



EGI Workload Management Service

*A. Tsaregorodtsev,
Aix Marseille Univ, CNRS/IN2P3, CPPM
ISGC Conference, 24th March 2021*



- ▶ EGI WMS infrastructure
- ▶ Services
 - ▶ Managing jobs
 - ▶ Managing computing resources
 - ▶ Managing data
 - ▶ Managing workflows
- ▶ Example usage
- ▶ Conclusions

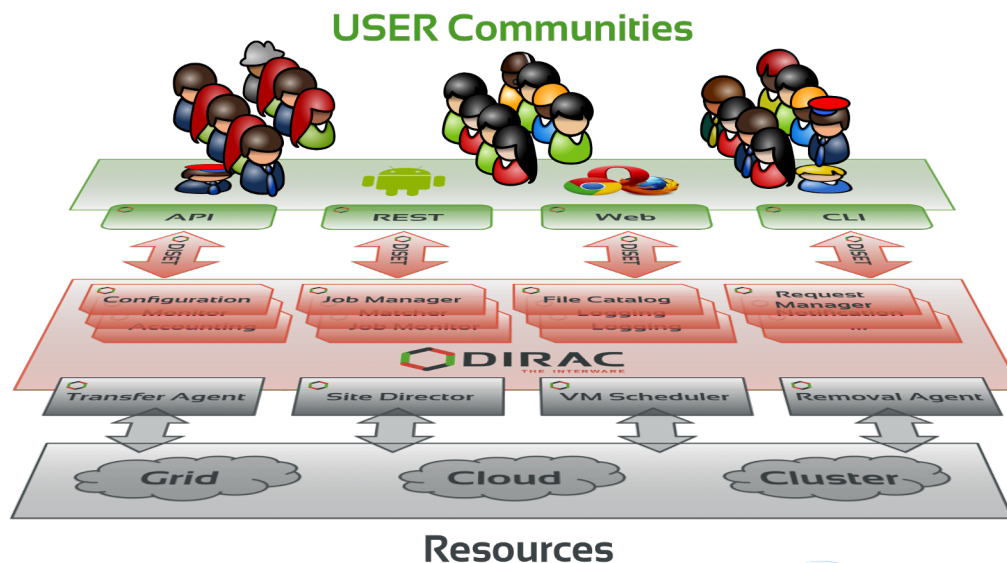
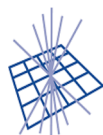


**EUROPEAN OPEN
SCIENCE CLOUD**

- ▶ EGI Workload Manager is one of the services in the EOSC Marketplace Catalog
 - ▶ <https://marketplace.eosc-portal.eu/services/egi-workload-manager>
 - ▶ Managing user jobs running on the EGI computing resources
 - ▶ Replacement for the gLite WMS service
 - ▶ Based on the DIRAC Interware distributed computing framework



- A software framework for distributed computing
- A **complete** solution to one (or more) user community
- Builds a layer between users and resources
- A *framework* shared by multiple experiments, both inside HEP, astronomy, and life sciences



- ▶ EGI Workload Manager Service was hosted until now by the CYFRONET project

- ▶ 5 medium size VM servers
- ▶ 3TB storage
- ▶ MySQL database service
- ▶ ~8 million user jobs in 2020

- ▶ Similar service was provided by the France-Grilles NGI

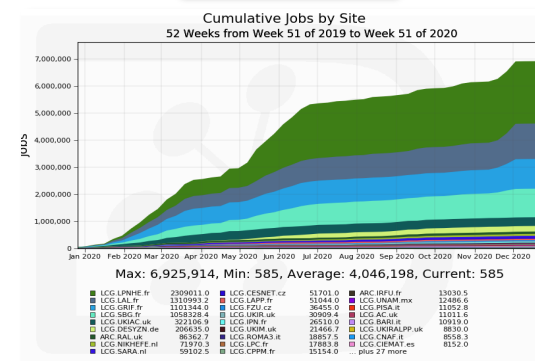
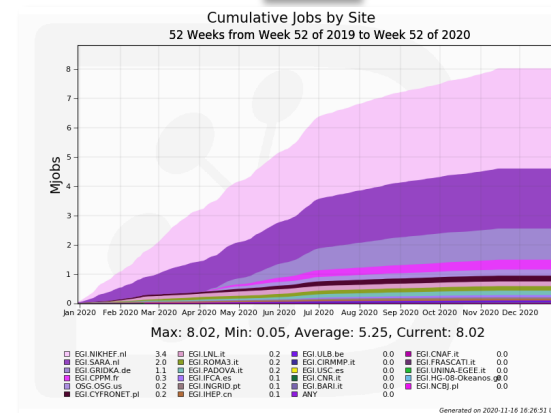
- ▶ Similar hardware infrastructure
- ▶ 7 million user jobs in 2020

- ▶ In March 2021 the two services were merged to provide a single one

- ▶ Optimized maintenance and operations
- ▶ Single administrator team
- ▶ The service is hosted in the IN2P3 Computing Center, Lyon
 - ▶ 11 virtual hosts, 8TB storage, using MySQL and Elasticsearch services provided by the computing center

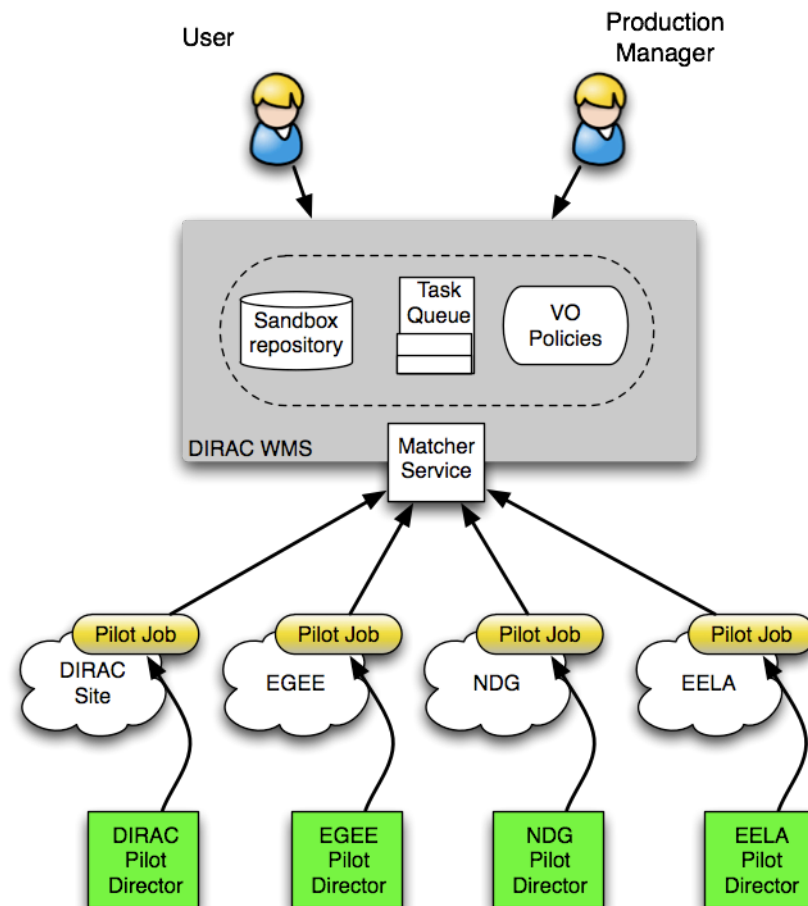
- ▶ The combined service inherited all the communities and users

- ▶ >40 registered VO's
- ▶ About 700 registered users
- ▶ Accessible via the same endpoint
 - ▶ <https://dirac.egi.eu>



Managing user jobs

- ▶ 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



► 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

- Starting from v7r2
Python3 client API
is supported

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

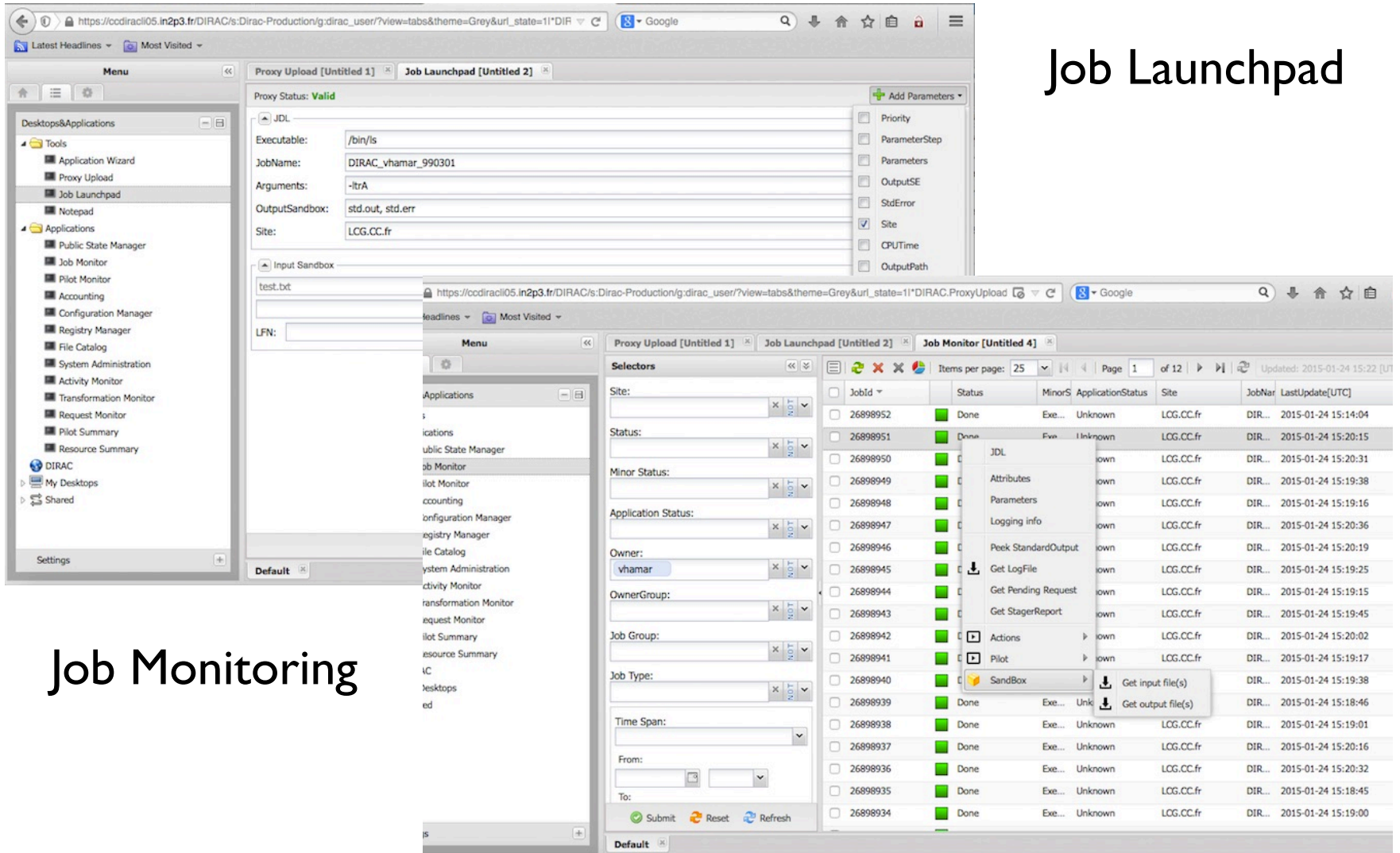
dirac = Dirac()
j = Job()

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

dirac.submitJob(j)
```

- ▶ Several methods to install the DIRAC client software on user workstations/laptops (Linux flavors)
 - ▶ **dirac-install** installer tool
 - ▶ Rather tedious (see tutorials)
 - ▶ Suitable for various flavors of Linux
 - ▶ **Docker** container (Linux, MacOS)
 - ▶ `docker run -it -v $HOME:$HOME -e HOME=$HOME diracgrid/client:egi`
 - ▶ **CVMFS** client installation (Linux)
 - ▶ `source /cvmfs/dirac.egi.eu/dirac/bashrc_egi`
 - ▶ **Conda** environment (Linux, MacOS)
 - ▶ `conda create -c conda-forge --name dirac ipython dirac-grid`
`conda activate dirac`
 - ▶ Installing as a regular Python package (starting from v7r2)
 - ▶ `pip install dirac`
 - ▶ `dirac-configure -S dips://dirac-config.egi.eu:9135/Configuration/Server`

Job Launchpad

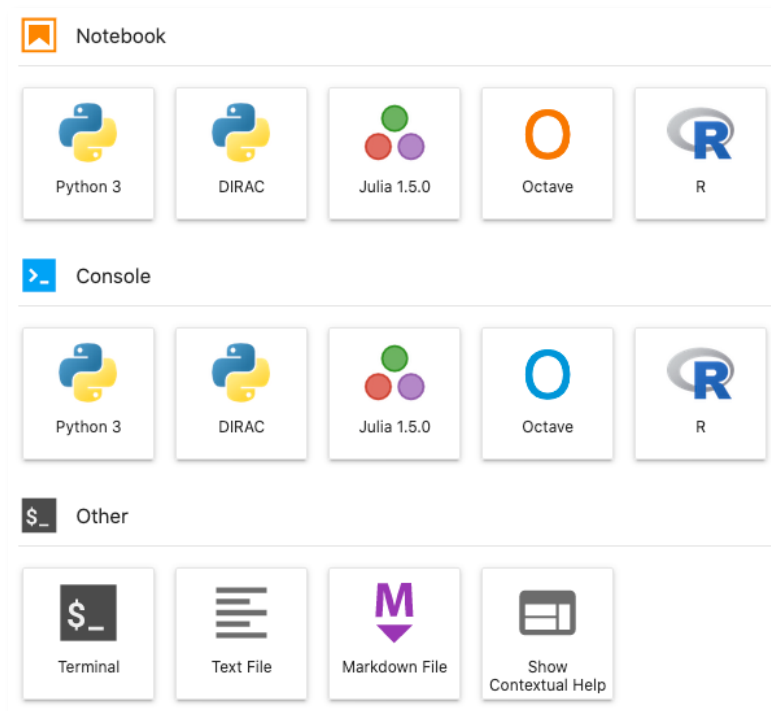


The screenshot displays the DIRAC web interface, specifically the Job Launchpad and Job Monitoring sections. The interface is divided into several panes:

- Menu:** A sidebar on the left containing various tools and applications, including 'Tools' (Application Wizard, Proxy Upload, Job Launchpad, Notepad) and 'Applications' (Public State Manager, Job Monitor, Pilot Monitor, Accounting, Configuration Manager, Registry Manager, File Catalog, System Administration, Activity Monitor, Transformation Monitor, Request Monitor, Pilot Summary, Resource Summary).
- Proxy Upload [Untitled 1]:** A pane showing the status of a proxy upload. It includes fields for Executable (/bin/ls), JobName (DIRAC_vhamar_990301), Arguments (-lra), OutputSandbox (std.out, std.err), and Site (LCG.CC.fr). The Proxy Status is indicated as 'Valid'.
- Job Launchpad [Untitled 2]:** A pane for launching jobs. It includes fields for Site, Status, Minor Status, Application Status, Owner (vhamar), OwnerGroup, Job Group, Job Type, and Time Span. There are buttons for Submit, Reset, and Refresh.
- Job Monitor [Untitled 4]:** A pane displaying a table of jobs. The table has columns for JobID, Status, MinorS, ApplicationStatus, Site, JobName, and LastUpdate[UTC]. A context menu is open over the table, showing options like JDL, Attributes, Parameters, Logging info, Peek StandardOutput, Get LogFile, Get Pending Request, Get StagerReport, Actions, Pilot, Sandbox, Get input file(s), and Get output file(s).

Job Monitoring

- ▶ REST API
 - ▶ A language neutral interface for job manipulation
- ▶ The HTTPS based DIRAC service interface is released
 - ▶ Will be shortly available to the EGI users
 - ▶ Will allow for a language neutral RPC interface
- ▶ Jupyter Notebook interface
 - ▶ In a prototype phase
 - ▶ DIRAC API enabled iPython shell
 - ▶ Terminal with DIRAC command line interface
 - ▶ Managing user credentials is being sorted out
 - ▶ Functional for users having grid certificates and registered in the Check-In SSO service



- ▶ There are multiple examples of SSO solutions
- ▶ The EGI Check-in service enables access to EGI services and resources using federated authentication mechanisms
 - ▶ A hub between federated Identity Providers (IdPs) and Service Providers (SPs) that are part of EGI

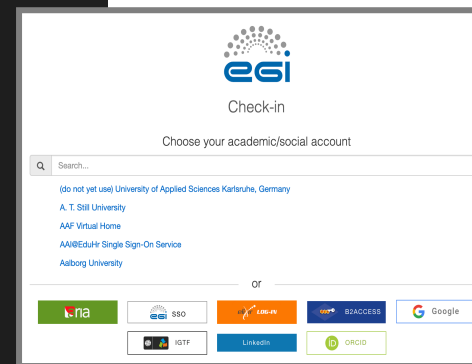


The image displays two screenshots of EGI authentication interfaces. The left screenshot shows the 'Janus - Gestion des identités' page, which includes a CERN Single Sign-On section with a 'Sign in with your CERN account' button and a 'Use credentials' section for entering a username or email address. The right screenshot shows the 'Login to EGI AAI Service Provider Proxy' page, featuring a 'Username' field (containing 'atsareg'), a 'Password' field (masked with dots), and a 'Login' button. Below the password field are checkboxes for 'Don't Remember Login' and 'Clear prior granting of permission for release of your information to this service.' The bottom of the right screenshot shows a 'Sign in with a public service account' section with a 'Facebook, Google, Live, etc.' button and a 'Sign in with your organization or institution account' section with a dropdown menu and a 'Go' button.

Web Portal functional prototype

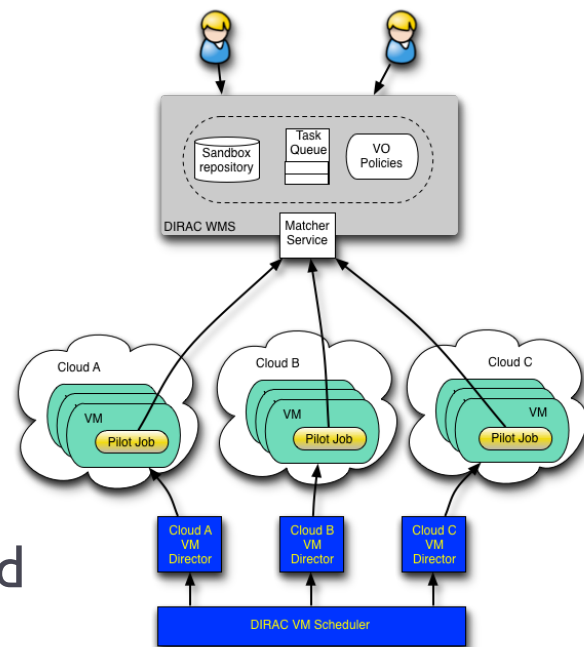


Command Line functional prototype

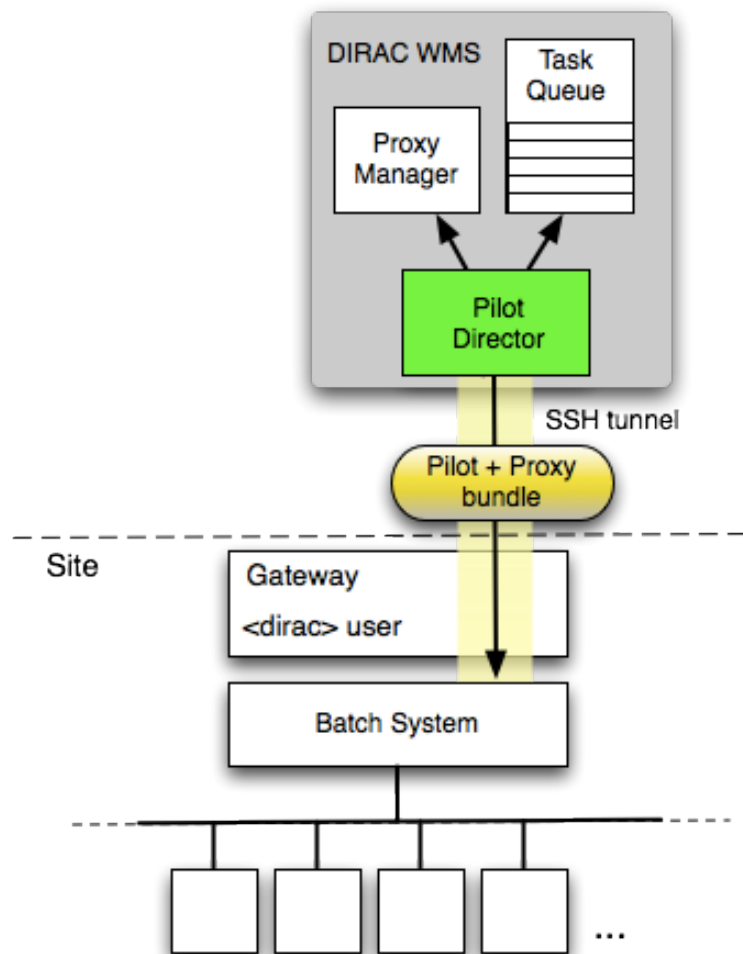


Managing user computing resources

- ▶ DIRAC was initially developed with the focus on accessing conventional Grid computing resources
 - ▶ WLCG grid resources for the LHCb Collaboration
- ▶ Grid infrastructures
 - ▶ E.g. EGI, WLCG, OSG
 - ▶ CREAM, HTCondorCE, ARC
- ▶ Cloud infrastructures
 - ▶ EGI Federated Cloud, France-Grilles cloud

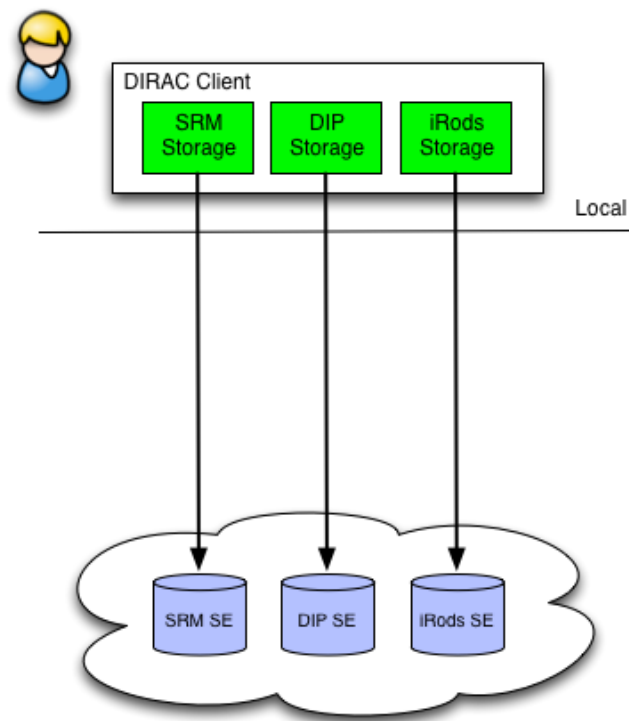


- ▶ Users can connect their own computing resources
 - ▶ Not making part of any grid infrastructure
- ▶ The user site can be:
 - ▶ a single computer or several computers without any batch system
 - ▶ a computing cluster with a batch system
 - ▶ LSF, BQS, SGE, PBS/Torque, Condor
 - Commodity computer farms
 - ▶ SLURM
 - HPC centers



Managing user data

- ▶ Storage element abstraction with a client implementation for each access protocol
 - ▶ DIPS – DIRAC data transfer protocol
 - ▶ FTP, HTTP, WebDAV
 - ▶ SRM, XROOTD, RFIO, DCAP, etc
 - ▶ HEP centers specific protocols
 - ▶ Using gfal2 library developed at CERN
 - ▶ S3, Swift, CDMI: cloud specific data access protocols
- ▶ Each SE is seen by users as a logical entity
 - ▶ With some specific operational properties
 - ▶ Archive, limited access, etc
 - ▶ SE's can be configured with multiple protocols
- ▶ New data access technologies require creating new specific plug-ins



- ▶ File Catalog is a service to keep track of all the physical file replicas in all the SE's
 - ▶ Stores also file properties:
 - ▶ Size, creation/modification time stamps, ownership, checksums
 - ▶ User ACLs
- ▶ DIRAC relies on a *central* File Catalog
 - ▶ Defines a single logical name space for all the managed data
 - ▶ Organizes files hierarchically like in common file systems
- ▶ VO's can ask for dedicated File Catalog services
 - ▶ No interference with other users
 - ▶ Customized behavior
 - ▶ Example: Eiscat 3D File Catalog
 - 117M files
 - Custom access policies

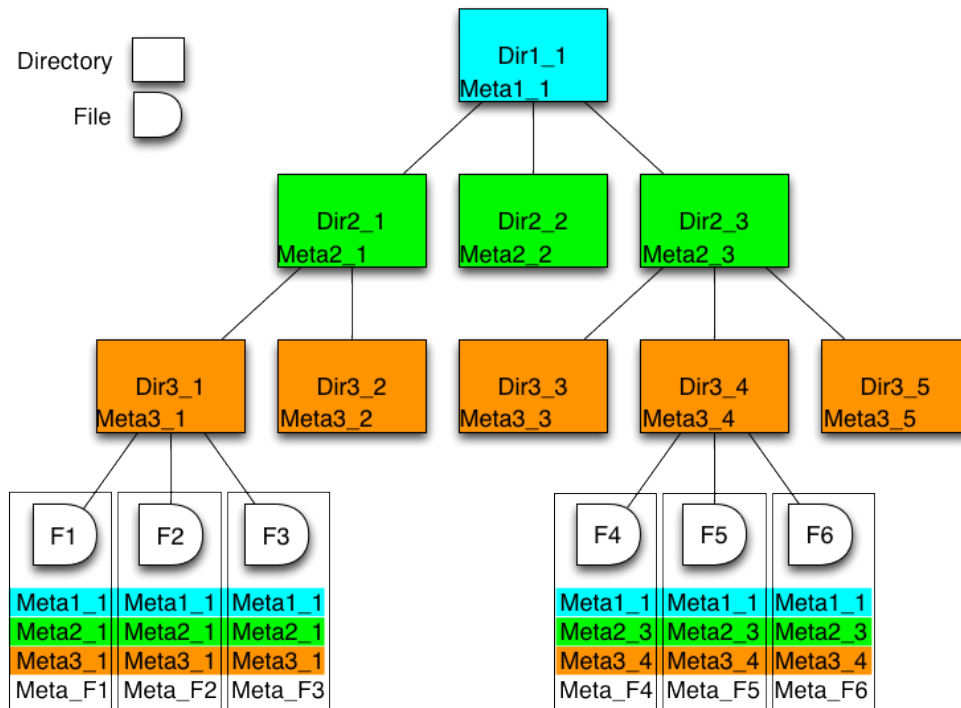
► COMDIRAC

- Representing the logical DIRAC file namespace as a parallel shell
 - **dls, dcd, dpwd, dfind, ddu**, etc commands
- Commands for file upload/download/replication
 - **dput, dget, drepl**

```
bash-4.2# dput test.jdl /enmr.eu/user/a/atsareg/test/test.jdl
bash-4.2# dls -L /enmr.eu/user/a/atsareg/test/test.jdl
-rwxrwxr-x 1 atsareg wenmr_user 256 2020-10-22 22:33:12 test.jdl
    CYFRONET-USER    dips://dirac-dms.egi.eu:9148/DataManagement/StorageElement/enmr.eu/user/a/atsareg/test/test.jdl
bash-4.2# rm test.jdl
bash-4.2# dget /enmr.eu/user/a/atsareg/test/test.jdl
bash-4.2# ls test.jdl
test.jdl
bash-4.2# drm /enmr.eu/user/a/atsareg/test/test.jdl

1 object(s) removed in total
```


- ▶ DFC is Replica and Metadata Catalog
 - ▶ User defined metadata
 - ▶ The same hierarchy for metadata as for the logical name space
 - ▶ Metadata associated with files and directories
 - ▶ Allow for efficient searches
 - ▶ Efficient Storage Usage reports
 - ▶ Suitable for user quotas



- ▶ Example query:
 - ▶ `find /lhcb/mcdata LastAccess < 01-01-2012`
`GaussVersion=v1,v2 SE=IN2P3,CERN Name=*.raw`

Path to start from: /

antenna: 32p

country: SW

Directory Metadata

- account
- antenna
- country
- end
- experiment_name
- start
- type

Updated: 2020-10-22 21:14 [UTC] | Items per page: 100 | Page 1 of 256

Directory: /eiscat.se/archive/2015/lt2e1_EASI_0.1_SW@32p/20150303_09 (100 Items)

File	Date	Size	Metadata
05302946.mat.bz2	2016-06-26 05:21:59	16663243	
05303410.mat.bz2	2016-06-26 05:21:59	16336868	
05303542.mat.bz2	2016-06-26 05:21:59	16326493	
05305260.mat.bz2	2016-06-26 05:21:59	16364777	
05305644.mat.bz2	2016-06-26 05:21:59	16353232	
05304370.mat.bz2	2016-06-26 05:21:59	16332666	
05304490.mat.bz2	2016-06-26 05:21:59	16325806	
05303794.mat.bz2	2016-06-26 05:21:59	16324414	
05306316.mat.bz2	2016-06-26 05:21:59	16366711	
05305816.mat.bz2	2016-06-26 05:21:59	16356926	
05302886.mat.bz2	2016-06-26 05:21:59	16746361	
05303810.mat.bz2	2016-06-26 05:21:59	16322298	
05304028.mat.bz2	2016-06-26 05:21:59	16327548	
05304022.mat.bz2	2016-06-26 05:21:59	16325224	
05302880.mat.bz2	2016-06-26 05:21:59	16763981	
05305860.mat.bz2	2016-06-26 05:21:59	16357369	
05305700.mat.bz2	2016-06-26 05:21:59	16351208	

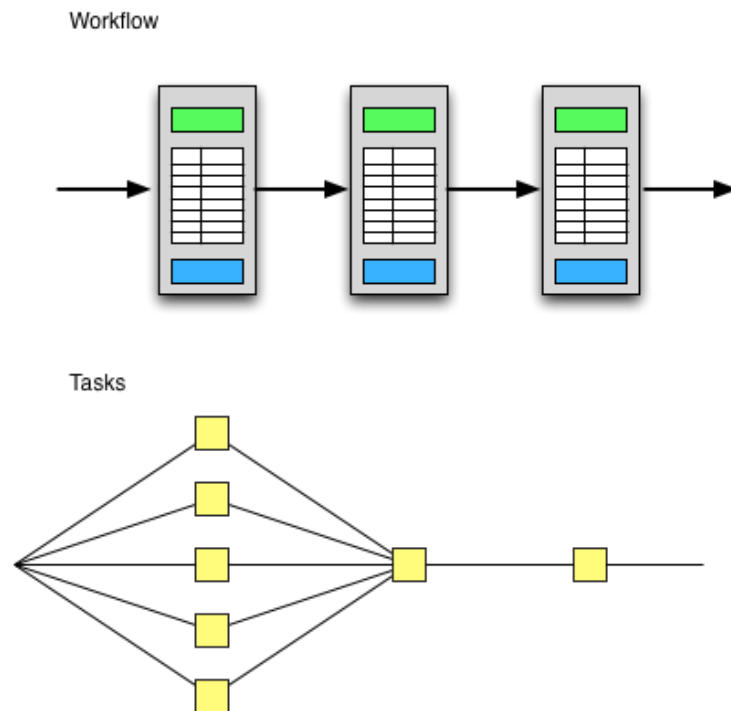
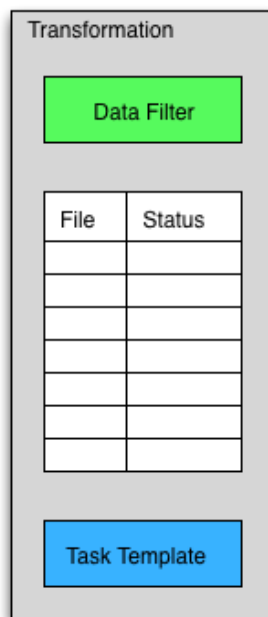
Sub... Refre... Cle...

Managing workflows

- ▶ DIRAC can deal with large numbers of jobs
 - ▶ > 100K simultaneously running jobs
 - ▶ > 10M jobs in the WMS
- ▶ DIRAC can deal with large volumes of scientific data
 - ▶ 10's of Petabytes
 - ▶ 10^7 - 10^8 of files and directories
- ▶ There is a need for massive (bulk) operations
 - ▶ Examples:
 - ▶ Submit and monitor 50K jobs
 - ▶ Replicate 10^5 files from SE A to SE B
 - ▶ Remove 10^5 files and all their replicas in all the storages
- ▶ Massive operations supported
 - ▶ Asynchronous execution
 - ▶ Request Management System, RMS
 - ▶ Automatic failure recovery
 - ▶ Automated data driven workflows

Transformation System for data driven workflows

- ▶ Data driven workflows as chains of data transformations
 - ▶ Transformation: input data filter + recipe to create tasks
 - ▶ Tasks are created as soon as data with required properties is registered into the system
 - ▶ Tasks:
 - ▶ Jobs submission
 - ▶ Data replication, removal
 - ▶ etc
- ▶ Transformations can be used for automatic data driven bulk data operations
 - ▶ Scheduling RMS tasks
 - ▶ Often as part of a more general workflow

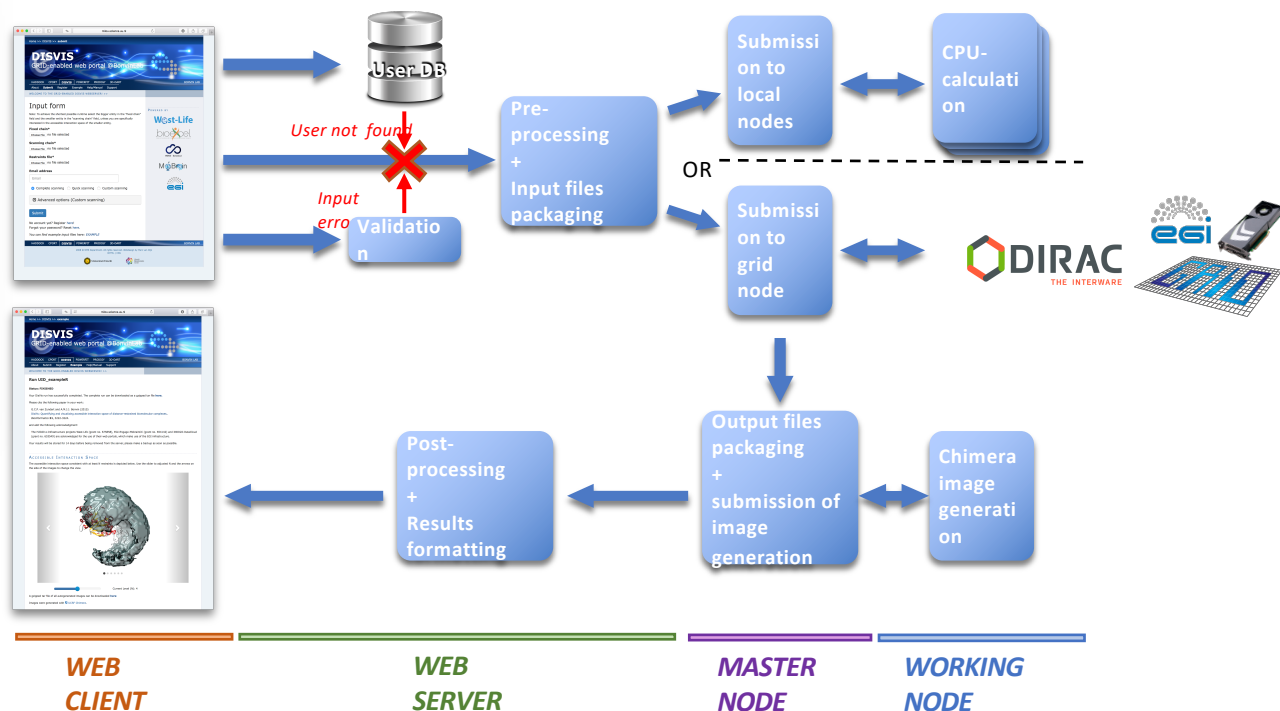


Example of usage

The WeNMR portal architecture

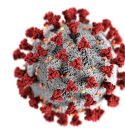


- Main use cases:**
- Disease-causing mutations
 - Engineer better molecules for material, health or food applications
 - Obtain a starting point for drug design to combat disease

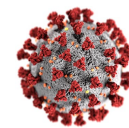


Alexandre M.J.J. Bonvin, Utrecht University

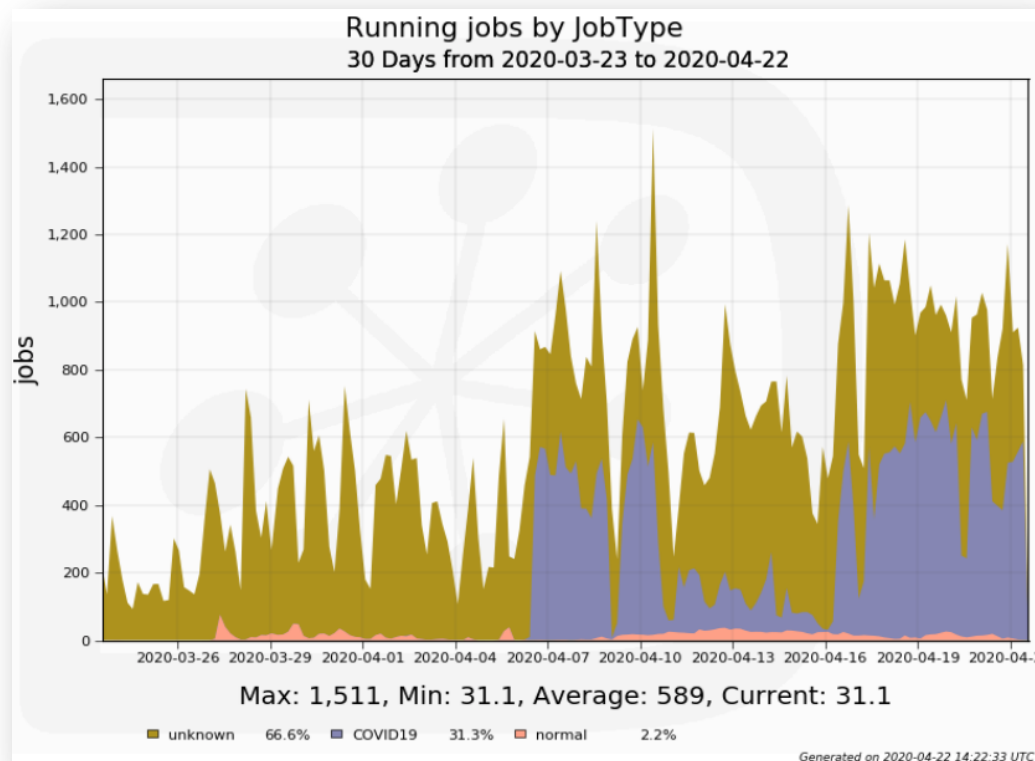
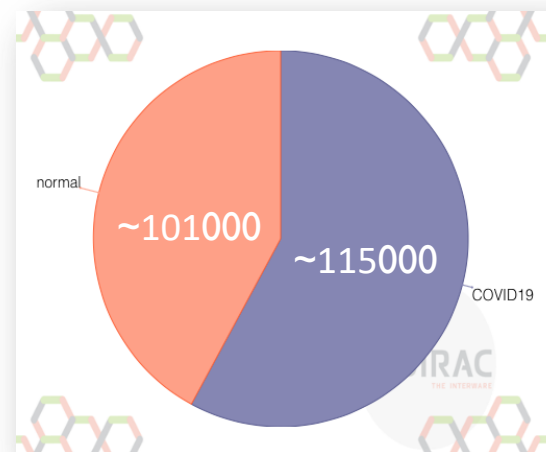
Job submission to the grid and cloud resources is performed using the EGI Workload Manager



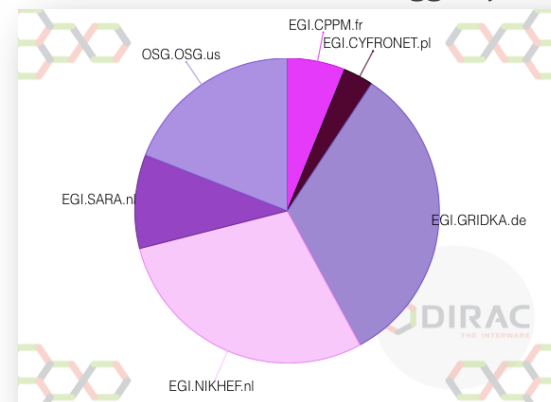
- **Multiple Centers provided resources to studies related to the COVID-19 research**
- **Administrators of OSG (National Grid Infrastructure in USA) created a special HTCondorCE gateway giving access to several sites**
 - The new resource was quickly connected to the EGI Workload Manager Service
- **Several WLCG centers opened access to their resources for the WeNMR payloads**
 - **Centre de Physique des Particules de Marseille (CPPM)**
 - **Karlsruhe Institute of Technology**
 - **Spanish LHCb Tier2 (USC-LCG2) site**
- **CYFRONET provided a 5TB disk storage of the S3 type for the WeNMR data**
 - An S3 storage plugin was quickly developed by the DIRAC team to incorporate it into the common infrastructure
- **The jobs were tagged as COVID-19 to allow giving these tasks high priority in the EGI Workload Manager Service**



#jobs last 7 days (Apr. 23-30)



Distribution of COVID-tagged jobs



- ▶ Large scientific communities have to employ various geographically distributed computing and storage resources
- ▶ DIRAC provides a framework for building distributed computing systems aggregating multiple types of resources
- ▶ EGI Workload Manager service provides an integrated solution with a reach set of ready to use services for managing computing resources, application workloads and data



This work is co-funded by EGI and the [EOSC-hub project \(Horizon 2020\)](#) under Grant number 777536



- ▶ DIRAC Project site: <http://diracgrid.org>
- ▶ Guides: <https://dirac.readthedocs.io/en/latest/>
- ▶ Tutorials:
<https://github.com/DIRACGrid/DIRAC/wiki/DIRAC-Tutorials>