DIRAC Project Status

A. Tsaregorodtsev,
CPPM-IN2P3-CNRS

NEC 2015, Budva, 1 October 2015

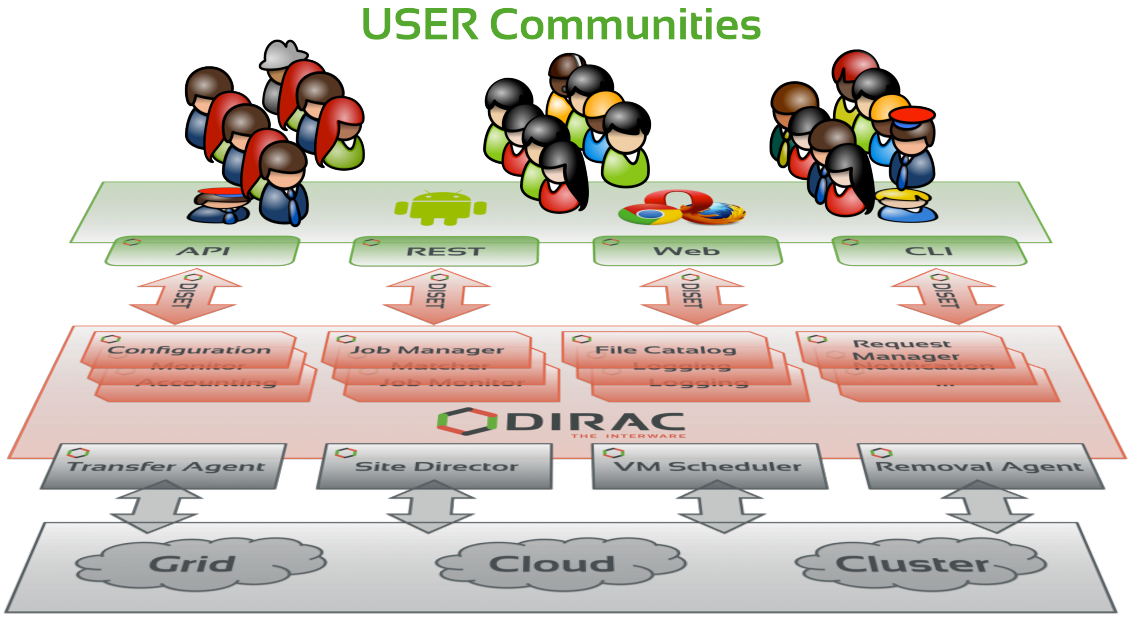
DIRAC
THE INTERWARE

Plan

- DIRAC Project reminder
- Pilot framework updated
- DIRAC Data Management Model
- Web Portal Framework
- DIRAC as a framework for integration
- Conclusions

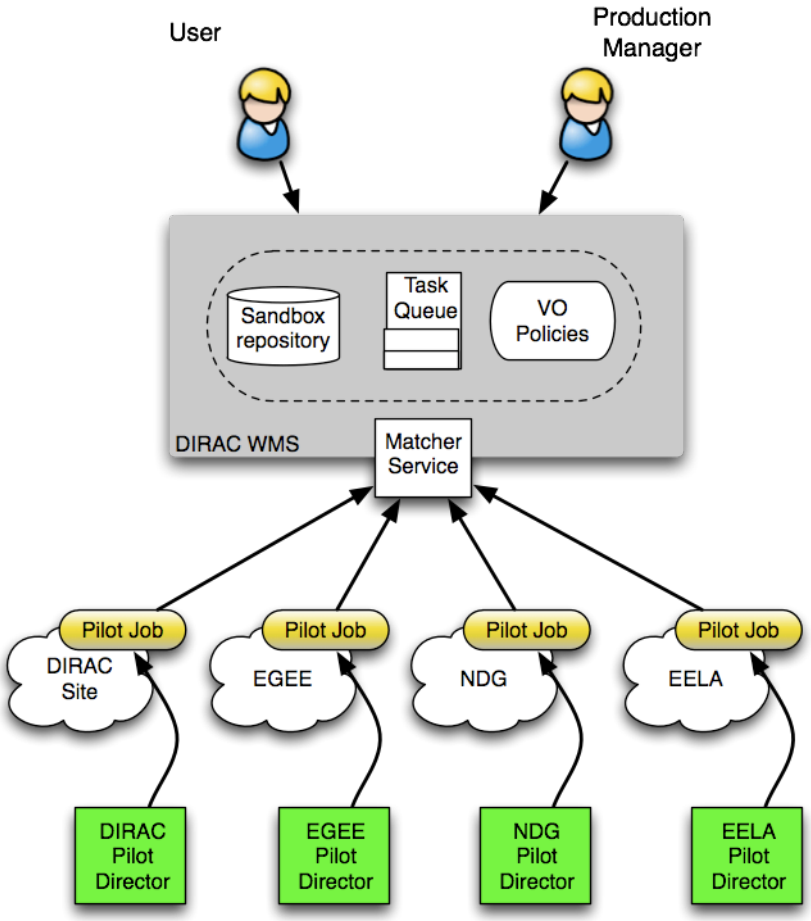
Interware

- DIRAC provides all the necessary components to build ad-hoc grid infrastructures **interconnecting** computing resources of different types, allowing **interoperability** and simplifying **interfaces**. This allows to speak about the DIRAC ***interware***.



Interware: WMS

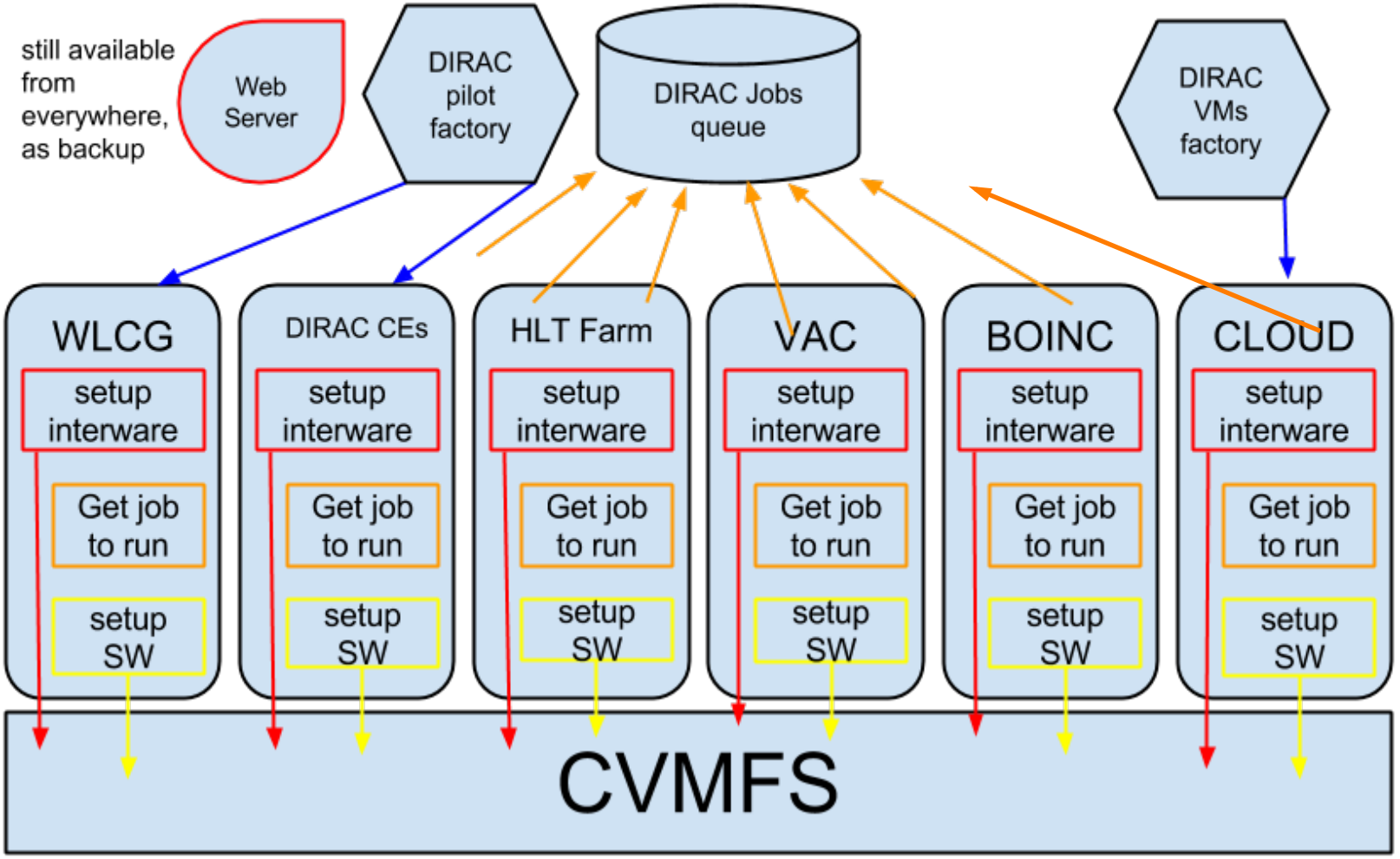
- Pilot based Workload Management
  - High user job efficiency
  - Suitable for usage with heterogeneous resources
  - Allowing application of community policies
- Pilot 2.0 framework
  - Modular for easy customizations for different environments, communities, etc



Pilots 2.0 framework

- Modularity: pilots are designed as a configurable sequence of commands
- Extensions: each community can provide custom commands for specific operations
  - Environment checks
  - Software installation
  - Monitoring reports
- Monitoring sensors
  - Checking the status of computing and storage resources in the same conditions as user payloads
  - Pilot sensors are being incorporated into the Resource Status System of DIRAC

LHCb example: "pilots to fly in all the sky's"

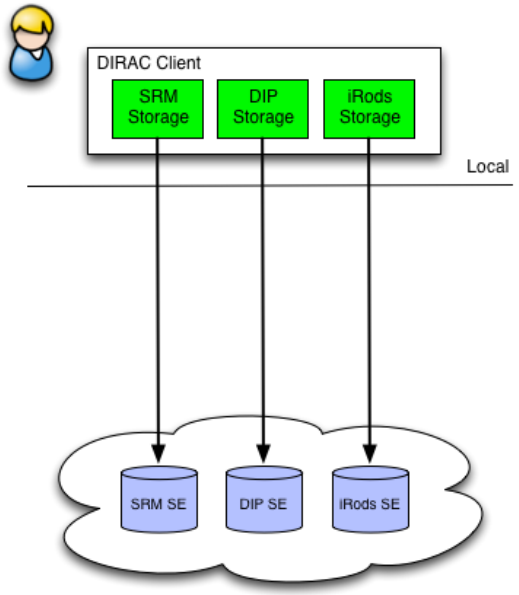


Computing resources available via pilots

- Grids
  - gLite/EMI: EGI (CREAMComputingElement)
  - VDT: OSG (GlobusComputingElement)
  - ARC: NDGF (ARCComputingElement)
- Standalone clusters
  - Using SSH/GSISSH tunnel
  - LSF, BQS, SGE, PBS/Torque, Condor(G), …
    - E.g. Yandex computing farm
  - HTCondor ComputingElement is in the test
- HPC centers:
  - OAR, SLURM
- Clouds
  - VM scheduler for EC2, OCCI, Nova, libcloud
    - Amazon, OpenNebula, OpenStack, CloudStack, Stratuslab
  - VAC, Vcycle
- BOINC
  - IDGF
  - Standalone, e.g. LHCb@HOME

Data Management Model

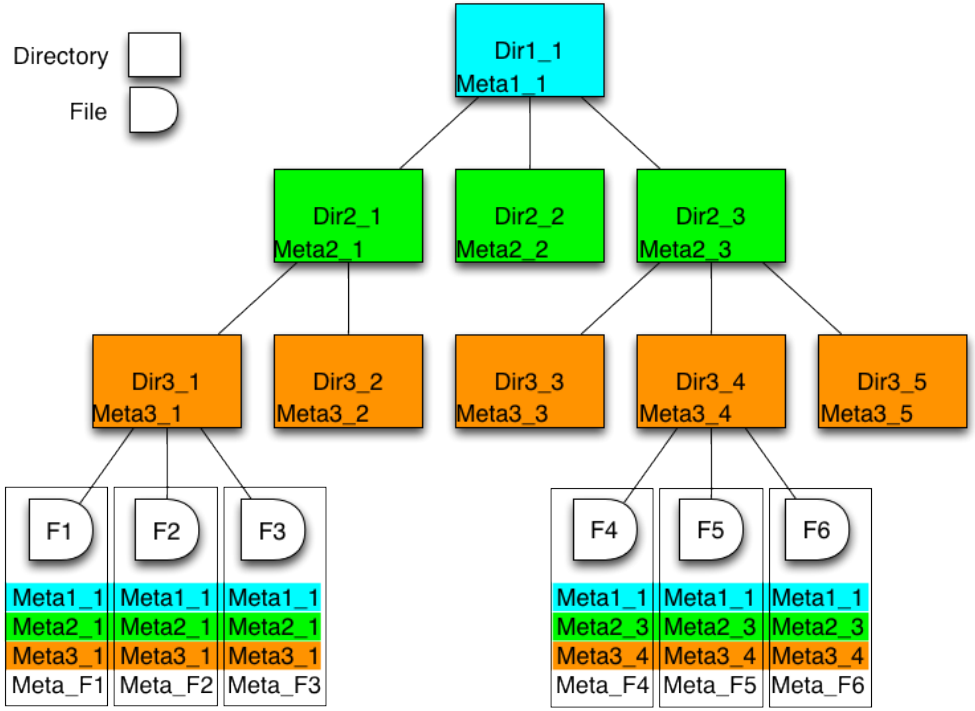
- Storage element abstraction with implementation for each access protocol (SRM, XROOTD, gfal2 based, etc)
- Each SE is seen by the clients as a logical entity
  - With some specific operational properties
  - New SE technologies, e.g. Federated Cloud, EOS are available after the proper configuration
  - SE's can be configured with multiple protocols



- Central File Catalog (DFC, LFC, ...) is maintaining a single global logical name space
  - Replica URLs are stored or constructed on the fly
- From the user perspective all the data are seen as stored in a single global file system
  - FSDIRAC – mount the "DIRAC File System" locally in user space using FUSE

File Catalog

- 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`

Support for datasets

- Datasets defined as a resulting list of files from a given metaquery
  - Particular case: all the files under a given directory
- Dataset objects are stored in the same directory hierarchy as files
  - ACLs, ownership, show up in the output of ***ls*** command as for files
- Datasets can be frozen in order not to change from one query to another
  - Can be refreshed by an explicit command, a quick check if changed since the last update
- Datasets can be annotated
- Operations on datasets
  - Replica lookup for all the files in a dataset
  - Total size, number of files report
  - Replication, removal, etc

DFC usage

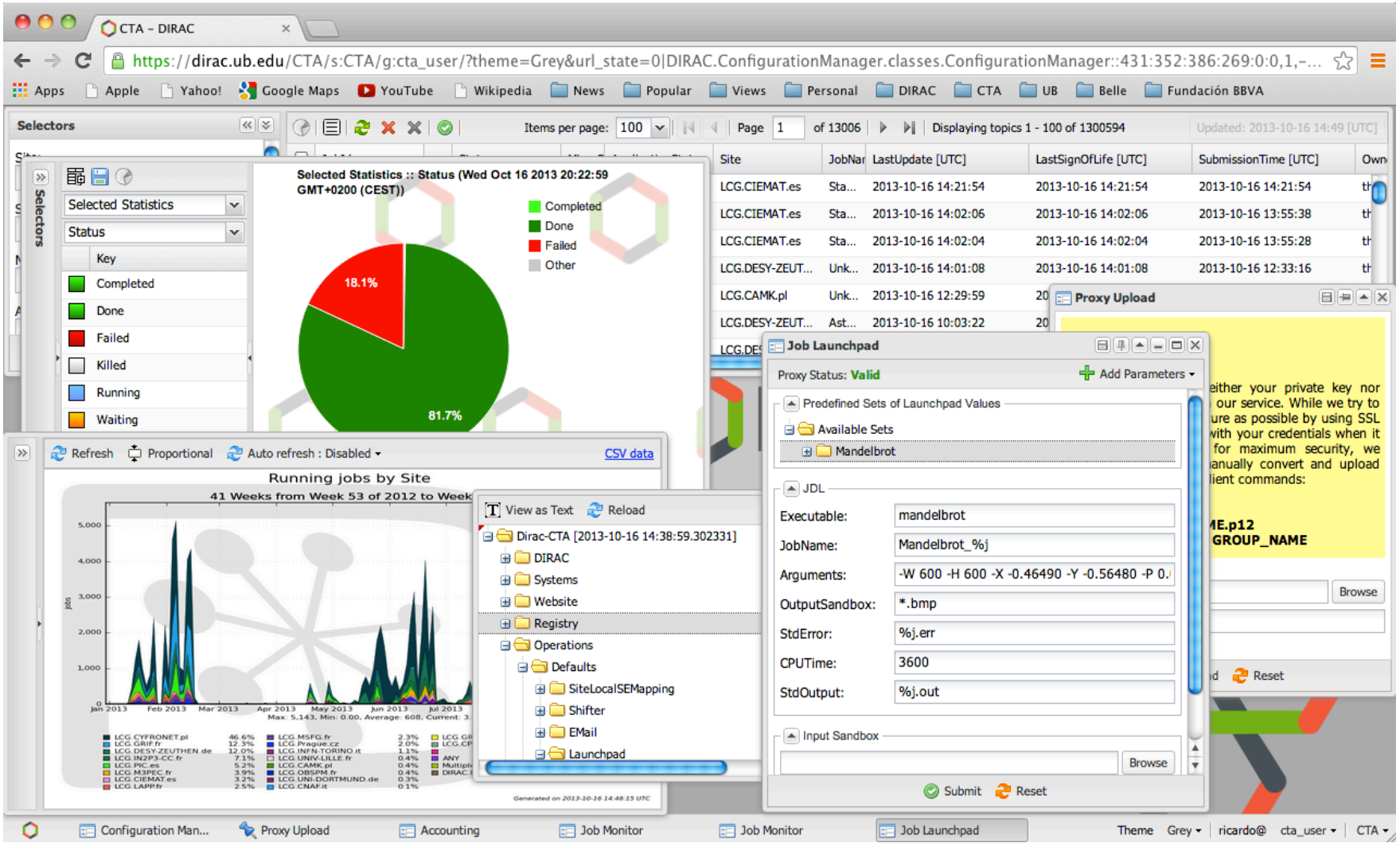
- LHCb accomplished migration from LFC to DFC
  - Needed to develop a specific ACL plugin where several DIRAC groups have same ACLs for a given data
  - Not using the Metadata features of the DFC except for the Storage Usage reports
  - Using Transformation System of DIRAC for bulk data driven operations (e.g. replication, processing tasks submission, etc)
- ILC, BES III, CTA use intensively DFC as both Replica and Metadata Catalog
  - BES III performed a detailed performance comparison with the AMGA metadata service
- Pierre Auger Observatory
  - Working on complex metadata queries and dataset algebra (dataset relations, intersections, unions, etc)

Interfaces

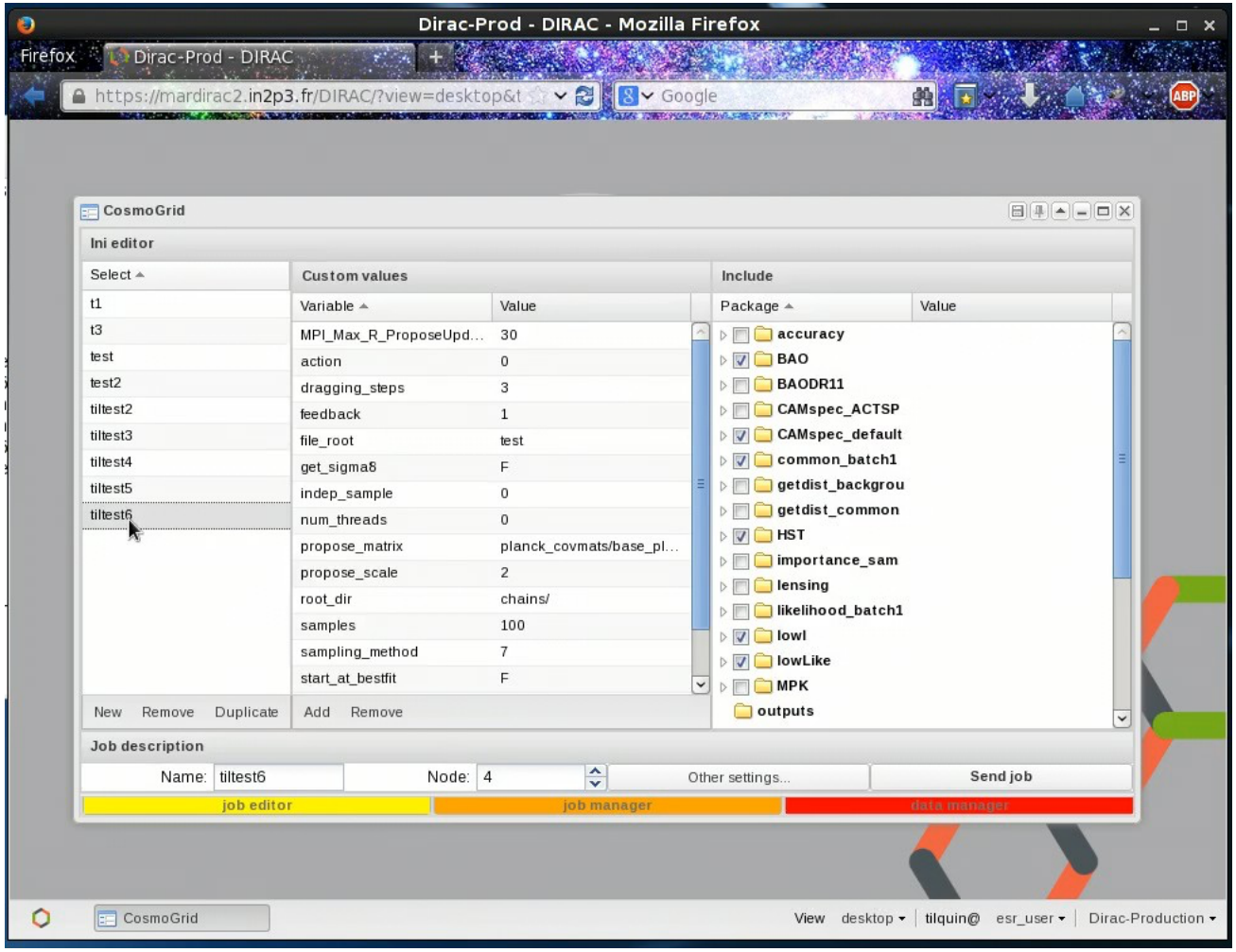
- Web Portal development framework
  - Tornado CMS, ExtJS GUI
  - Secure with X509 certificates
- Desktop GUI paradigm
  - Natural for non-expert users
- Support of most of the user tasks (jobs, data, monitoring, management)
  - A dedicated “Application” for each task
- Applications for specific communities are being developed
  - LHCb Production Management Console
  - Example KosmoUI portal for astrophysics MC

    https://www.youtube.com/watch?v=-kSAoYsSX2o

Web Portal: example interfaces



KosmoUI Web Application



Task input, job submission and monitoring, results visualization (*Xiaobo Li*)

DIRAC
THE INTERWARE

Distributed Computer

- DIRAC is aiming at providing an abstraction of single computer for massive computational and data operations from the user perspective
  - Logical Computing and Storage elements
  - Global logical name space
  - Desktop-like GUI

Development framework

- The DIRAC software architecture is designed with extensibility in mind
  - Mechanisms to provide new components (service, agents, command lines, etc) as extensions to the core DIRAC software
    - Standard distribution, deployment, configuration, discovery
- Example extensions
  - LHCb:
    - File Catalog plugins for ACLs, Directory Tree engine
    - ElasticSearch database with Kibana based visualization of the service monitoring information
  - BES III
    - Resource Status Monitoring sensors, File Catalog Dataset engine
  - CTA
    - Complex data-driven Transformations (plugins to the DIRAC workflow engine)
- Some developments done as extensions are imported into the DIRAC Core package

Quang's case

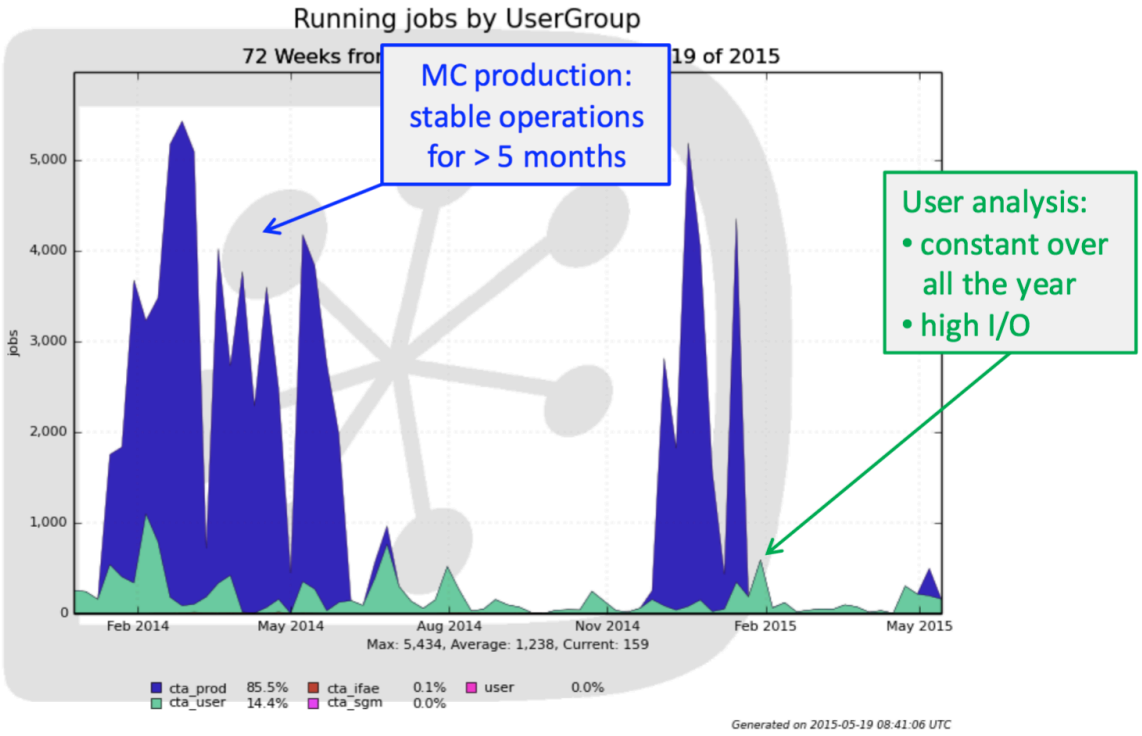
- Quang Bui – vietnamese PhD student in the Clermont-Ferrand University
  - Thesis topic: scheduling algorithms minimizing the time span of bulk (multi-job) computations
  - The mathematical modeling as well as simulation was relatively easy
  - Testing the algorithms in a real working environment was nearly impossible
    - Need for dedicated resources, WMS service, etc

- Finally the scheduling policies were implemented as a plugin to the DIRAC job matching service (dispatcher) applicable only for the jobs of a dedicated group of users with Quang as a single member
  - Allowed testing in a real environment without affecting other user activities

Users

- Dedicated installations for the large user communities
  - LHCb, Belle II, BES III, ILC, CTA

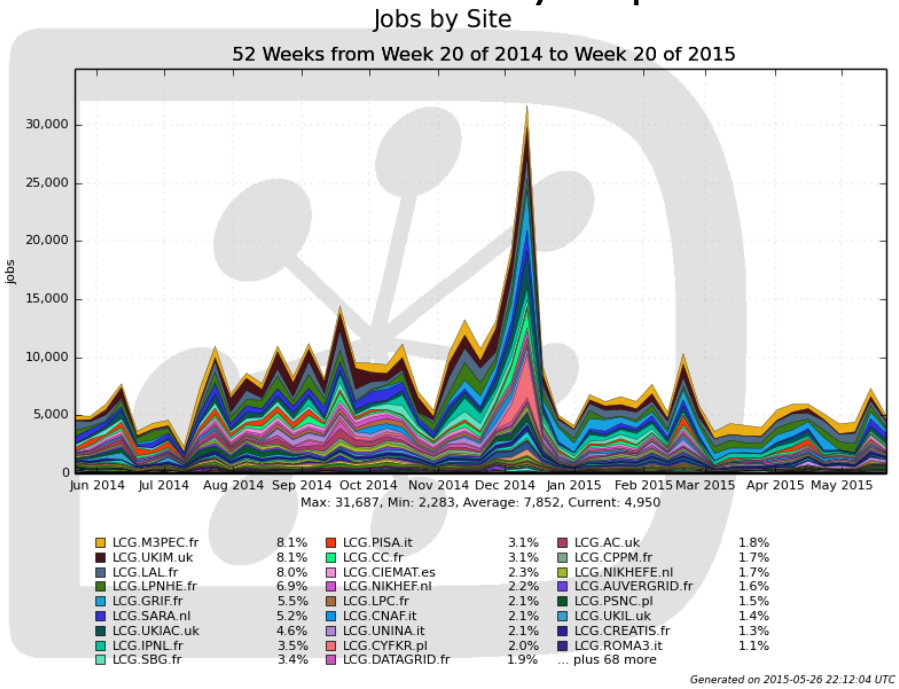
CTA MC Production run



Users

- Multi-Community services
  - DIRAC4EGI
    - 10 Virtual Organizations
    - > 1 million jobs processed in the last 2 months
  - National services: supported by the national grid infrastructure projects: France, Spain, UK, Romania, …

FG-DIRAC activity snapshot



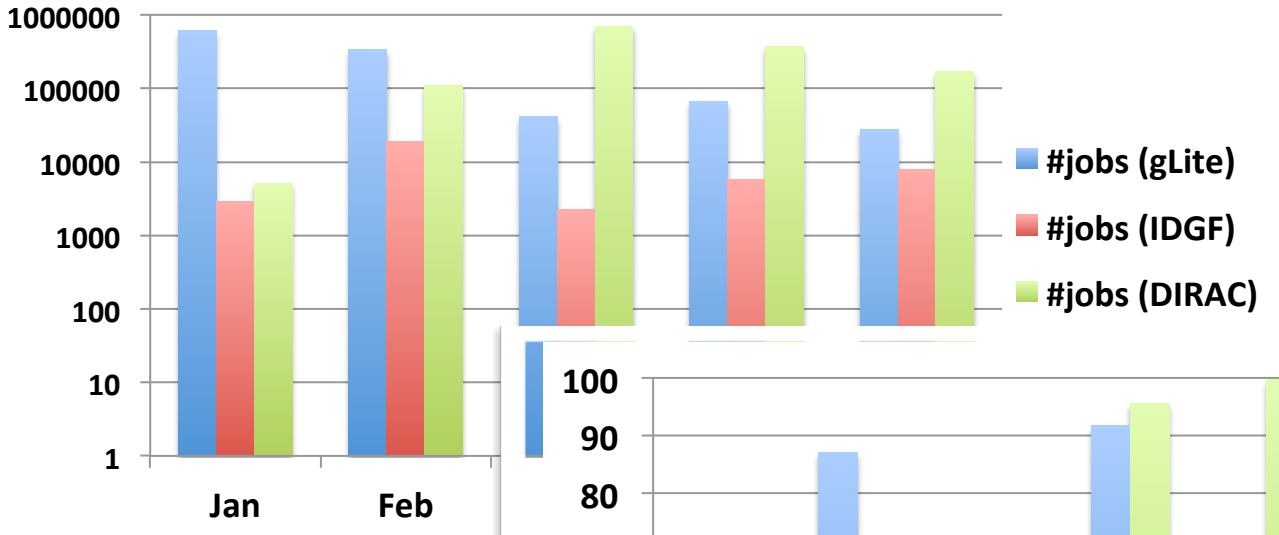
HADDOCK goes DIRAC

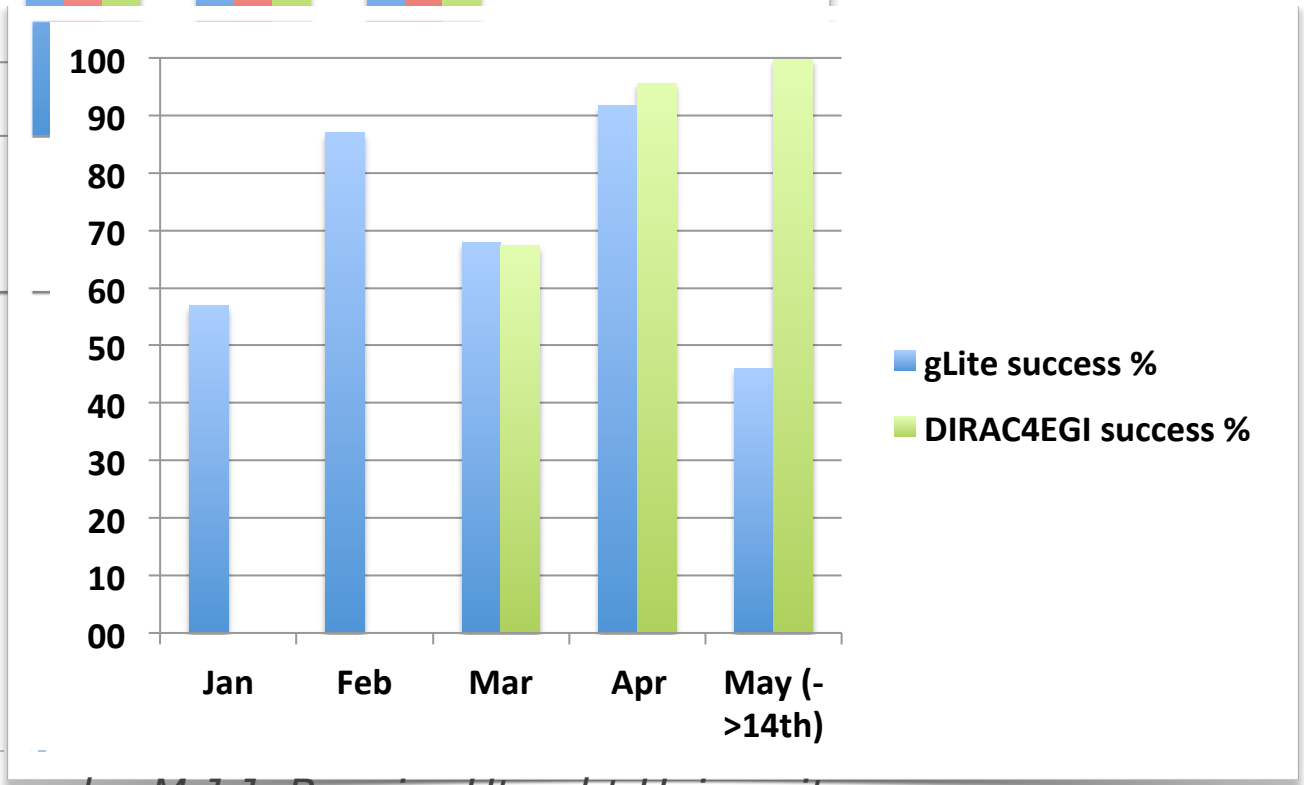
- **DIRAC submission enabled at minimum cost!**
  - In one afternoon, thanks to the help or Ricardo  and Andrei 
  - Clone of the HADDOCK server on a different machine
  - No root access required, no EMI software installation required
- **Minimal changes to our submission and polling scripts**
  - Requirements and ranking no longer needed, only CPUTime

```
JobName = "dirac-xxx";
CPUTime = 100000;
Executable = "dirac-xxx.sh";
StdOutput = "dirac-xxx.out";
StdError = "dirac-xxx.err";
InputSandbox = {"dirac-xxx.sh","dirac-xxx.tar.gz"};
OutputSandbox = {"dirac-xxx.out", "dirac-xxx.err","dirac-xxx-result.tar.gz"};
```

- **Very efficient submission** (~2s per job – without changing our submission mechanism)**, high job throughput**

Some WeNMR statistics





DIRAC
THE INTERWARE

Users

- A prototype DIRAC service in LIT/JINR
  - Installed since July 2015
  - Basic functionality is demonstrated
    - What's next is being discussed
- Some dedicated services are extended to support other communities
  - ILC service extended to support the Calice VO
    - GEANT 4 to join ? CERN service ?
  - BES III service is to support Juno, CEPC, LHAASO, HXMT, Daya Bay experiments

Multi-VO Services

- There is a clear need for services like DIRAC for an increasing number of communities with a low expertise in (distributed) computing and with high demands for computing resources

- Important goal is to lower the threshold for scientists of these communities
  - Training is one of the main purposes of all the infrastructures deploying DIRAC services
  - Examples of training projects
    - vo.formations.idgrilles.fr (FG-DIRAC)
    - training.egi.eu (DIRAC4EGI)
      - The next DIRAC tutorial is at the EGI Community Forum, November
    - Distributed computing student courses
    - CERN@school project using the GridPP DIRAC service
  - Assistance in porting application to the (DIRAC) grids

Conclusions

- DIRAC is providing a framework to bring together various services and computing resources in a single coherent system
- From the user perspective the whole system is seen as a single computer with an intuitive (graphical) interface
- DIRAC is extensible to easily include new technologies and specific developments in managing data and workflows
- Multi-VO DIRAC services is an excellent way to open access to distributed computing resources for non-expert user communities.

http://diracgrid.org