EGI Federated Cloud Tutorial

Enol Fernández, Yin Chen

EGI User Community Support Team

http://go.egi.eu/cloud

EGI-Engage is co-funded by the Horizon 2020 Framework Programme of the European Union under grant number 654142

The trainers

- Enol.fernandez@egi.eu
- Santander, CSIC – Spanish NGI
- Cloud Architect
- Support for new communities

- Yin.Chen@egi.eu
- Amsterdam, EGI.eu
- Senior Technical Outreach Expert
- Support for new communities

Training goals

1. Learn basic concepts of cloud computing and cloud federations
2. Learn how to use EGI's standard-based cloud federation
   - With pre-defined images
   - With custom deployments (contextualisation)
   - With your own images
3. Guide to become user of the EGI Federated Cloud

Outline

- Introduction to EGI, Clouds, EGI Federated Cloud (20') (Y)
- Application best practices and examples (15') (E)
- Introduction to training infrastructure and first exercises (10') (E)
- Exercises (45')
  - Exercise 1: Compute management – Setup a Wiki
  - [Optional] Exercise 2: Persistent storage – Wiki with block storage

BREAK (~15')

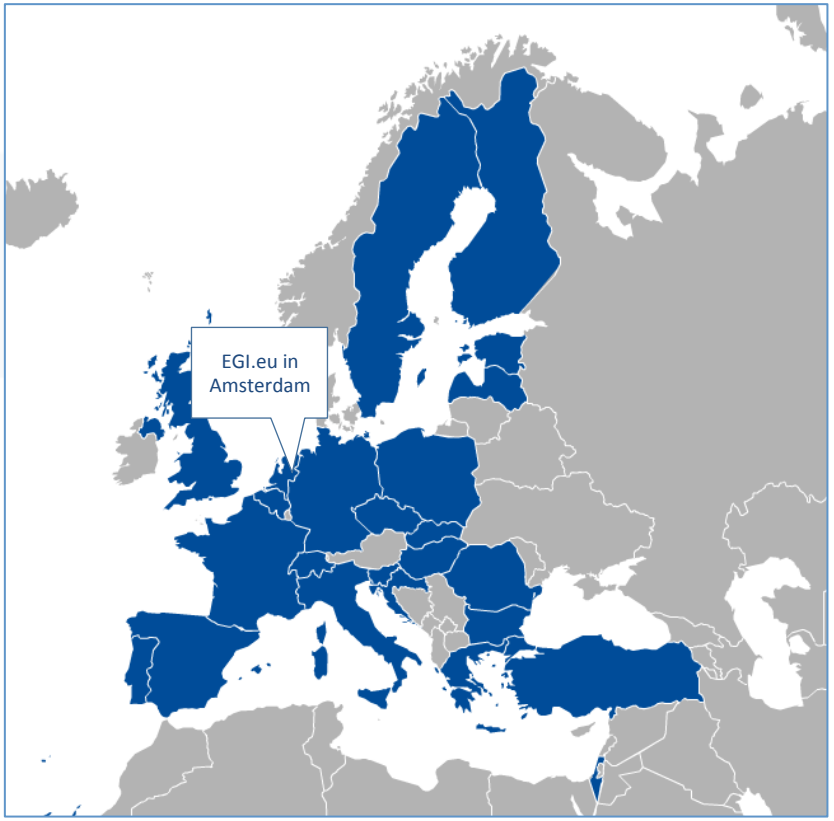
- Introduction to contextualisation (15') (E)
- Exercise 3: Contextualised compute – Fractal application (30')
- Talk + Demo: Creating your own Virtual Machine Image (15') (E)
- FINISH EXERCISES (buffer time)
- Next steps for you - How to become a user (5') (E)
- EGI's plans with the Federated Cloud (5') (E)
- Feedback forms

EGI-Engage

Introduction to EGI, Clouds, EGI Federated Cloud

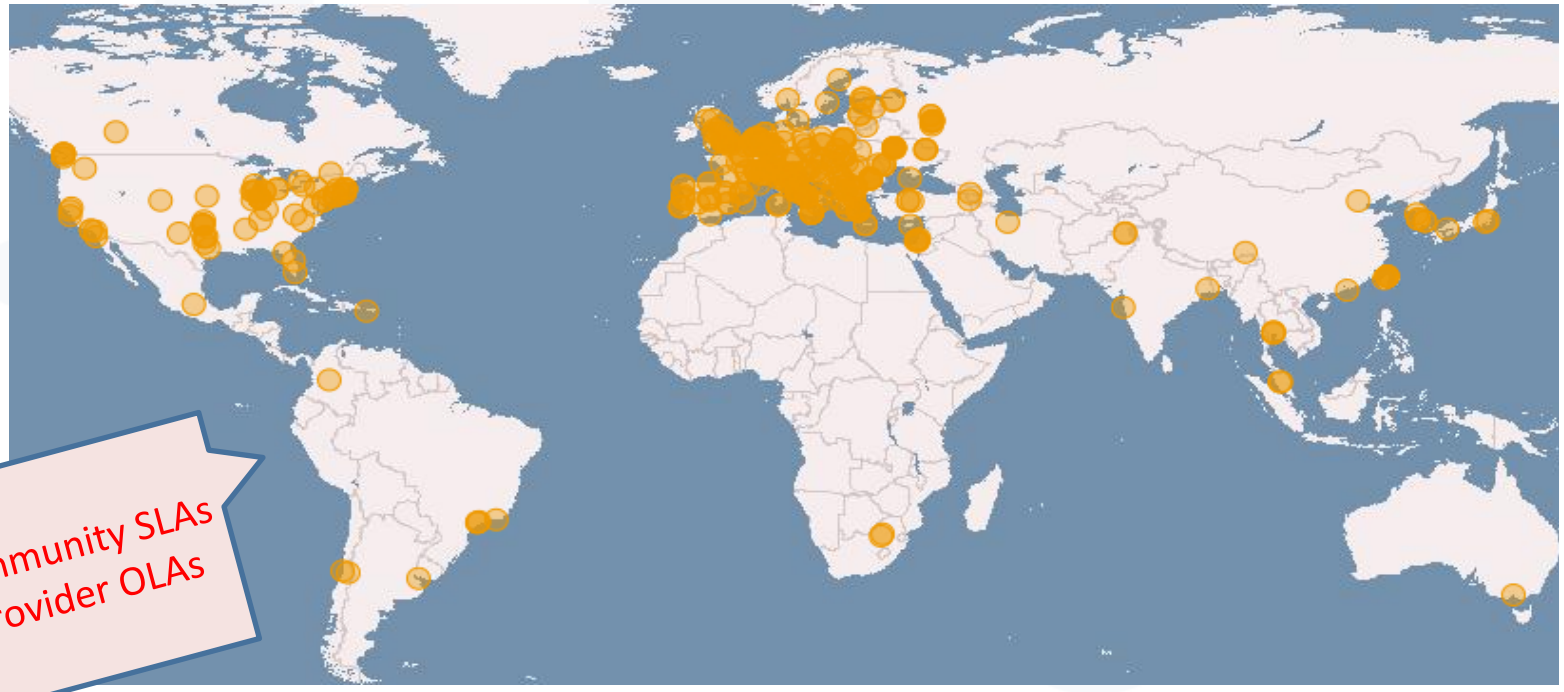
EGI ~~(European Grid Infrastructure)~~

- EGI Council
  - 26 participants: 24 NGIs and 2 EIROs (CERN, EBI)
  - Opening membership to research communities
  - Affiliation programme for countries
- Shared interest in
  - Developing and providing e-infrastructure services that enable open science
- Sustainability
  - Sustainable core services (HW, SW, human!)
  - Project-driven innovation
    - H2020: EGI-Engage, AARC, INDIGO, ELI-Trans, etc.
    - FP7: BioVeL, FitSM, SCI-BUS, Cloud-SME, …



Membership under discussion
Armenia, Austria, Belarus, Denmark, Moldova, Norway, Russia, Ukraine

Enabling Global Infrastructures



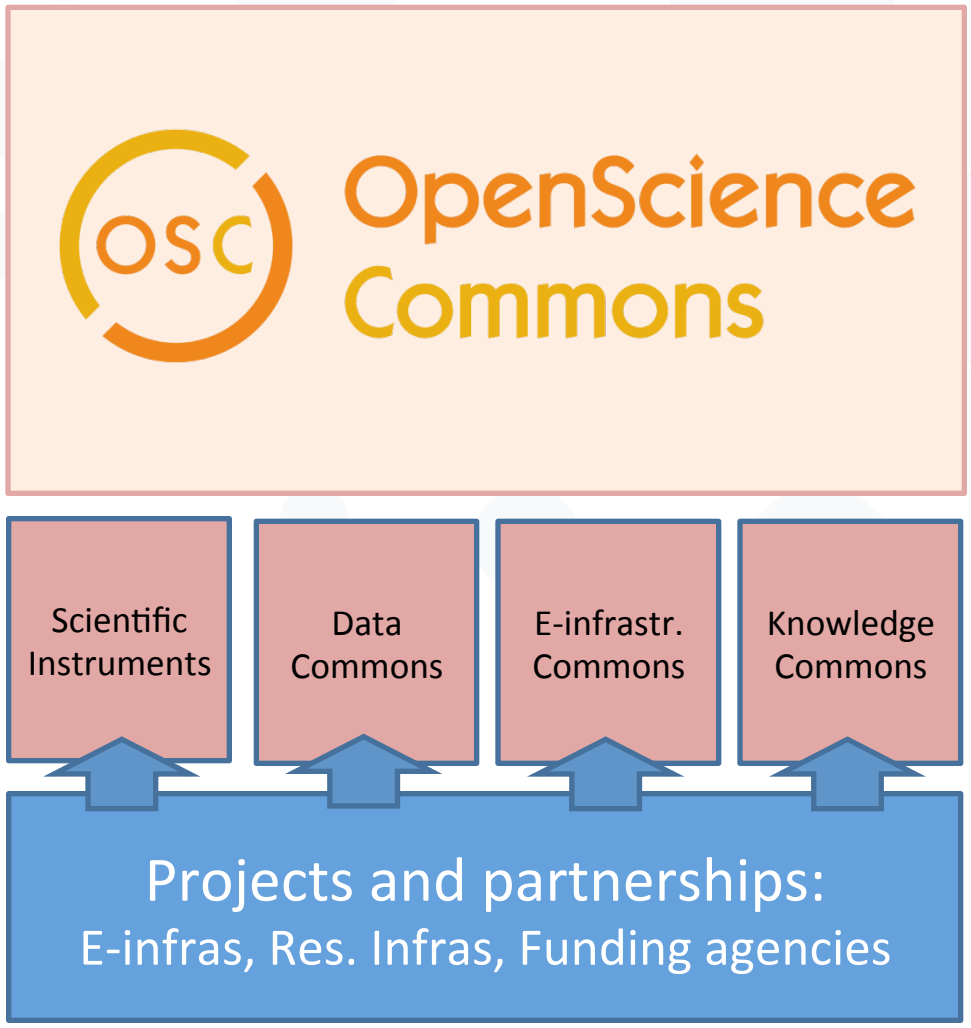
Community SLAs
Provider OLAs

- Distributed, federated storage and compute facilities
- Compute platforms (Grid, Cloud)
- Virtual Research Environments
- > 200 user research projects

Total capacity (grid + cloud):

- 340 resource centres in 54 countries
- 550,000 logical CPU cores
- 290 PB disk, 180 PB tape

Implementing the Open Science Commons



Researchers from all disciplines have easy, integrated and open access to the advanced digital services, scientific instruments, data, knowledge and expertise they need to collaborate to achieve excellence in science, research and innovation.

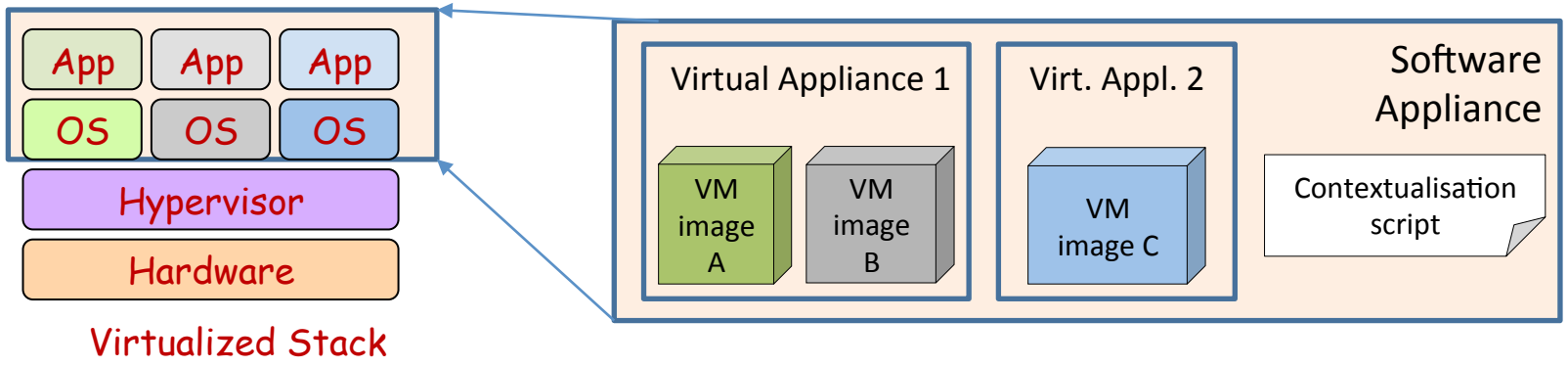
www.opensciencecommons.org

EGI-Engage H2020 project:

- 30 months, Start: 1/March/2015
- 43 partners
- 8 m Euro EC contribution

Cloud computing & Key terms

- Services and solutions that are delivered and consumed in real time over the Internet
- (Some of the) benefits for EGI & Open Science
  - Virtualisation – Platform-independence; Self-servicing
  - Scalability – Pay-as-you-go; Multi-tenant allocation
  - Predictability – Versioning of VMs and contextualisation scripts
  - Abstractions – IaaS, PaaS, SaaS
  - Open source – KVM, OpenStack, OpenNebula, TOSCA, Hadoop,…



What is a cloud federation? – In General

- Practice of interconnecting cloud service providers.
  - Data locality; Data privacy; Shared investment; Distributed expertise, …
- Multiple cloud sites with some sort of interconnection(s), for example:
  - Every cloud registered in a single catalogue
  - Single VM image catalogue for users
  - Support for the same image format
  - Automated replication of VM Images to clouds
  - Single sign-on for users
  - Shared operational practices
    - Harmonised configuration, shared monitoring, accounting, etc.
  - Same support model
    - Ticketing system, consultancy, training
  - …

GI Federated Cloud

TOPIC OF THIS TUTORIAL

EGI Fe oud is a collaboration of communities developing, innovating, operating and usin federations for research and education. The collaboration facilitates innovation a stainability of participating clouds and related communities.

Public Cloud (1)

- **Open to any research community, maintained by EGI FC Task Force**
- **Open Standards**: use of open standards to implement federation: OCCI, OVF, GLUE2, APEL, (CDMI), … Standards required.
- **Stronger integration profile**: Cloud Computing integrated into the existing production infrastructure.

Community Clouds (n)

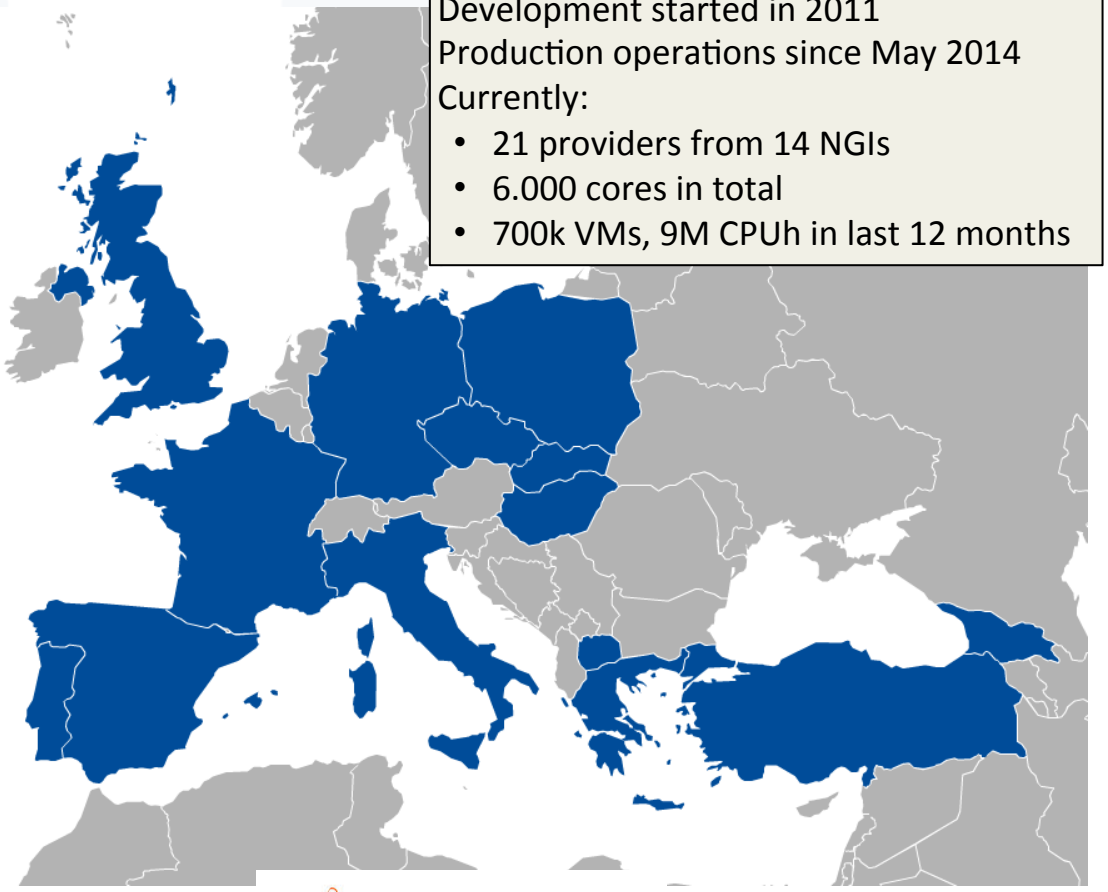
- **Available for specific communities, maintained by them**
- **Community choices:** Services and APIs to implement federation is community choice. Standards encouraged.
- **Looser Federation profile**: Based on a subset of EGI components (accounting, monitoring, …)

EGI Federated cloud

EGI-Engage

Development started in 2011
Production operations since May 2014
Currently:

- 21 providers from 14 NGIs
- 6.000 cores in total
- 700k VMs, 9M CPUh in last 12 months



IPHC Institut Pluridisciplinaire Hubert CURIEN STRASBOURG

INFN Istituto Nazionale di Fisica Nucleare

CYFRONET

IFCA Instituto de Física de Cantabria

LIP

SAS Slovak Academy of Sciences

grnet Networking Research and Education

CETA-Ciemat Centro Extremeño de Tecnologías Avanzadas

JÜLICH Forschungszentrum

MetaCentrum

Ss. Cyril and Methodius University in Skopje Faculty of Computer Science and Engineering

MTA SZTAKI

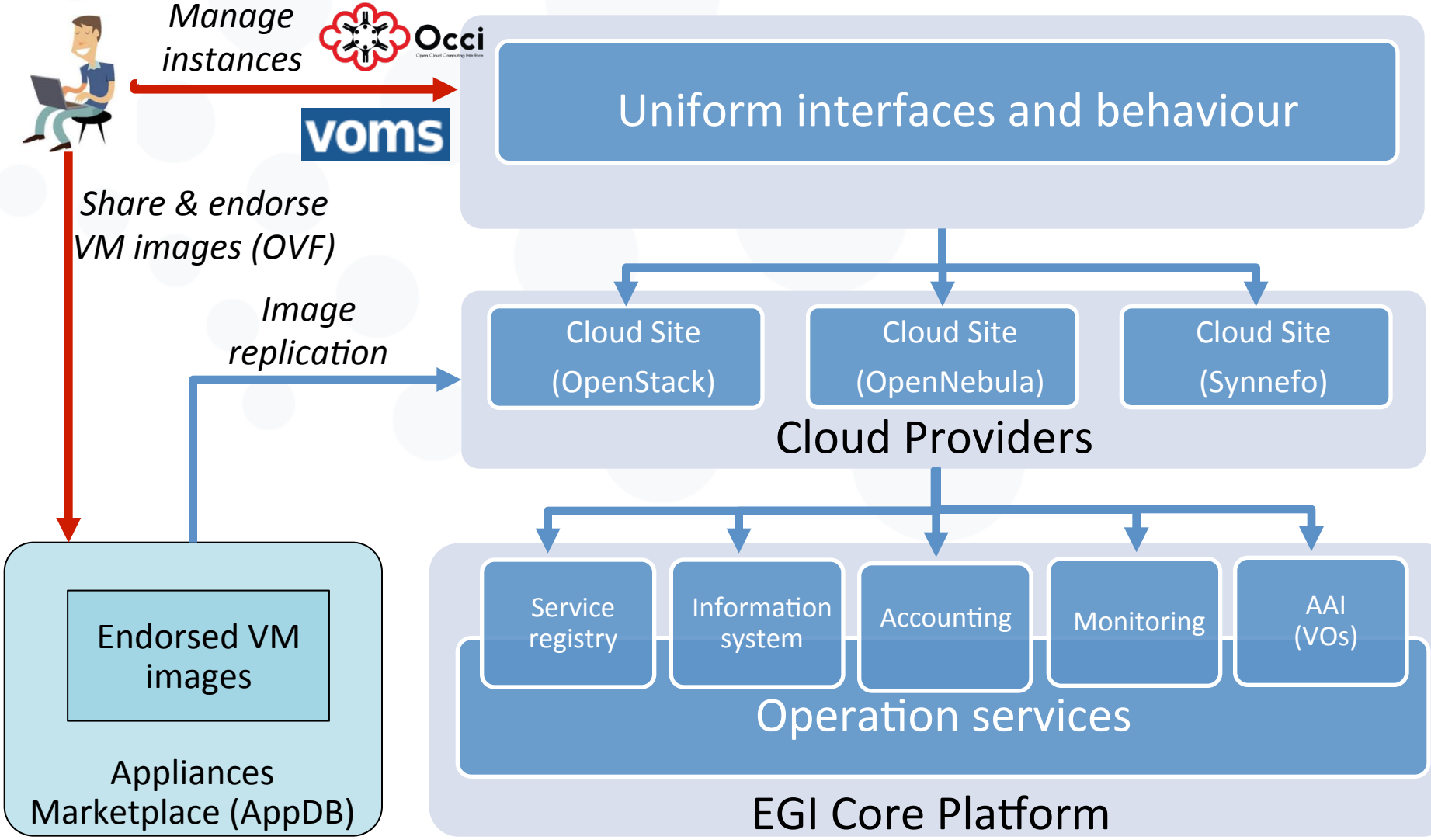
TÜBİTAK ULAKBİM

Grid y Computación de Altas Prestaciones GRyCAP

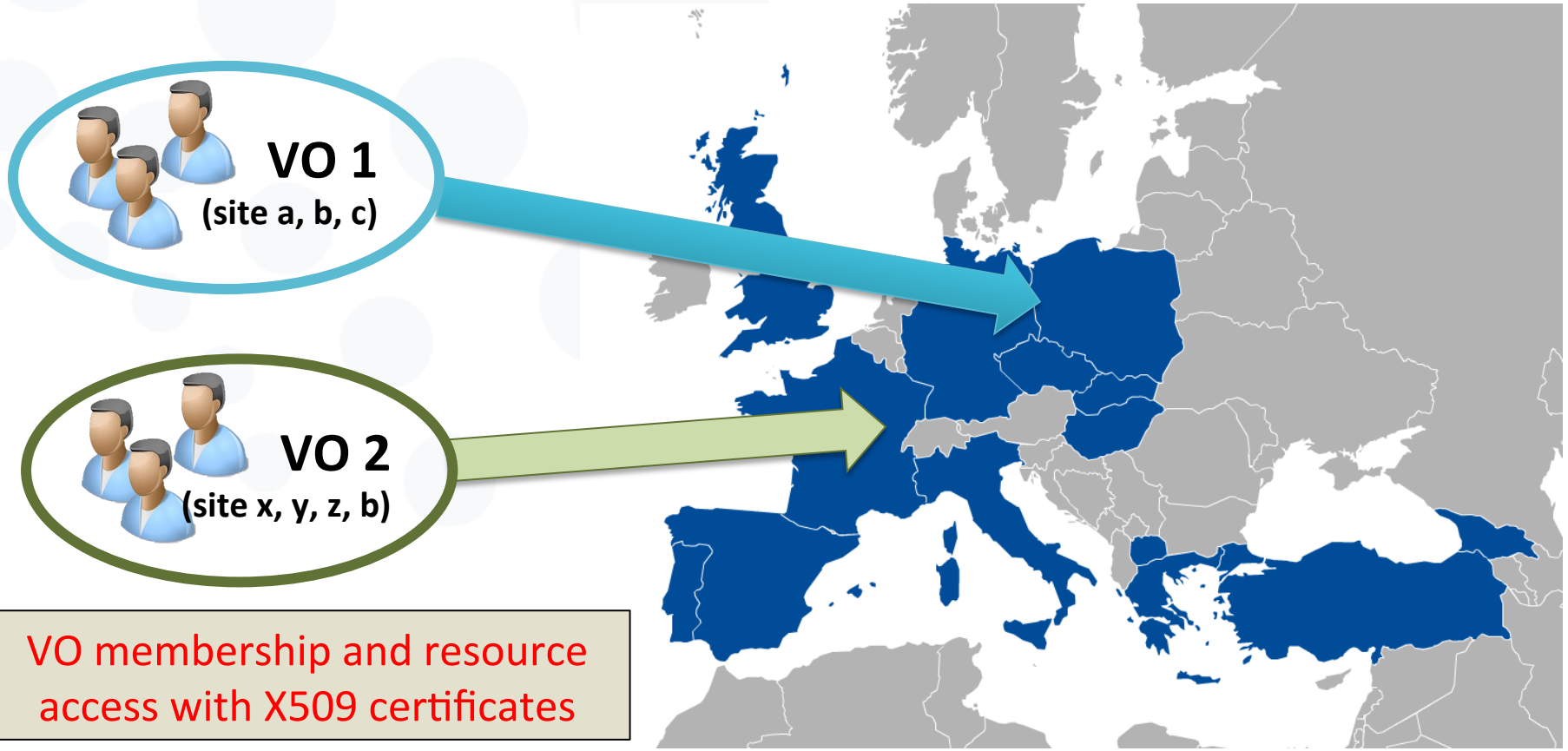
bifi

100%IT

Architecture of the EGI Federated Cloud



AAI and access to the Federated Cloud: Virtual Organisations



VO membership and resource access with X509 certificates

1. Generic VOs – e.g. fedcloud.egi.eu → Incubator for new users
2. Community-specific VOs – e.g. CHIPSTER, Highthroughtputseq, EISCAT, etc.
3. Training VO = training.egi.eu → To be used today

VO search tool at http://operations-portal.egi.eu/vo/search

High Level Tools (PaaS & SaaS)

- Today we focus on IaaS with OCCI (rOCCI cmd. line), but…
- …PaaS and SaaS tools exist on top of this
  - Alternatives to OCCI client and API – but usually much more!
  - External contributions – EGI is open for additional tools! Suggest/integrate!
  - Guidance on tools: https://wiki.egi.eu/wiki/HOWTO10. Currently:

Catania Science Gateway	• SaaS • Identity Federation	VMDIRAC	• Abstraction on top of various HPC/HTC/cloud
Slipstream	• PaaS for automating deployments • Helix Nebula	WS-PGRADE	• Workflow development and enactment
COMPSs	• Framework for auto-parallelisation	Vcycle	• VM lifecycle manager

OCCI and rOCCI

- **OCCI** (Open Cloud Computing Interface, OGF, 2011)
  - For VM Management
  - Text-based protocol and API focusing on cloud interoperability
  - Currently v. 1.1: three parts with focus on IaaS: Core, Infrastructure, HTTP Rendering
  - V1.2 released for public comments until end of July*
    - Comp.Template, PaaS, SLAs, Monitoring, Text rendering, JSON rendering
- **rOCCI** (OCCI command-line client; r for Ruby)
  - Interacts with the OCCI servers deployed on cloud sites
  - Supports EGI AAI (X.509 certificates + VOMS)
  - Available in source and VM image
  - To be used today
- (jOCCI: Java API for OCCI → not today)

Main commands to be used during Exercises

Command	Explanation
voms-proxy-info	Check the lifetime of your proxy
ssh-keygen	Generate key-pairs for password-less SSH
occi --endpoint A --auth B --action C –resource D	Perform action C on resource D of cloud site A authenticating as B
--action list	
--action create	
--action describe	
--resource compute	
--resource storage	

rOCCI quick reference guide: https://gist.github.com/arax/4de4a41fb0fa67719856

EGI-Engage

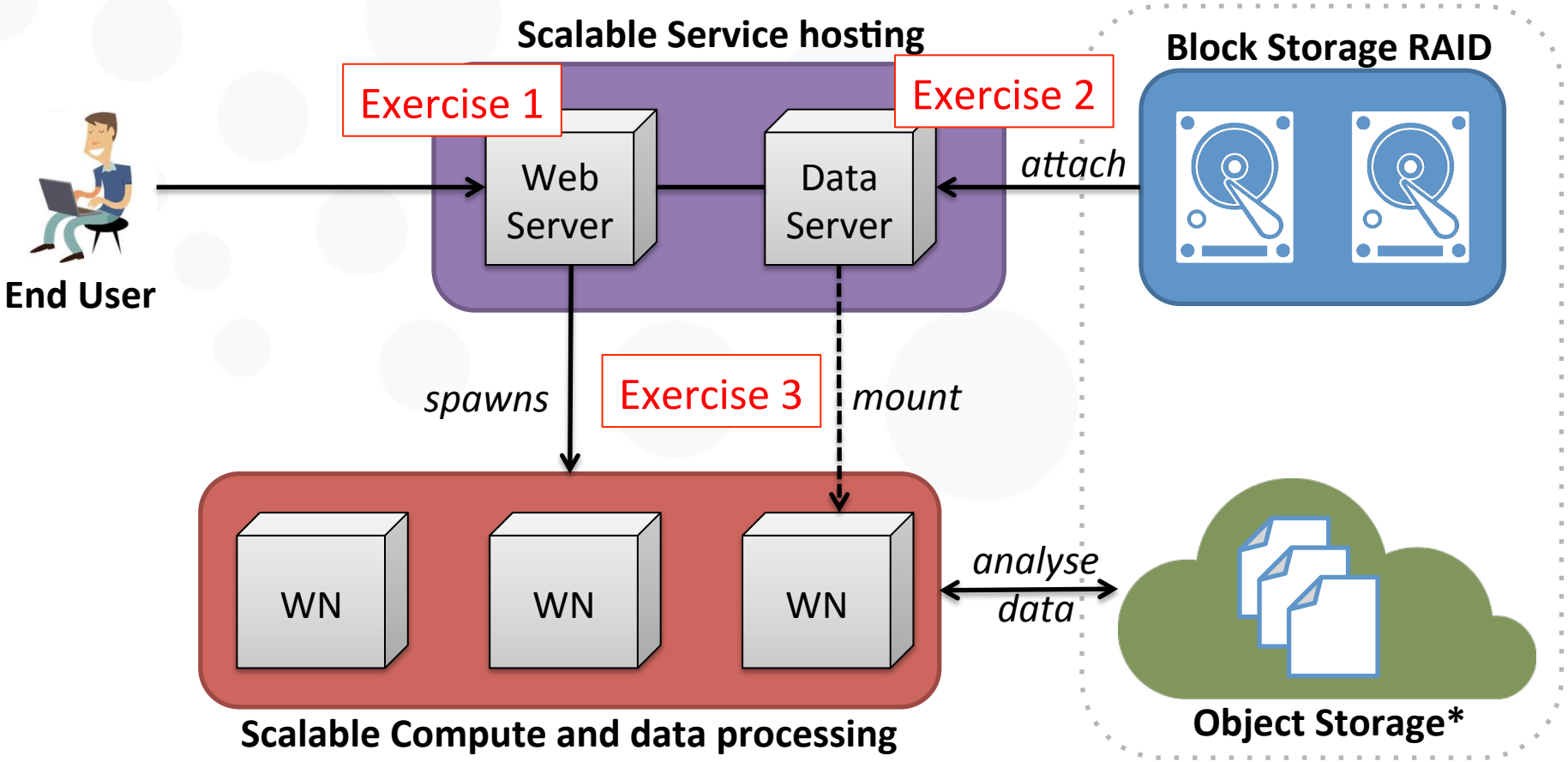
Application best practices and examples

Typical usage models (I)

- **Service Hosting**
  - Long-running services (e.g. web server, database, application server)
- **Compute and data intensive workloads**
  - Batch and interactive (e.g. IPython, R, Matlab) with scalable and customized environments (not limited to batch job model!)
- **Datasets repository**
  - Store and manage large datasets
- **Disposable and testing environments**
  - Host training environments, test applications

Typical usage models (II)

Combine usage models in a single application



* Object storage (CDMI or other) is not available on every site

Integration Strategies

- Manual Server setup
- Basic OS image with contextualization
- Custom OS image
- Brokers/High Level Tools

https://wiki.egi.eu/wiki/HOWTO10

Manual Server setup

- Start a virtual server with a pre-defined environment (eg. Ubuntu 12.04, CentOS 6.4, Wiki, etc...)
- Login into it (via SSH) and install your own software or configure pre-deployed software
- **Advantage:**
  - Easiness
- **Disavantage:**
  - Unsuitable for complex deployments
- **Recommended for:**
  - Tests, Simple applications, 'Disposable' applications

Exercise 1 and 2 today

Basic OS image with contextualization

- Contextualisation script automatically executed at startup to install and configure application inside virtual server
- **Advantages:**
  - Repeatability
  - Faster application configuration
  - Better portability
- **Disavantages:**
  - Extra effort to build the contextualization script
  - Slow startup
- **Recommended for:**
  - Simple applications, Development tests, Unknown configuration parameters until deployment time.

Exercise 3 today: Fractal application

Custom OS image

- Packaging your application in a custom VM image
- **Advantages:**
  - Build the virtual disk directly from a legacy machine (~cloning)
  - Speed-up the deployment for applications with complex installation packages
- **Disavantages:**
  - Compatibility issues with hardware drivers
  - Machine requires manual updates
- **Recommended for:**
  - Services that are started very frequently; Special OS flavours; Complex application setup

Demo today

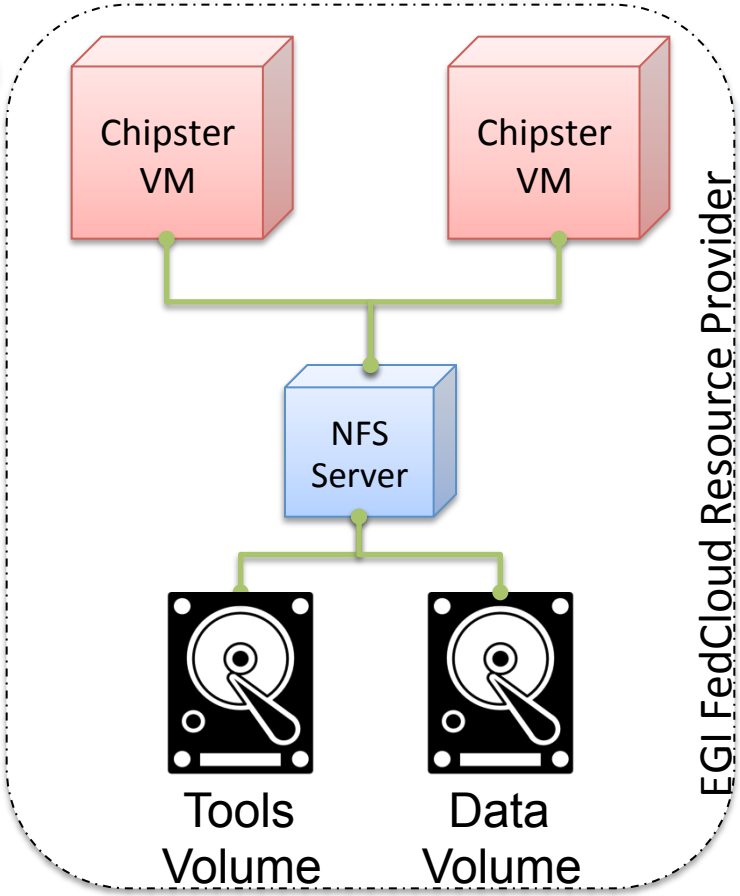
Brokers/High Level Tools

- **Infrastructure Broker:** run a full deployment of multiple servers (in the order of thousands or more)
  - usually via a web interface or custom APIs
  - using contextualization to setup the component dependencies
- **Application Broker:** abstracts the Cloud APIs and frees you from the need to control the deployment of the application
  - take care of starting the virtual servers according to the workflow/workload

See ‘High-level tools’ slide earlier

Example: Chipster

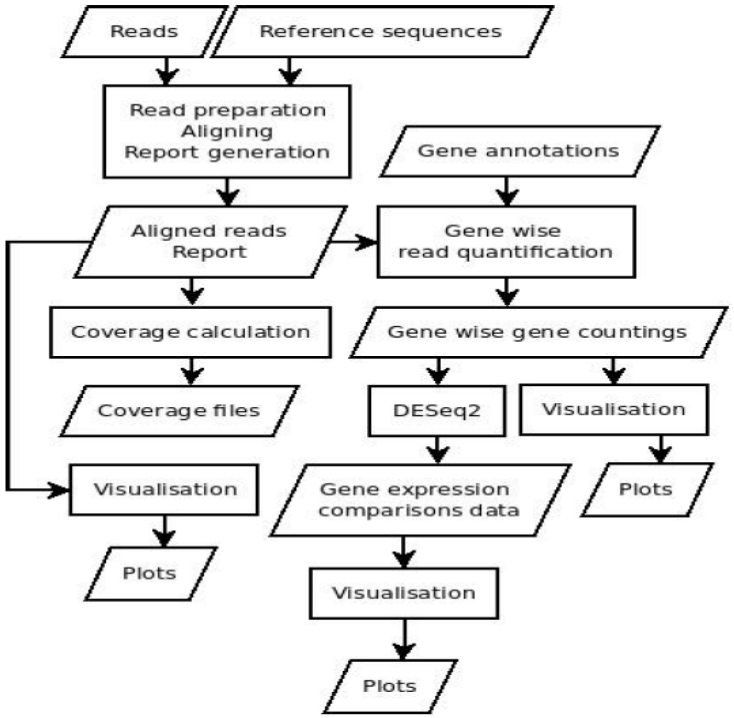
Analysis software contains over 300 analysis tools for NGS, microarray, proteomics and sequence data.



Usage Model	• **Web service** • **Heavy computation and large memory** • **Manage large datasets**
Scientific Disciplines	• **Bioinformatics**
Deployment in the FedCloud	• **Complex deployment through contextualisation** • **shared block storage exported as NFS up to 1 TB**

Example: READemption

Pipeline for the computational evaluation of RNA-Seq. data

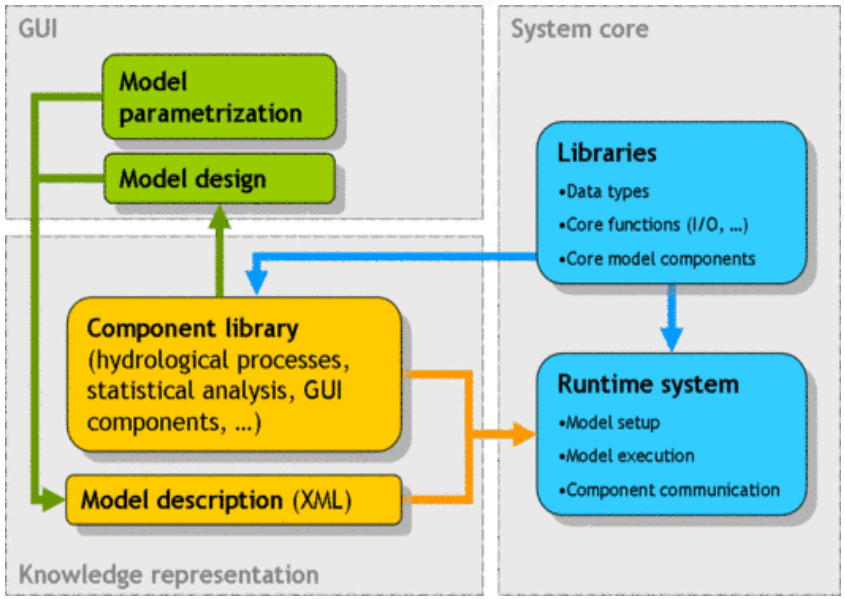


Source: Konrad U. Förstner

Usage Model	• **Heavy computation** • **Large Memory**
Scientific Disciplines	• **Bioinformatics**
Deployment in the FedCloud	• **VMs with 24 cores, 128 GB of RAM** • **Block storage up to 3 TB**

Example: JAMS
Jena Adaptable Modelling System

Platform for process-based hydrological model development

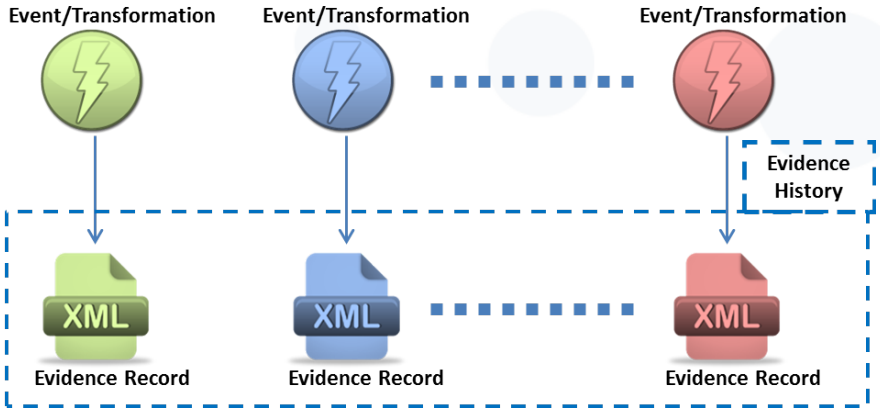


Usage Model	• **Web Service** • **Heavy computation** • **Large Memory**
Scientific Disciplines	• **Hydrology**
Deployment in the FedCloud	• **VMs with > 8 cores, > 16 GB of RAM** • **Image available in the marketplace**

Example: HAPPI

Supports the archive manager and curator to capture and manage part of the Preservation Descriptive Information

SCIDIP-ES
HAPPI Toolkit



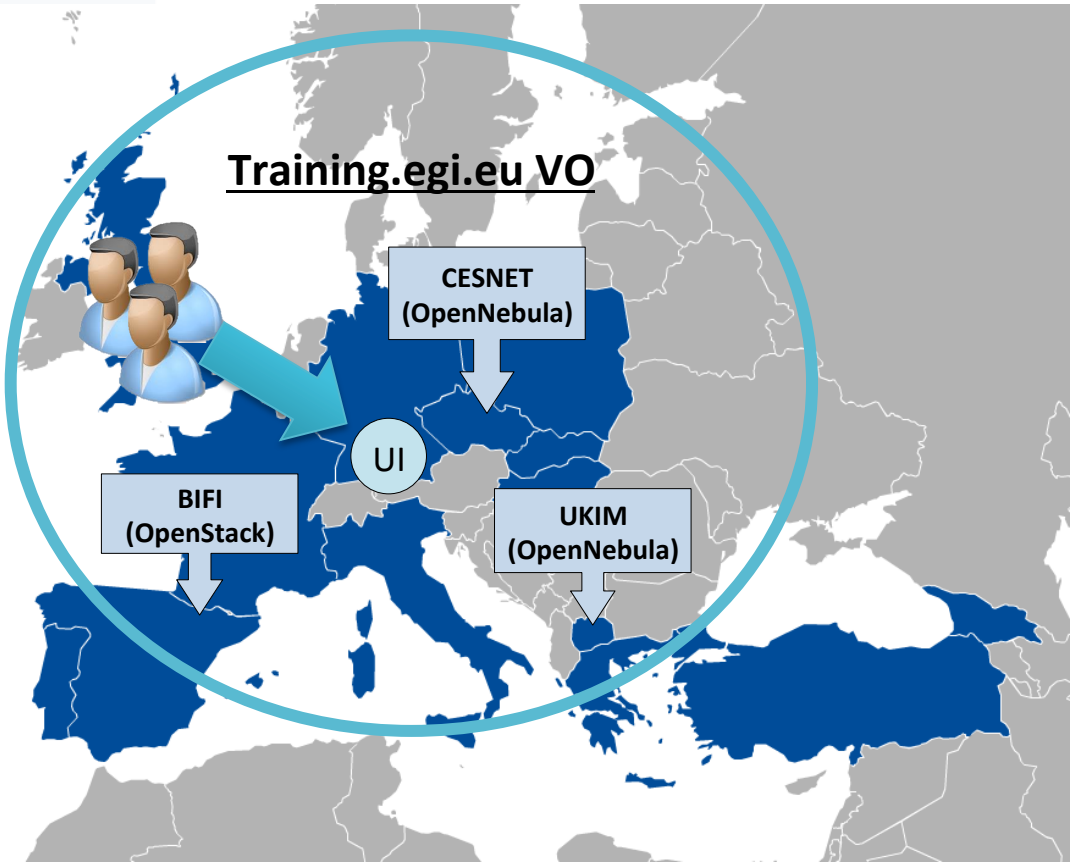
Usage Model	• **Web Service**
Scientific Disciplines	• **Digital Archives**
Deployment in the FedCloud	• **VMs with 2 cores, 4 GB of RAM** • **Block Storage**

EGI-Engage

Training infrastructure and first exercises

Training infrastructure: training.egi.eu Virtual Organisation

Site	Available capacity in the VO
CESNET (CZ)	64 vCPUs 110 GB of RAM 1 TB of persistent storage
BIFI (ES)	50 vCPUs 50 GB of RAM 50 storage volumes 50 public IP addresses
UKIM (MK)	48 vCPUs 48 GB of RAM 48 public IP addresses



- Trainers join VO with X509 personal certificates → Generate own proxy for access
- Trainees get proxies from trainers. **Your proxy is valid for 24 hours**
  - You will need personal certificate from a recognised CA for the long-term – More later!

Accessing the training VO

- Ubuntu 14.04 with rOCCI client
- Configured by trainers

- 1 account / trainee
- 1 proxy / account

Login with SSH

UI

Get IDs of
- Cloud endpoint
- VA image
- Resource template

Access services
(e.g. SSH, Web)

Execute OCCI commands

EGI AppDB

VM

Block storage

Cloud Resource Providers

http://appdb.egi.eu → Cloud Marketplace

Exercise 1 and 2

Managing VMs and block storage:

1. Host a simple wiki on an EGI Cloud site
2. Use persistent storage for wiki contents

Exercise sheet: Wiki setup

Start a wiki on FedCloud

2. login

UI

3. Manage VM

1. Get IDs of
- Cloud resource prov.
- Resource template
- VM image

4. Edit wiki

EGI AppDB

Download Wiki VA

Wiki VM

Cloud Resource Provider

Browsing AppDB

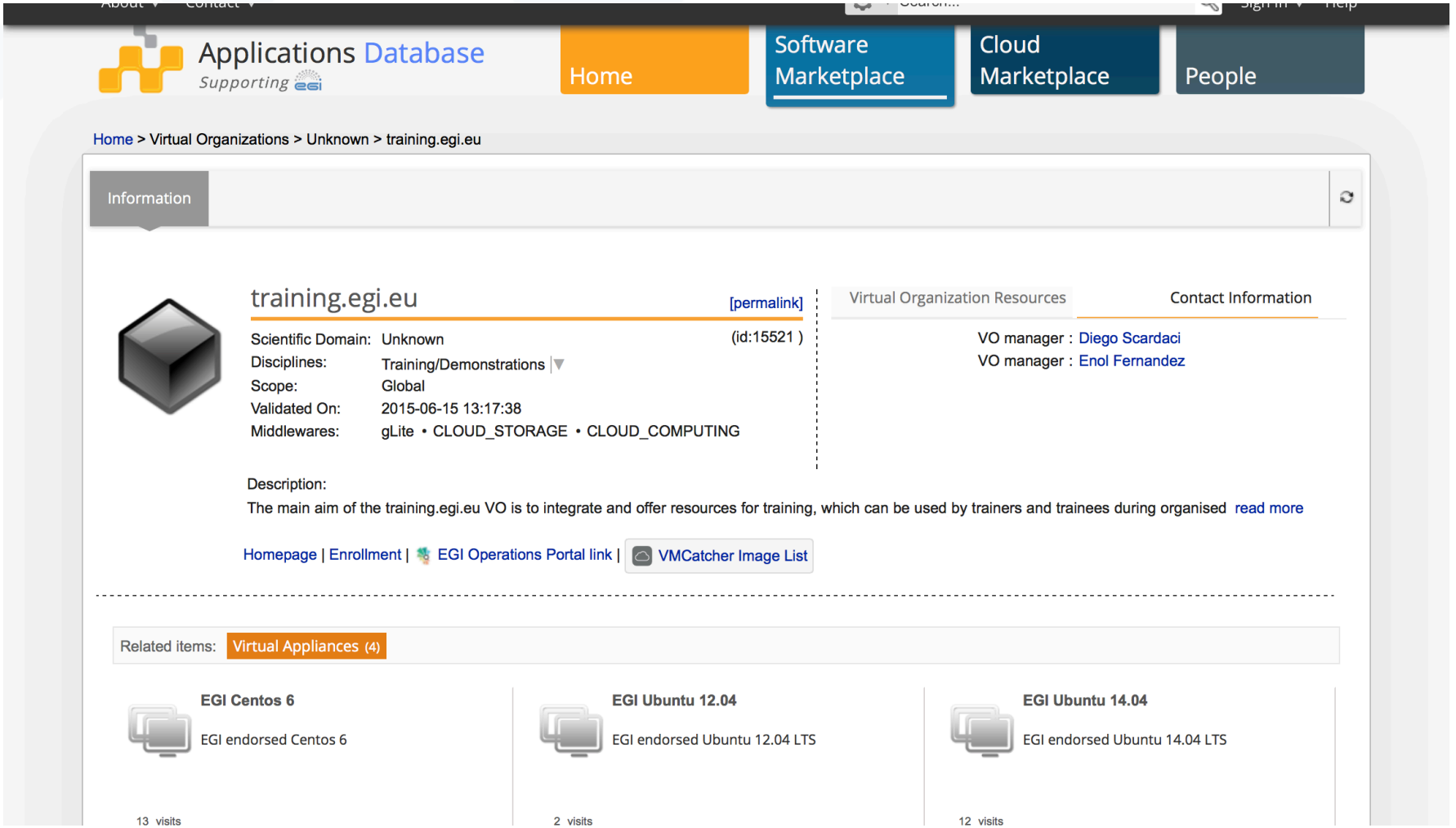
- Go to AppDB:
  - http://appdb.egi.eu
  - Cloud Mp → Virtual Organizations → training.egi.eu
- Choose MoinMoin VA and a specific site
  - See request on next slide!
- VAs and SAs in this VO:
  - Baseline OS appliances
    - Minimal OS images
    - Centos6, Ubuntu 12.04, Ubuntu 14.04
  - Specific appliances
    - FedCloud tools: Ubuntu 14.04 with FedCloud clients ready to use
    - MoinMoin wiki: Ubuntu 14.04 image with MoinMoin installed and configured to run on startup
  - Software appliances
    - Use contextualization to deliver the functionality

REQUEST

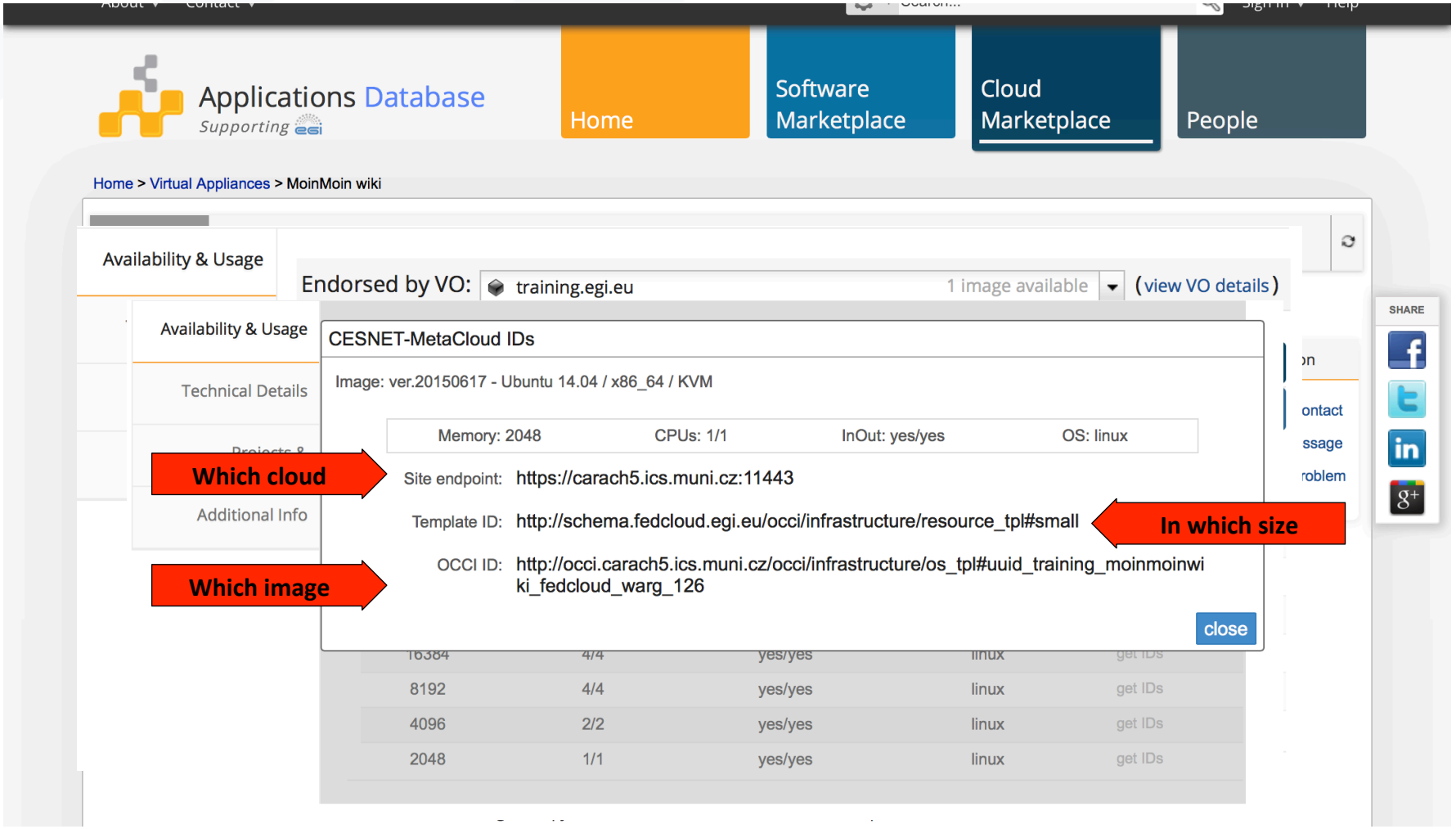
- Instantiate VMs based on the smallest resource templates during the whole tutorial
  - I.e. Use the following **Template IDs**:

Site	Template name	Template ID
CESNET	Small	http://schema.fedcloud.egi.eu/occi/infrastructure/resource_tpl#small
BIFI	Tiny	resource_tpl#m1-tiny-ephemeral
UKIM	Small	resource_tpl#small

AppDB


Applications Database
Supporting egi
Home
Software Marketplace
Cloud Marketplace
People
Home > Virtual Organizations > Unknown > training.egi.eu
Information
training.egi.eu
[permalink]
(id:15521)
Scientific Domain: Unknown
Disciplines: Training/Demonstrations
Scope: Global
Validated On: 2015-06-15 13:17:38
Middlewares: gLite • CLOUD_STORAGE • CLOUD_COMPUTING
Virtual Organization Resources
Contact Information
VO manager : Diego Scardaci
VO manager : Enol Fernandez
Description:
The main aim of the training.egi.eu VO is to integrate and offer resources for training, which can be used by trainers and trainees during organised read more
Homepage | Enrollment | EGI Operations Portal link | VMCatcher Image List
Related items: Virtual Appliances (4)
EGI Centos 6
EGI endorsed Centos 6
13 visits
EGI Ubuntu 12.04
EGI endorsed Ubuntu 12.04 LTS
2 visits
EGI Ubuntu 14.04
EGI endorsed Ubuntu 14.04 LTS
12 visits

Getting information on the VA



Log into the UI

- Log into the User Interface

  <DETAILS TO BE PROVIDED DIRECTLY TO TRAINEES>

- Check your proxy file

```
~$ echo $X509_USER_PROXY
```

- Check the lifetime of your credential

```
~$ voms-proxy-info –all
```

Get ready to access your VMs with SSH

- VMs are (normally) accessible through SSH
  - But password logins are disabled
  - Instead use key pairs
- Create a ssh key to access:

```
~$ ssh-keygen
```

(defaults are ok, can be left without password for the tutorial)

Basic check: dump OCCI model

```
~$ ENDPOINT=<Copy here Site Endpoint information from AppDB>
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --dump-model
```

Site	Endpoint	Cloud Middleware
CESNET	https://carach5.ics.muni.cz:11443/	OpenNebula
BIFI	http://server4-epsh.unizar.es:8787	OpenStack Grizzly
UKIM	https://occi.nebula.finki.ukim.mk:443	OpenNebula

What does dump-model return?

occi --endpoint $ENDPOINT --auth x509 --voms --user-cred $X509_USER_PROXY --dump-model

@kinds – different types of resources available at the site (“compute”, “storage”)

@mixins – mixins extend the kinds with additional attributes or actions, e.g.: a “compute” resource can have a given “resource_tpl” for specifying its size and a “os_tpl” for specifying its OS image

@actions – list of actions (e.g. start/suspend) that can be triggered on a resource

@resources – existing resources, e.g. started VMs (normally empty if just dumping the model)

@links – existing links, e.g. block storage attached to a VM (empty if just dumping the model)

Available templates: list & describe

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action list --resource os_tpl

~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action list --resource resource_tpl
```

```
~$ OS_TPL=XXX
```

Choose a value from the above result list

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action describe –resource $OS_TPL
```

Create your first compute appliance

Use MoinMoinWiki VA values from AppDB!

```
~$ RESOURCE_TPL=<copy here the Template ID from AppDB>

~$ OS_TPL=<copy here the OCCI ID from AppDB>

~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action create --resource compute \
        --mixin $RESOURCE_TPL --mixin $OS_TPL \
        --attribute occi.core.title="wiki$(date +%s)" \
        --context public_key="file:///$HOME/.ssh/id_rsa.pub"

~$ COMPUTE_ID=...
```

Save the ID in an Env. variable

List and describe your VM instances

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action list --resource compute
```

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action describe --resource $COMPUTE_ID
```

This returns lot of info, including the IP Address of your VM!

`occi.networkinterface.address = …`

→ It's not so simple ☹ See next slide!

Accessing the appliance

- If the VM does not have a public IP (on BIFI endpoint):

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action link --resource $COMPUTE_ID \
        --link /network/public
```

- Obtain the IP address from the output of the describe command.
- Check in your browser:
  - http://<your vm ip>/
  - And edit your wiki!

Logging into the appliance

- ssh with ubuntu user:

```
~$ ssh ubuntu@<your vm ip>
```

- Once logged in, check the size of the image:

```
~wiki $ cat /proc/cpuinfo

~wiki $ cat /proc/meminfo
```

- Check the wiki edit history

```
~wike $ cat /org/mywiki/data/edit-log
```

EGI-Engage

Exercise sheet
Wiki with persistent storage

Making wiki data persistent

- When a VM is deleted all its disks are also deleted
  - If you need persistency for your data you must use a storage volume
- Let's try it with our wiki:
  1. Create a volume
  2. Attach volume to our wiki VM
  3. Create FS in the volume and copy wiki contents
  4. Detach volume and delete VM
  5. Create new VM with the created volume attached
  6. Mount the volume and check the wiki contents are still there

Create the volume and describe it

- Create a volume

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action create --resource storage \
        --attribute occi.storage.size="num(1)" \
        --attribute occi.core.title="wikidata_$(date +%s)"
~$ STORAGE_ID=...
```

Save the ID in an Env. variable

- Describe it

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action describe --resource $STORAGE_ID
```

Attach to VM

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action link --resource $COMPUTE_ID \
        --link $STORAGE_ID
```

See attach information

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action describe --resource $COMPUTE_ID

[…]

Links:

    [[ http://schemas.ogf.org/occi/infrastructure#storagelink ]]
    >> location: /storage/link/c17e204e-c96f-40ff-
aebe-671351254a5e_1e0162cb-2805-4fe7-8c4e-997a5ddf02ff
    occi.core.source = /compute/c17e204e-c96f-40ff-aebe-671351254a5e
    occi.core.target = /storage/1e0162cb-2805-4fe7-8c4e-997a5ddf02ff
    occi.core.id = /storage/link/c17e204e-c96f-40ff-
aebe-671351254a5e_1e0162cb-2805-4fe7-8c4e-997a5ddf02ff
    occi.storagelink.deviceid = /dev/vdb

[…]
```

LINK_ID

We will need this at the VM to manage the volume

Move wiki contents to new volume

```
~$ ssh ubuntu@<your wiki ip>

~wiki $ sudo mkfs.ext3 /dev/vdb

~wiki $ sudo mount /dev/vdb /mnt

~wiki $ sudo service apache2 stop

~wiki $ sudo cp –a /org/mywiki /mnt

~wiki $ sudo umount /mnt

~wiki $ sudo mount /dev/vdb /org

~wiki $ sudo service apache2 start
```

Clean up and stop the VM

- Umount the volume

```
~wiki $ sudo umount /org
```

- Detach the volume:

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $LINK_ID
```

- Delete VM:

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $COMPUTE_ID
```

Create a new wiki with the volume

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action create --resource compute \
        --mixin $RESOURCE_TPL --mixin $OS_TPL \
        --attribute occi.core.title="wiki$(date +%s)" \
        --link $STORAGE_ID \
        --context public_key="file:///$HOME/.ssh/id_rsa.pub"

~$ COMPUTE_ID2=...
```

Save the ID in an Env. variable

Use the volume

- Login into the VM and mount the volume at /org

```
~$ ssh ubuntu@<your wiki ip>

~wiki $ sudo service apache2 stop

~wiki $ sudo mount /dev/vdb /org

~wiki $ sudo service apache2 start
```

- Test it in your browser, it should have all the contents from previous VM!

Once done, delete your instances

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $COMPUTE_ID2
```

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $STORAGE_ID
```

BREAK

www.egi.eu

EGI-Engage is co-funded by the Horizon 2020 Framework Programme
of the European Union under grant number 654142

EGI-Engage

Contextualisation

Contextualization

- What?
  - Contextualization is the process of installing, configuring and preparing software upon boot time on a pre-defined virtual machine image
  - e.g. hostname, IP address, ssh keys, …
- Why?
  - Configuration not known until instantiation (e.g. data location)
  - Private Information (e.g. host certs)
  - Software that changes frequently or under development
  - Not practical to create a new VM image for every possible configuration

Contextualization in FedCloud (I)

- Contextualization requires passing some data to the VMs on instantiation (the context)
- OCCI v1.1 (current version) API lacks a way to pass that context to the VMs
- FedCloud proposed two mixins to support this feature
  - public_key and user_data mixins
  - Implementation available for all currently supported cloud management frameworks
  - Deployed at all resource providers

Contextualization in FedCloud (II)

- OCCI extensions specify how to pass context to the VM
- BUT not how the data will be available!
- Each RP has different mechanisms to provide it
  - metadata server at a known location
  - iso filesystem attached to the VM
  - file injection into the VM image
  - …
- cloud-init is the preferred tool to abstract this

cloud-init

- cloud-init abstracts the different ways of providing context to the VM and defines a format for the data
- cloud-init is able to:
  - configure network, users, **ssh keys**, filesystems,
  - **install packages,**
  - execute arbitrary commands,
  - **execute user provided scripts,**
  - invoke puppet or chef for configuration
  - …
- And can be easily extended!

cloud-init support

- Cloud-init is the de-facto standard for contextualization of VMs
- Supports:
  - Most commercial providers (Amazon EC2, Azure, Rackspace,…)
  - Cloud management frameworks in fedcloud:
    - OpenStack
    - OpenNebula (use version ≥ 0.7.5)
    - Synnefo
- Packages available for most Linux distributions: ubuntu/debian, SL5/SL6 (in EPEL), SUSE, …

Use with rOCCI CLI

- Use --context option to specify
  - public_key
  - user_data

EXAMPLE – NO NEED TO EXECUTE

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action create --resource compute \
        --mixin $RESOURCE_TPL --mixin $OS_TPL \
        --attribute occi.core.title="wiki$(date +%s)" \
        --context user_data="file:///$PWD/context"  \
        --context public_key="file:///$HOME/.ssh/id_rsa.pub"

~$ COMPUTE_ID=...
```

Meta data vs user data

Meta data

- Basic predefined information on the VM
  - VM Identifier
  - Hostname, IP
  - User Public Keys

User data

- User data is treated as opaque data: what you give is what you get back. It is up to the instance (cloud-init) to be able to interpret it.

cloud-init uses both meta-data and user-data to contextualize the VMs

Public Key injection with cloud-init

- SSH with key pairs is so common that treated specially
  - Goes into the VM meta-data
  - By default, cloud-init will add it to the authorized keys of the default user (ubuntu for Ubuntu, centos for CentOS, see description of the VM in AppDB)
  - You can inject more keys to other users with the user-data but adding it here makes it independent to errors of the user-data → access the VM to debug issues

User data in cloud-init

- Some of the supported formats:
  - user script: execute it. (begins with #!)
  - Cloud Config Data: cloud-config is the simplest way to accomplish some things via user-data. Using cloud-config syntax, the user can specify certain things in a human friendly format. (begins with #cloud-config)
  - include file: contains a list of urls, one per line. Each of the URLs will be read, and their content will be passed through this same set of rules. (begins with #include)
  - gzipped content → uncompress and use as it were not compressed.

Contextualize with a script

EXAMPLE – NO NEED TO EXECUTE

Create a simple script, called script.sh

```
#!/bin/sh
echo "Hello World." > /root/context.txt
echo "The time is now $(date -R)!" >> /root/context.txt
```

Instantiate VM

```
~$ occi -e $ENDPOINT -n x509 -X -x $X509_USER_PROXY \
        -a create -r compute  \
        [...]
        --context user_data="file://$PWD/script.sh"
```

Check results

```
~$ ssh -i test.key ubuntu@155.210.71.129 "sudo cat /root/
context.txt"
Hello World
The time is now Sat, 20 Jun 2015 18:01:29 +0100
```

cloud-config

- Cloud-config is cloud-init own configuration format.
- Uses YAML (invalid syntax will make the contextualization fail!)
- Examples:
  - Create users and groups
  - apt-get upgrade should be run on first boot
  - Install packages
  - Run commands
  - …
- cloud-init documentation contains examples for all the supported options: http://cloudinit.readthedocs.org/

Sample cloud-config file

```
#cloud-config

users:
  - default
  - name: myuser
    sudo: ALL=(ALL) NOPASSWD:ALL
    lock-passwd: true
    ssh-import-id: myuser
    shell: /bin/bash
    ssh-authorized-keys:
      - <your key here>

package_upgrade: true

packages:
  - ca-policy-egi-core
  - occi-cli
  - voms-clients
```

Tells cloud-init this is a cloud-config file

Configure default user (e.g. ubuntu)

Create a user called "myuser" able to sudo and with a ssh key

Run apt-get upgrade (or yum upgrade)

Install some packages

What about Windows?

- cloud-init is linux-only, but
- **cloudbase-init** can help!
- Features:
  - setting hostname
  - user creation
  - group membership
  - static networking
  - SSH user's public keys
  - user_data custom scripts running in various shells (CMD.exe / Powershell / bash)

cloudbase-init

- Installation:
  - Get installer at https: //github.com/stackforge/cloudbase-init#binaries
  - Installer can also execute sysprep if needed.

- Contextualization
  - user-data can only be a script
  - but cloudbase-init will use all the meta-data (networking, hostname, keys)

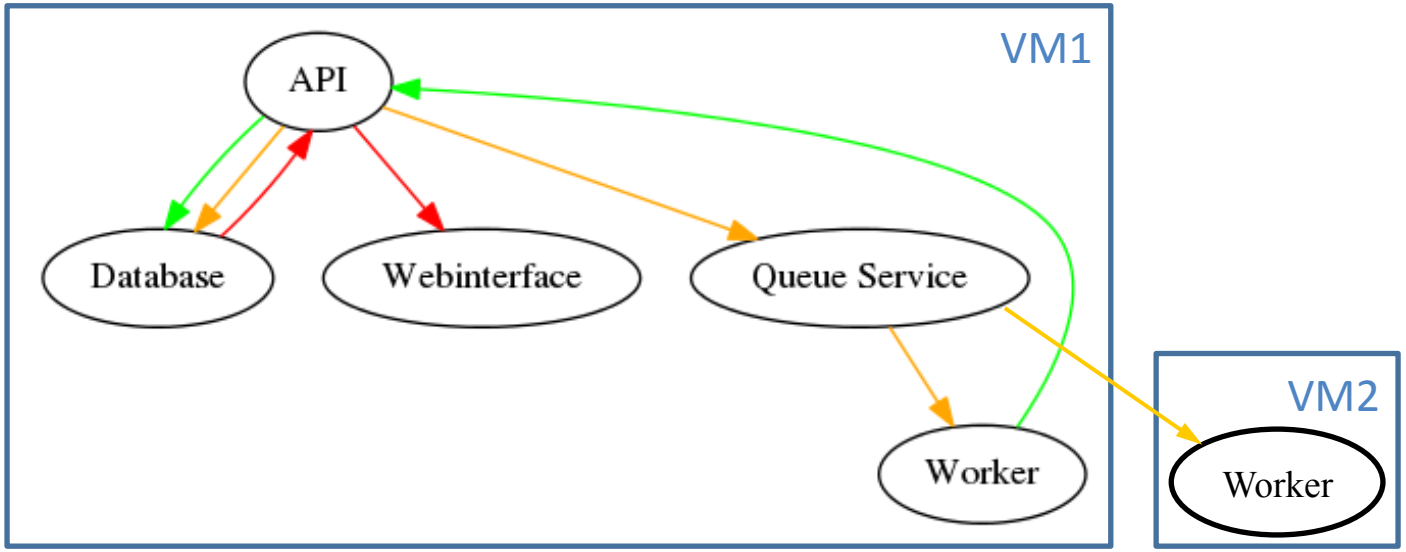
EGI-Engage

Exercise sheet
Fractal application with contextualisation

Fractal application on FedCloud

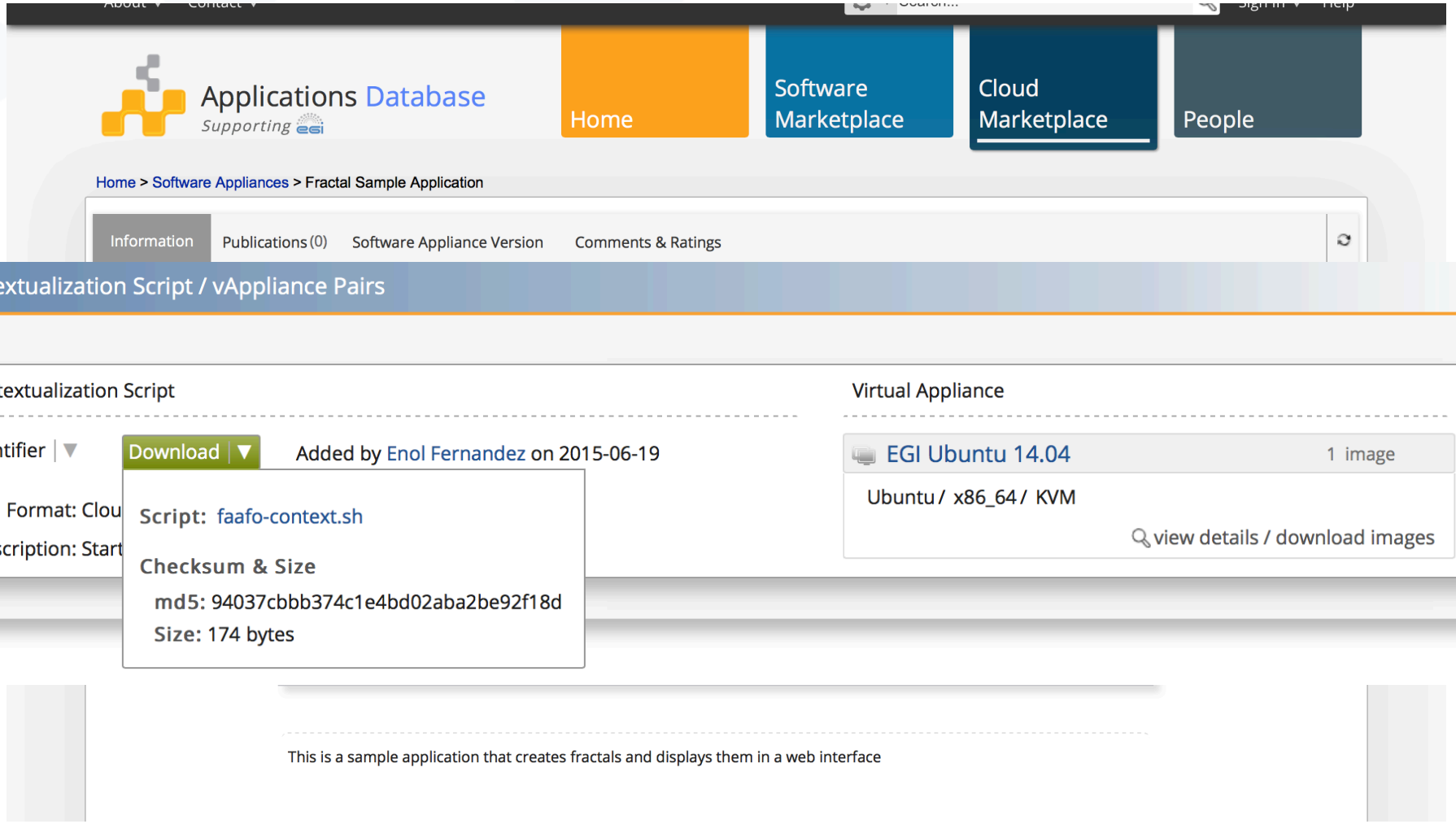
- Sample application borrowed from OpenStack “First Application For OpenStack” tutorial*
- Compute & display fractals using a set of worker nodes that get the tasks using a queue service

- A single binary includes all services.
- Configuration is controllable with command-line parameters

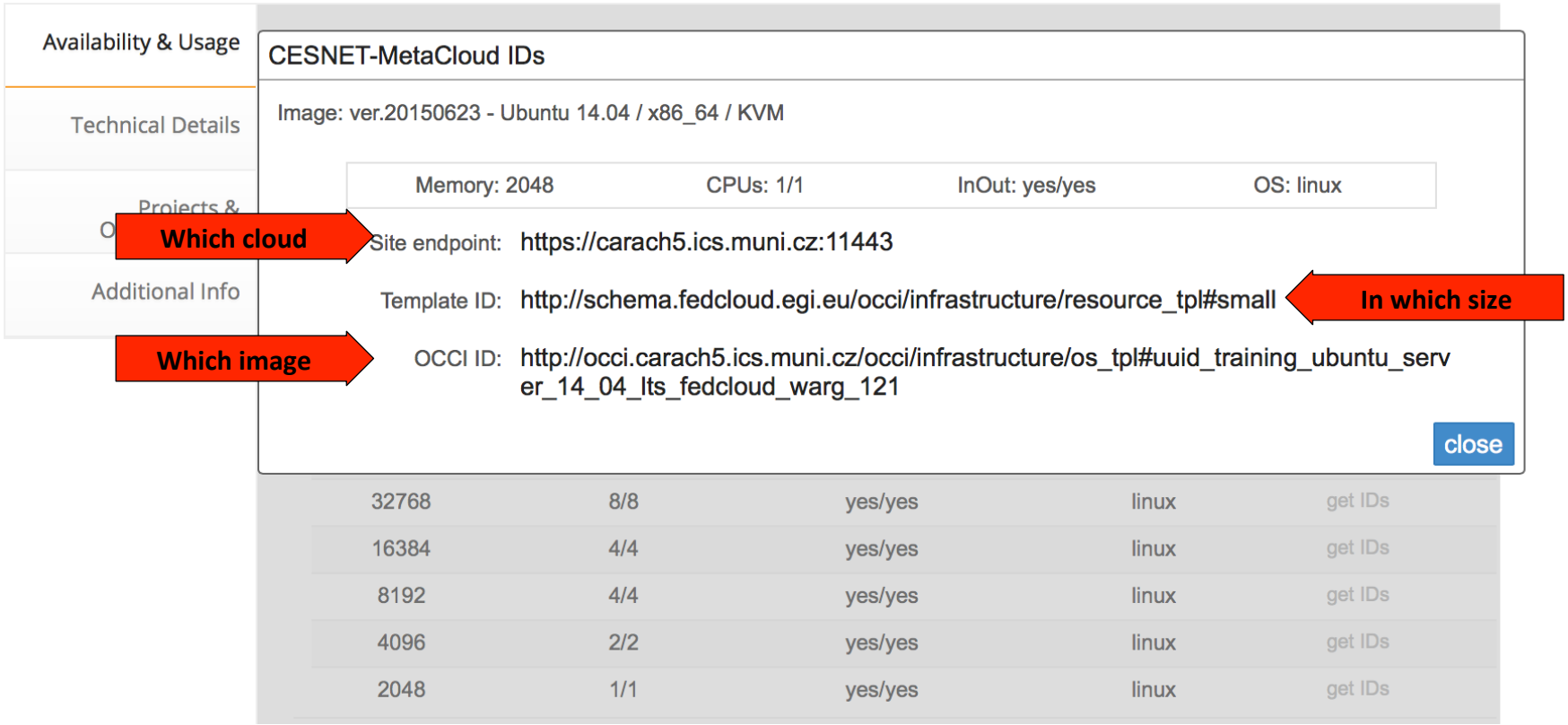


*Licensed under Creative Commons Attribution 3.0 License http://creativecommons.org/licenses/by/3.0/legalcode.

Where to find contextualised VAs?
Software Appliances in AppDB



VA details


Availability & Usage
Technical Details
Projects &
Additional Info
CESNET-MetaCloud IDs
Image: ver.20150623 - Ubuntu 14.04 / x86_64 / KVM
Memory: 2048
CPUs: 1/1
InOut: yes/yes
OS: linux
Which cloud
Site endpoint: https://carach5.ics.muni.cz:11443
Template ID: http://schema.fedcloud.egi.eu/occi/infrastructure/resource_tpl#small
In which size
Which image
OCCI ID: http://occi.carach5.ics.muni.cz/occi/infrastructure/os_tpl#uuid_training_ubuntu_server_14_04_lts_fedcloud_warg_121
close
32768 8/8 yes/yes linux get IDs
16384 4/4 yes/yes linux get IDs
8192 4/4 yes/yes linux get IDs
4096 2/2 yes/yes linux get IDs
2048 1/1 yes/yes linux get IDs

Workflow

1. Get the VA details from AppDB
2. Get the contextualization script from AppDB
3. Start a VM that holds the complete application and test it → VM1
4. Modify the contextualization script for a worker
5. Start a worker VM that listens on the queue service → VM2
6. Create some fractals
7. Check that two workers generate fractals

Start Application

```
~$ OS_TPL=<OCCI ID from AppDB>

~$ RESOURCE_TPL=<Template ID from AppDB>

~$ curl <context script URL from AppDB> > context.sh

~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action create --resource compute \
        --mixin $RESOURCE_TPL --mixin $OS_TPL \
        --attribute occi.core.title="fractal$(date +%s)" \
        --context user_data="file:///$PWD/context.sh" \
        --context public_key="file:///$HOME/.ssh/id_rsa.pub"

~$ MASTER_ID=...
```

From the Fractal VA

Save the ID in an Env. variable

List and describe your VM instance

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action list --resource compute
```

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action describe --resource $MASTER_ID
```

Accessing the appliance

- If the VM does not have a public IP (BIFI endpoint):

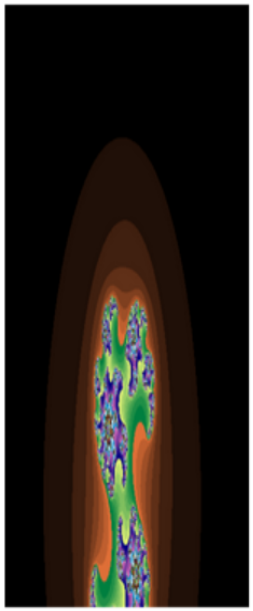
```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action link --resource $MASTER_ID \
        --link /network/public
```

- Login with your ssh key and create some fractals:

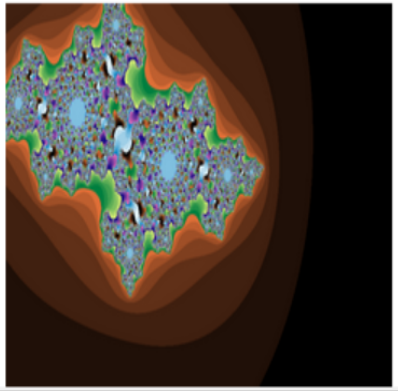
```
~$ faafo --endpoint-url http://localhost --verbose create
```

- Check in your browser:
  - http://<your vm ip>/

Give the VM some minutes for contextualization! It installs and configures several software packages



UUID	702e6c72-36a8-41ac-91bf-0e7420cbe5de
Duration	1.35273122787 seconds
Dimensions	316 x 629 px
Iterations	423
Parameters	xa = -2.97293091744 xb = 3.20980065971 ya = -2.7268038736 yb = 0.576010426675
Filesize	17317 bytes
Checksum	a707671566077d8dff9c7f1d823f760c5bbac5f626e914fd2597aefee693bda6
Generated by	stoor14



UUID	512a91f1-f60b-4713-aa1a-452d2a072f74
Duration	3.96059513092 seconds
Dimensions	774 x 971 px
Iterations	157
Parameters	xa = -1.23113331702 yb = 2.82781937233

Add workers to the app

- Adapt the contextualization script
  - not start the whole application, but just the worker
  - connect to the existing VM for messaging queue
- Create worker.sh with the following content:

```
#!/usr/bin/env bash
curl -L -s http://git.openstack.org/cgit/stackforge/faafo/
plain/contrib/install.sh | bash -s -- \
    -i faafo -r worker -e 'http://<VM IP>' -m 'amqp://
guest:guest@<VM IP>:5672/'
```

Start Worker – On the same, or on a different cloud site

```
~$ OS_TPL=<OCCI ID from AppDB>

~$ RESOURCE_TPL=<Template ID from AppDB>

~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action create --resource compute \
        --mixin $RESOURCE_TPL --mixin $OS_TPL \
        --attribute occi.core.title="worker$(date +%s)" \
        --context user_data="file:///$PWD/worker.sh" \
        --context public_key="file:///$HOME/.ssh/id_rsa.pub"

~$ WORKER_ID="..."
```

Only if different cloud site is used for Master and Worker

Testing the worker

1. [Optional] disable the generation of fractals by the master

Edit /etc/supervisor/conf.d/faafo.conf (use sudo for opening the editor) and comment out the [program:faafo-worker] entry:
[program:faafo_worker]
command=/usr/local/bin/faafo-worker
priority=20

Restart the supervisor: sudo service supervisor restart.

2. Create some fractals at the master:

```
~master $ faafo --endpoint-url http://localhost \
                --verbose create --tasks 20
```

3. Check in the browser the “Generated by” of new fractals

Remember to delete your VMs!

```
~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $MASTER_ID

~$ occi --endpoint $ENDPOINT \
        --auth x509 --voms --user-cred $X509_USER_PROXY \
        --action delete --resource $WORKER_ID
```

EGI-Engage

Preparing your own images

Custom Image vs Contextualization

Custom Images

- Time consuming, needs virtualization software
- Static configuration
- Fast startup of VM
- Requires moving large files to sites
- Easier to debug

Contextualization

- Fast to use, just add user data to VM
- Configuration on creation
- Slow startup of VM
- Works on top of existing images
- Hard to debug if fails

- **EGI.eu maintains core VM images in AppDB.**
- **These can be used as starting point in both scenarios.**
- **More later!**

Image Creation

- Create VM with some virtualization software:
  - VirtualBox (available for most OS, easy to use)
  - KVM
  - Xen
  - VMWare
  - …
- Once VM is configured as needed, export to the proper format
  - OVF is the preferred format in the EGI Federated Cloud

Image containers and disk images

- A VM is a set of disk images within a container
  - Normally one disk image per VM
- Disk images: contain the actual disk for the VM
  - raw: a direct copy of the disk into a file
  - qcow2: copy-on-write format that allows fast duplication of images
  - VMDK: sparse file format created by VMWare, now open format
  - …
- Containers: package the disks + metadata
  - raw: no container, just the disk image
  - OVF (Open Virtualization Format): open format by the Distributed Management Task Force (DMTF)
  - …

OVF, OVA and VMDK

- OVF is a specification for packaging and distributing software appliances
- An OVF package consists of:
  - OVF description (XML file with .ovf extension) that contains the metadata (name, hardware requirements, etc.)
  - One or more disk images: any format can be used, but in practice every implementation uses VMDK
  - Optional auxiliary files (certificates, checksums, …)
- An OVA (OVF archive) is a tar file of a OVF package
  - Easier for distribution than a directory with the files

Packer

- Packer is a tool for creating machine and container images for multiple platforms from a single source configuration
  - Reproducible builds
  - Creates VM in your virtualization platform (or cloud)
  - Executes scripts on top to install/configure
  - Can apply the same scripts to different platforms (so all images are the same at the end)

Packer Image Template

- JSON file containing the description of what to build.
- Builders
  - Define how to install/start the VM for a given platform
  - Supports AWS, OpenStack, VirtualBox, qemu, VMWare, …
- Provisioners
  - Define how to install and configure software into the image
  - Several types: shell scripts, uploading files, puppet, chef, ansible, …

Packer builder

```
"builders": [{
    "type": "virtualbox-iso",
    "guest_os_type": "Ubuntu_64",
    "disk_size": 2000,
    "iso_url": "http://archive.ubuntu.com/ubuntu/dists/trusty/main/installer-amd64/
current/images/netboot/mini.iso",
    "iso_checksum":
"bc09966b54f91f62c3c41fc14b76f2baa4cce48595ce22e8c9f24ab21ac8d965",
    "iso_checksum_type": "sha256",
    "ssh_username": "root",
    "ssh_password": "rootpasswd",
    "ssh_wait_timeout": "90m",
    "shutdown_command": "shutdown -h now",
    "http_directory": "httpdir",
    "http_port_min": 8500,
    "http_port_max": 8550,
    "boot_command": [
        "<esc>",
        " install auto=true priority=critical preseed/url=http://{{ .HTTPIP }}:
{{ .HTTPPort }}/ubuntu.cfg",
        "<enter>"
    ],
    "vm_name": "Ubuntu.14.04.20150623"
  }],
```

Packer provisioners

```
"provisioners": [
    {
        "type": "file",
        "source": "provisioners/cloud.cfg",
        "destination": "/root/cloud.cfg"
    },
    {
        "type": "file",
        "source": "provisioners/sshd_config",
        "destination": "/root/sshd_config"
    },
    {
        "type": "shell",
        "script": "provisioners/script.sh"
    }
 ]
```

Provisioner: script

```
#!/usr/bin/env bash

# update already installed packages
apt-get update

if [ "x$(lsb_release -rs)" == "x12.04" ]; then
  apt-get --assume-yes install python-software-properties
  add-apt-repository -y ppa:iweb-openstack/cloud-init
fi

apt-get update
apt-get --assume-yes upgrade
apt-get --assume-yes install cloud-init curl

# move configuration files to their right place
mv /root/sshd_config /etc/ssh/sshd_config
mv /root/cloud.cfg /etc/cloud/cloud.cfg

ln -s /dev/null /etc/udev/rules.d/75-persistent-net-generator.rules

# remove ssh keys
rm -f /etc/ssh/ssh_host_*

# lock root password
passwd -l root

# clean bash history and cloud init logs
rm -f ~/.bash_history
rm -f /var/log/cloud-init*

# Remove virtualbox things
rm -f VBoxGuestAdditions.iso
```

EGI Endorsed VM images

- Process to assure that a Virtual Machine Image (VMI)/ Virtual Appliance (VA) published in AppDB is *well-configured*, *secure* and *up-to-date*
  - *These have 'EGI' in their title in AppDB!*
- Guide on how to create, configure, harden and publish images into AppDB
  - https://wiki.egi.eu/wiki/Virtual_Machine_Image_Endorsement
- Currently using packer for creating the images:
  - See https://github.com/EGI-FCTF/VMI-endorsement

Video (~5')

- Build (video) of Ubuntu 14 for EGI endorsed image:
  - Packer installs the OS
  - Waits until it can be accessed
  - Executes the provisioners
  - Exports the Image as OVF
- Video is available at https://documents.egi.eu/document/2553

Make your VM available via AppDB

- Package the OVF + disk into OVA
  - https://github.com/EGI-FCTF/VMI-endorsement/blob/master/tools/ovf2ova.sh
- Upload the image to a repository
  - http://appliance-repo.egi.eu/images/ available for fedcloud.egi.eu VO members
- Register image in AppDB
  - Create a VA entry
    https://wiki.appdb.egi.eu/main:faq:how_to_register_a_virtual_appliance
  - Create a new version within the VA and point to your image location
    https://wiki.appdb.egi.eu/main:guides:guide_for_managing_virtual_appliance_versions_using_the_portal
- Request VA endorsement in your VO
  (so VMI gets distributed to the infrastructure)
  - Endorsement requests are dealt with by designated VO members
    https://wiki.appdb.egi.eu/main:guides:notify_virtual_organization_representatives

EGI-Engage

Next steps to become a user

Documentation

EGI Federated Cloud User Support doc. entry page:

- https://wiki.egi.eu/wiki/Federated_Cloud_user_support

Federated Cloud user support

Main | Roadmap and Innovation | Technology | **For Users** | For Resource Providers | Media

Contents [hide]

Technical support is available via the EGI.eu Support Team

Users of the EGI Federated Cloud are scientists working in many fields, who can benefit of a flexible environment for running their workloads. Also, the EGI cloud is suitable to projects aiming to provide services and platforms to the scientific community.

Concept [edit]

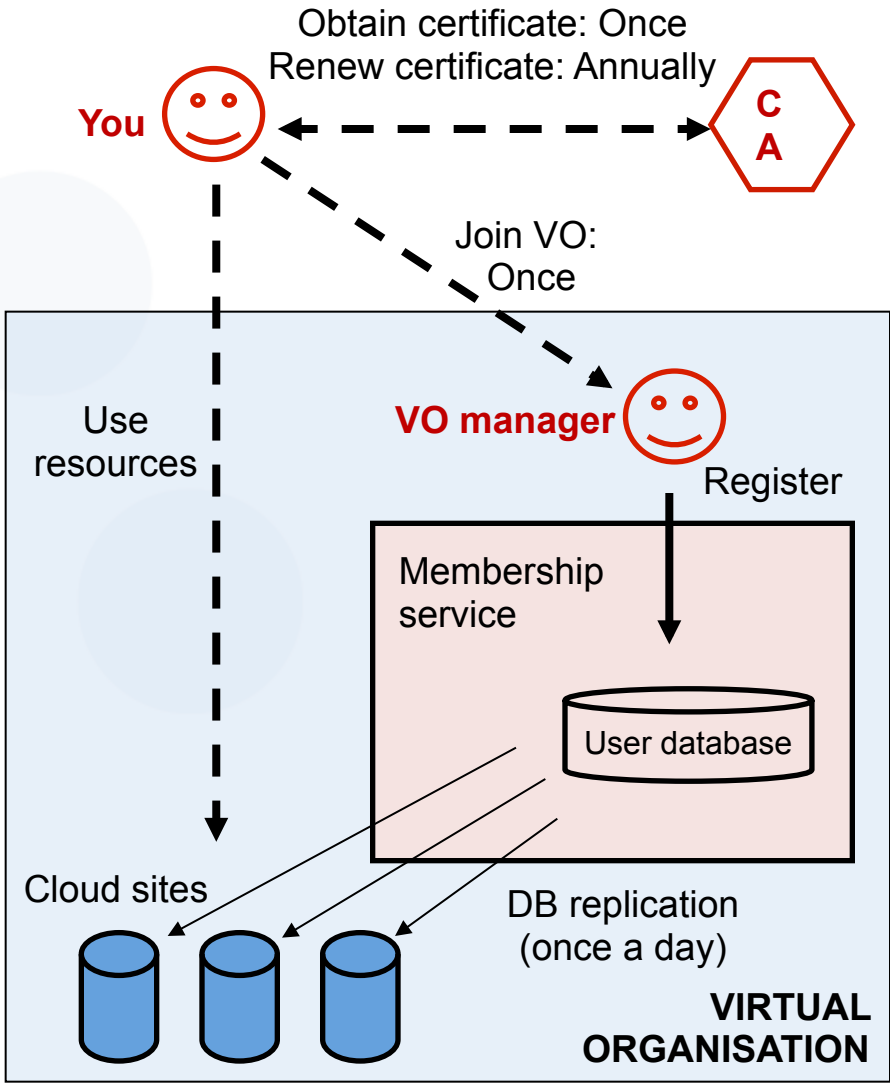
The EGI Federated Cloud is a seamless grid of academic private clouds and virtualised resources built around open standards and focusing on the requirements of the scientific community. The result is a new type of research e-infrastructure based on the mature federated operations services that make EGI a reliable resource for science. When using EGI Federated Cloud resources, researchers and research communities can count on:

- Total control over deployed applications
- Elastic resource consumption based on real needs
- Immediately processed workloads – no more waiting time
- An extended e-Infrastructure across resource providers in Europe
- Service performance scaled with elastic resource consumption
- Single sign-on at multiple, independent providers

Getting access

Your steplist:

1. Obtain certificate from your national CA: http://www.igtf.net
2. Register at the VO
   - fedcloud.egi.eu is a good starting point
   - Other VOs: http://operations-portal.egi.eu/vo/search
3. VO manager authorizes You
   - Membership DB updated
   - Identity replicated to resource within 1 day
4. Interact with the resorurces
   - rOCCI
   - High-level tool



Support for the Federated Cloud

Dedicated technical consultancy for each community (Request at support@egi.eu)

F2F/Web Meetings	• Identify suitable setup • Allocate technical experts • Define milestones
Doc	• Step by step guides • Tutorials • Examples
Continuous tracking and support	• Technical integration • Periodic meetings
EGI VM Images	• Main OS versions • Secure, up-to-date • Contextualisation
Fedcloud.egi.eu VO	• Resources for prototyping • Enabled on all sites • Usable for 2x6 months
Migration into production	• Identifying committed resource providers • Support for VO setup

Support through the NGIs

http://www.egi.eu/about/ngis/

- EGI's federated support model
  - National support teams (NGIs)
  - Topic/discipline-specific support teams (see next slide)
  - EGI.eu UCST – primarily coordination & support for supporters (support@egi.eu)

Keep in mind!

- You have **root** access to your virtual machines
- Your virtual machines are often visible from the Internet
- It is up to you to keep your virtual machines updated and secure
- **DO NOT USE** password-based authentication for remote access
- You should terminate your virtual machine as soon as it is not needed anymore

EGI-Engage

EGI plans

Plans

- In the IaaS layer you saw today:
  - New abstractions in the rOCCI client (2016 Q1)
  - VA instantiation interface in AppDB (2016-17)
  - Create VM snapshots, resize VMs on sites (OCCI extension to v1.2)
- In the PaaS layer:
  - Tutorials and SLAs for high-level services
  - Complete the integration and guides of emerging tools, e.g. OCCO
  - Extend ‘proxy factory concept’ from tutorials to production users
- In the SaaS layer
  - Community specific service developments:
    - BBMRI, ELIXIR, DARIAH, MoBrain, EISCAT_3D, LifeWatch, EPOS, Disaster Mitigation
    - HumanBrainProject, Marine and Fisheries, etc.
  - Operation of community SaaS based on SLAs
- Collaborations with cloud federations
  - Canfar, FogBow, HARNESS, Nectar, CERN, JetStream, etc.
  - Technology exchange; Interoperability; User support and training

Next EGI Community Event

- EGI Community Forum
- Bari, Italy
- 2015, Nov 10-13.
- http://cf2015.egi.eu/
- Types of contributions
  - Presentations, tutorials, workshops, demos, posters
- Main topics:
  - Community Engagement and Innovation
  - Virtual Research Environments
  - Data and Computing
  - Identity provisioning, Authentication, Authorization and Accounting
  - Open Science Commons

Thank you for your attention.

Questions?

PLEASE RETURN THE FEEDBACK FORMS!

This work by Parties of the EGI-Engage Consortium is licensed under a Creative Commons Attribution 4.0 International License.