Distributed Infrastructure for Scientific Applications

A.Tsaregorodtsev,
Aix Marseille Univ, CNRS/IN2P3, CPPM
Belle II Computing Workshop, CC-IN2P3, Lyon
15 January 2020

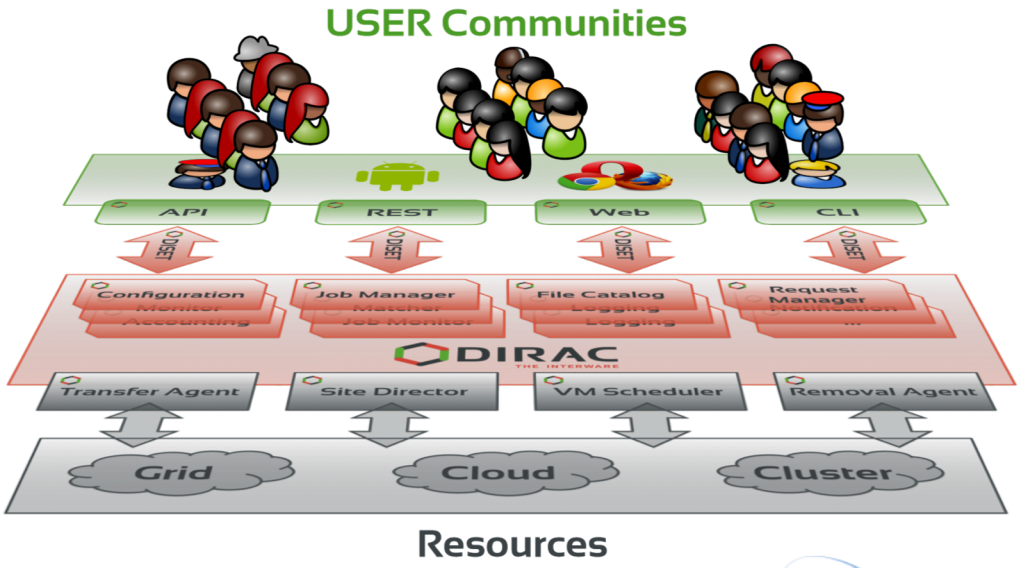
DIRAC
THE INTERWARE

Outline

- DIRAC Project
- DIRAC Interware
  - WMS and computing resources
  - DMS and storage resources
  - Interfaces
  - Framework for distributed computing systems
- Conclusions

Interware

- 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



An open source project

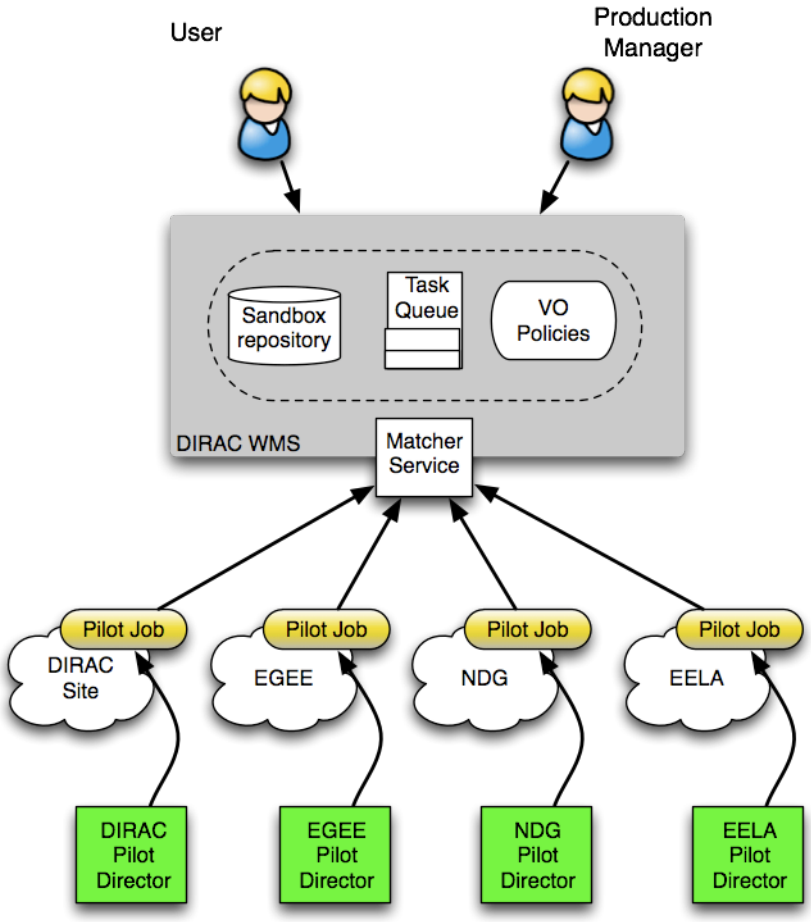
- Started as an LHCb project, became experiment-agnostic in 2009
  - First users (after LHCb) end of 2009
- Developed by communities, for communities
  - Open source (GPL3+), GitHub hosted, python 2.7
  - No dedicated funding for the development of the "Vanilla" project
  - Publicly documented, active assistance forum, yearly users workshops, open developers meetings
  - 4 FTE as core developers, a dozen contributing developers
- The DIRAC consortium as representing body
  - CNRS, CERN, IHEP, KEK
  - PNNL, University of Montpellier, Imperial College

DIRAC
THE INTERWARE

Workload Management

Job scheduling

- 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



Computing resources

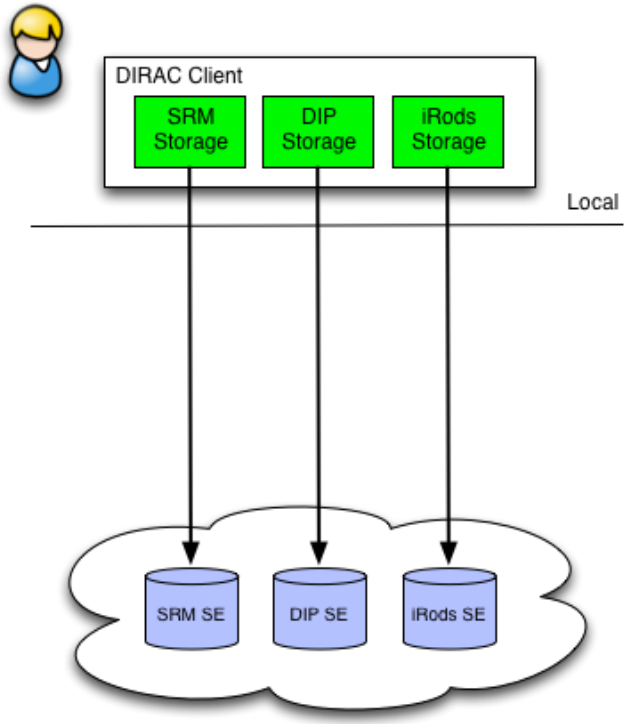
- Computing Grids
  - CREAM/HTCondorCE/ARC Computing Elements
  - Globus ComputingElement (OSG)
- Clouds
  - Dynamic VM life cycle management
  - Openstack, OpenNebula
  - Amazon EC2
- Standalone computing clusters
  - Access through SSH tunnel
  - LSF, BQS, SGE, PBS/Torque, Condor
    - Commodity computer farms
  - OAR, SLURM
    - HPC centers

DIRAC
THE INTERWARE

Data Management

Storage access: plugins

- 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
- Like with CE's, each SE is seen by the clients as a logical entity
  - With some specific operational properties
    - Archive, limited access, etc
  - SE's can be configured with multiple protocols
- Including new data access technologies requires creating new specific plug-in

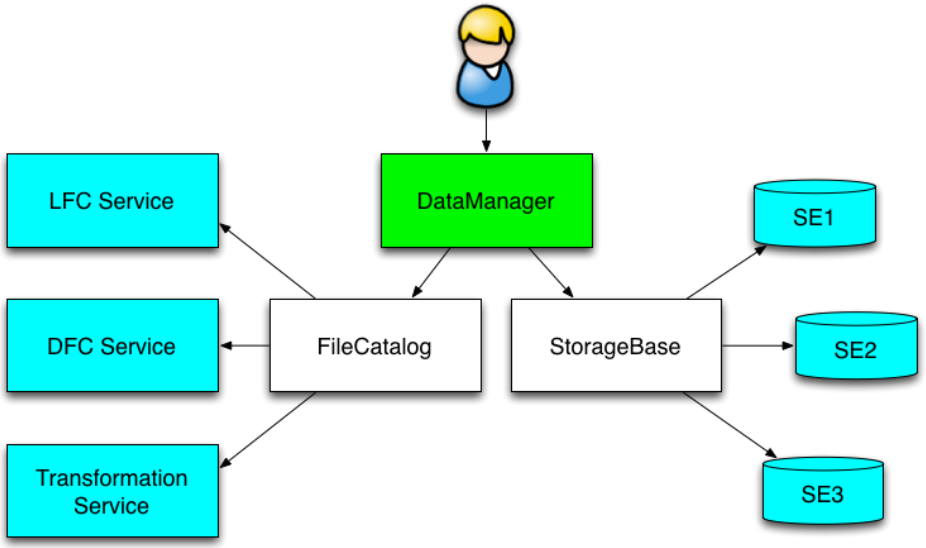


File Catalog Service

- 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
  - Other projects, e.g. distributed file systems, keep file data in multiple distributed databases
    - More scalable
    - Maintaining data integrity is very difficult

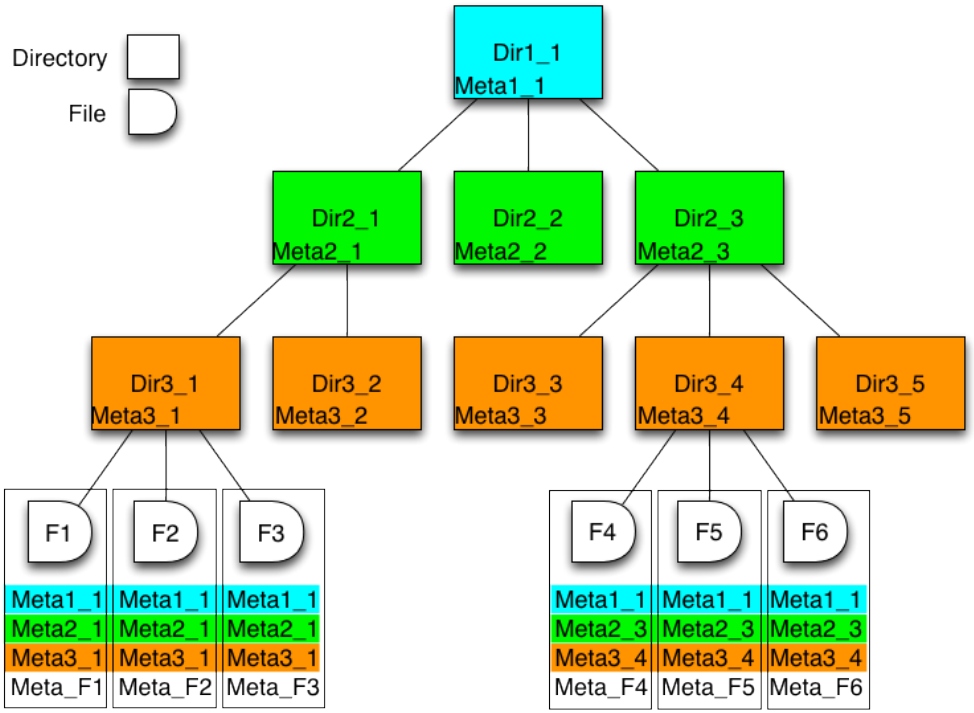
Combined data API

- Together with the data access components DFC allows to present data to users as a single global file system
- DataManager API is a single client interface for logical data operations



File Catalog: Metadata

- 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`

- Result of file search is a precise list of corresponding files
  - Unlike Google index

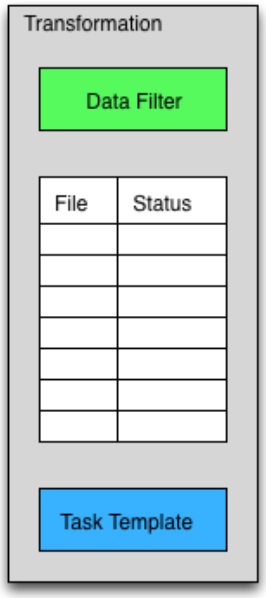
DIRAC
THE INTERWARE

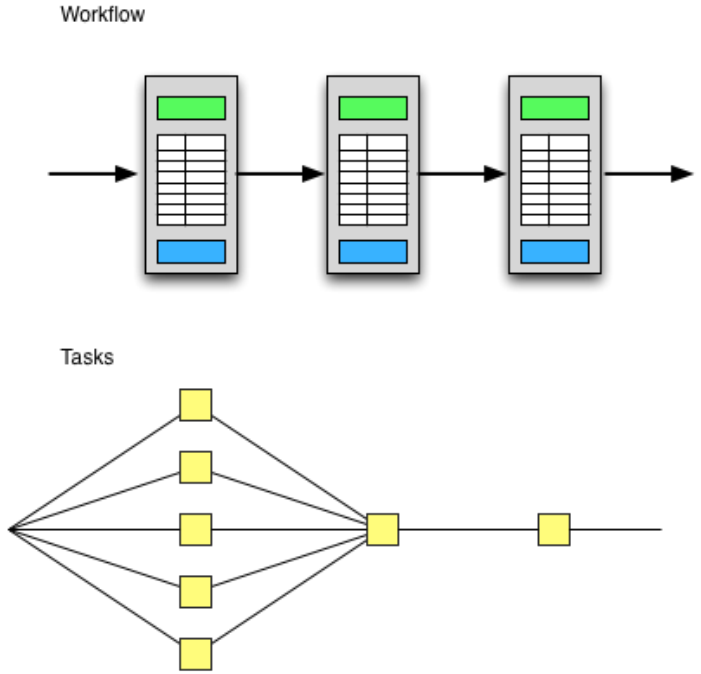
Data operations

- StorageManagement System
  - Support for data staging
- RequestManagement System - massive data operations
  - Data replication, removal, etc
  - Asynchronous execution
  - Automatic failure recovery
- Transformation System
  - Automated data driven workflows

Transformation System for data driven workflows

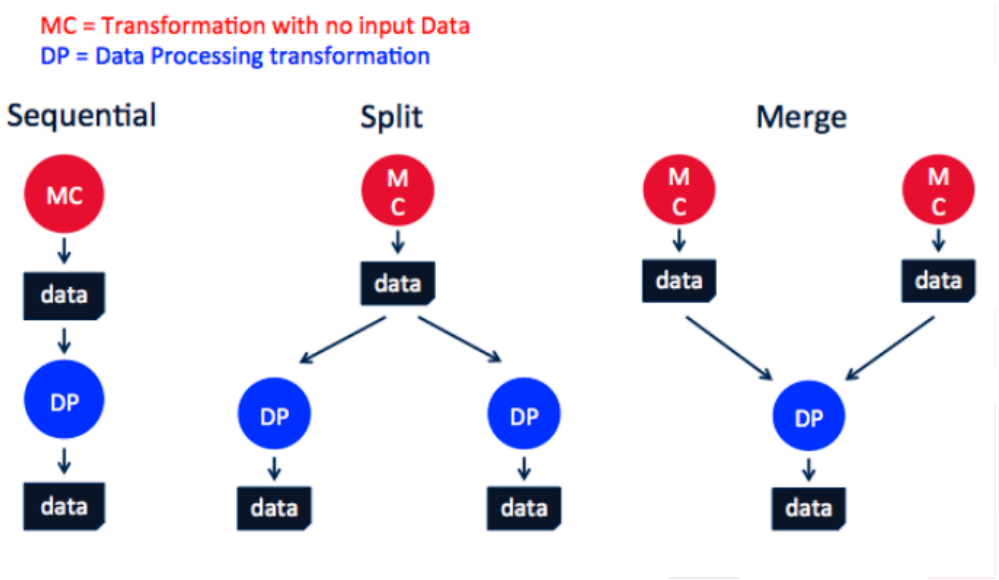
- Data driven workflows as chains of data transformations
  - Transformation: input data filter + recipe to create tasks
  - Tasks are created as soon as data with required properties is registered into the system
  - Tasks: jobs, data replication, etc
- Transformations can be used for automatic data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





Production System

- Available in DIRAC version v7r0
- Automatic transformation instantiation based on the production definition
- Fully data-driven
- Tested for simple workflow schemes and their combinations



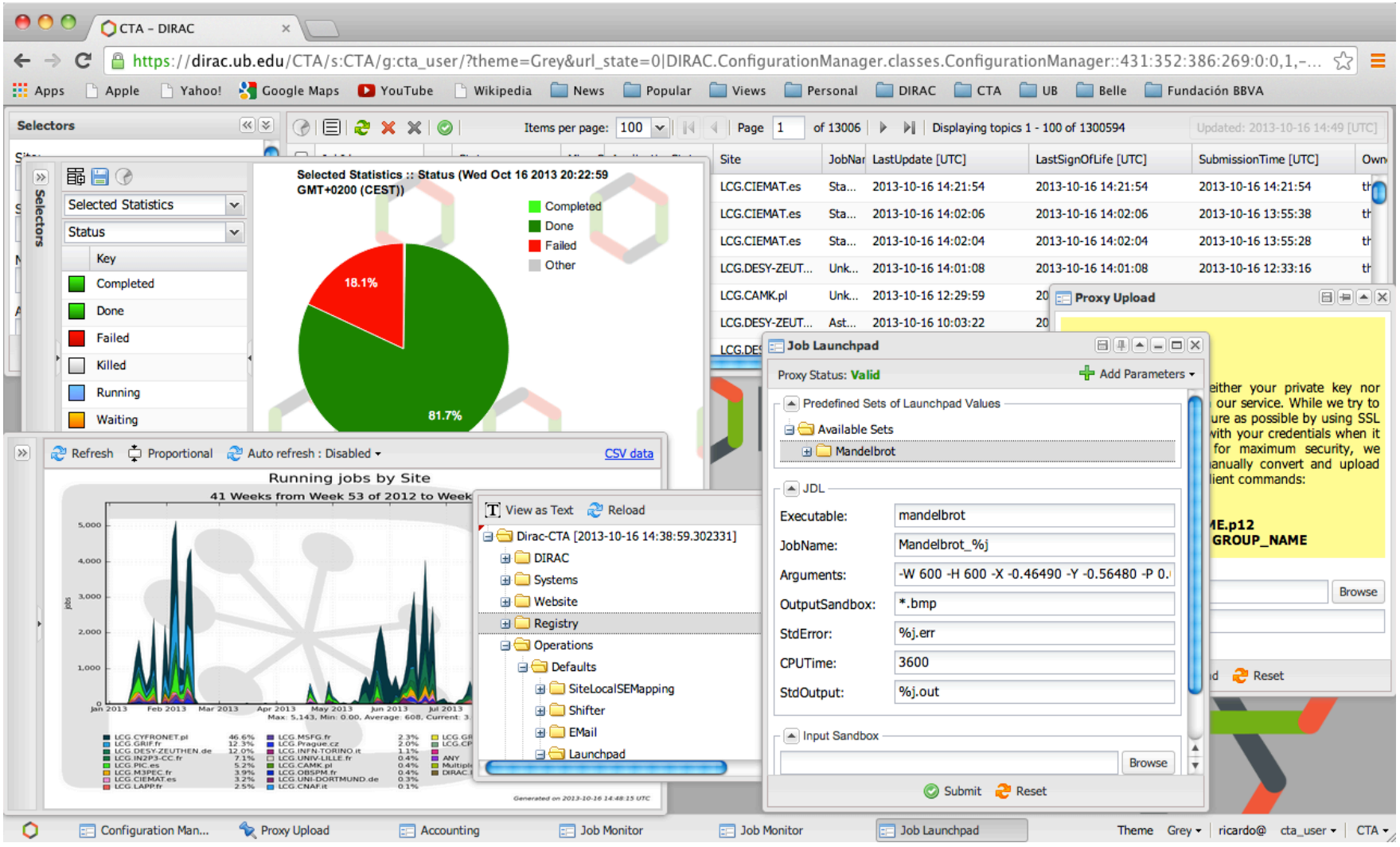
DIRAC
THE INTERWARE

Interfaces

DM interfaces

- Command line tools
  - Multiple dirac-dms-… commands
- COMDIRAC
  - Representing the logical DIRAC file namespace as a parallel shell
  - **dls, dcd, dpwd, dfind, ddu** etc commands
  - **dput, dget, drepl** for file upload/download/replication
- REST interface
  - Suitable for use with application portals
  - Multiple application portals are interfaced to DIRAC this way

Web Portal applications



DIRAC
THE INTERWARE

Development Framework

DIRAC software

- Python 2.7
  - Python 3 migration is planned
- GPLv3 license
- GitHub repository
- External dependencies are compiled and provided as part of the DIRAC distribution
  - DIRACOS package
- DIRAC extensions are managed by communities
  - E.g. CTADIRAC
    - GitHub repository
- External service used
  - MySQL, ElasticSearch, Stomp enabled MQ's
- Software is available as
  - installable tar archives
    - dirac-install tool
  - preinstalled in CVMFS
  - Docker containers

DIRAC development framework

- Modular architecture
  - Much functionality is implemented as plugins, e.g.
    - Job policies
    - Data access rules
    - Web applications
- Multiple community specific extensions exist already
  - BelleDIRAC, LHCbDIRAC, CTADIRAC, ILCDIRAC, etc
- Community specific services can run as part of a general purpose DIRAC service
  - E.g. Eiscat 3D specific File Catalog as part of the EGI DIRAC service

Highlights of ongoing developments

- Support for OAuth2/OIDC based AuthN/AuthZ frameworks
  - Federated Identity Providers, Proxy Providers
- Replacing the custom DIRAC client/server protocol by an HTTP based protocol
  - Standards based, better flexibility, and scalability
- Python 3 migration
- Rucio interface (RucioFileCatalog)

DIRAC
THE INTERWARE

Conclusions

- DIRAC is a complete solution for a large scientific community to build its distributed computing system
- DIRAC is continuously developed to keep it up to date with the technology advancements
- DIRAC together with some specific extension can suit the needs of the Belle II experiment

DIRAC
THE INTERWARE

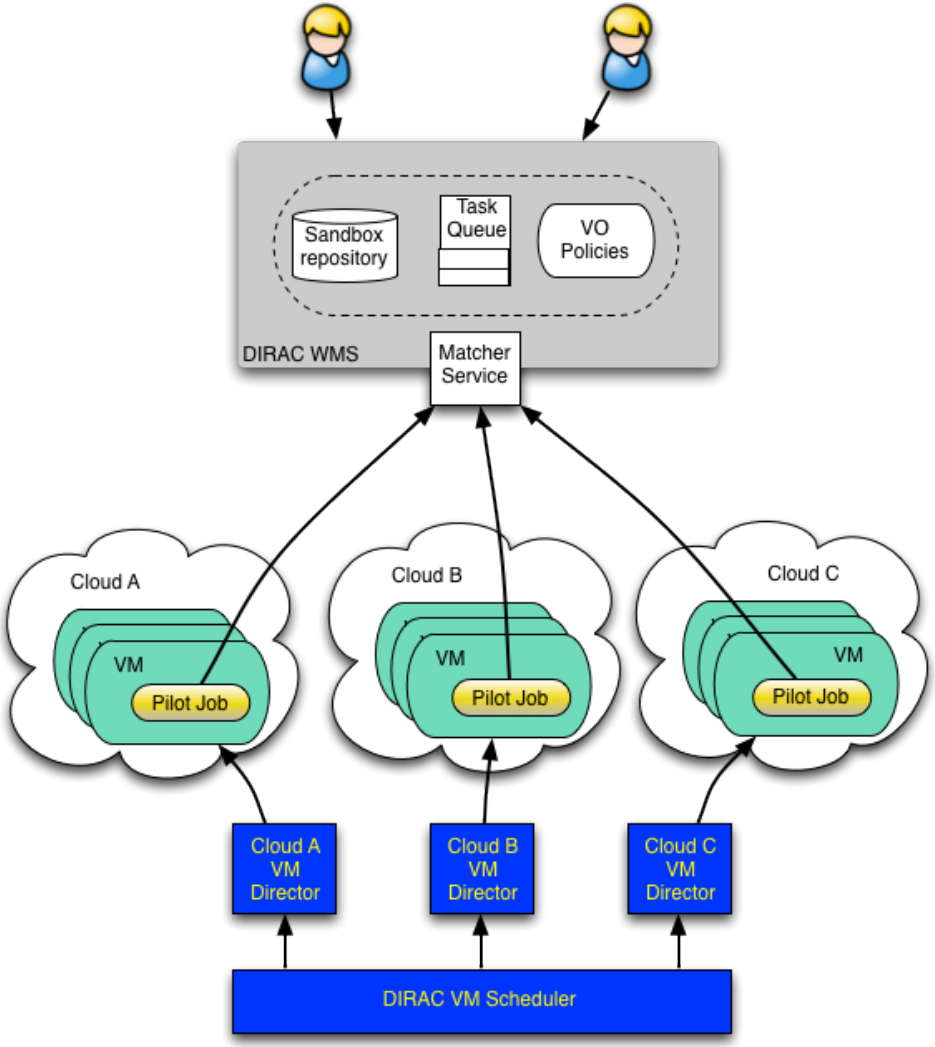
Backup slides

Resources: Computing Grids

- DIRAC was initially developed with the focus on accessing conventional Grid computing resources
  - WLCG grid resources for the LHCb Collaboration
- It fully supports different middleware based grids
  - European Grid Infrastructure (EGI), WLCG, etc
    - DIRAC is an officially supported WMS service for the EGI infrastructure
      - CREAM/HTCondorCE/ARC ComputingElement's
  - Northern American Open Science Grid (OSG)
    - GlobusComputingElement's
- Other types of grids can be supported
  - As long we have customers needing that

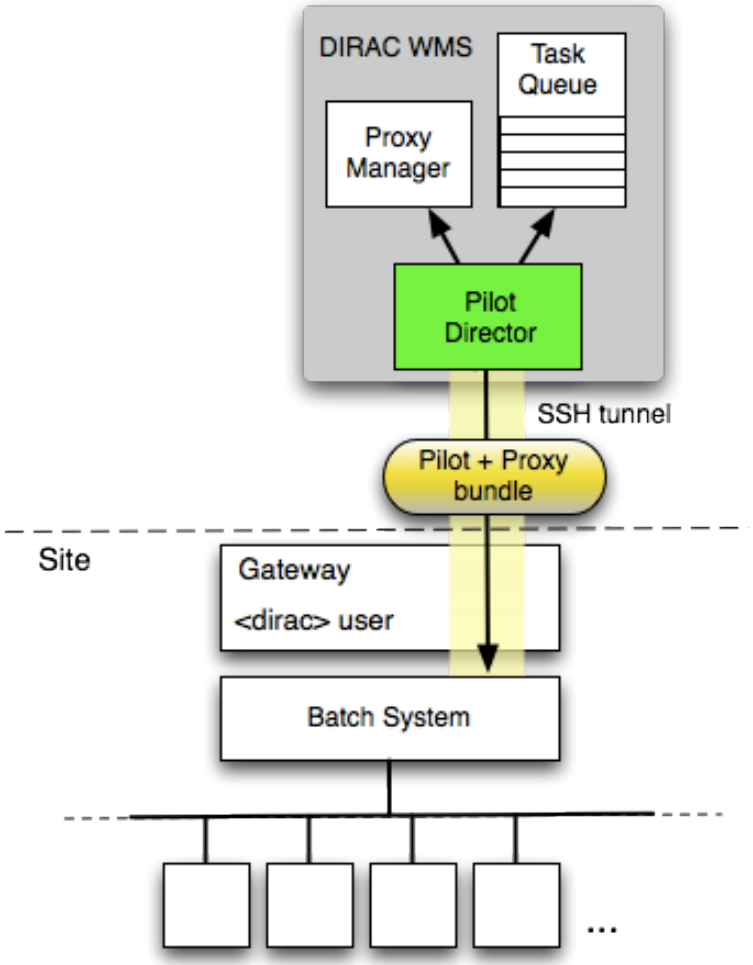
Resources: Clouds

- 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
  - …

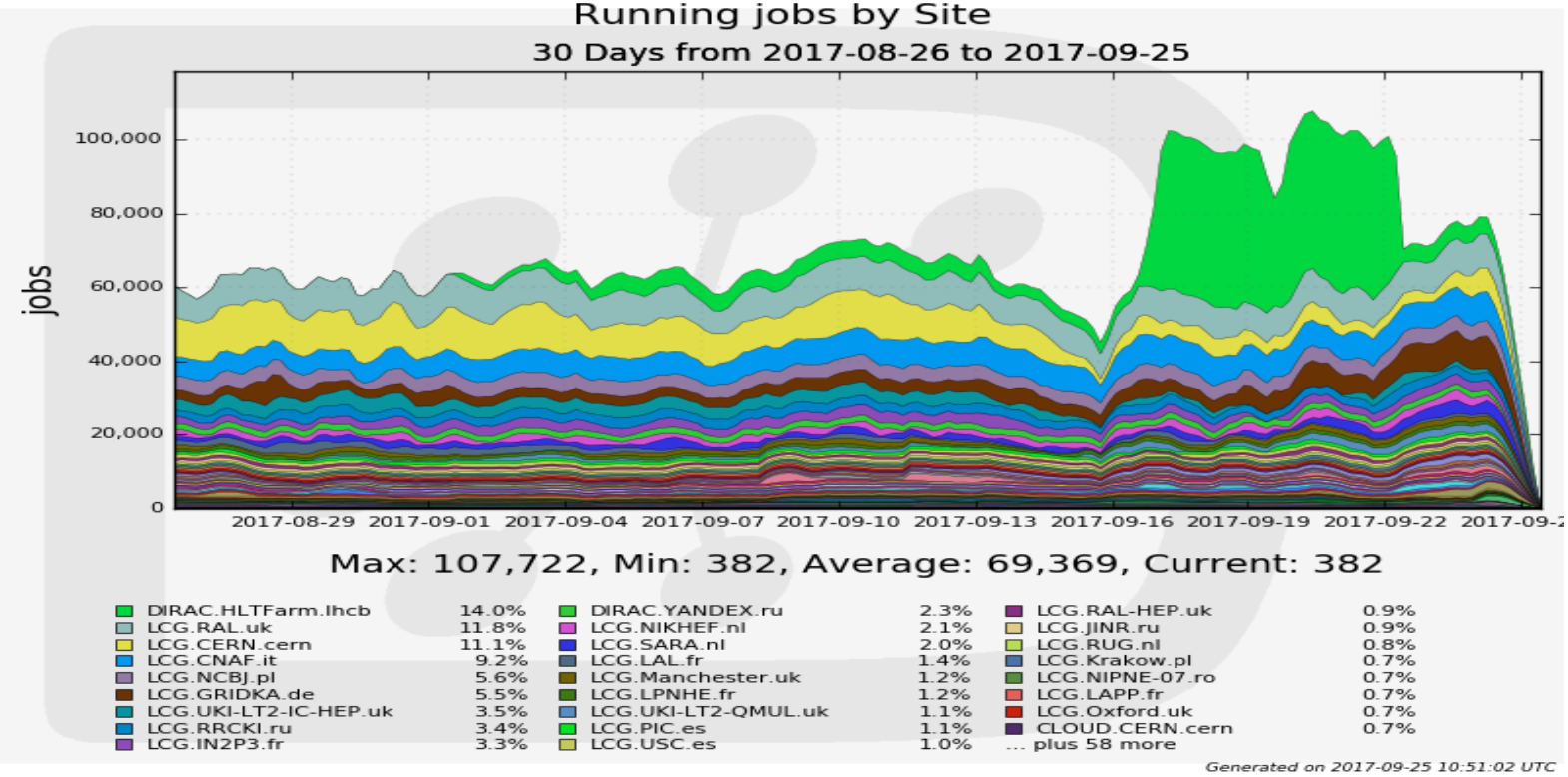


Resources: standalone computing clusters

- Off-site Pilot Director
  - Site delegates control to the central service
  - Site must only define a dedicated local user account
  - The payload submission through an SSH/GSISSH tunnel
- The 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
    - OAR, SLURM
      - HPC centers
- The user payload is executed with the owner credentials
  - No security compromises with respect to external services



LHCb Collaboration

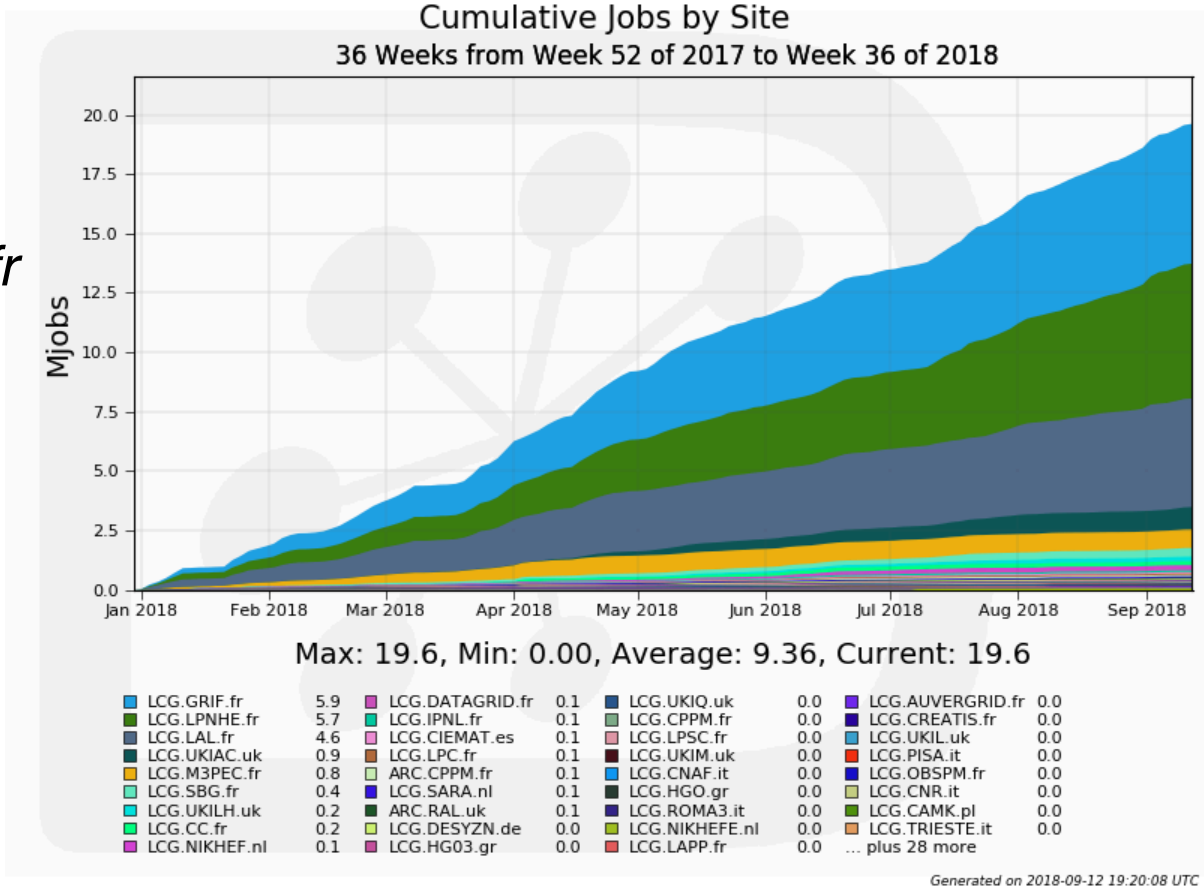


- More than 100K concurrent jobs in ~120 distinct sites
  - Limited by available resources, not by the system capacity
- Further optimizations to increase the capacity are possible
  - Hardware, database optimizations, service load balancing, etc

DIRAC
THE INTERWARE

FRANCE GRILLES

- Hosted by the CC/IN2P3, Lyon
  - *dirac.france-grilles.fr*
- Distributed administrator team
  - 5 participating universities
- In production since May 2012



- About 5 active communities – complex-systems, biomed, vo.france-grilles.fr, …
- > 20M jobs executed this year at 90 sites

DIRAC4EGI service

- Partners
  - Operated by EGI
  - Hosted by CYFRONET, Krakow
  - DIRAC Project providing software, consultancy
  - Supported via the EOSC-Hub H'2020 grant
  - *dirac.egi.eu*
- 20 Virtual Organizations
  - enmr.eu
  - virgo
  - eli-beams
  - eiscat.se
  - fedcloud.egi.eu
  - …
- Usage
  - > 5 million jobs processed this year

DIRAC4EGI activity snapshot