is like a registry of images metadata. Unlike the other RPs, in the metadata we don't specify network nor storage elements. Instead in the VM templates it's mandatory to specify identifier image URL from the marketplace in the SOURCE filed. (e.g. SOURCE=http://marketplace.egi.eu/metadata/HGSxEvjFP0TUo1mMcT-M63Y-2KF/airaj@lal.in2p3.fr/2013-02-12T14:11:55Z). NB. In StratusLab, OpenNebula is used only as VMM service, and in the near future this will be replaced with integration done directly with libvirt.

The current configuration will works till the StratusLab 13.05 release, OpenNebula as VMM will be dropped in the StratusLab 13.08 release, in August 2013.

5.5 WNoDeS

WNoDeS cloud release 3.0.0-1 offers computing service as well as information management services to keep trace of all the virtual machines currently running for each hypervisor and all the virtual

[33] https://wiki.egi.eu/wiki/Fedclouds_BDII_instructions

machines stored in the configured repository [R 5, R 6]. Details of how to deploy, install and configure WNoDeS to support cloud provisioning are specified in the system administration guide [R 7]. This release is part of the EMI-3 distribution [R 8]. Up to now, WNoDeS has developed a subset of the OCCI 1.1 interface that is computing specific.

To technically integrate a WNoDeS cloud with the EGI Cloud Federation the following steps have been performed:

Integration with the accounting service APEL

The client for accounting service APEL has been implemented and included in the WNoDeS release 3.0.0-1.

Integration with the marketplace

The CLOUD CLI has been upgraded in order to get metadata image information not only from the WNoDeS information service but also from the EGI Marketplace endpoint. This component has been included in the WNoDeS release 3.0.0-1.

5.6 Synnefo

Synnefo (http://www.synnefo.org/) is open source cloud software used to create massively scalable IaaS clouds. It uses Google Ganeti for the low level VM management and also talks to the outside world through the OpenStack APIs with extensions for advanced operations. Synnefo in conjunction with Google GANETI (https://code.google.com/p/ganeti/) is the software that empowers GRNETs ~Okeanos service (https://okeanos.grnet.gr) that currently supports 2100 users with 2941 VMs and 10119 Virtual cores. ~Okeanos is only partially integrated with the Federated cloud infrastructure using snf-occi (http://www.synnefo.org/docs/snf-occi/latest/index.html), an implementation of the OCCI specification on top of synnefo’s API kamaki. Development for the rest of the modules required is currently foreseen for the near future but due to lack of manpower and parallel developments of the synnefo API there is no estimate for the date of delivery for each module.

6 CONCLUSION

This document allows a provider of cloud infrastructures for research to understand both the technical and policy requirements that are placed upon them by membership of the EGI Federated Cloud. Using the input from this document a provider can make a balanced decision on the type of cloud software they wish to deploy, how much work is required on top of the cloud installation procedure is needed to federate the cloud resource with others, and where the different other services that are needed to connect to the infrastructure are used within the federation. It has also been shown how the resource provider, to enhance the services they are able to provide, may broaden the types of research communities and applications that they are able to support.

This document also captures the state of the cloud federation at the end of the third development phase as the Federated Cloud moves towards a production infrastructure. This also shows how the external operations and structure for the support of services within EGI integrate with possible internal services that the provider may operate to support other communities outside of EGI.

The experiences, changes to technologies etc. are all tested with real experiences by providers that have deployed the various different technologies that are described within this document.

7 REFERENCES

R 1	R. Nyren, A. Edmonds, A. Papaspyrou, and T. Metsch, “Open Cloud Computing Interface - Core," GFD-P-R.183, April 2011. [Online]. Available: http://ogf.org/documents/GFD.183.pdf
R 2	T. Metsch and A. Edmonds, “Open Cloud Computing Interface - HTTP Rendering," GFD-P-R.185, April 2011. [Online]. Available: http://ogf.org/documents/GFD.185.pdf
R 3	“Open Cloud Computing Interface - Infrastructure," GFD-P-R.184, April 2011. [Online]. Available: http://ogf.org/documents/GFD.184.pdf
R 4	R. Mach, R. Lepo-Metz, S. Jackson, L. McGinnis, “Usage Record – Format recommendation" GFD-P-R.98, September 2006. https://www.ogf.org/documents/GFD.98.pdf
R 5	Elisabetta Ronchieri, Giacinto Donvito, Paolo Veronesi, Davide Salomoni, Alessandro Italiano, Gianni Dalla Torre, Daniele Andreotti, Alessandro Paolini, “*Reource Provisioning through Cloud and Grid Interfaces by means of the Standard CREAM CE and the WnoDeS Cloud Solution*,” PoS(EGICF12-EMITC2)124.
R 6	Davide Salomoni, Alessandro Italiano, Elisabetta Ronchieri, “*WNoDeS, a Tool for Integrated Grid and Cloud Access and Computing Farm Virtualization*,” 2011 Journal of Physics: Conference Series Volume 331 Part 5: Computing Fabrics and Networking Technologies.
R 7	System Administration Guide, http://web2.infn.it/wnodes/images/stories/doc/wnodes-sysadminguide_v_1_1_0_2.pdf
R 8	Cristina Aiftimiei, Andrea Ceccanti, Danilo Dongiovanni, Alberto Di Meglio, Francesco Giacomini, “*Improving the quality of EMI Releases by leveraging the EMI Testing Infrastructure*,” 2012 Journal of Physics: Conference Series Volume 396 Part 5.