



EGI-InSPIRE

EVALUATION AND WORK PLAN FOR CLOUDCAPS USE CASES

Document identifier:	2013-06-17-Use-Case-Evaluation MD.docx
Date:	07/08/2013
Activity:	SA4
Lead Partner:	JUELICH
Document Status:	FINAL
Dissemination Level:	PUBLIC
Document Link:	https://documents.egi.eu/document/1824

Abstract

This document is about the initial evaluation of cloud capabilities and use cases. We have matched cloud capabilities with use cases in order to derive the next steps of the project. We selected two use cases, which will be the primary focus of our support during the next months.

I. COPYRIGHT NOTICE

Copyright © Members of the EGI-InSPIRE Collaboration, 2010-2014. See www.egi.eu for details of the EGI-InSPIRE project and the collaboration. EGI-InSPIRE (“European Grid Initiative: Integrated Sustainable Pan-European Infrastructure for Researchers in Europe”) is a project co-funded by the European Commission as an Integrated Infrastructure Initiative within the 7th Framework Programme. EGI-InSPIRE began in May 2010 and will run for 4 years. This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, and USA. The work must be attributed by attaching the following reference to the copied elements: “Copyright © Members of the EGI-InSPIRE Collaboration, 2010-2014. See www.egi.eu for details of the EGI-InSPIRE project and the collaboration”. Using this document in a way and/or for purposes not foreseen in the license, requires the prior written permission of the copyright holders. The information contained in this document represents the views of the copyright holders as of the date such views are published.

II. DELIVERY SLIP

	Name	Partner/Activity	Date
From	Björn Hagemeyer Boris Parak Miroslav Ruda		17/06/2013
Reviewed by	Moderator: Reviewers: Michel Drescher		
Approved by	AMB & PMB <<To be completed by project office on submission to EC>>		

III. DOCUMENT LOG

Issue	Date	Comment	Author/Partner
1		First draft	
2	21 Jun 2013	Review	Michel Drescher/EGI.eu
3	06 Aug 2013	Implemented recommended changes	Björn Hagemeyer

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 document is about the initial evaluation of cloud capabilities and use cases. We have matched cloud capabilities with use cases in order to derive the next steps of the project.

In the following sections, we have reported our findings about image characteristics as they stand at the start of the project. Following that, we briefly describe the questionnaire that we sent to the user communities and a summary of their responses. After a description of cloud capabilities, we mention the use cases and how they could benefit from better exploiting cloud capabilities. In the end, we selected the WeNMR and BioVel use cases for further improvement.



TABLE OF CONTENTS

1 INTRODUCTION	7
2 IMAGE CHARACTERISTICS.....	8
3 QUESTIONNAIRE.....	9
4 CLOUD CAPABILITIES	10
4.1 Core Capabilities.....	10
4.2 Advanced Capabilities	10
5 ALL USE CASES.....	13
5.1 OpenModeller/BioVel (Boris).....	13
5.2 WeNMR (Björn).....	14
5.3 BNCWeb (Boris)	15
5.4 WS-PGRADE (Boris).....	16
5.5 DIRAC (Boris).....	16
6 USE CASE SELECTION	17
7 SUMMARY	18
8 APPENDIX A.....	19
9 REFERENCES	24



1 INTRODUCTION

The goal of mini-project CloudCaps (Transforming Scientific Use Cases to Exploit Cloud Capacity) is to help scientific user communities adapt their use cases to cloud computing in general, with a particular focus on the EGI Cloud Infrastructure Platform. With cloud capabilities, we refer to features such as scalable object storage, attachable block storage, and on-demand elasticity, to mention a few. Another topic is the preparation of minimal base images to avoid extremely large images and at the same time allow for the maximum possible usability in all federated resource providers. Our project follows a two-fold approach. First of all, we will enable select applications to make use of cloud-specific features. Secondly, we will derive best practices from our activity and provide these as generic documentation tailored towards scientific applications using the EGI Cloud Infrastructure Platform.

This document is about the initial evaluation of cloud capabilities and use cases. We have matched cloud capabilities with use cases in order to derive the next steps of the project.

In the following sections, we have reported our findings about image characteristics as they stand at the start of the project. Following that, we briefly describe the questionnaire that we sent to the user communities and a summary of their responses. After a description of cloud capabilities, we mention the use cases and how they could benefit from better exploiting cloud capabilities. We conclude with the use cases that we select for our further work and a brief summary.

2 IMAGE CHARACTERISTICS

One of the motivations to start our project were recurring VM Images stemming from our User Communities that shipped with huge amounts of unneeded and/or static data that could have been stored in other locations than the images themselves. Both lead to very large images that severely slow down the instantiation process and don't scale to large numbers of instances due to size constraints in the resource providers' compute nodes.

Essentially, we found that the images provided by the user communities were often ordinary desktop installations containing a graphical user environment and software that would otherwise only be used in an office environment, for example OpenOffice, web browsers, even games. In one image, the games and some other packages had been removed, but their configuration remained. As these images were all intended for automated usage and management, hence not requiring interactive access via a graphical environment, many of these packages are superfluous and subject to removal, thus reducing the transfer time and storage resources required for their use.

In addition to the large number of packages installed, the images were equipped with locally (i.e. manually) installed software, most of which was certainly required for the actual application that the image had been prepared for. However, even here we found software that was clearly targeted at development environments, e.g. the Jenkins continuous integration server, which had already created an extra 2G of data that was shipped along with the image.

Depending on the type of image (e.g. raw or qcow2) some cloud management frameworks (e.g. OpenStack) convert the image prior to instantiating a virtual instance from it. In one case, the transfer time and conversion time added up to more than 30 minutes. Naturally, this conversion and transfer only has to be done once on each compute node, depending on policy and installation details of the site. From our point of view, this tremendous amount of time can be avoided by following simple guidelines that are subject of our project.

More information about the characteristics of images can be found at [4]. This information will be used to measure the success of this project. We will compare the original images as obtained at the start of the project with those that will be created according to our suggestions.



3 QUESTIONNAIRE

During May 2013, we distributed a questionnaire to use cases interested in participating in and using resources of FCTF. The questionnaire had been carefully prepared and tailored to each use case, in order to maximize the number of responses. Questionnaires were delivered to the following user communities:

- OpenModeller/BioVel
- WeNMR
- BNCweb
- PeachNote
- WS-PGRADE
- DIRAC
- DCH-RP

The questionnaire contained questions from various areas, including image preparation and management, workload management, authentication and contextualization, data management.

The full set of questions and the project specific questions including answers can be found in appendix A.

4 CLOUD CAPABILITIES

Among others, one of the initial tasks of the project was to identify typical cloud capabilities in order to match them with use cases and make educated suggestions, how user communities could benefit the most from each of the capabilities.

4.1 Core Capabilities

It was apparent from early on that there is a natural separation between basic and advanced capabilities. For example, basic capabilities can be used from within virtual instances and only require minimal configuration outside the instance. The capabilities we considered here were block storage and object storage.

Block storage can be used transparently by the application inside the virtual instance, as it appears like any other block device attached to the machine. Typically, one would format the block device with a file system and thus make it available to the application.

Object storage is scalable storage in which data objects can be stored. The FCTF [5] have agreed on CDMI to be the access mechanism of choice [6]. Applications within virtual instances need to make explicit use of object storage in order to make use of it. This is a key difference in comparison to block storage. A potential solution to overcome this limitation is to try and mount object storage as a file system, e.g. via a FUSE driver. There are hints that such solutions exist. In the case of OpenStack Swift, object storage can also be accessed via plain HTTP methods, making it available even without dedicated clients and also without the requirement for authentication. Object storage could thus also be used as a content provider for static data, such that it does not need to be shipped with virtual machine images. This will also avoid the problem of having to update large images again and again in case new layer data is added.

4.2 Advanced Capabilities

These capabilities are what can also be described as cloud platform services. They are more complex to configure and most often require extra actions to be taken. The following is a list of advanced capabilities that can be offered, if use cases indicate their need for them. At the time of writing this document, these capabilities do not have direct integrations with any of the cloud management platforms, but there is some work ongoing to provide and integrate them in the future, e.g. OpenStack Heat is an activity to provide AWS like CloudFormation with the Havana release due in September.

Most of the columns in the following table are self-explanatory, but the accounting column deserves a short explanation. From a provider's perspective, it is interesting to know whether the use of a capability can be accounted for and how. This is what we try to convey in this column.

ID	Capability	Description	Accounting	Technical Solution(s)
3	Messaging	A messaging system can be used to decouple communication among distributed components of cloud "applications". At the same time, this capacity can be used at the infrastructure level. Messaging is a higher level	Yes, number of messages, maybe their size. SQS accounts number of sends and receives separately. Bulk send/receive	AMQP <ul style="list-style-type: none"> • ActiveMQ • RabbitMQ Need to be careful with AMQP protocol versions, 0.9.1 seems entirely different from 1.0. ActiveMQ

		<p>service, which can be offered/used in several modes of operation:</p> <ol style="list-style-type: none"> 1. Directly within a VM, either a specific one deployed by the user or as part of a larger VM containing multiple components and services 2. As a platform service offered by the cloud. Users could simply rely on its availability and make use of it. <p>Examples of this kind of system are: Amazon SQS [7]</p>	cheaper.	support 1.0, RabbitMQ 0.9.1 (and will stick to it).
4	CloudFormation	CloudFormation is the capacity to deploy collections of related resources.		CloudFormation is the capacity to deploy collections of related resources. OpenStack Heat implements this for OpenStack and is scheduled for the Havana release in September
5	AutoScaling	<p>Many applications can benefit from automatic scaling either horizontally or vertically. We refer to vertical scaling as the resizing of existing cloud instances, whereas horizontal scaling means adding or removing instances on demand.</p> <p>AutoScaling can be supported by additional services such as rule engines and messaging</p>		



		services to provide the data and decisions for scaling.		
6	Database	Databases as a service, be it SQL or no-SQL.	Based on number of requests and size of DB.	

5 ALL USE CASES

5.1 *OpenModeller/BioVel*

OpenModeller is a generic framework that includes various modelling algorithms and is compatible with different data formats. Its functionality is available through the command line, desktop application and web service interfaces, allowing integration with other specialized services providing input data for the modelling process. This use case aims to setup and operate an instance of the web service in Europe using resources from the EGI Federated Cloud and from the EGI Service Availability Monitoring infrastructure. The instance would serve the Biodiversity Virtual e-Laboratory (BioVel) that supports research on biodiversity issues using large amounts of data from cross-disciplinary sources [8].

OpenModeller/BioVel use case could benefit from the following cloud capabilities:

1. **Block storage** -- The OpenModeller service is designed to work with a wide spectrum of data formats used in environmental modelling. Its outputs vary from relatively small XML files (products of model creation and testing) to large rasterized files (model projections) that have to be stored locally until they can be retrieved and stored in a more permanent storage element. This requires potentially large amounts of temporary non-persistent storage available locally to every OpenModeller instance. This makes OpenModeller/BioVel use case an ideal candidate for the block storage cloud capability.
2. **Object storage** -- The OpenModeller service uses so-called environmental layers as input for the modelling procedure. Environmental layers are currently stored as files and their copy is part of each OpenModeller image. Since they vary in size from a few hundred megabytes to tens of gigabytes and the OpenModeller service accesses them in a read-only mode, they could be stored in common object storage and retrieved on demand by the OpenModeller service instances. This would reduce image size, make data management easier and help with horizontal scaling across multiple resource providers. Leaving this data out of the images also eases the update of images, i.e. updates are smaller and no update is needed only because new layer data is available.
3. **Horizontal auto-scaling** -- The OpenModeller service processes jobs submitted by an external framework. When multiple service endpoints exist (are registered with the framework), available workload is distributed across all of them evenly. This allows for easy horizontal auto-scaling with a significant improvement in over-all processing time.

The provided OpenModeller/BioVel image is very well prepared. It doesn't contain a desktop environment; it's minimalistic and even in its raw format takes up only about 10 GB of space (including free space and a local copy of environment layers, both of which can be easily removed using the cloud capabilities mentioned above). All necessary software is already installed and configured. The image contains a few unnecessary packages related to the X window system and GCC compiler, however it is unclear whether they can be removed or are among required dependencies for some of the installed software. The image takes advantage of basic (platform-specific) contextualization to inject public SSH keys. The layer data inside the original image is only a subset of the production set, which is roughly 100GB.

We suggest the following improvements that would make the OpenModeller/BioVel image more cloud-friendly:

1. **Automated software installation**
2. **Automated software configuration**
3. **Removal of unnecessary packages and unused kernels/ramdisks/modules**
4. **The use of attached block storage for local computation results**
5. **The use of shared object storage for read-only environmental layers**

5.2 WeNMR

The objective of WeNMR [9] is to optimize and extend the use of the NMR and SAXS research infrastructures through the implementation of an e-infrastructure in order to provide the user community with a platform integrating and streamlining the computational approaches necessary for NMR and SAXS data analysis and structural modelling. Access to the e-NMR infrastructure is provided through a portal integrating commonly used software and GRID technology.

WeNMR could benefit from the following cloud capabilities:

1. **Horizontal auto-scaling** -- perfect candidate, as the input set of this community can be easily decomposed, and work units descriptions are pulled from a central token server.

The original image provided by this user community is quite large, despite the fact that it had already been reduced. The reduced size is roughly 8.6GB compressed. The reduction seemed to only have targeted the image file itself rather than its contents. It contains a full desktop operating system installation with automatic login and software packages that are not needed for the mode of operation described in their use case. The version of Ubuntu (11.04) is not supported anymore and would require an update. A number of locally installed packages and data exist in the main user's home directory, the need for which is at least questionable. Further details about the packages that we see in excess of the minimal requirements are available in our project wiki [10].

We suggest the following actions to improve the situation:

1. **Reduce image size** either by removing obsolete packages or creating a new minimal image from scratch. First candidates for removing software would cover much of the graphical environment as well as office related software and some games. We have generic instructions available to create minimal images with little effort and start working from there.
2. If in (1) above we choose to remove software rather than starting from scratch, **the following need to be considered:**
 1. There is a non-empty /swapfile1 of 1GB that is not even used in the default configuration. It should be removed.
 2. The directory /lib/modules contains kernel modules for 13 kernel versions, adding up to another 1G of data. Most of them can safely be removed.
 3. Another ~1G worth of data is located in /opt/intel, which we assume is an Intel Compiler. Is this strictly required?
 4. Sort out the really required, locally installed packages in /home/i/workspace, adding up to several GB of data, some of which may only be needed in a development environment.
3. **Automated configuration** of the instances, i.e. **contextualization**. At the moment, the token



server is hard-coded in the image, such that work items get processed independently of the user who instantiated the image. This is not always desirable.

5.3 BNCWeb

BNCWeb is an interface to the British National Corpus, a dataset of 100 million words, carefully sampled from a wide range of texts and conversations to provide a snapshot of British English in the late 20th century. It is used in a wide variety of computational linguistic applications. BNCWeb offers powerful search and analysis functions for searching the text and exploiting the detailed textual metadata. It is an open source project, and the BNC is freely available for educational and research purposes [11].

The BNCWeb use case could benefit from the following cloud capabilities:

1. **Vertical auto-scaling** -- The BNCWeb service is implemented as a GUI-based web-service. As it is intended for direct interaction with users (i.e. users are expected to access a specific endpoint that provides the user interface and performs all computation necessary to answer their queries), only vertical auto-scaling is feasible without changing the internal architecture of the service itself.
2. **Database** -- The BNCWeb service performs queries on a large read-only data set of linguistic data. This data set is copied locally inside the BNCWeb image and could be off-loaded to an external (and possibly horizontally auto-scalable) SQL database to reduce image size and improve the performance and data management capabilities in multi-node deployments. This recommendation requires further discussion with BNCWeb developers since we do not fully understand all of BNCWeb's internal workings.

The provided image is quite large, in excess of 28 GB. This makes dynamic deployment almost impossible. It is necessary to reduce its size considerably. It contains a full desktop environment deployment, which is not needed and is a cause for further security-related concerns. It also contains kernel modules for a large number of unused kernels, VBoxGuestAdditions and a full apt cache. The BNCWeb image is distributed with locally installed MySQL database full of read-only corpus data. Removal and/or cleanup of these parts of the image could reduce its size by half.

We suggest the following improvements that would make the BNCWeb image more cloud-friendly:

1. **Automated software installation**
2. **Automated software configuration**
3. **Removal of unnecessary packages and kernel modules**
4. **The use of external (shared) SQL-based database for read-only data**
5. **The use of attached block storage for caching**

5.4 WS-PGRADE

WS-PGRADE is a portal environment for the development, execution and monitoring of workflows and workflow based parameter studies on different Distributed Computing Infrastructures (DCI). Various scientific collaborations and support teams in Europe and beyond use it in their infrastructures. WS-PGRADE is integrated with Service Grids (gLite, UNICORE, ARC, Globus), Desktop Grids (BOINC) and cluster (PBS, LFS) middleware through the components of the gUSE service stack.



There are no WS-PGRADE-specific images, however there are some generic suggestions that every image provider/author/owner should consider when providing a new cloud-aware image:

- 1. Automated software installation**
- 2. Automated software configuration**
- 3. Support for contextualization**
- 4. The use of block storage for local computation results and caching**
- 5. The use of shared read-only file systems for applications**
- 6. The use of object storage or shared databases for data sets**

5.5 DIRAC

The DIRAC interware project provides a framework for building ready to use distributed computing systems. It has been proven to be a useful tool for large international scientific collaborations integrating in a single system, their computing activities and distributed computing resources: Grids, Clouds and HTC clusters. In the case of Cloud resources, DIRAC currently integrates with Amazon EC2, OpenNebula, OpenStack and CloudStack. It provides high-level scientific services on top of these by using the DIRAC framework. The new design has been adopted by a federated hybrid cloud architecture (Rafhyc). Initial integration and scaling tests demonstrate the architecture is suitable for managing federated hybrid IaaS clouds providing eScience SaaS. LHCb DIRAC adopted this solution for LHCb computing on federated clouds, using end-points just like another computing resource.

There are no DIRAC-specific images, however there are some generic suggestions that every image provider/author/owner should consider when providing a new cloud-aware image:

- 1. Automated software installation**
- 2. Automated software configuration**
- 3. Support for contextualization**
- 4. The use of block storage for local computation results and caching**
- 5. The use of shared read-only file systems for applications**
- 6. The use of object storage or shared databases for data sets**



6 USE CASE SELECTION

The goal of this document is to select user communities to work with during the next few months and define how they could benefit from cloud capabilities. Some of them could be ruled out quite early, as they have not been very communicative and did not indicate particular interest in our services. Lack of interest in collaborating with the EGI Federated Clouds Task had already been reported prior to the project for BNCweb and PeachNote. We did not receive any feedback to our questionnaire from any of these projects. With an image for BNCweb being available, we evaluated it at any rate.

For the DCH-RP community, it seems quite early to make any concrete plans. Before we offer our help in making use of cloud resources, the approach of this community must materialize. We may be able to offer our help in a later phase.

GaiaSpace was rejected funding for a proposal at the beginning of May, therefore we cannot expect any requirements or input from this user community.

An evaluation of the WS-PGRADE images as well as the DIRAC framework lead to the conclusion that we would only support them for concrete use cases, for which we could provide valuable information. Of course, they will be able to benefit from our documentation about basic image creation.

The remaining two cases are BioVel and WeNMR, for which we believe that we can provide valuable input regarding both general image setup and exploitation of cloud capabilities. The details are mentioned in the respective sections above. The providers of these use cases have been the most communicative, which makes us believe that we can successfully help them improve their use of the EGI Cloud Infrastructure Platform



7 SUMMARY

In this report, we have identified the cloud capabilities as well as the user communities that we look at in this project. Furthermore, we have matched capabilities and user communities/use cases, in order to determine the next steps that are required to help User Communities make use of these capabilities. Additionally, we have evaluated the existing virtual machine images that were available prior to the project. We have gathered a number of characteristic properties of these images that we see unfit for cloud computing, e.g. because the images are larger than needed.

We have set the path for the next months and will start our activities by reconstructing the images provided by the WeNMR and BioVel communities. The lessons learnt during these activities will be reported again after the next phase of the project. An intermediate report about our activities will be presented at the EGI Technical Forum in Madrid in September.

8 APPENDIX A

This appendix conveys the questions and answers for each of the use cases from which we received a response.

- How did you create the image? (from scratch, basic installation, full installation etc.)
 - **OpenModeller**: From scratch, installing the needed libraries and tools as COMPSs, OpenModeller, rOCCI client, etc. and then we saved it.
 - **WeNMR**: the image was taken by Wouter from a previous developer not working anymore at CMBI. We initially planned to re-create it from scratch by following best practices as suggested e.g. in [1], however there was no time to do it before the FedCloud demo deadline and we simply shrunk the existing one as much as possible.
 - **WS-PGRADE**: We create basic images for different operating systems (e.g SL 6, Debian, Ubuntu, CentOS) and then we fork the images in order to customize them (e.g install and configure WS-Pgrade).
 - **DIRAC**: A DIRAC run includes a VMScheduler which contains at least one Running Pod. A Running Pod defines the relationship between a single DIRAC VM image with a list of end-points. A DIRAC VM image is a Boot image with the necessary for a particular Contextualization method. Currently the following contextualization methods are implemented:
 - Ad-hoc image: A ready to go image without further contextualization. This image has to be prepared to run in a specific endpoint and a particular DIRAC configuration (VMDIRAC server to connect, DIRAC release to use ...) We have run with CentOS and Ubuntu images supporting platform dependencies for Belle, Alice and Auger HEP software, both in private CloudStack IaaS and at Amazon EC2 commercial cloud.
 - HEPiX Contextualized image to allow an image management with a golden image separated of the context specifics, which are automatically managed by VMDIRAC.
 - OpenNebula - HEPiX contextualization: golden image CernVM-batch-node. DIRAC configuration provided by a ISO context image, which is generic for all the OpenNebula IaaS sites of the cloud aggregation. End-point configuration provided through the on-the-fly Open Nebula context section environment, which is specific of each Open Nebula IaaS end-point and selected on submission time from the DIRAC Configuration Server
 - OpenStack - HEPiX contextualization: golden image CernVM-batch-node. DIRAC configuration provided by the amiconfig tools, sending the scripts in nova 1.1 userdata. End-point configuration provided through nova 1.1 metadata, which is specific of each

OpenStack IaaS end-point and selected on submission time from the DIRAC Configuration Server.

- Generic contextualization using ssh. Whatever image with a ssh daemon listening in a port with inbound connectivity. The VM boots, the VMDIRAC polls the active sshd port and runs the DIRAC and the end-point configuration using sftp and ssh connections.
- Is everything required for computation already installed in the image? (software, tools etc.)
 - **OpenModeller**: Yes, even if we could have done it at VM creation time
 - **WeNMR**: the software is there, the input data (listed in the ToPoS token) are copied by the VM before processing doing wget from [3]
 - **WS-PGRADE**: Yes, it contains everything.
 - **DIRAC**: For the software area VMDIRAC is using cvmfs remote software repository from CERN with the LHCb repo with software and Conditions DB and also cvmfs repository from USC with software and tools for Alice and Auger. VMDIRAC uses the cvmfs included at CernVM images with a particular configuration, but also Ubuntu and CentOS images which have been prepared with a cvmfs client. Of course, an eventual user may set up an ad-hoc image with every software and tools prepared for a particular run.
 - About the data, VM uses transparently third-party storage systems
- How will you distribute your image and its updates? (vmcaster/vmcatcher, automated using a different tool, by hand)
 - **OpenModeller**: I uploaded the image in the EGI repository and then I endorsed it at CESNET. I don't know how the other providers published it.
 - **WeNMR**: by hand
 - **WS-PGRADE**: Marketplaces, vmcaster/vmcatcher
 - **DIRAC**: Any third-party automated distribution can be used. Currently VMDIRAC is manually configured with the required image catalog metadata.
- What are your resource requirements? (CPU, memory, storage and network)
 - **OpenModeller**: We never measured minimum requirements for running openModeller, and this can also be quite variable depending on the experiment. However, since the EGI Use Case is related with the BioVeL project, which is currently using a modelling service hosted at CRIA, I think BioVeL expects that the new service instance in Europe should provide at least a similar performance, if not better. The whole service is running on a single machine here:
 - Dell PowerEdge 1800 (2 processors Intel Xeon CPU 3.80GHz, 4GB Memory DDR2, 400MHz, 6 HDs SCSI of 146GB, 2 HDs SCSI of 300GB)
 - **WeNMR**: 2GB of RAM per CPU-core are enough, storage is not critical (1GB of free space on the VM is enough), network is needed to copy input and output data from/to web repositories, and to communicate with ToPoS server

- **WS-PGRADE:** The WS-PGrade needs 2 CPU, 4GB memory and 8-16GB of storage.
- **DIRAC:** The VM DIRAC image configuration allows to specify the VM flavor to run. LHCb has a work in progress to take advantage of multicore processing with different CPU and memory requirements depending on the specific software to run on the VMs.
- How do you submit work to running instances? (pilot framework or local workload)
 - **OpenModeller:** We use the VENUS-C/COMPSs framework. An endpoint is provided to user communities and accessed through Taverna for BioVeL users, through a Virtual Research Environment in EUBrazil-OpenBio.
 - **WeNMR:** using the ToPoS pilot framework
- Does your application support horizontal (more instances) and vertical (more resources for a single instance) auto-scaling?
 - **OpenModeller:** Horizontal
 - **WS-PGRADE:** not yet
 - **DIRAC:** A VM submitting policy is configured to each end-point:
 - endpoint vmPolicy "static" -> slots driven, indicated by maxEndpointInstances, all the end-point available slots are used to create VMs.
 - endpoint vmPolicy "elastic" -> jobs driven, one by one if there are jobs on the DIRAC Task Queued. The elasticity of this policy is tuned using the Running Pod configuration parameter namely CPUPerInstance. If the current required CPU in the jobs of the DIRAC task queue divided by the currently running VMs is lower than the CPUPerInstance, then no more VMs are submitted. A CPUPerInstance can be set to the contextualization time of a specific setup, and in this manner if the necessary average required time to run the jobs of the DIRAC Task Queue is lower than the contextualization time, then no more VMs are submitted. This is a compromise solution to use the available resources in a more efficient manner (saving creation overheads), and at the same time can be setup to use all the available resources to finish the production in a shorter total time, but with more resource costs (additional overhead).
 - A VM stoppage policy is configured to each end-point:
 - endpoint vmStopPolicy "never"
 - endpoint vmStopPolicy "elastic" -> no more jobs + VM halting margin time
 - In any case, VMs can be stopped by the VMDIRAC admin or by the HEPiX stoppage in the CernVM images (which is the responsibility of each cloud site admin). If a running VM is required to be stopped, then the VM stops in an ordered manner, declaring the running job stopped in

DIRAC (which can be resubmitted), then halting the VM.

- How do you access your virtual machines once they have been launched?
 - **OpenModeller**: COMPSs access the VMs for execution.
- Are you using contextualization? (how and where or why not)
 - **OpenModeller**: COMPSs at the moment of VM creation, copies the SSH keys
 - **WeNMR**: some contextualisation was needed for Wnodes and CESNET cloud providers, as described in [2]. The user know the password of the "i" account, but there is no need from the user to login in the VM because the application starts automatically after the boot. However the ssh key of the "i" account of the NMR server has to be present in the VM in order to copy there through rcp the produced output data.
 - **WS-PGRADE**: We are using EC2 like contextualization.
 - **DIRAC**: Only in the generic ssh contextualization, the rest of the cases VMDIRAC uses outbound connectivity.
- What's the character of your data? (size, format, read-only vs. read-write)
 - **OpenModeller**: The local set of environmental layers has ~32GB. Please note this is just a limited set of the most popular files, so this number can easily increase over the time. Environmental layers are one of the inputs for the modelling procedure, together with a set of species occurrences points. Both are only read by openModeller to generate, test or project models.
 - **WeNMR**: read-only input data zipped tar balls O(5 MB) accessed through web
- How are you accessing your data? (copied locally vs. accessed remotely)
 - **OpenModeller**: copied locally
 - **WeNMR**: copied locally
 - **WS-PGRADE**: remotely.
 - **DIRAC**: access data on VMDIRAC: Accessing "remotely" in the site by SRM. Accessing on Amazon using S3. Copied locally. Input and Output Sandox.
- How much space do you need for a single computation?
 - **OpenModeller**: Results can either be small XML files generated by creating or testing models (few KB) or a raster file generated by projecting a model (from a few MB up to a few GB, depending on spatial extent, resolution and format). In the new service API there's a new operation that will accept multiple model creations, tests and projections in a single request, which makes this question even more complicated to answer.
- Have you considered using object storage to access your data and store the results?
 - **WeNMR**: no(t yet?)
- Could environmental layers be stored in object storage?
 - **OpenModeller**: Environmental layers are rasters that are usually stored as regular files, but they can also be stored in relational databases, such as TerraLib or rasdaman do. Apparently Oracle Spatial can store rasters using object storage, but I have no experience with this.

- How are you gathering results and what's their character? (size, format, sensitivity)
 - **OpenModeller**: The service is asynchronous: clients send job requests and need to retrieve results later when the job is finished. Results are stored for a certain period of time configured by the sysadmin. They are all regular files. Here we keep them for a couple of weeks. It's hard to tell about the size because it depends on the number of requests in that period and on the type of requests. Right now the results in our server are only taking a few MB, but it would be a good idea to reserve some GB for this task. There's no security mechanism provided by the service. Results are retrieved by providing a ticket that is generated for the initial request (a random combination of numbers and characters).
- Are you dealing with sensitive data?
 - **WeNMR**: data are public, while parts of the code in the VM is something that the developers do not want make publicly available, because there is strong competition among the bio-NMR groups in designing the best algorithms for structure calculations
- Do you support or actively use any dynamic cloud-like environment? (which, how and why)
- Are you exposing any services to the outside world? (i.e., listening on public interfaces)
 - **OpenModeller**: Yes, there is an Extended Open Modeller Web Service exposed to the users and deployed outside EGI
 - **WS-PGRADE**: Our users should connect to the portal via http.
 - **DIRAC**: On the particular case of the generic ssh contextualization, public key from the VMDIRAC service is used, the ssh connections are disabled after the configuration of the VM.
- How are they protected from unauthorized use?
 - **OpenModeller**: The endpoint is public (at the moment, Renato don't know if you plan something about security) but then the access to the VENUS-C/COMPSs middleware is protected with x509 certificates security.
 - **WS-PGRADE**: It has a liferay portal framework and it uses its own authentication methods.

9 REFERENCES

R 1	http://docs.openstack.org/trunk/openstack-compute/admin/content/creating-custom-images.html
R 2	https://wiki.egi.eu/wiki/Fedcloud-tf:FedCloudWeNMR
R 3	http://nmr.cmbi.ru.nl/NRG-CING/data/
R 4	https://wiki.egi.eu/wiki/VT-CloudCaps:OriginalImages
R 5	https://wiki.egi.eu/wiki/Fedcloud-tf:FederatedCloudsTaskForce
R 6	https://wiki.egi.eu/wiki/Fedcloud-tf:WorkGroups:Scenario2
R 7	http://aws.amazon.com/sqs/
R 8	https://wiki.egi.eu/wiki/FedCloudOPENMODELLER
R 9	http://www.wenmr.eu/
R 10	https://wiki.egi.eu/wiki/VT-CloudCaps:OriginalImages#WeNMR
R 11	https://wiki.egi.eu/wiki/FedCloudBNCweb