

WMS and Computing Resources

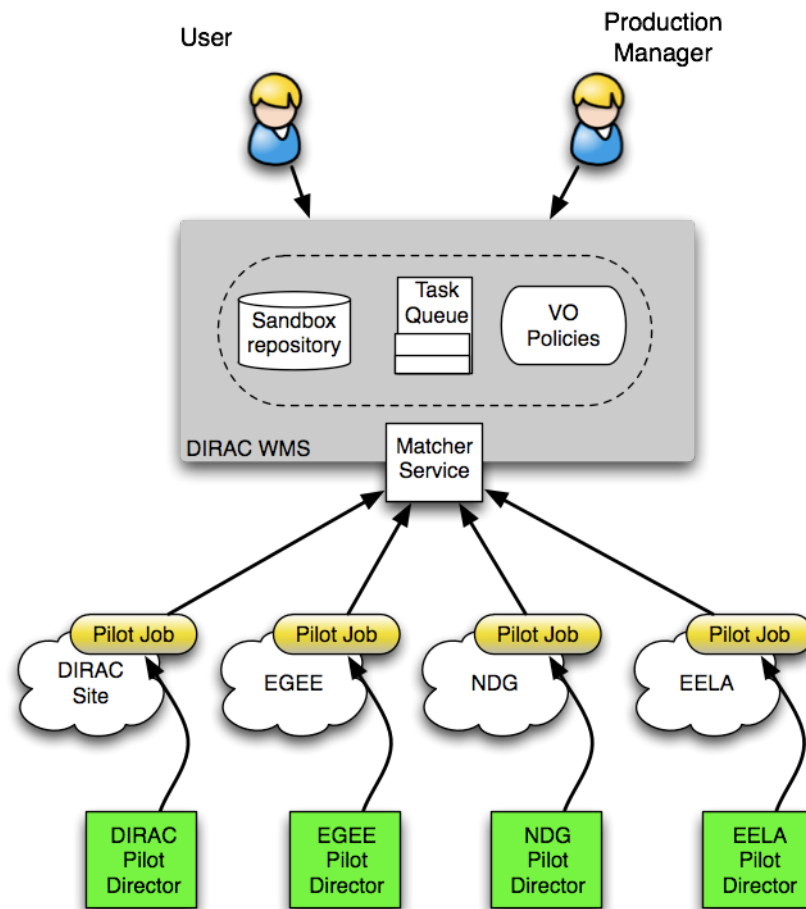
*A. Tsaregorodtsev,
CPPM-IN2P3-CNRS, Marseille,
10th virtual DIRAC User Workshop,
10 May 2021 London*



- ▶ WMS overview
- ▶ Computing resources
- ▶ Interfaces
- ▶ Conclusions

- ▶ No revolutionary changes in the last year
 - ▶ Stable system, proven architecture
 - ▶ Many optimizations are done and still ongoing
- ▶ No use of grid WMS resource brokers for most grid infrastructures
 - ▶ E.g. replaces grid WMS for the EGI infrastructure
 - ▶ Pilot factory (SiteDirectory) **SubmissionMode** option has gone !

- ▶ Pilot jobs are submitted to computing resources by specialized Pilot Directors
- ▶ Pilots retrieve user jobs from the central Task Queue and steer their execution on the worker nodes including final data uploading
- ▶ Pilot based WMS advantages:
 - ▶ increases efficiency of the user job execution
 - ▶ allows to apply efficiently community policies at the Task Queue level
 - ▶ allows to integrate heterogeneous computing resources

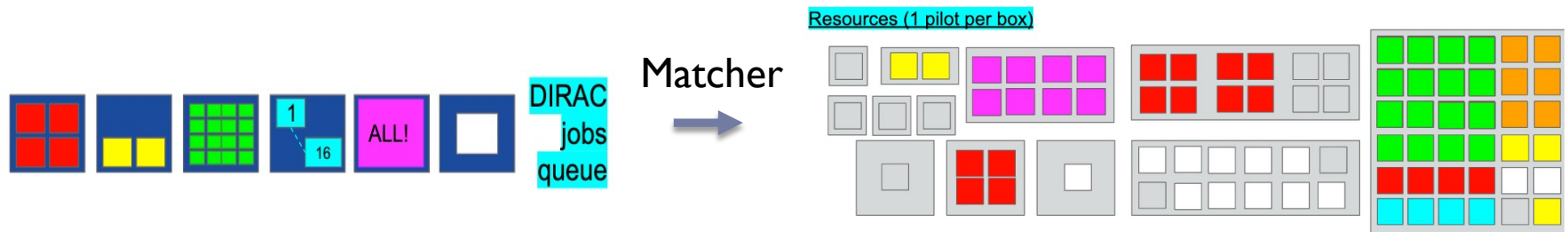


- ▶ **DIRAC Pilot Factory**
 - ▶ Submits pilot jobs for a given VO according to the status of the Task Queue
 - ▶ Gets the pilot status
 - ▶ Renews proxy delegation on CE's
 - ▶ Optionally retrieves pilot job outputs
- ▶ **Refactored for optimization and simplification**
 - ▶ Submitting Pilot3 jobs by default
 - ▶ Parallel execution for some operations
 - ▶ E.g. pilot status update
 - ▶ Simplified evaluation of numbers of pilots to submit
 - ▶ Easier to debug pilot submission problems
 - ▶ Dropped ***SubmitPool*** option usage
- ▶ **MultiProcessorSiteDirector is discontinued**
 - ▶ Use pilots starting PoolComputingElement instead

- ▶ Pilot3 is a separate DIRAC subproject to encapsulate code used in pilot jobs
 - ▶ Pilots are running in a DIRAC-free environment
 - ▶ All the new developments are going to Pilot3. Pilot2 is discontinued as of v7r2
- ▶ Many (but not all) installations use Pilot3 package for running pilot jobs
 - ▶ All the DIRAC installation administrators are invited to move to using Pilot3 (if not yet done)
- ▶ Pilot3 software is bundled and stored on a web server by running a special agent.
 - ▶ No special configuration for the web server is required for file uploads

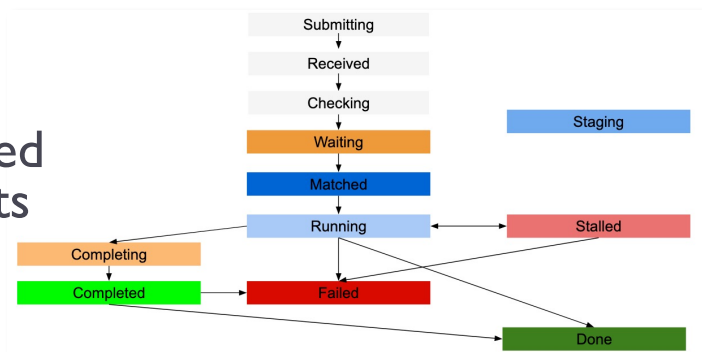
- ▶ Pilots are launching user jobs on WN's to "inner" Computing Elements
- ▶ **InProcess** CE – execution in the same process as JobAgent
- ▶ **Sudo** CE – execution in a spawned process with a different user ID
 - ▶ Used on VMs to isolate pilot environment from the user job
- ▶ **Singularity** CE
 - ▶ The user job is executed inside a Singularity container
 - ▶ Isolation of the pilot environment
 - ▶ Possibility to update the environment for user job execution, e.g. reinstall DIRAC client with different options.
- ▶ **Pool** CE

- ▶ Pilots can exploit multi-core nodes use **PoolCE** “inner” Computing Element
 - ▶ On-WN batch system
 - ▶ Flexible strategy with prioritized job requests to the Matcher, e.g.:
 - ▶ First, ask for jobs requiring WholeNode tag
 - ▶ If none, ask for jobs requesting as many cores as available
 - ▶ If none, ask for jobs with MultiProcessor requirement
 - ▶ If none, ask for single-core jobs
 - ▶ The goal is to fill the nodes with payloads fully exploiting there multi-core capacity



- ▶ Evaluation of the time left in the reserved job slot
 - ▶ Allow to present the CPU time in the job request to the Matcher service
- ▶ The implementation depends on the information obtained from the execution environment – batch systems
- ▶ Multiple problems due to failures to get batch system numbers in pilots
- ▶ Reverting in case of problems to the time left estimation only with numbers available in the pilot:
 - ▶ initial queue time length
 - ▶ CPU/Wallclock time consumed
- ▶ Increase in efficiency
 - ▶ Especially for the case of large numbers of short jobs

- ▶ WMS jobs are proceeding through a chain of states from **Submitting** to **Done/Failed**
 - ▶ In some cases illegal state transitions happened due to a desynchronization of the state reports
- ▶ Introduction in v7r3 of a strict JobState machine (in development)
 - ▶ Forbid state transitions which are not allowed in the state machine definition
 - ▶ E.g. Failed -> Running
- ▶ In v7r1 The **Completed** status is split into:
 - ▶ **Completing**: the user application is done but the job finalization is ongoing, e.g. output sandboxes/data uploading
 - ▶ **Completed**: the user job is done but there are pending requests in the RMS remaining



- ▶ Computing resources in DIRAC are represented by logical Computing Elements with various implementations
 - ▶ CREAM: obsoleted and is progressively discontinued
 - ▶ Main grid CE types:
 - ▶ HTCondorCE
 - ▶ ARC
 - ▶ SSH Computing Element
 - ▶ No-CE access
- ▶ HPC (see Alexandre's presentation)

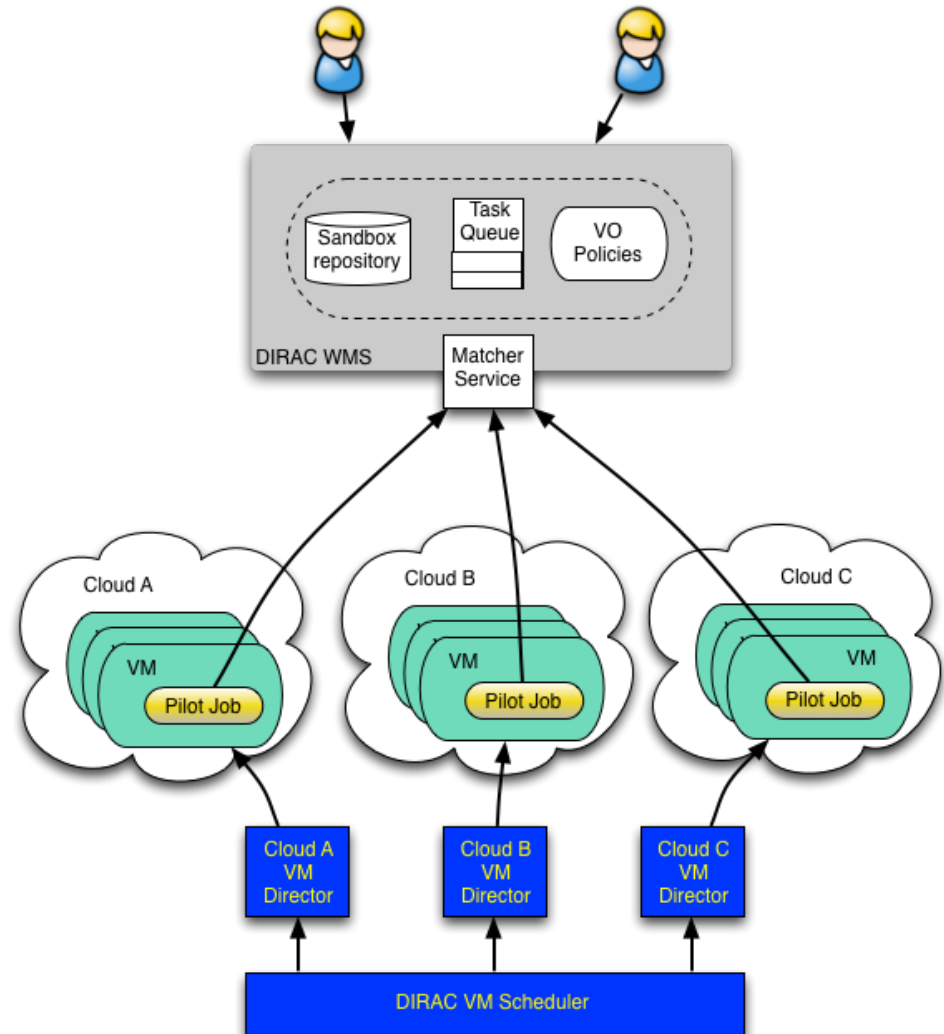
- ▶ HTCondorCE Computing Element is implemented using the *condor* command line interface:
 - ▶ Both local and remote *schedd condor* service can be used
- ▶ Stable operation in general
- ▶ Using remote *schedd* service requires job files to be kept after the submission command executed
 - ▶ Needs asynchronous clean-up of large number of files
 - ▶ Could result in SiteDirector blocking in this operation
 - ▶ Fixed in recent patches
- ▶ Allow for token-based authentication

- ▶ **ARC Computing Element in DIRAC**
 - ▶ Implementation is based on the python API encapsulating calls
 - ▶ **ldap** service for job operations
 - ▶ **gridftp** for file operations
 - ▶ Difficult to debug in case of problems, dependency on the python API provided by the ARC developers
- ▶ **ARC6 version of the software offers a RESTful interface**
 - ▶ Job and data operations
 - ▶ Proxy delegation renewal
 - ▶ Access authenticated with OIDC tokens
- ▶ **Developing access to ARC6 CE's with the REST interface is ongoing**

- ▶ Information about Computing Elements as defined by system administrators is kept in the BDII database
 - ▶ *ldap* based service
- ▶ BDII2CSAgent automatically updates the DIRAC Configuration Service for the CE parameters
- ▶ The agent was updated to use the Glue2 BDII information schema
 - ▶ Enabled by a configuration option
 - ▶ Will become the only option starting from v7r3

- ▶ VM scheduler
 - ▶ Dynamic VM spawning taking Task Queue state into account
 - ▶ Discarding VMs automatically when no more needed

- ▶ The DIRAC VM scheduler by means of dedicated VM Directors is interfaced to
 - ▶ Public:
 - ▶ OpenStack, OpenNebula
 - ▶ Amazon EC2
 - ▶ ...



- ▶ The VMDIRAC package encapsulated codes for cloud management
 - ▶ Simon Fayer, IC, has taken over the responsibility for the package
- ▶ Only minor changes in the past year:
 - ▶ Bug/compatibility patches.
 - ▶ Started improving documentation.
 - ▶ Python3 readiness.
 - ▶ Enabled standard set of DIRAC tests + (minimal) unit testing.
 - ▶ VMDIRAC now included on certification server.
 - ▶ Basic functionality is verified with new releases.



Short-term goals:

- ▶ More bug-fixes and documentation!
- ▶ Use the EGI FedCloud Marketplace service to discover appropriate images

Long-term goals:

- ▶ Make it easier to debug issues in cloud jobs.
- ▶ Make cloud-init start-up the default method and deprecate others
- ▶ Look at feasibility of merging cloud-type endpoints back into core DIRAC.
- ▶ Reduce dependence on X.509 when tokens are available.
 - ▶ May be neat to do this alongside other long-term tasks as most will need major code refactor.



- ▶ Tags were introduced to declare special capabilities of computing resources
 - ▶ E.g. Tags = GPU to declare GPU applications support
 - ▶ Tags can be requested in job's JDLs in order to limit them only to sites with special capabilities
 - ▶ Site queues can also define jobs with which tags they accept
- ▶ Tags are not statically predefined and can be used for flexible tuning
 - ▶ Example: site queues offering resources for the biomed VO but only for COVID19 related jobs define:
RequiredTag = COVID19
VO = biomed

- ▶ Users are managing jobs using various tools

- ▶ Command line (batch system like interface):

```
bash-4.2# dsub /bin/echo "Hello world"
53917277
bash-4.2# dstat
JobID      Owner      JobName    OwnerGroup JobGroup   Site           Status   MinorStatus  SubmissionTime
=====
53917277  atsareg    Unknown    wenmr_user  NoGroup   EGI.NIKHEF.nl  Running  Application  2020-10-22 19:06:24

bash-4.2# doutput 53917277
bash-4.2# ls -l 53917277
total 4
-rw-r--r-- 1 71139 2062 12 Oct 22 19:06 std.out
```

- ▶ Python API

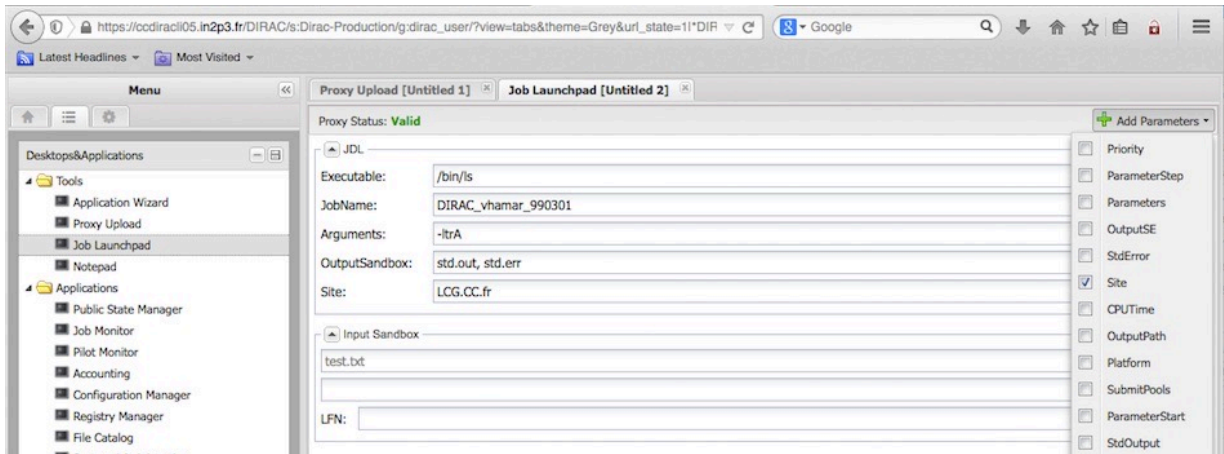
```
from DIRAC.Interfaces.API.Job import Job
from DIRAC.Interfaces.API.Dirac import Dirac

dirac = Dirac()
j = Job()

j.setCPUTime(500)
j.setExecutable('/bin/echo hello')
j.setExecutable('/bin/hostname')
j.setExecutable('/bin/echo hello again')
j.setName('API')

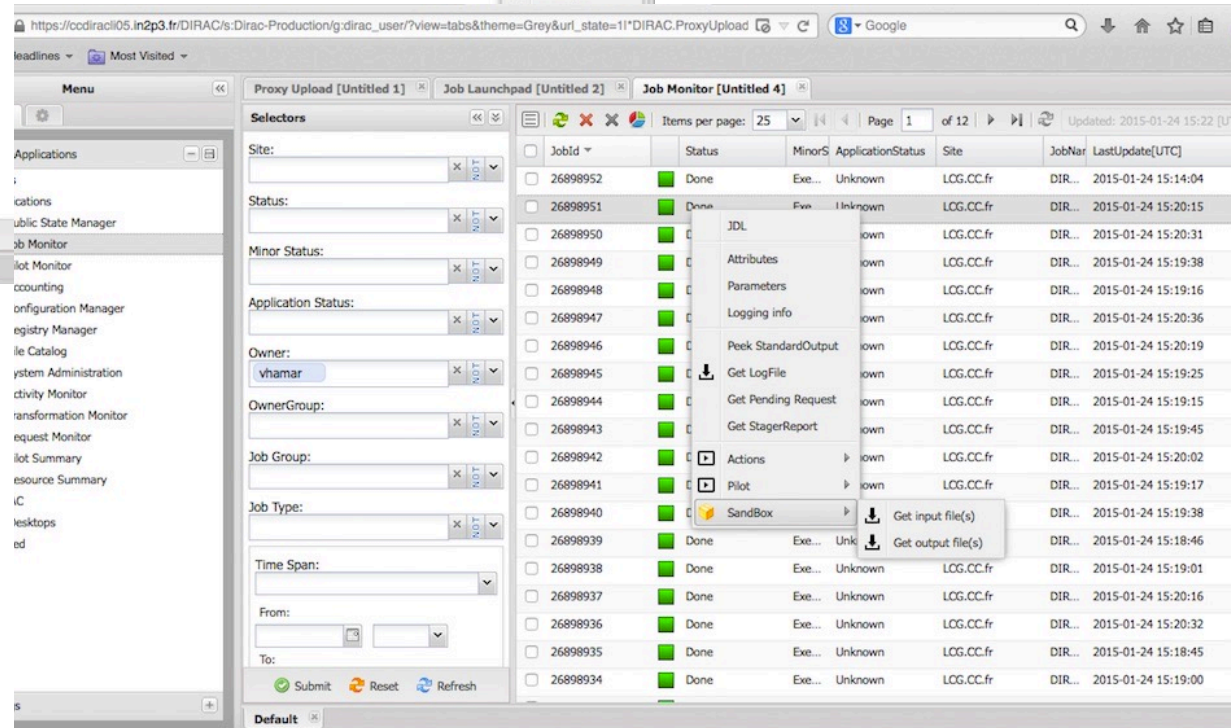
dirac.submitJob(j)
```

Job Launchpad



Proxy Status: **Valid**

Executable: /bin/ls
 JobName: DIRAC_vhamar_990301
 Arguments: -ltrA
 OutputSandbox: std.out, std.err
 Site: LCG.CC.fr



JobID	Status	MinorS	ApplicationStatus	Site	JobName	LastUpdate[UTC]
26898952	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:14:04
26898951	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:20:15
26898950	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:20:31
26898949	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:19:38
26898948	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:19:16
26898947	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:20:36
26898946	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:20:19
26898945	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:19:25
26898944	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:19:15
26898943	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:19:45
26898942	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:20:02
26898941	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:19:17
26898940	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:19:38
26898939	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:18:46
26898938	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:19:01
26898937	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:20:16
26898936	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:20:32
26898935	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:18:45
26898934	Done	Exe...	Unknown	LCG.CC.fr	DIR...	2015-01-24 15:19:00

Job Monitoring

- ▶ Getting computing resources eligible for a given VO
 - ▶ As defined in the DIRAC CS
 - ▶ After synchronization with the BDII index

```
[atsareg:-] $ dirac-resource-info --vo auger -C
```

	Site	CE	CEType	Queue	Status
1	EGI.INFN-LECCE.it	ce.le.infn.it	CREAM	cream-lsf-auger	Active
2	EGI.CESNET.cz	ce1.grid.cesnet.cz	HTCondorCE	condor	Active
3		ce2.grid.cesnet.cz	HTCondorCE	condor	Active
4	EGI.CAFPE.es	cream-cafpegrid.ugr.es	CREAM	cream-pbs-auger	Active
5				cream-slurm-auger	Active
6	EGI.RWTH-Aachen.de	grid-ce.physik.rwth-aachen.de	CREAM	cream-pbs-auger	Active
7		grid-ce-1-rwth.gridka.de	HTCondorCE	condor	Active
8	EGI.OBSPM.fr	cream-ce-grid.obspm.fr	CREAM	cream-pbs-auger	Active
9	EGI.M3PEC.fr	ce0.m3pec.u-bordeaux1.fr	CREAM	cream-pbs-auger	Active
10		ce0.bordeaux.inra.fr	CREAM	cream-pbs-auger	Active

- ▶ Get CEs/Queues matching the job requirements
 - ▶ Lists eligible Sites/CEs/Queues
 - ▶ Attempts to give a reason of no match for non-eligible Sites/CEs/Queues

```
[atsareg:~/test] $ dirac-wms-match -F test.jdl
```

	Site	CE	Queue	Status	Match	Reason
1	DIRAC.EDGI.fr	cr2.edgi-grid.eu	cream-pbs-edgidemo	Inactive	Yes	
2	DIRAC.EDGI.fr	cr2.edgi-grid.eu	cream-pbs-homeboinc	Inactive	Yes	
3	DIRAC.EDGI.fr	cr2.edgi-grid.eu	cream-pbs-edgidemo_dirac	Inactive	Yes	
4	DIRAC.LSST.fr	cclsst.in2p3.fr-sl6	verylong	Active	Yes	
5	EGI.AC.cy	ce101.grid.ucy.ac.cy	cream-pbs-biomed	Active	Yes	
6	EGI.AC.uk	hepgrid6.ph.liv.ac.uk	condor	Active	No	Job CPUTime requirement not satisfied
7	EGI.AC.uk	hepgrid10.ph.liv.ac.uk	cream-pbs-long	Active	Yes	

- ▶ Do not take into account input data yet
- ▶ Can help in understanding “why my jobs are not running this site ?”

- ▶ Pilot based WMS proven to be efficient in the HEP experiments is now available for the users of dedicated and multi-VO DIRAC services
- ▶ A large variety of heterogeneous computing resources can be federated due to the pilot job mechanism
- ▶ Ongoing effort to make new non-grid resources conveniently available (HPC, Cloud)
- ▶ Keeping uniform resource access interfaces for the users – single DIRAC computer