

Empowering Cloud with latest VMDIRAC improvements

DIRAC User Workshop, Ferrara, 2015



Víctor Méndez Muñoz
vmendez@uab.es

- A single context to drive them all
- Cloud flexibility: Hybrid and multi-layered
- Use Case: S3 1K genome processing
- Conclusions

A single context to drive them all

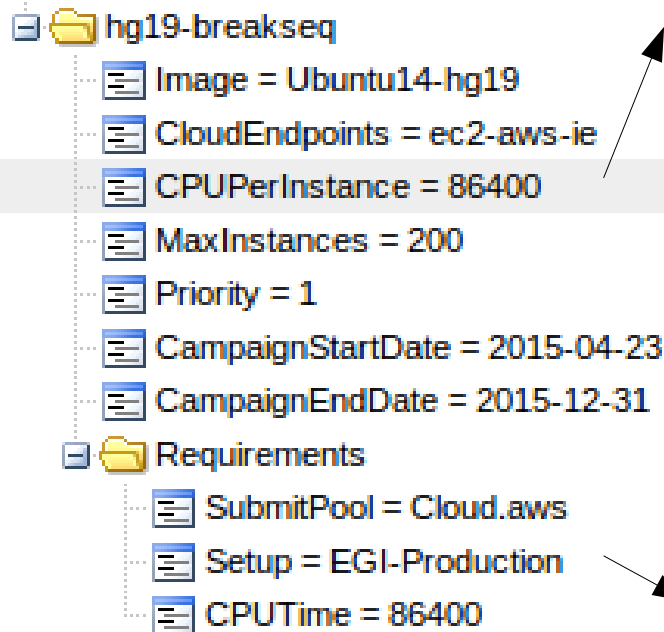
VMDIRAC v1r3 -April 2015- includes a single cloudinit method for

- OpenStack
 - OpenNebula
 - AWS
-
- A RunningPod will bind Job Requirements to be allocated in the Vms + A DIRAC Image + List of cloud endpoints
 - DIRAC Image in the Configuration System, includes the Image contextualization
 - Cloud endpoint should be setup in the DIRAC Configuration

A single context to drive them all

RunningPod setup:

Resources | VirtualMachines | RunningPods



minimal overall CPU of the DIRAC jobs waiting in the task queued to submit a new VM, elasticity:

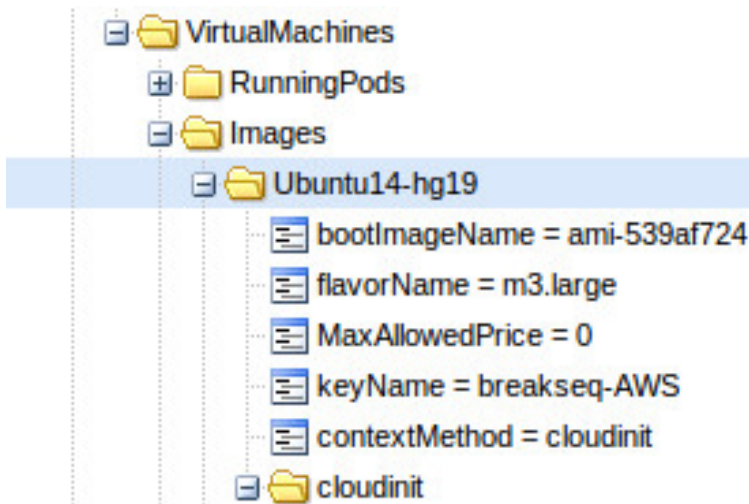
- 0 → no minimal CPU in tasks queue, best total wall time
- Average required CPU → compromise between VM efficiency and total wall time
- A very large value to maximize the efficiency in terms of VM creation overhead

Requirements you want to put in /opt/dirac/etc/dirac.cfg in the Vms Will be matched with the jobs

A single context to drive them all

DIRAC Image setup

Resources | VirtualMachines | Images



Your golden image name in the IaaS provider

Flavor = Instance Type, depending on your IaaS provider

If MaxAllowedPrice = 0 or not defined then not spot instances are used

KeyName in your IaaS to push public key in the Vms (deploying / debugging matters)

A single context to drive them all

DIRAC Image setup: Single cloudinit contextualization
Resources | VirtualMachines | Images

```
cloudinit
vmCertPath = /opt/dirac/VMcertkey/server.vmdirac.ub.short.cert.pem
vmKeyPath = /opt/dirac/VMcertkey/server.vmdirac.ub.short.key.pem
vmRunJobAgentURL = https://github.com/vmendez/VMDIRAC/raw/master/WorkloadManagementSystem/private/bootstrap/run.job-agent
vmRunVmMonitorAgentURL = https://github.com/vmendez/VMDIRAC/raw/master/WorkloadManagementSystem/private/bootstrap/run.vm-monitor-agent
vmRunVmUpdaterAgentURL = nouse
vmRunLogAgentURL = https://github.com/vmendez/VMDIRAC/raw/master/WorkloadManagementSystem/private/bootstrap/run.log
vmDiracContextURL = https://github.com/vmendez/VMDIRAC/raw/master/WorkloadManagementSystem/private/bootstrap/breakseq-DIRAC-context.sh
vmCvmfsContextURL = nouse
vmContextualizeScriptPath = /opt/dirac/pro/VMDIRAC/WorkloadManagementSystem/private/bootstrap/cloudinit-static-template.bash
```

➔ VM host cert/key

➔ DIRAC/VMDIRAC agents to push place in the VM

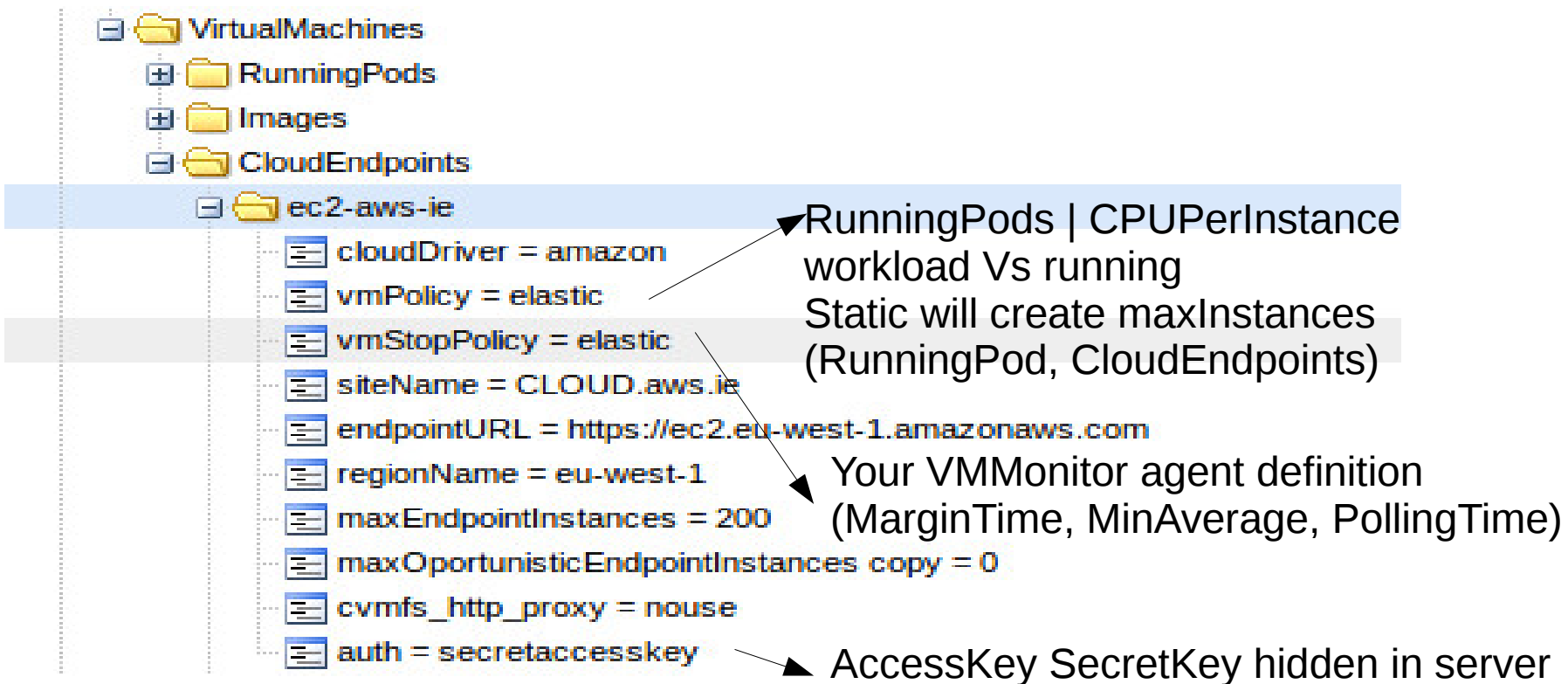
➔ Your CVMFS client contextualization script

➔ Your cloudinit static script + All Image/Endpoint setup → cloudinit booting VM

A single context to drive them all

Cloud endpoint setup: AWS

Resources | VirtualMachines | CloudEndpoints



The image shows a configuration tree for a cloud endpoint. The tree structure is as follows:

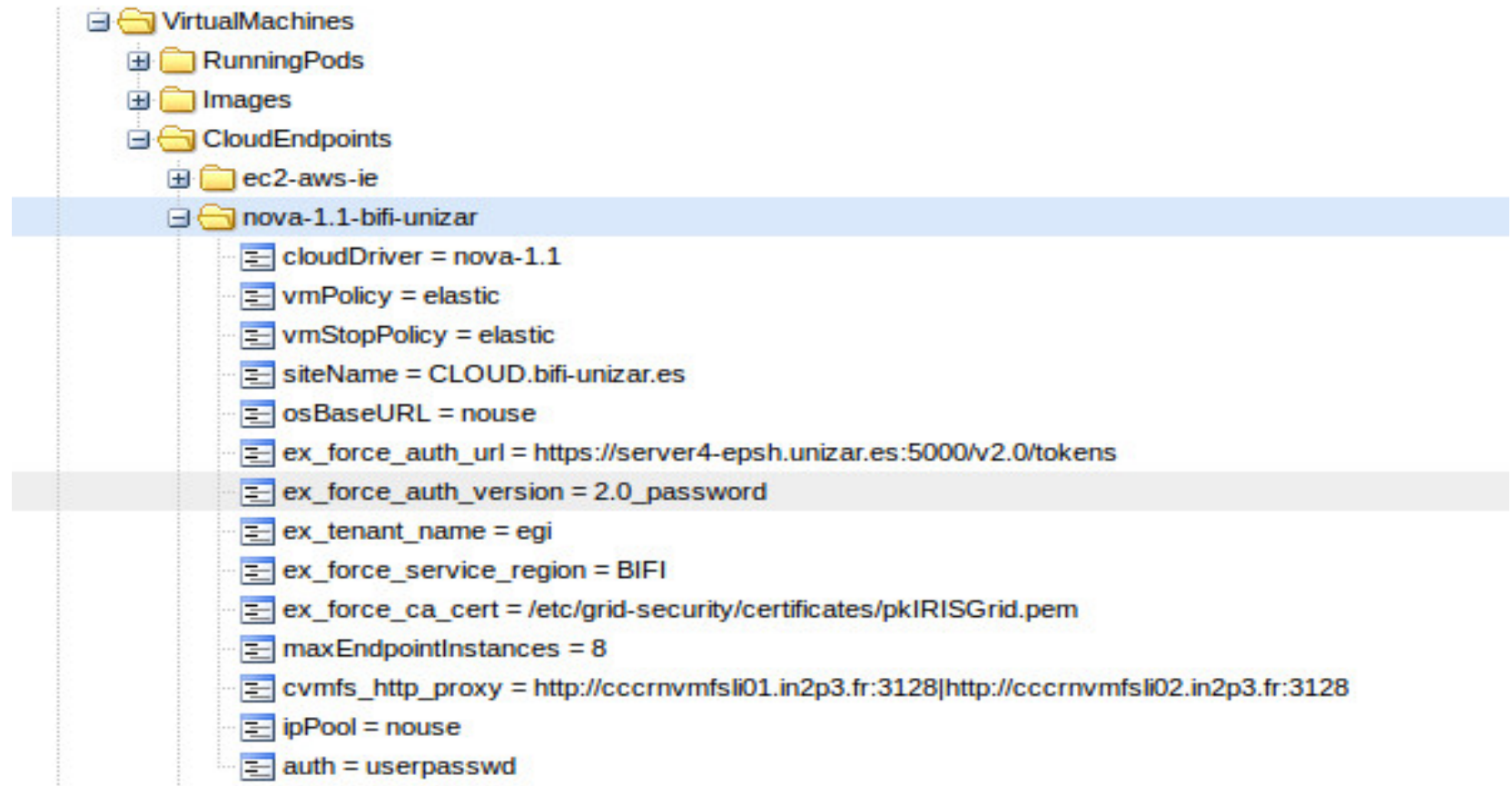
- VirtualMachines
 - RunningPods
 - Images
 - CloudEndpoints
 - ec2-aws-ie**
 - cloudDriver = amazon
 - vmPolicy = elastic
 - vmStopPolicy = elastic
 - siteName = CLOUD.aws.ie
 - endpointURL = https://ec2.eu-west-1.amazonaws.com
 - regionName = eu-west-1
 - maxEndpointInstances = 200
 - maxOpportunisticEndpointInstances copy = 0
 - cvmfs_http_proxy = nouse
 - auth = secretaccesskey

Annotations with arrows pointing to specific configuration items:

- An arrow points from the text "RunningPods | CPUPerInstance workload Vs running Static will create maxInstances (RunningPod, CloudEndpoints)" to the `vmPolicy = elastic` and `vmStopPolicy = elastic` lines.
- An arrow points from the text "Your VMMonitor agent definition (MarginTime, MinAverage, PollingTime)" to the `maxEndpointInstances = 200` line.
- An arrow points from the text "AccessKey SecretKey hidden in server" to the `auth = secretaccesskey` line.

A single context to drive them all

Cloud endpoint setup: OpenStack
Resources | VirtualMachines | CloudEndpoints

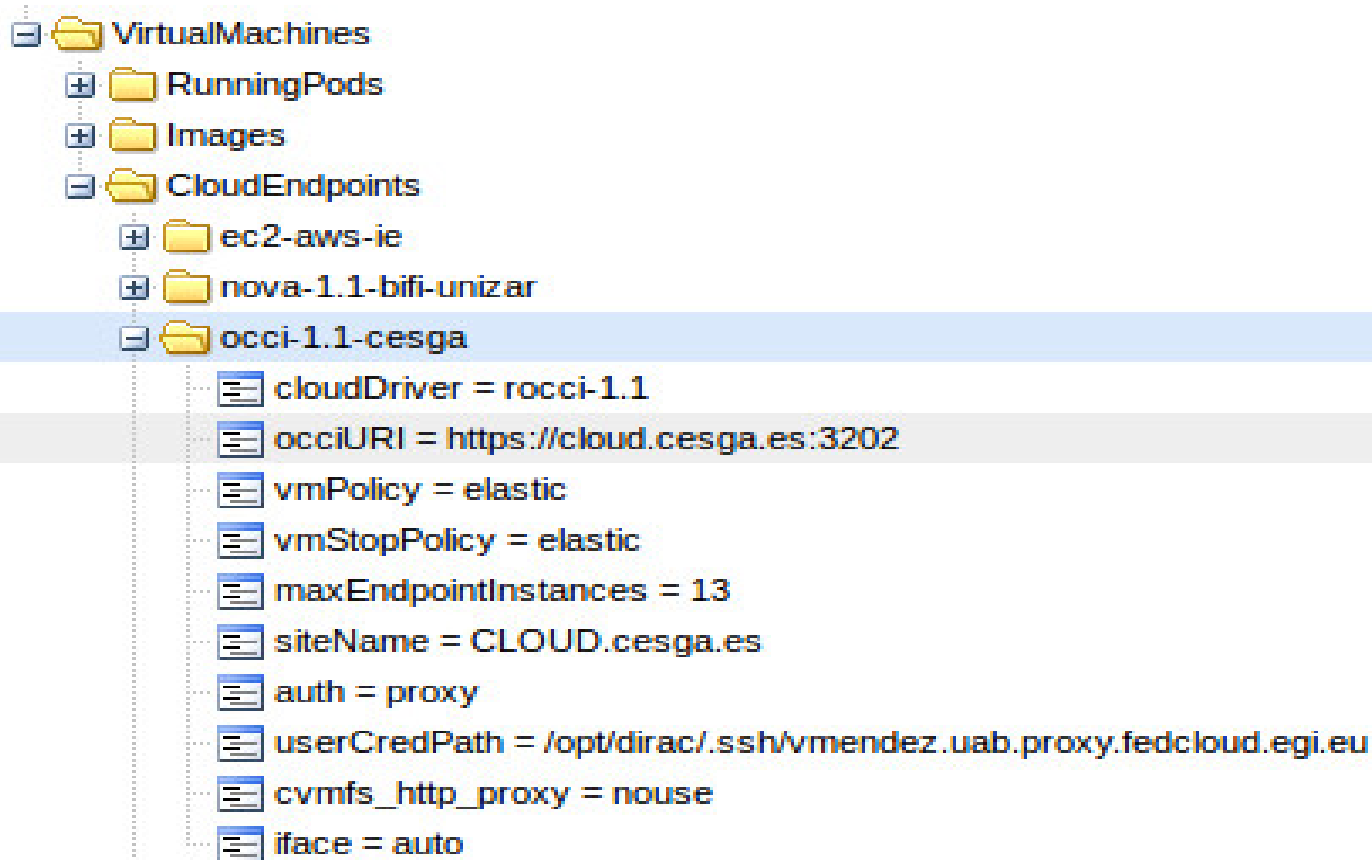


The image shows a hierarchical tree view of configuration resources. The tree is expanded to show the configuration for a specific CloudEndpoint named 'nova-1.1-bifi-unizar'. The configuration parameters are listed below the folder name.

- VirtualMachines
 - RunningPods
 - Images
 - CloudEndpoints
 - ec2-aws-ie
 - nova-1.1-bifi-unizar**
 - cloudDriver = nova-1.1
 - vmPolicy = elastic
 - vmStopPolicy = elastic
 - siteName = CLOUD.bifi-unizar.es
 - osBaseURL = nouse
 - ex_force_auth_url = https://server4-epsh.unizar.es:5000/v2.0/tokens
 - ex_force_auth_version = 2.0_password**
 - ex_tenant_name = egi
 - ex_force_service_region = BIFI
 - ex_force_ca_cert = /etc/grid-security/certificates/pkIRISGrid.pem
 - maxEndpointInstances = 8
 - cvmfs_http_proxy = http://cccrnvmfsl01.in2p3.fr:3128|http://cccrnvmfsl02.in2p3.fr:3128
 - ipPool = nouse
 - auth = userpasswd

A single context to drive them all

Cloud endpoint setup: OpenNebula
Resources | VirtualMachines | CloudEndpoints



Hybrid Cloud

- Usual Categorization:
 - Public and Private nature of IaaS resources
 - Public is any Cloud owned by third party
 - Private Cloud is a user infrastructure
- Other Categorizations:
 - Community Cloud: Is owned and / or managed by a particular user community.
 - Public: like some of EGI Fedcloud resources owned by providers which are EGI members
 - Private
 - Hybrid: f.e. public AWS + private OpenStack

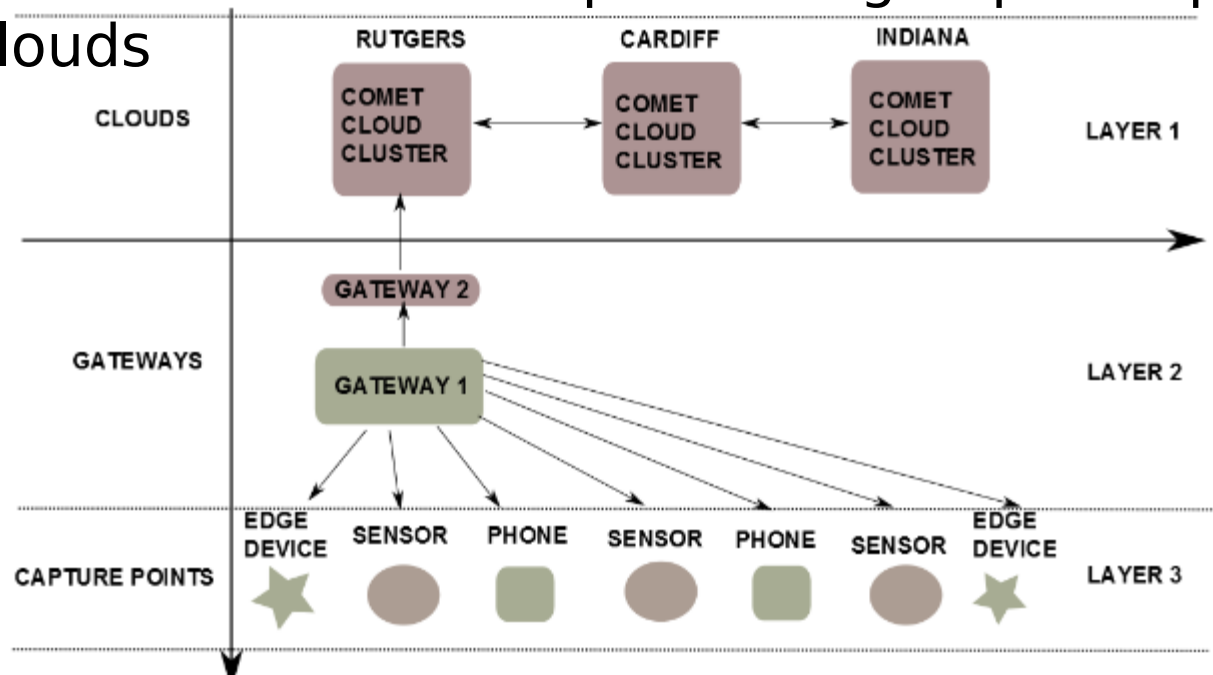
Multi-layered Cloud

- Taking advantage of pre-processing / post-processing to filter data out of the main processing in public pay-per-use Clouds

Example:

Input data
Filtered

Sensors:
Input data



CometCloud mobile data agregation in multi-layered cloud

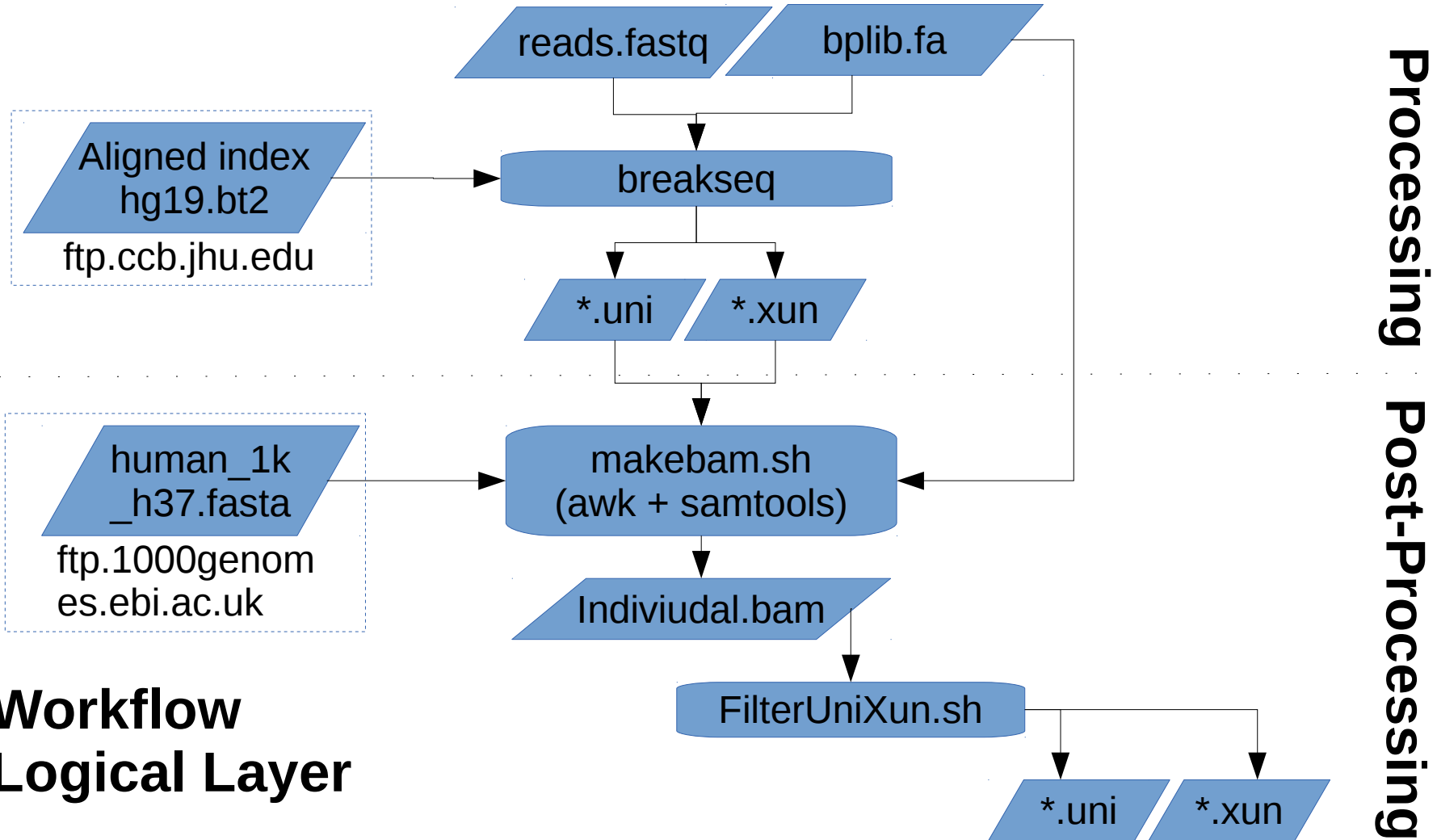
<http://nscac.rutgers.edu/CometCloud/sites/nscac.rutgers.edu/CometCloud/files/petri-diaz-FiCloud14-cm.pdf>

Use Case: S3 1k genome processing

Individual mapping of breakpoint junctions

- Breakseq provides genome processing, allowing mapping of reads with breakpoint junctions
- Human individuals specific mapping:
 - Data Patterns:
 - Homo sapiens raw genome: human_g1k_v37.fasta
 - Homo sapiens aligned genome sequence: .bt2
 - Breakpoint junctions of interest: bplib.fa
 - Input Data
 - Individuals to be mapped: reads.fastq
 - Output Data
 - Filtered allele inverted : .uni
 - Filtered reference allele: .xun

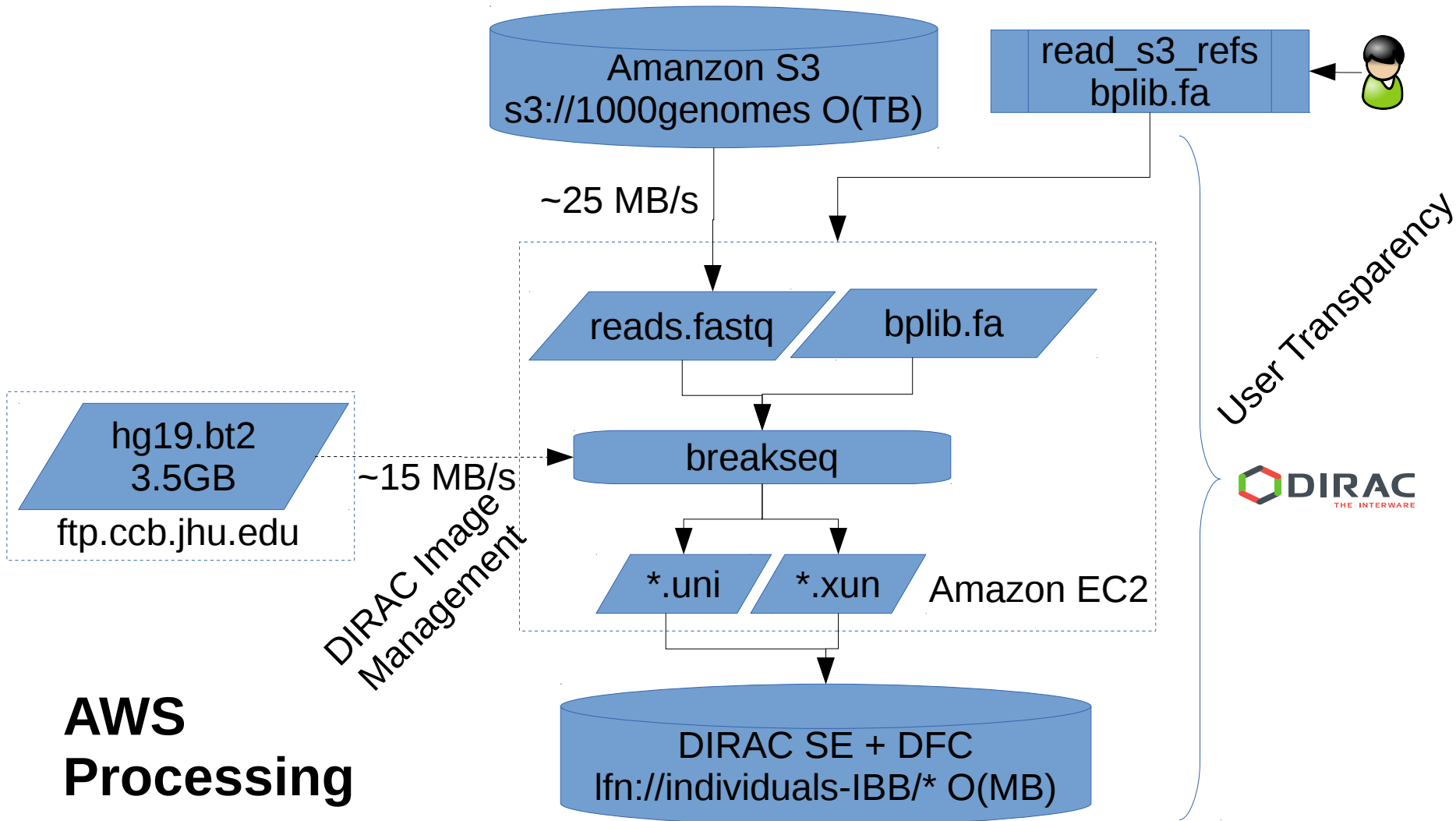
Use Case: S3 1k genome processing



**Workflow
Logical Layer**

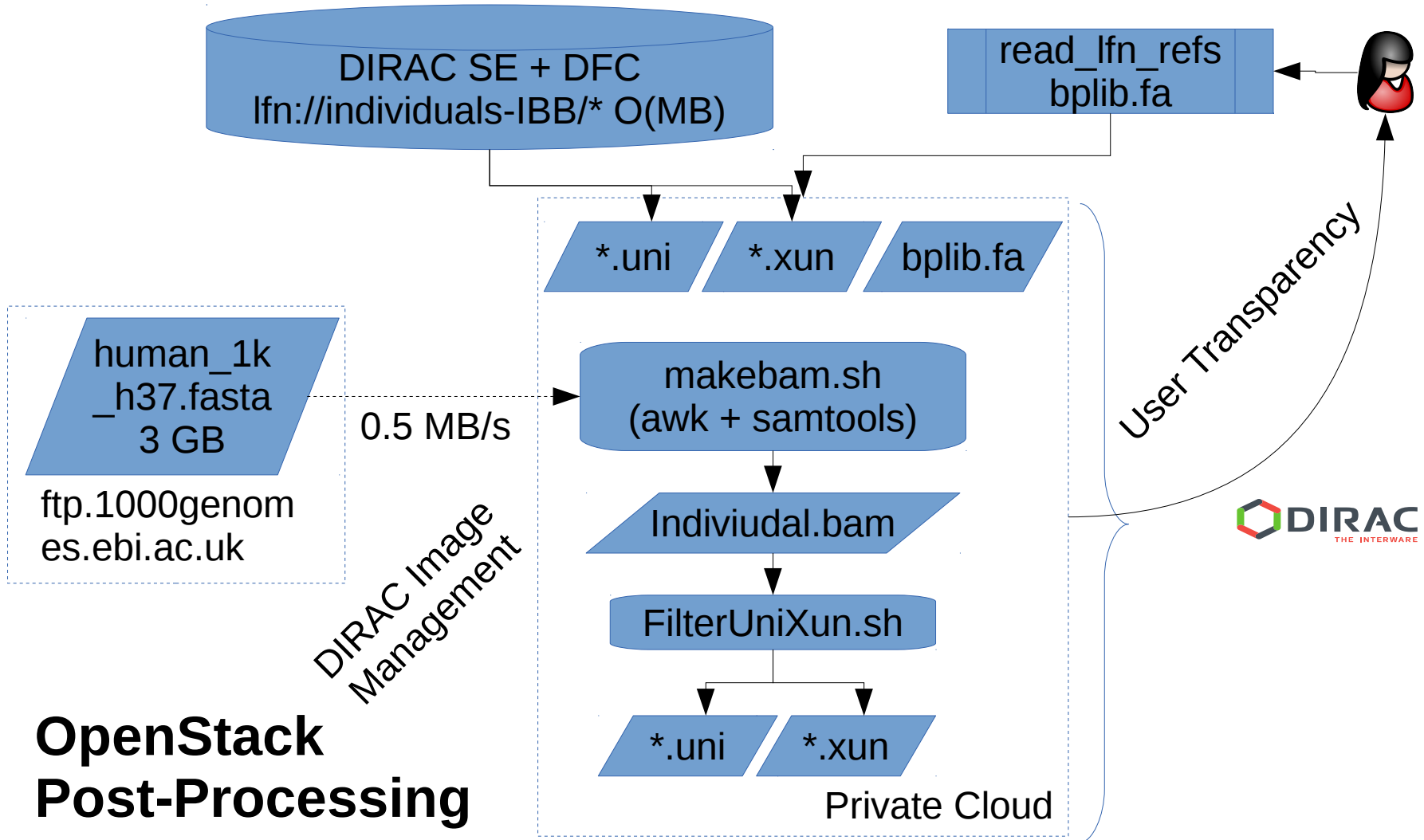
**Processing
Post-Processing**

Use Case: S3 1k genome processing



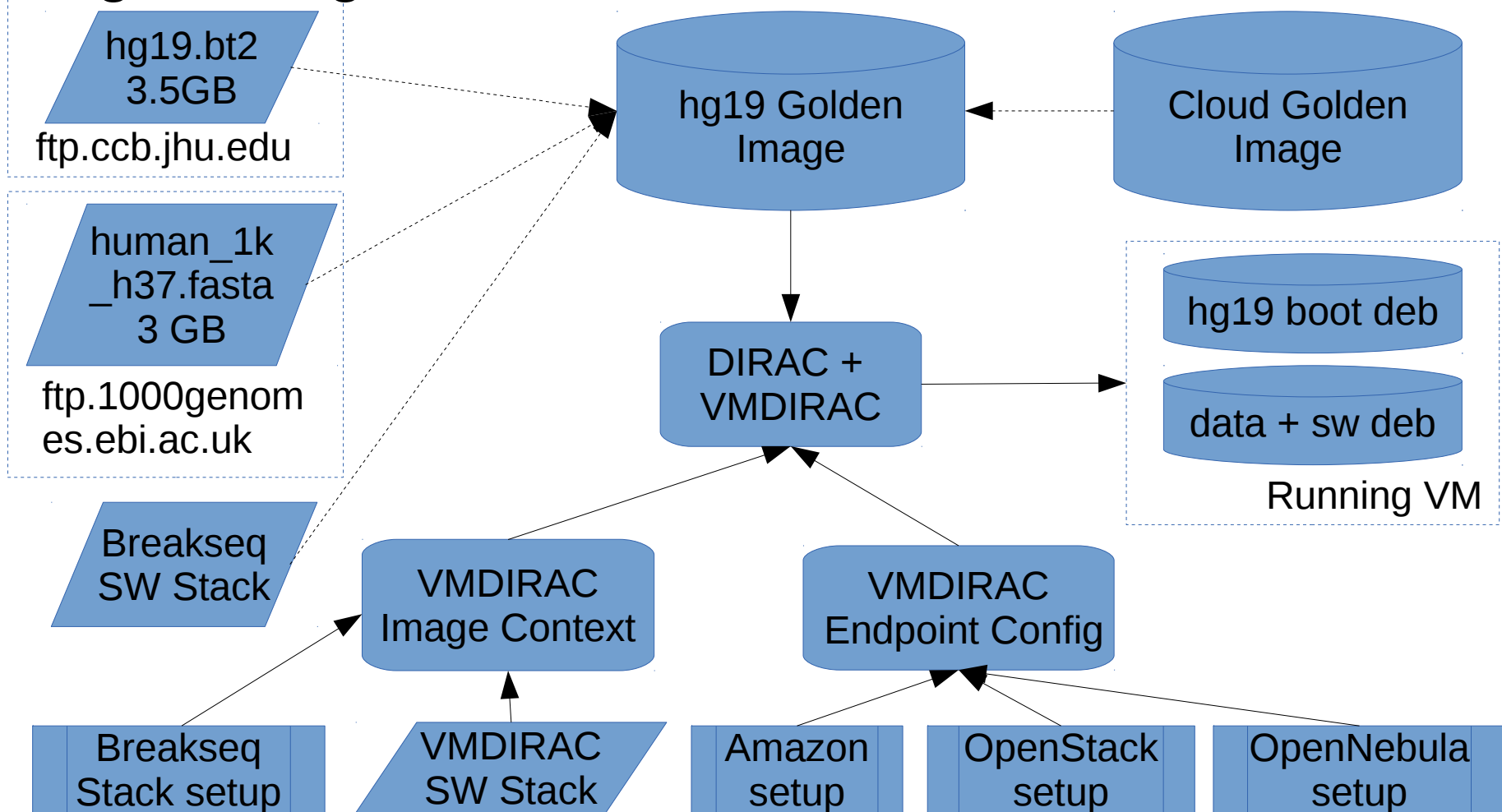
AWS Processing

Use Case: S3 1k genome processing



Use Case: S3 1k genome processing

Image management



- Cloudinit allows a single Image contextualization script for all your IaaS providers
- Cloudinit enables straight forward integration of the pilot 2.0, cloudinit script = pilot 2.0 script
- VMDIRAC setup allows hybrid cloud and multi-layer cloud setup to optimize computational costs and commercial cloud prices
- Everything is setup and managed by dirac administrator, enabling user transparency in the use of cloud resources

- Please, cite related papers in the link:

<https://scholar.google.com/citations?user=2RVFZrwAAAAJ&hl=es>

Thanks