Distributed Computing Framework

A.Tsaregorodtsev,
CPPM-IN2P3-CNRS, Marseille,
Plekhanov University of Economics, Moscow
NorduGRID'17, 28 June 2017, Tromsø

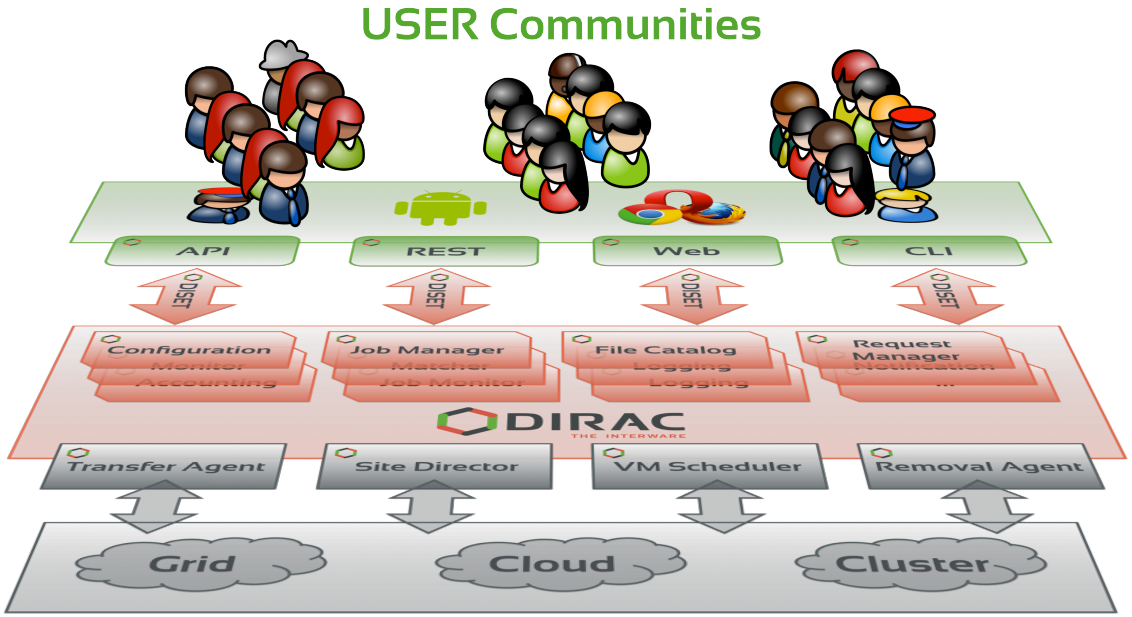
DIRAC
THE INTERWARE

Plan

- DIRAC Project overview
- Computing and Storage resources
- Users
- Services
- Development framework
- Conclusions

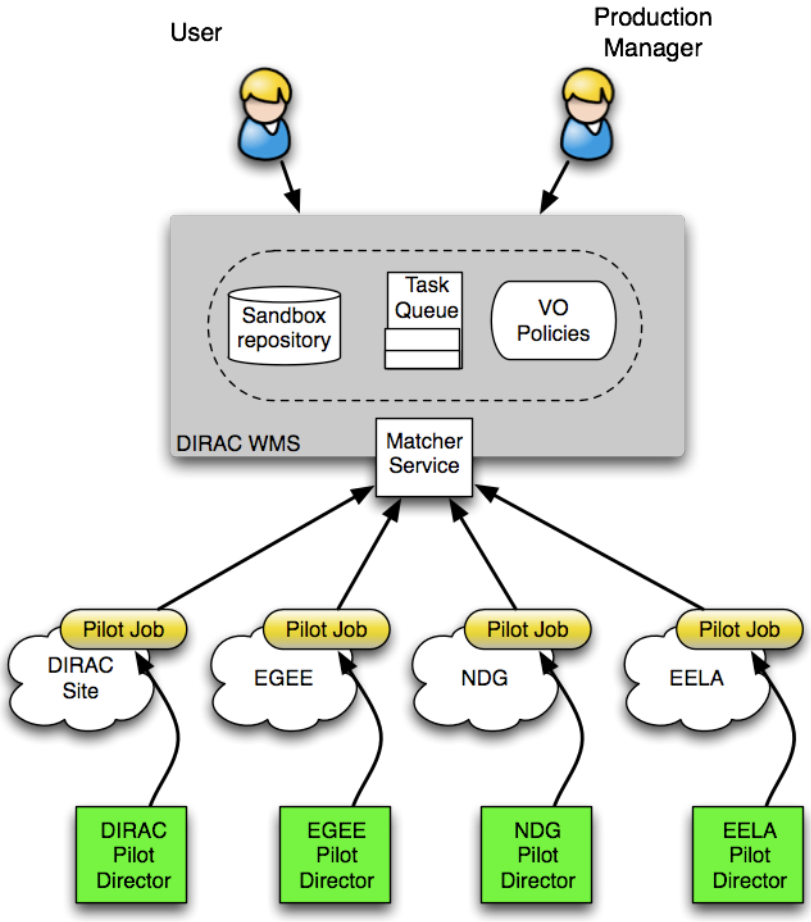
Interware

- 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***.



Job scheduling

- Pilot jobs are submitted to computing resources by specialized Pilot Directors
- After the start, Pilots check the execution environment and form the resource description
  - OS, capacity, disk space, software, etc
- The resources description is presented to the Matcher service, which chooses the most appropriate user job from the Task Queue
- The user job description is delivered to the pilot, which prepares its execution environment and executes the user application
- In the end, the pilot is uploading the results and output data to a predefined destination



Computing Resources

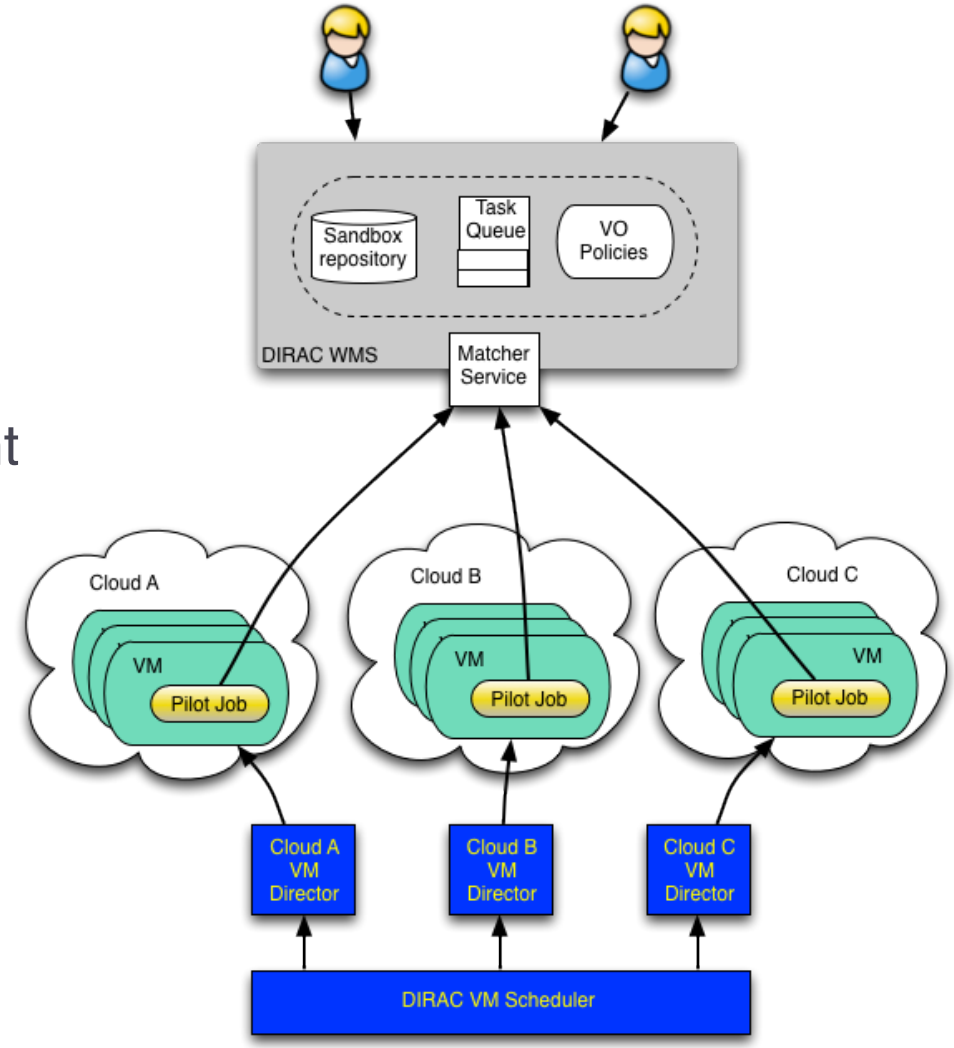
- DIRAC was initially developed with the focus on accessing conventional Grid computing resources
  - WLCG grid resources for the LHCb Collaboration
  - It fully supports multiple grid middlewares and infrastructures
    - EGI, WLCG, OSG, NorduGRID, etc
  - Other types of grids can be supported
    - As long we have customers needing that
- Standalone clusters
  - Access through SSH/GSISSH tunnel
  - Batch systems supported: LSF, BQS, SGE, PBS/Torque, Condor, OAR, SLURM
    - Used to access HPC centers
- BOINC Volunteer resources
  - Running pilots on volunteer machines
  - Separation of secure and unsecure parts, plugins for results validation

ARC Computing Element

- Access to ARC CE services via a corresponding ComputingElement plugin
  - Using ***arc*** python binding
    - Job submission, getting results, killing
  - Using BDII (ldap) commands to discover job and CE statuses
- Frequent problems with BDII look-up
  - Can not retrieve the CE occupancy, especially with respect to a particular community
  - Alternatively, using PilotAgentsDB of DIRAC to evaluate the load on a given ARC CE
- Considering using the ARC REST interface
  - Avoid ***arc*** python binding dependency

VMDIRAC extension

- VMDIRAC extension developed for Belle MC production system
  - Dynamic VM spawning taking Amazon EC2 spot prices and Task Queue state into account
- Now VMDIRAC is a general purpose service for VMs life cycle management
  - Creation
  - Monitoring
  - Discarding



VM submission

- Cloud endpoint plugins to interact with particular cloud provides
- Cloud endpoint abstraction
  - Implementations (*IHEP, Beijing*)
    - Apache-libcloud
      - Catch-all library, but not really…
    - Rocci
      - Using command line interface
      - Allow connections with GSI proxies
    - EC2
      - Boto python API
  - More implementations are in the works
    - OCCI, Google, Azur, IBM, …
    - Preferring RESTful interfaces

VM submission

- CloudDirector – VMDIRAC way
  - Similar to SiteDirector for grid jobs submission
  - VM submission based on the Task Queue status
    - If there are waiting user payloads
    - VM properties corresponding to payload requirements
- Vac/Vcycle (*A. McNab*)
  - Used by LHCb
  - Spawning VMs without a priori knowledge about the state of the Task Queue
- Similar contextualization and pilots
  - Separate development subproject to provide pilots running in DIRAC-free environments

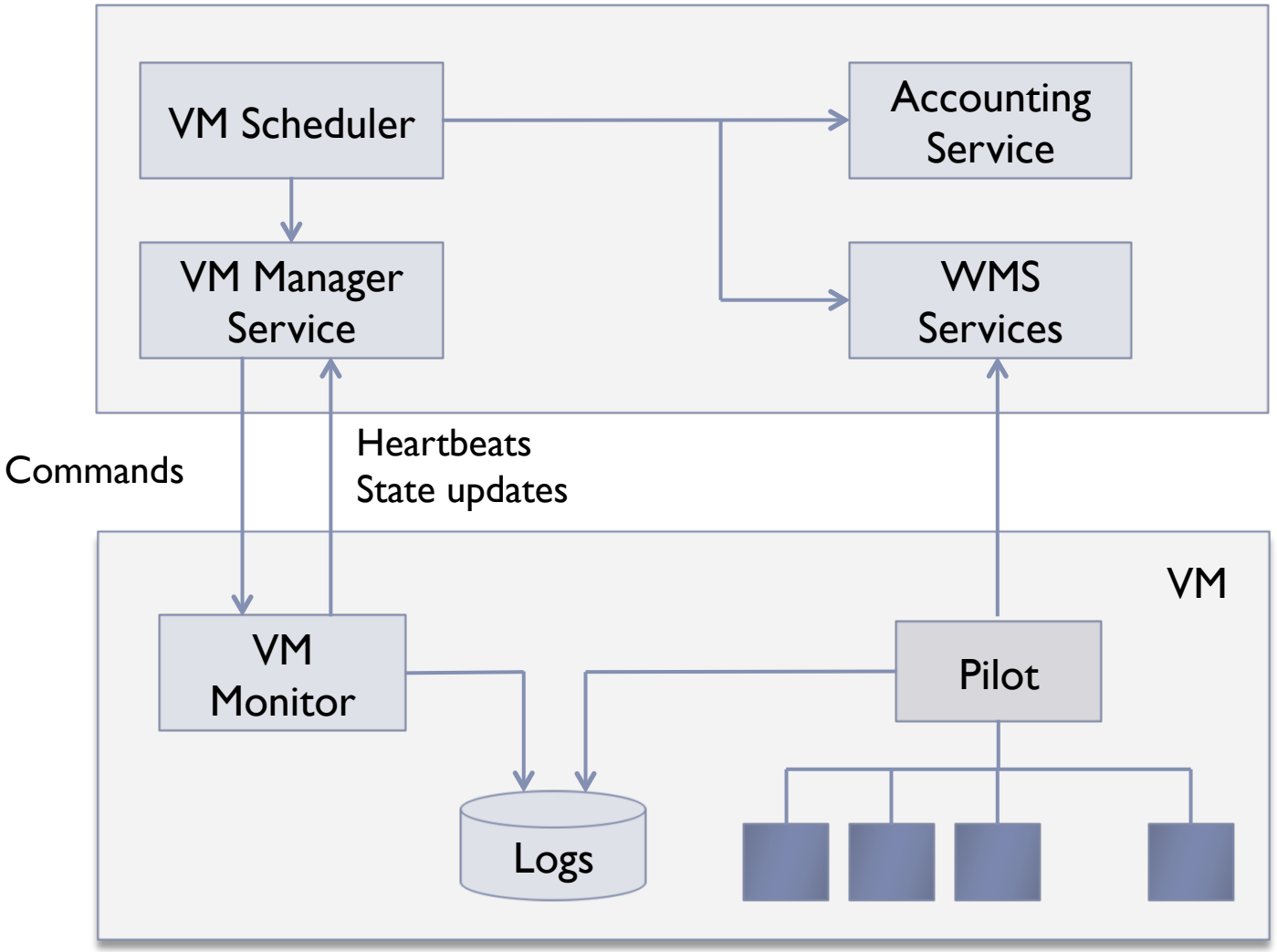
Pilots in the VMs

- Same as any other pilots
  - DIRAC Pilot 2.0 framework
    - A set of commands for the DIRAC environment installation and setup, starting Job Agents interacting with the WMS central service
    - User communities can provide custom pilot commands in addition and/or in replacement of the standard omes
- Managing the VM CPU cores scenarios
  - Launching as many pilots as they are cores
    - Suitable for single-core payloads, *à la* grid jobs
  - Launching single pilot
    - Suitable for multi-core payloads occupying the whole VM
  - Single pilot with a PoolComputingElement plugin for payloads execution
    - Simple "batch system" to manage VM job slots
    - Can execute payloads with any requirements to the number of cores: single, exact number of cores or whole node occupancy

Managing VM life cycle

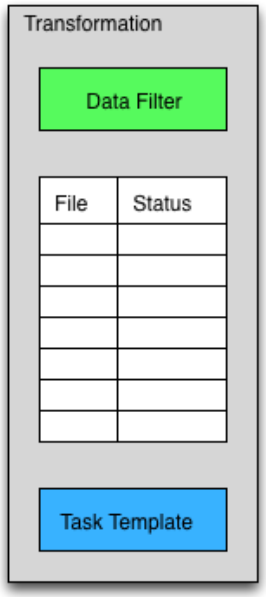
- *VM Monitor Agent* is launched in parallel with the pilot process during the VM bootstrapping
  - This is a watchdog for activities on the VM
  - Sends heartbeats and VM status information to the central VM Manager service
  - Can receive instructions from the central service as a response to the heartbeat
    - E.g., halt, drain and other commands
  - Monitors the VM status
  - Can be configured to halt the VM with different policies
- *VM Scheduler* orchestrates spawning and halting virtual machines depending on the Task Queue status, Accounting history
  - Necessary for fair sharing of cloud resources
  - Work in progress

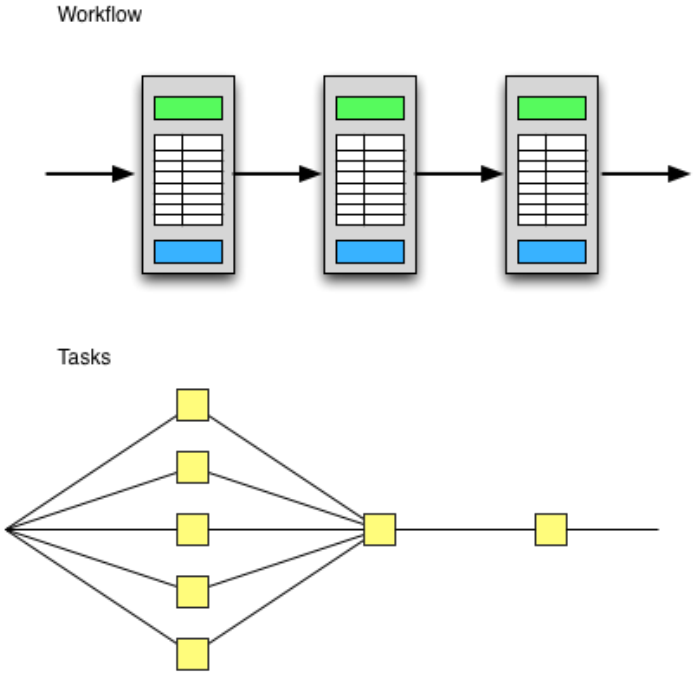
VM management



Transformation System

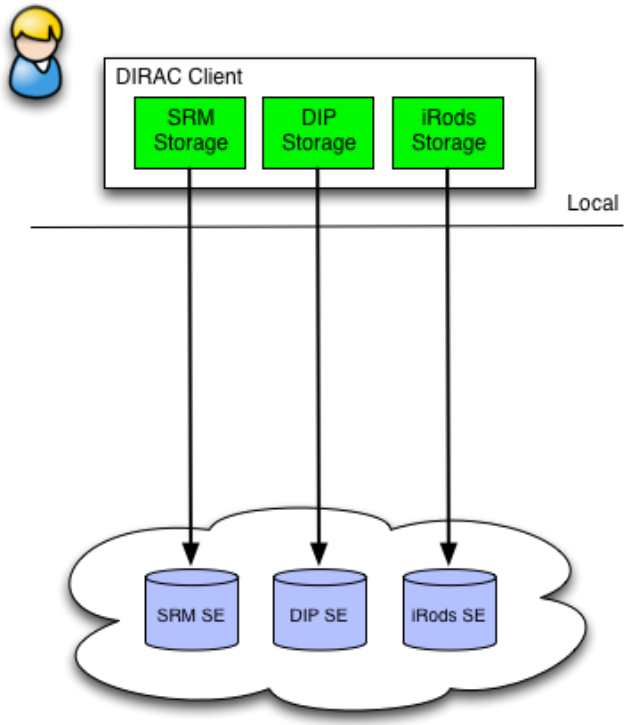
- Data driven workflows as chains of data transformations
  - Transformation: input data filter + recipe to create tasks
  - Tasks are created as soon as data with required properties is registered into the system
  - Tasks: jobs, data operations, etc
- Transformations can be used for automatic data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





Storage plugins

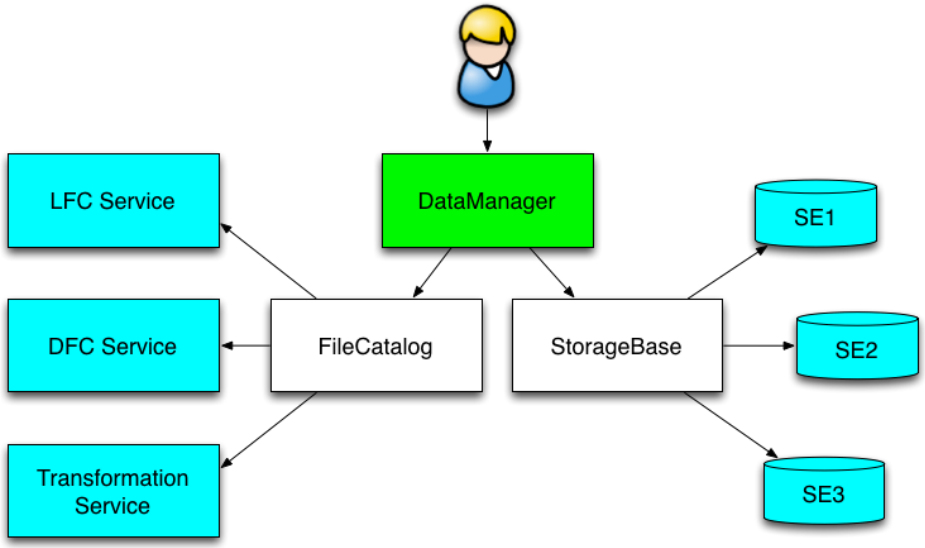
- Storage element abstraction with a client implementation for each access protocol
  - DIPS, SRM, XROOTD, RFIO, etc
  - gfal2 based plugin gives access to all protocols supported by the library
    - HTTP, DCAP, WebDAV, S3, …



- Each SE is seen by the clients as a logical entity
  - With some specific operational properties
  - SE's can be configured with multiple protocols

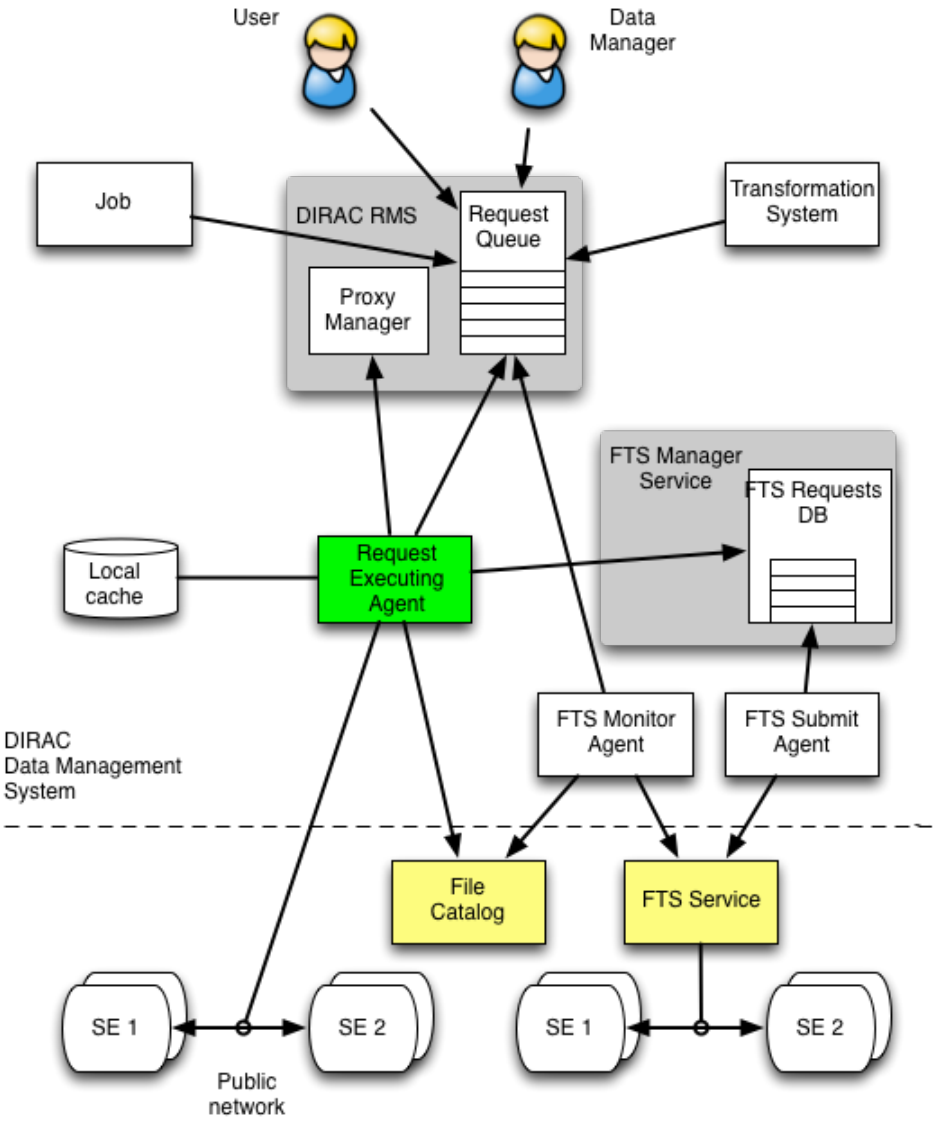
File Catalog

- Central File Catalog (DFC, LFC, …) is maintaining a single global logical name space
- Several catalogs can be used together
  - The mechanism is used to send messages to "pseudocatalog" services, e.g.
    - Transformation service (see later)
    - Bookkeeping service of LHCb
  - A user sees it as a single catalog with additional features
- DataManager is a single client interface for logical data operations

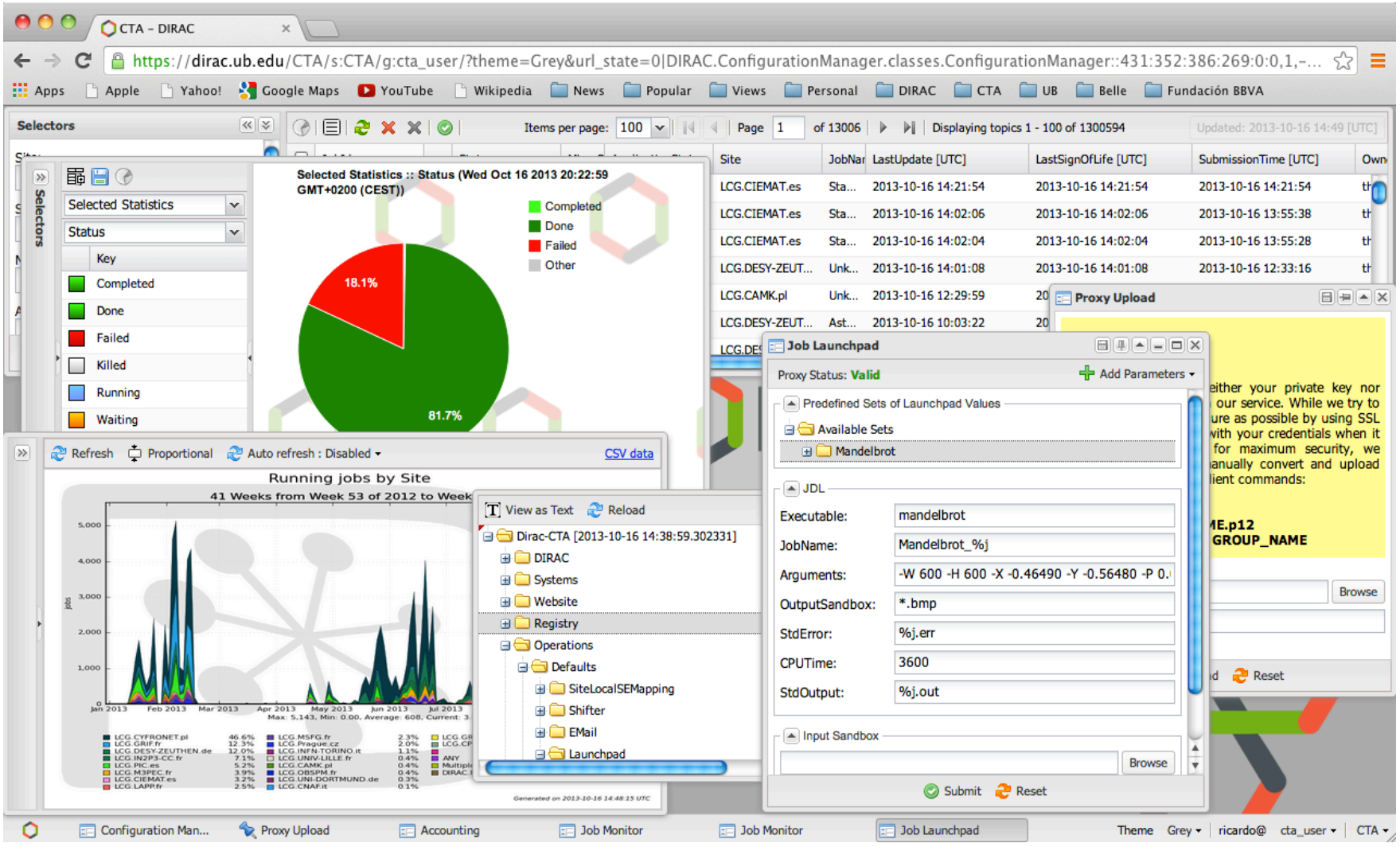


Bulk transfers

- Replication/Removal Requests with multiple files are stored in the RMS
  - By users, data managers, Transformation System
- The Replication Operation executor
  - Performs the replication itself or
  - Delegates replication to an external service
    - E.g. FTS
  - A dedicated FTSManager service keeps track of the submitted FTS requests
  - FTSMonitor Agent monitors the request progress, updates the FileCatalog with the new replicas



Web Portal examples

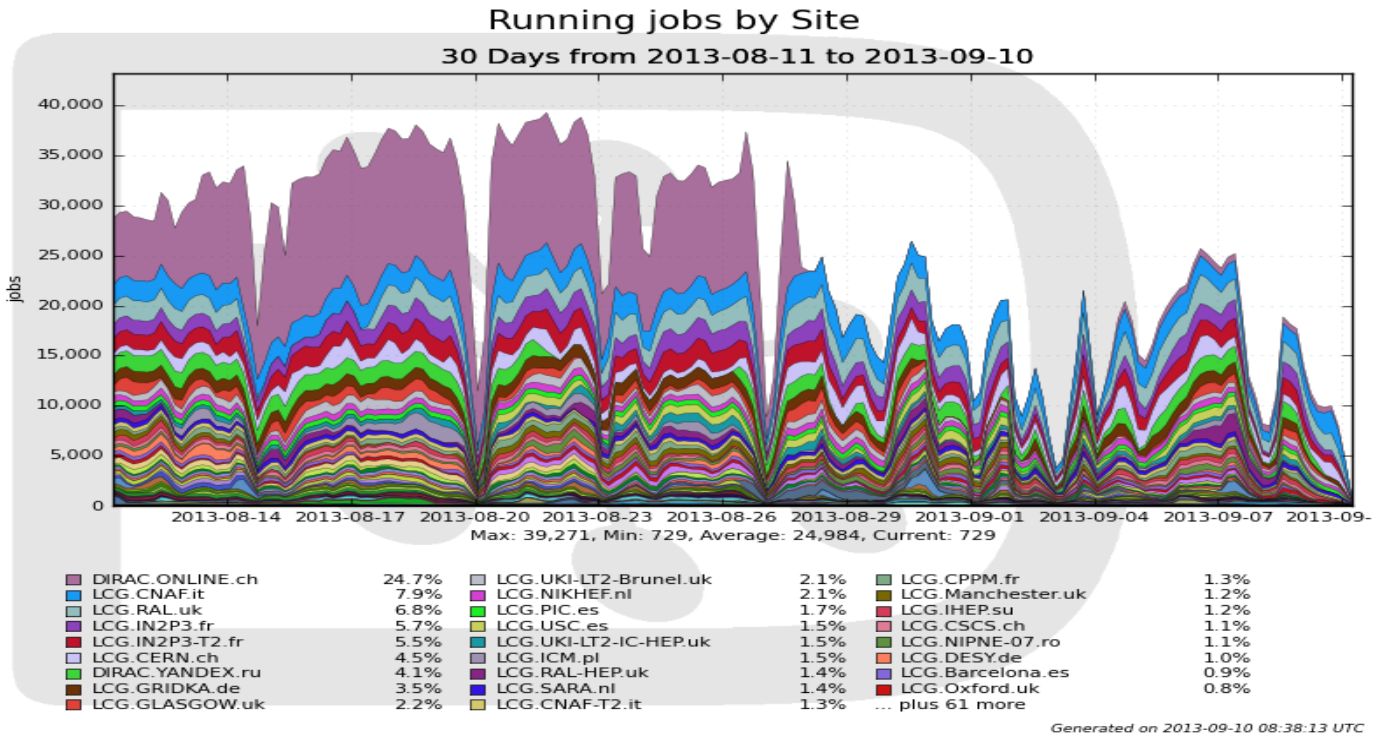


DIRAC
THE INTERWARE

Distributed Computer

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

LHCb Collaboration



- More than 100K concurrent jobs in ~120 distinct sites
  - Equivalent to running a virtual computing center with a power of 100K CPU cores
- Further optimizations to increase the capacity are possible
  - Hardware, database optimizations, service load balancing, etc

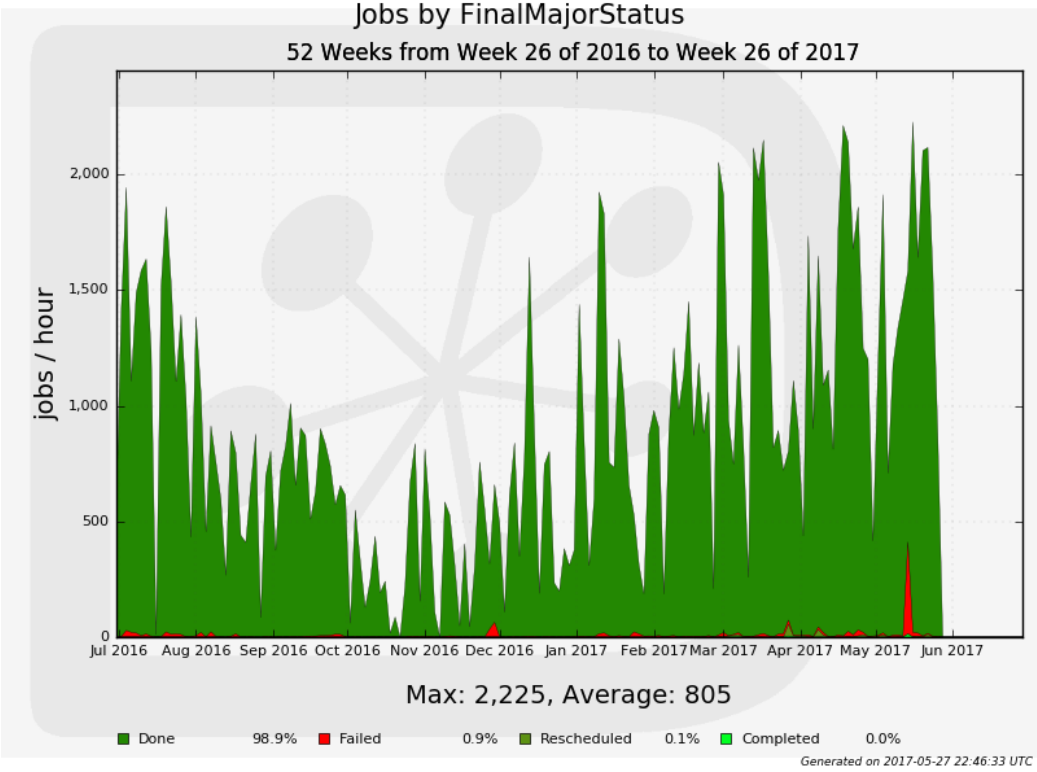
DIRAC services

- Dedicated installations
  - LHCb, Belle II, CTA
- Multi-community services
  - CERN: ILC, CALICE
  - IHEP: BES III, Juno, CEPC
  - FG-DIRAC
  - GridPP
  - DIRAC4EGI
- New services
  - PNNL: Belle II, Project8, MiniCLEAN, SuperCDMS, nEXO
  - DIRAC@JINR: NICA, Dubna University
- Several DIRAC evaluations are ongoing
  - Auger, ELI, …

DIRAC4EGI service

- In production since 2014
- Partners
  - Operated by EGI
  - Hosted by CYFRONET
  - DIRAC Project providing software, consultancy
- 10 Virtual Organizations
  - enmr.eu, vlemed, eiscat.se
  - fedcloud.egi.eu
  - training.egi.eu
- Usage
  - > 6 million jobs processed in the last year
  - Data Management solution
    - Eiscat 3D
- Starting from 2018 DIRAC becomes Core Service of EGI
  - WMS replacement
  - Serving both Grid and FedCloud resources
  - Part of H'2020 EINFRA-12 proposal

DIRAC4EGI activity snapshot



DIRAC
THE INTERWARE

EGI ACCOUNTING PORTAL

Normalised CPU time [units 1K.SI2K.Hours] by DATE and VO												
DATE	alice	atlas	belle	biomed	cms	compchem	ilc	lhcb	virgo	vo.cta.in2p3.fr	Total	%
Nov 2015	83,043,071	213,187,021	29,633,040	2,992,249	107,998,028	812,409	3,051,240	44,495,710	365,193	5,203,790	**490,781,751**	**8.60%**
Dec 2015	81,681,064	167,642,164	30,755,315	2,771,463	81,200,999	1,197,402	10,250,775	42,772,247	4,370	9,643,804	**427,919,603**	**7.50%**
Jan 2016	100,472,899	212,596,116	8,254,706	2,221,994	99,768,667	2,869,544	3,904,455	32,614,451	329,113	8,746,790	**471,778,735**	**8.27%**
Feb 2016	80,340,391	202,531,157	48,965	1,312,309	100,330,129	1,220,127	2,704,948	44,547,976	1,962,465	5,563,528	**440,561,995**	**7.72%**
Mar 2016	108,810,699	172,663,251	3,412,262	2,286,939	75,113,354	1,623,540	2,049,130	83,154,401	1,917,611	1,539,919	**452,571,106**	**7.93%**
Apr 2016	111,707,745	211,516,946	496,969	1,622,314	67,855,621	1,970,394	3,051,624	78,821,567	3,517,152	3,079,316	**483,639,648**	**8.47%**
May 2016	88,434,699	229,055,135	457,771	3,055,283	64,161,648	3,990,478	4,366,309	70,550,242	11,311,493	669,299	**476,052,357**	**8.34%**
Jun 2016	91,963,895	220,222,321	10,039,317	1,375,916	104,040,606	1,755,334	2,097,169	66,545,602	2,558,741	1,103,183	**501,702,084**	**8.79%**
Jul 2016	113,408,142	187,198,001	3,614,046	2,152,445	104,373,741	1,614,892	1,596,155	65,898,735	8,005,698	7,794,153	**495,656,008**	**8.69%**
Aug 2016	88,278,412	212,942,846	34,225	6,500,219	51,366,225	3,474,177	5,538,912	72,803,805	2,919,127	5,410,036	**449,267,984**	**7.87%**
Sep 2016	88,164,653	309,040,532	7,314,602	514,897	90,018,815	2,602,763	3,297,430	106,365,999	1,770,213	6,487,567	**615,577,471**	**10.79%**
Oct 2016	68,902,764	167,532,717	1,528,430	467,733	82,329,281	1,301,416	5,324,702	71,019,670	2,752,272	104,325	**401,263,310**	**7.03%**
Total	**1,105,208,434**	**2,506,128,207**	**95,589,648**	**27,273,761**	**1,028,557,114**	**24,432,476**	**47,232,849**	**779,590,405**	**37,413,448**	**55,345,710**	**5,706,772,052**	
Percentage	**19.37%**	**43.91%**	**1.68%**	**0.48%**	**18.02%**	**0.43%**	**0.83%**	**13.66%**	**0.66%**	**0.97%**		

- 5 out of Top-10 EGI communities used heavily DIRAC for their payload management in the last year
  - 4 out of 6 top communities excluding LHC experiments
    - belle, biomed, ilc, vo.cta.in2p3.fr
    - compchem will likely join the club

DIRAC
THE INTERWARE

DIRAC Software Framework

Software Framework

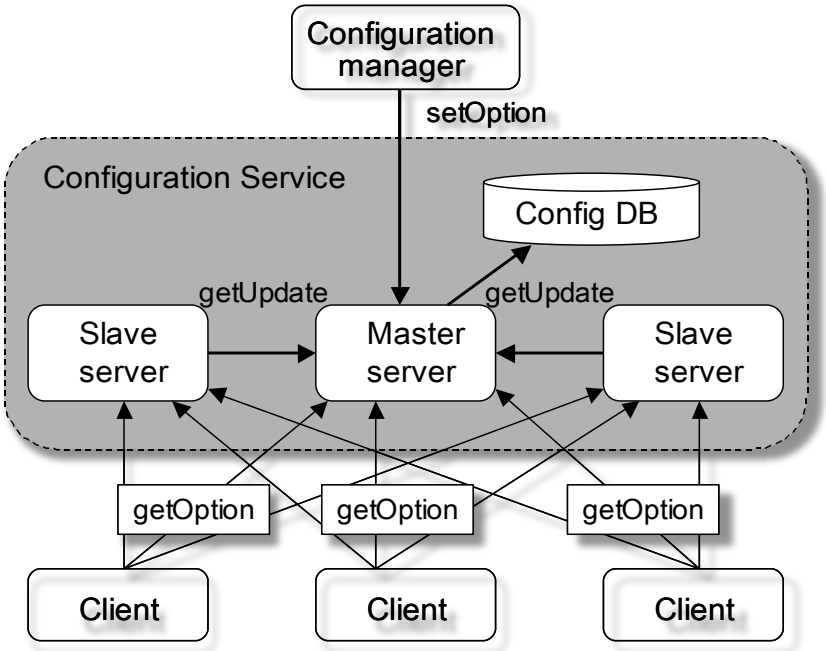
- DIRAC software architecture is based on well defined components with clear recipes for developing
  - *Services*
    - passive components reacting to client request
    - Keep their state in a database
  - *Agents*
    - Light permanently running distributed components, animating the whole system
  - *Clients*
    - Used in user interfaces as well as in agent-service, service-service communications
- All the communications between the distributed components are secure
  - DISET custom client/service protocol
    - Focus on efficiency
    - Control and data transfer communications
  - X509, GSI security standards

Software Framework

- The framework allows to easily build DIRAC components concentrating on the business logic of the applications
  - Starting from basic skeletons
  - Development environment: Python
    - Several non-core Python modules are used, e.g. M2Crypto, SQLAlchemy
- Third party dependencies
  - MySQL
    - Replacement by MariaDB is being tested
  - ElasticSearch DB
    - Activities monitoring, accounting
  - Message Queues (abstraction layer with RabbitMQ implementation)
    - Alternative inter-component protocol
    - Centralized logging

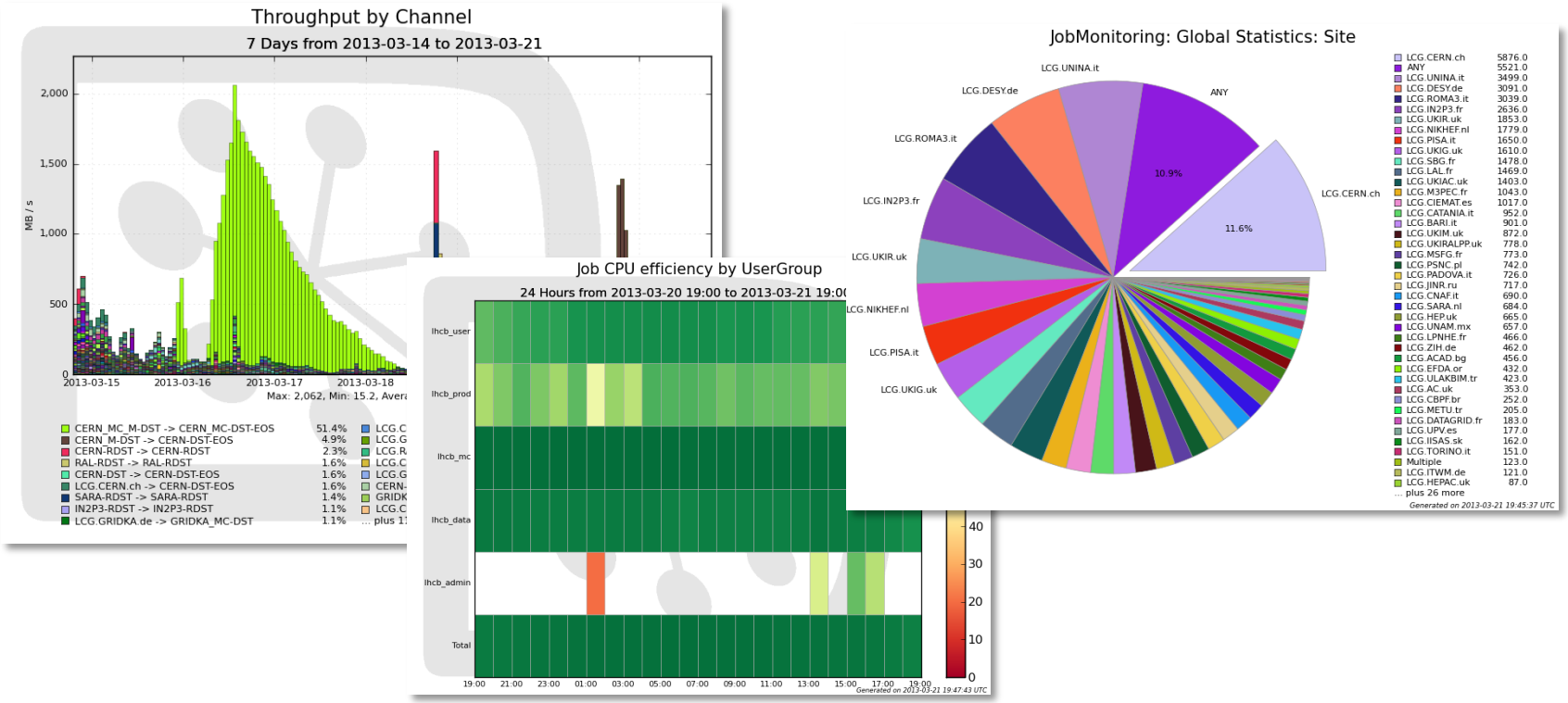
DIRAC base services

- Redundant Configuration Service
  - Provides service discovery and setup parameters for all the DIRAC components
- Full featured proxy management system
  - Proxy storage and renewal mechanism
- System Logging service
  - Collect essential error messages from all the components
- Monitoring service
  - Monitor the service and agents behavior
- Security Logging service
  - Keep traces of all the service access events



Accounting

- Comprehensive accounting of all the operations



- Publication ready quality of the plots
  - Plotting service can be used by users for there own data

Customizing DIRAC

- DIRAC Extensions
  - Specific functionality can be provided as custom components and plugin modules, e.g.
    - Data access policies
    - Job scheduling policies
  - Standard rules for packaging specific components
    - Using standard release and deployment tools
    - Autodiscoverring custom components at run time
      - Possibility to override behavior of core components
  - Multiple extensions are created
    - LHCb, Belle, ILC, BES, CTA, Eiscat, …

Contributing new code

- DIRAC software repository in the Github service
  - https://github.com/DIRACGrid
- Multiple means for efficient collaborative development
  - Strict branching model
  - Review process for each new contribution
  - Automated testing with
    - Multiple unit tests (Travis CI)
    - Continuous integration (Jenkins)
- Automated coding conventions and coverage evaluation
- Automated documentation builds for each new release
- Regular releases
  - Weekly patch releases
  - 3-4 major releases per year

Conclusions

- DIRAC provides a framework for building distributed computing systems aggregating multiple types of computing and storage resources
- Multiple large HEP and astrophysics collaborations adopted DIRAC for their production systems. Multiple evaluations are ongoing
- Multiple multi-community DIRAC services are provided by large grid infrastructures. DIRAC becomes an EGI core service replacing gLite WMS starting from 2018.
- DIRAC software framework facilitates development of extensions to its functionality, some of which are accepted into the core code base

http://diracgrid.org