
The DIRAC interware

current, upcoming and planned capabilities and technologies

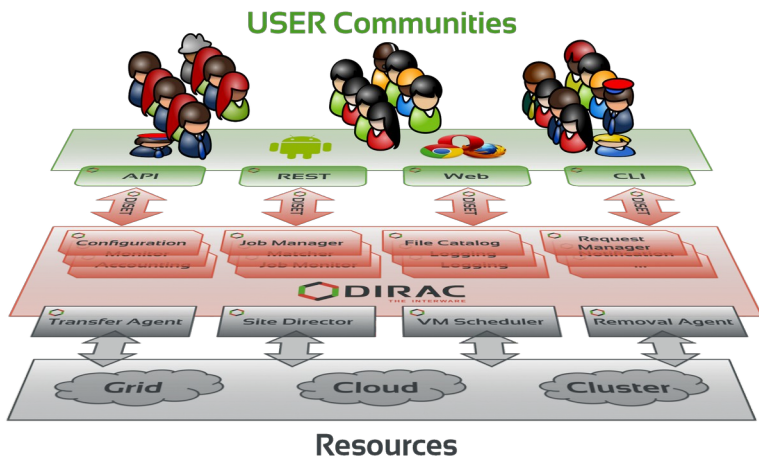


Federico Stagni
DIRAC technical coordinator
on behalf of the DIRAC consortium

federico.stagni@cern.ch

DIRAC: the interware

- A software framework for distributed computing
- A **complete** solution to one (or more) user community
- Builds a layer between users and resources

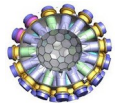


- Started as an LHCb project, experiment-agnostic in 2009
- Developed by communities, for communities
 - Open source (GPL3+), [GitHub](#) hosted
 - Python 2.7 (python 3 in development)
 - No dedicated funding for the development of the “Vanilla” project
 - Publicly [documented](#), active [assistance forum](#), yearly [users workshops](#) (2020 canceled...), open [developers meetings](#) and [hackathons](#)
- The DIRAC consortium as representing body

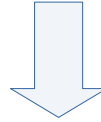


A framework shared by multiple experiments/projects,
both inside HEP, astronomy, and life science

Experiment agnostic
Extensible
Flexible



Jobs and files

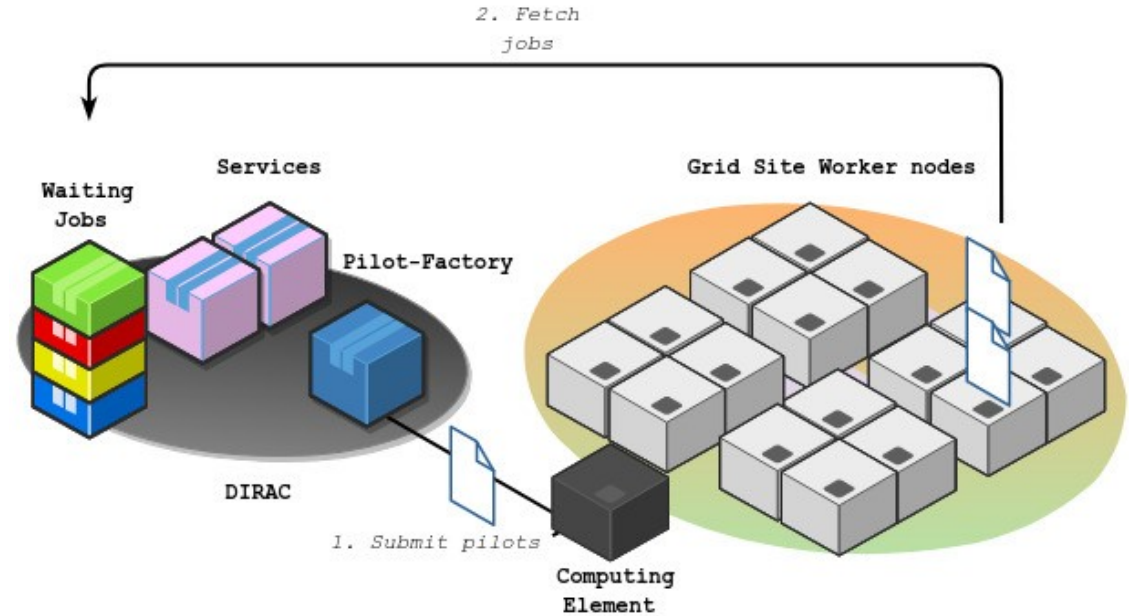


WMS and DMS

Workload and Data Management Systems

[WMS] “typical” GRID operation

WMS is for
managing **single**
jobs



[WMS] resources federation

**Pilots are the
“federators”**

Send them

as “pilot jobs” (via a CE)

Or just Run them!

e.g. as part of the contextualization of a (V)M

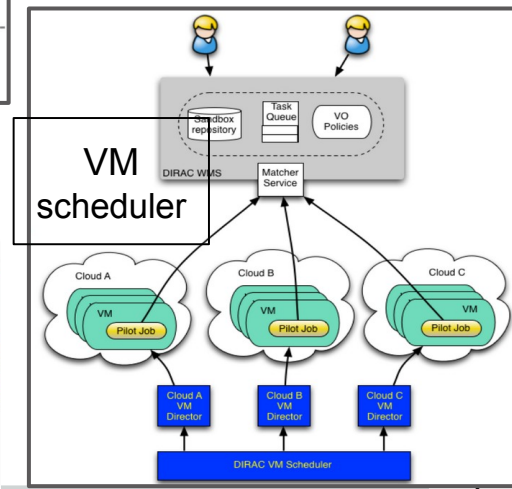
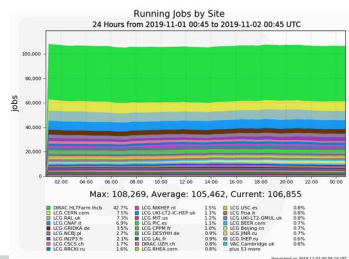
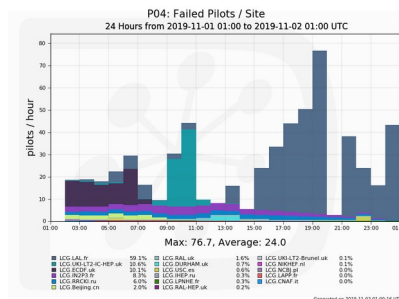
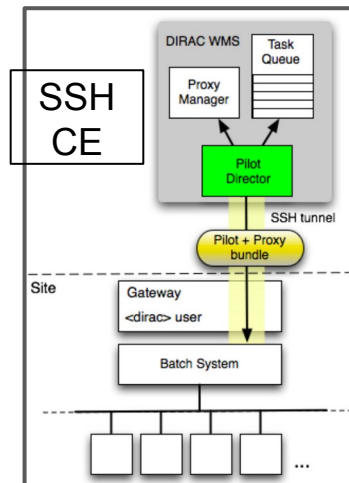
OR

“Make a machine a pilot machine,
and you are done”

Caveat: pilots require Internet access

[WMS] Computing resources

- **Grids** (EGI, OSG, NorduGRID)
 - CREAM, HTCondorCE, ARC
- **Clusters** behind a Batch System
 - access through SSH/GSISSH tunnel
 - a really thin layer that we call “SSH CE”
 - Interfacing with the BS (LSF, BQS, SQL, PBS, Condor)
- **Vacuum** (No CE/BS to start the pilots)
 - VAC/vcycle resources
 - Volunteer resources (e.g. BOINC) with trust gateway
 - HLT farm (LHCb)
- **VMs scheduler**
 - Openstack, Keystone v2 & v3, OpenNebula XML-RPC, Amazon EC2 (boto2), Apache libcloud, rocci cli, OCCI REST
 - Contextualization from standard images
 - with, at least, the DIRAC pilot
- **HPC sites**
 - No catch-all solution
 - OAR, Slurm support

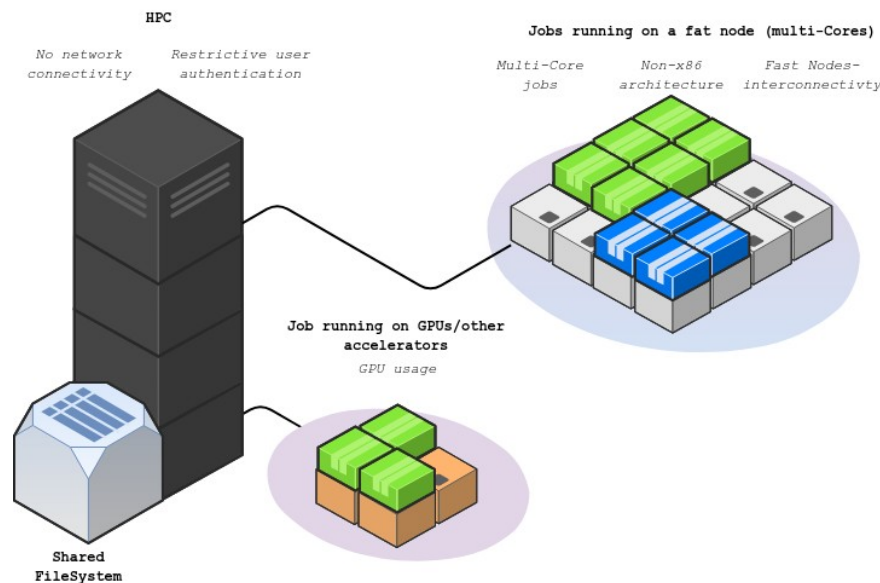


[WMS] The case of HPCs

Non-homogeneous,
"new" challenges, e.g.:

- Fat nodes, e.g. 200+ logical processors per node
 - DIRAC solution: PoolCE
 - In the work: adaptive benchmark
- Absence of CEs
 - DIRAC solution: SSH CE
 - In the work: DIRAC \longleftrightarrow slurm for multi-fat nodes allocation
- Limited connectivity with login node
 - DIRAC solution: local pilot factory
- Absence of network connectivity from the Wns
 - No DIRAC solution yet
- Absence of CVMFS (but CVMFS is not a DIRAC requirement)
- Non-x86 architectures

For HPCs we might be in situations where:



BASICS of DMS (managing single files):

- **LFNs**: unique identifier within DIRAC of a file

- **Logical File Name**
- (described as paths)

- LFNs are registered in **catalog(s)**

- and there are implementations like the **DFC (DIRAC File Catalog)**
- → and you can connect as many catalogs as you want
 - e.g. the LFC, and **Rucio** Catalog plugin is coming

- LFNs *may* have **PFNs**, stored in **SEs**.

- **Physical File Name on Storage Elements**
- (and SEs are monitored, within the DIRAC Resource Status System)

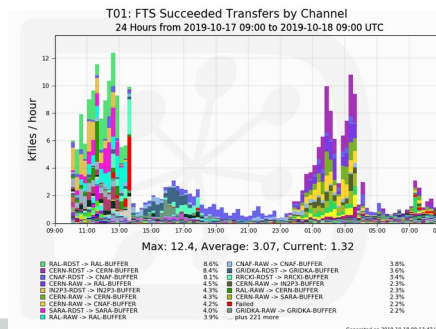
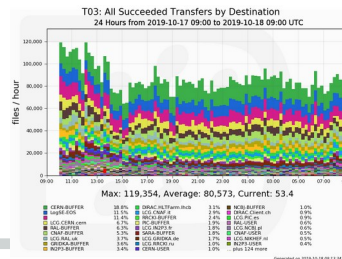
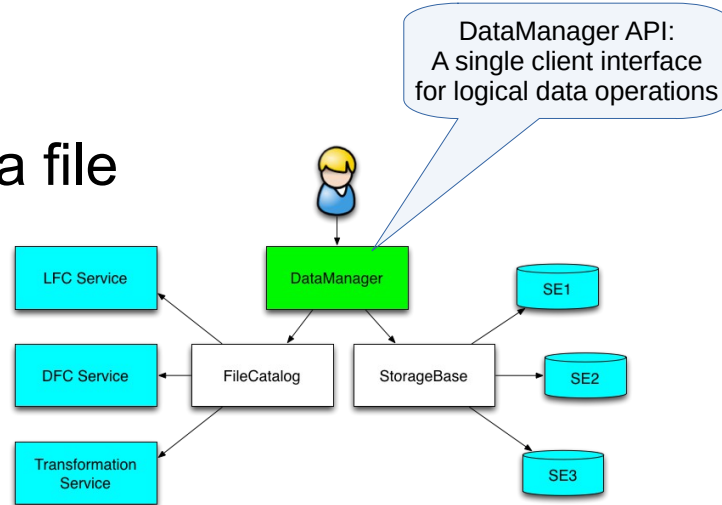
- You can access those PFNs with several

protocols (and can also be brought online - i.e. staged)

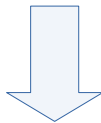
- e.g. root, gsiftp, srm, http, WebDav, rfio, dcap, dip, S3.
- Using **gfal2** library

- Use **FTS** for transfers

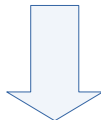
- DIRAC \longleftrightarrow FTS interface



Grouping jobs and files



Productions and datasets

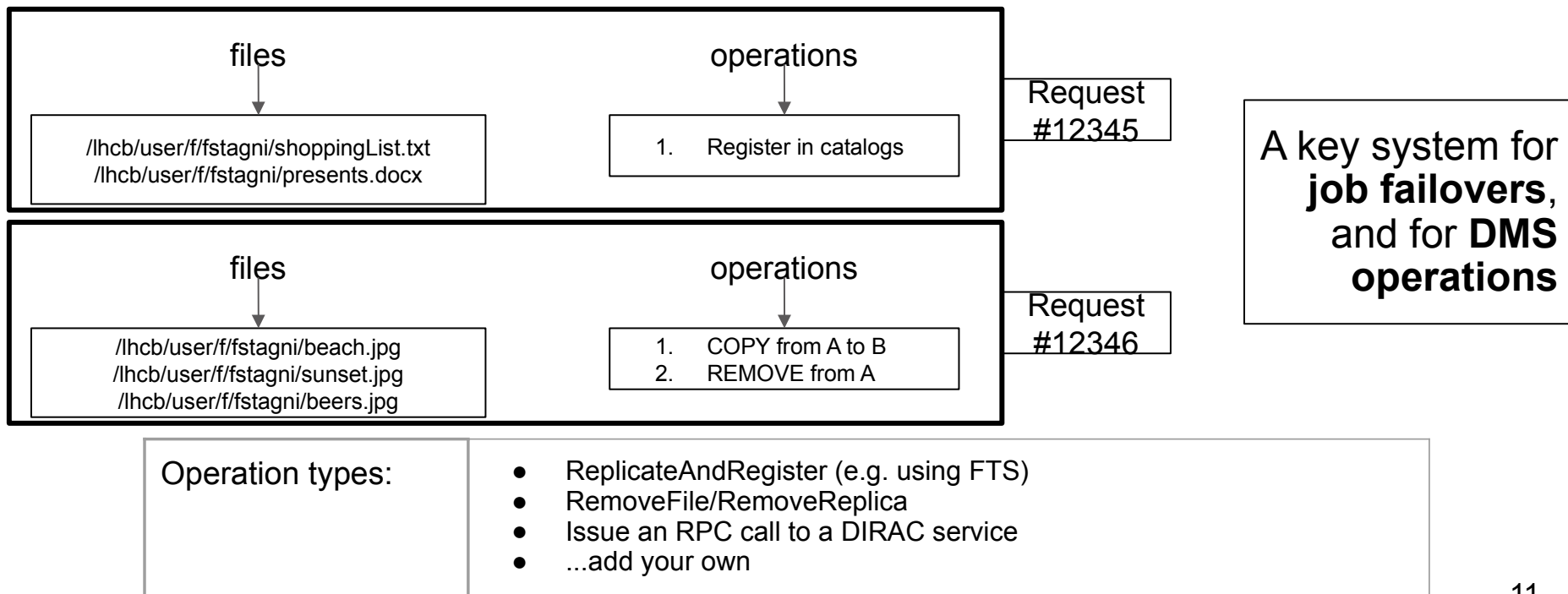


RMS and TS

Request Management and Transformation Systems

Request management system

A generic, flexible system, which can be used for queueing *operations* (on files, but not only)
like a to-do list



A generic system for queueing similar *operation types* on certain *datasets* and forward them to the appropriate *systems*

An *operation type* can be, e.g.:

- a simulation workflow
- a reconstruction workflow
- a replication
- a removal
- ...

A *dataset* is split into groups, based on criterias defined by *plugins*, e.g.:

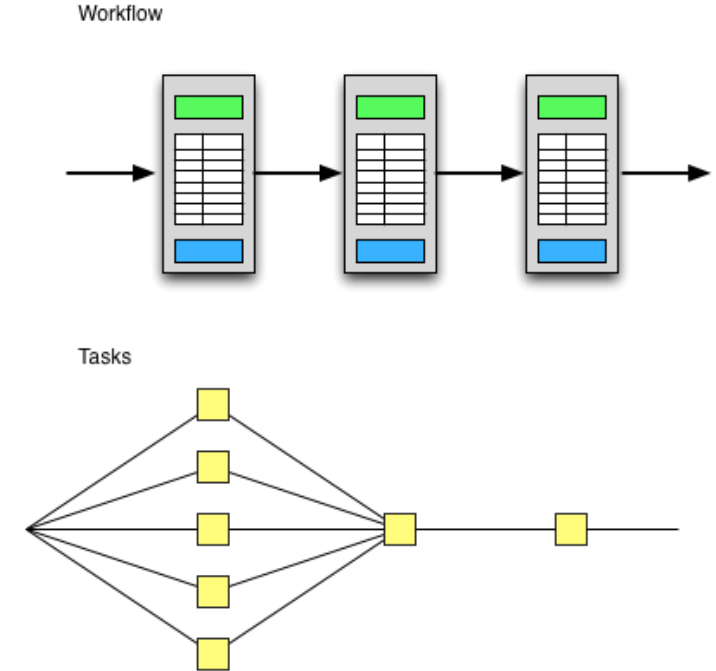
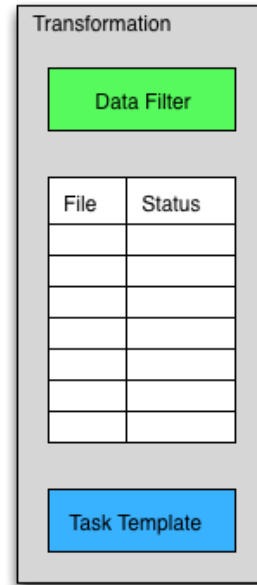
- split by size
- by destination
- by metadata
- ... [code it]

A *system* is either (today) the DIRAC WMS (for productions) or the DIRAC RMS (for dataset management operation types)

[DMS] example (for dataset management): Take all my holidays pictures from 2018 with tag='sunset', make sure that there is one copy on tape and one on disk, distributed on all the sites according to free space, and group the operations by group of at most 100 files.

[WMS] example (for jobs productions): Take all my holidays pictures from 2018 with tag='sunset', make sure to run (only once) the 'red-enhancer' workflow on each one of them, using only Tier2 sites.

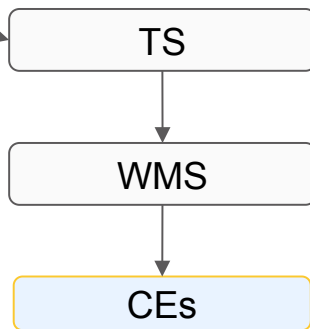
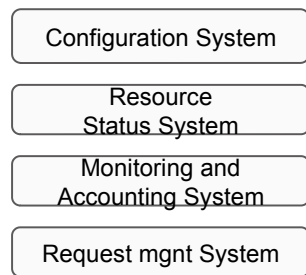
- In summary, Transformations can be used for automatic data driven bulk operations
- Scaling to millions of files



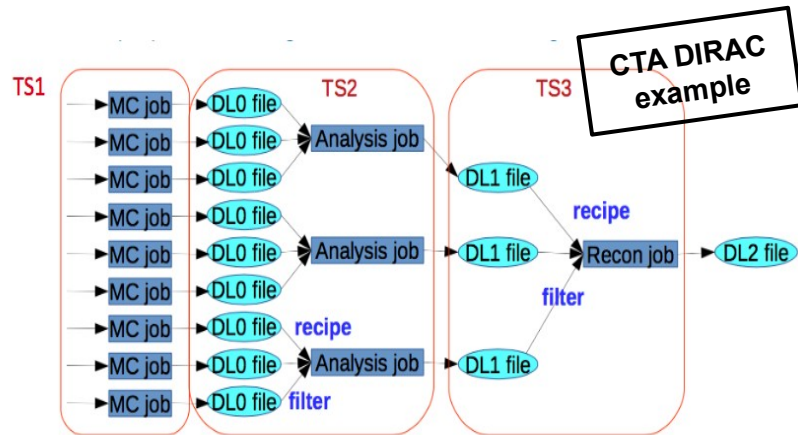
[PMS] Productions management

DIRAC Transformation System + WMS: for single productions

(Transformation APIs)



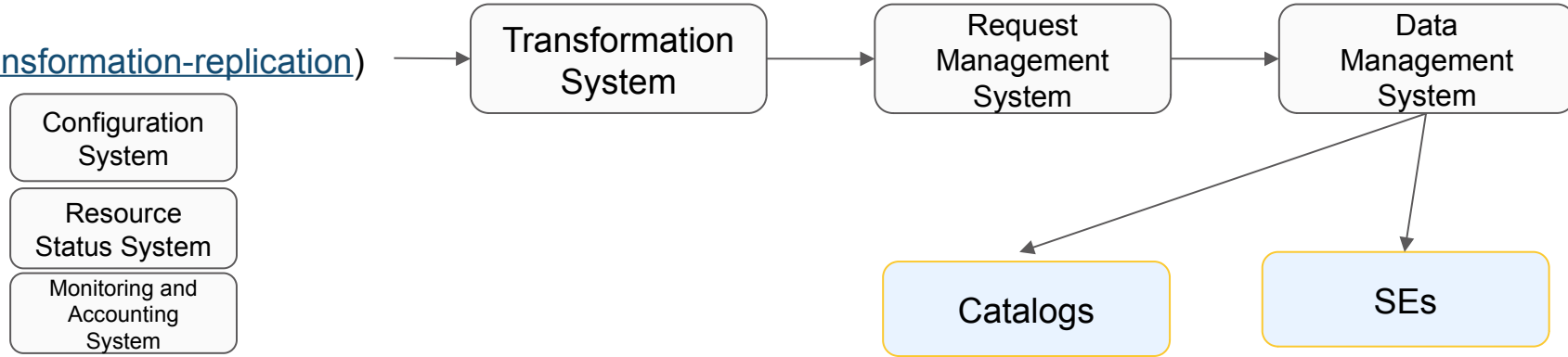
DIRAC Productions System: chaining job productions together



Job										Summary										Details										History									
Job ID	Job Name	Job Type	Job Status	Job Priority	Job Age	Job Size	Job Time	Job Cost	Job CPU	Job Mem	Job Disk	Job Net	Job I/O	Job Error	Job Exit	Job Reason	Job Comment	Job ID	Job Name	Job Type	Job Status	Job Priority	Job Age	Job Size	Job Time	Job Cost	Job CPU	Job Mem	Job Disk	Job Net	Job I/O	Job Error	Job Exit	Job Reason	Job Comment				
1000000000	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000000	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0				
1000000001	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000001	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000002	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000002	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000003	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000003	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000004	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000004	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000005	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000005	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000006	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000006	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000007	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000007	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000008	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000008	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000009	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000009	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000010	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000010	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000011	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000011	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000012	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000012	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000013	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000013	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000014	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000014	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000015	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000015	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000016	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000016	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000017	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000017	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000018	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000018	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000019	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000019	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000020	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000020	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000021	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000021	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000022	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000022	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000023	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000023	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000024	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000024	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000025	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000025	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000026	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000026	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000027	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000027	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000028	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000028	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000029	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000029	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000030	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000030	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000031	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000031	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000032	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000032	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000033	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000033	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000034	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000034	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000035	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000035	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000036	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000036	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000037	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000037	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000038	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000038	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000039	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000039	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000040	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000040	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000041	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000041	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000042	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000042	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000043	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000043	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000044	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	1000000044	MC job	MC	Active	100	0	1000000000	0	0	0	0	0	0	0	0	0	0	0	0			
1000000045	MC job																																						

[DMS] Dataset management

([dirac-transformation-replication](#))



Selections: [Active] [Completed] [Idle] [Clearing]

Agent Type: [All] [Production] [Request] [Replication]

Group: [All] [Production] [Request] [Replication]

Plugin: [All] [Production] [Request] [Replication]

Time Span: [From:] [To:]

Production(DTs): []

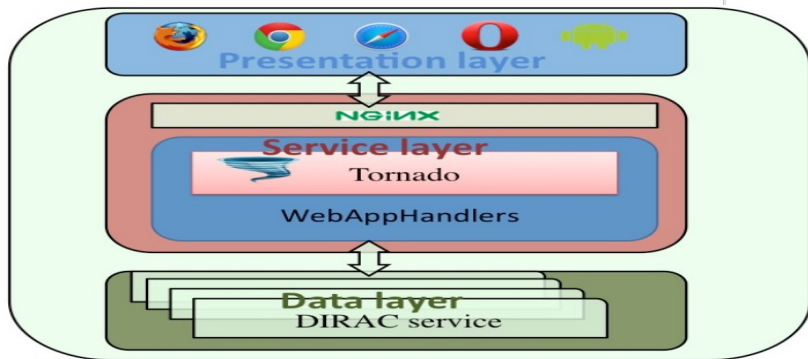
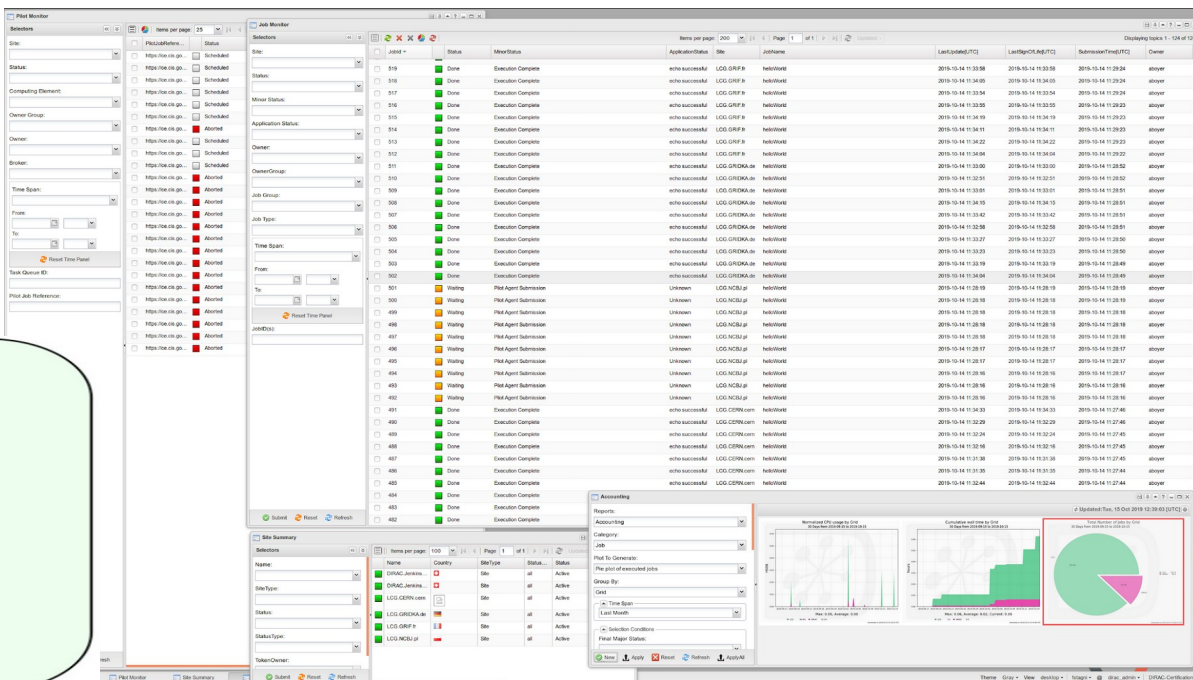
Request(DTs): []

Show not productions only: []

ID	Status	Agent Type	Type	Name	Files	Processed (%)	Created	Total Created	Submitted	Matched	Checking	Waiting	Staging	Rescheduled	Killed	Running	Scheduled	Done	Completed	Failed
97270	Active	Automatic	Replication	Replication...	186670	57.2	0	186670	0	0	0	0	0	0	0	0	811	1078	0	0
95031	Active	Automatic	Replication	Replication...	388731	99.1	0	388731	0	0	0	0	0	0	0	0	33	3909	0	2
95037	Idle	Automatic	Replication	Replication...	25	100.0	0	14	0	0	0	0	0	0	0	0	0	14	0	0
94872	Idle	Automatic	Replication	Replication...	216	100.0	0	13	0	0	0	0	0	0	0	0	0	10	0	3
93883	Idle	Automatic	Replication	Replication...	2216	100.0	0	945	0	0	0	0	0	0	0	0	0	944	0	2
93724	Idle	Automatic	Replication	Replication...	6	100.0	0	6	0	0	0	0	0	0	0	0	0	6	0	0
95913	Idle	Automatic	Replication	Replication...	6	100.0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
95912	Idle	Automatic	Replication	Replication...	18	100.0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
95911	Idle	Automatic	Replication	Replication...	999	100.0	0	10	0	0	0	0	0	0	0	0	0	10	0	0
95910	Idle	Automatic	Replication	Replication...	914	100.0	0	10	0	0	0	0	0	0	0	0	0	10	0	0
93035	Idle	Automatic	Replication	Replication...	170	100.0	0	2	0	0	0	0	0	0	0	0	0	2	0	0
97450	Com...	Manual	Replication	Replication...	322	100.0	0	7	0	0	0	0	0	0	0	0	0	7	0	0
97479	Com...	Manual	Replication	Replication...	1338	100.0	0	14	0	0	0	0	0	0	0	0	0	14	0	0
97478	Com...	Manual	Replication	Replication...	16	100.0	0	2	0	0	0	0	0	0	0	0	0	2	0	0
97201	Idle	Automatic	Replication	Replication...	16	100.0	0	7	0	0	0	0	0	0	0	0	0	7	0	0
97200	Active	Automatic	Replication	Replication...	93	92.4	0	78	0	0	1	0	0	0	0	0	6	71	0	0
97289	Active	Automatic	Replication	Replication...	3074	99.3	0	506	1	0	0	0	0	0	0	0	3	502	0	0
97288	Active	Automatic	Replication	Replication...	11659	99.9	0	141	0	0	0	0	0	0	0	0	1	140	0	0
97287	Idle	Automatic	Replication	Replication...	15	100.0	0	2	0	0	0	0	0	0	0	0	2	0	0	0
97286	Idle	Automatic	Replication	Replication...	31	100.0	0	7	0	0	0	0	0	0	0	0	7	0	0	0
97285	Active	Automatic	Replication	Replication...	1302	99.9	0	198	1	0	0	0	0	0	0	0	1	196	0	0
97284	Idle	Automatic	Replication	Replication...	541	100.0	0	9	0	0	0	0	0	0	0	0	0	9	0	0
97283	Active	Automatic	Replication	Replication...	1760	97.2	0	543	9	0	0	0	0	0	0	0	42	492	0	0
97282	Idle	Automatic	Replication	Replication...	349	100.0	0	16	0	0	0	0	0	0	0	0	0	16	0	0
97281	Idle	Automatic	Replication	Replication...	771	100.0	0	70	0	0	0	0	0	0	0	0	0	70	0	0
97280	Idle	Automatic	Replication	Replication...	185	100.0	0	4	0	0	0	0	0	0	0	0	0	4	0	0
97279	Idle	Automatic	Replication	Replication...	738	100.0	0	23	0	0	0	0	0	0	0	0	0	23	0	0
97278	Active	Automatic	Replication	Replication...	21341	99.9	0	506	1	0	0	0	0	0	0	0	4	501	0	0
97277	Idle	Automatic	Replication	Replication...	40	100.0	0	19	0	0	0	0	0	0	0	0	0	19	0	0
97276	Idle	Automatic	Replication	Replication...	264	100.0	0	15	0	0	0	0	0	0	0	0	0	15	0	0
97275	Idle	Automatic	Replication	Replication...	179	100.0	0	22	0	0	0	0	0	0	0	0	0	22	0	0
97274	Active	Automatic	Replication	Replication...	662	99.5	0	23	0	0	0	0	0	0	0	0	2	27	0	0
97273	Idle	Automatic	Replication	Replication...	101	100.0	0	11	0	0	0	0	0	0	0	0	0	11	0	0
97272	Active	Automatic	Replication	Replication...	56.8	100.0	0	14	0	0	0	0	0	0	0	0	1	13	0	0
97271	Idle	Automatic	Replication	Replication...	603	100.0	0	26	0	0	0	0	0	0	0	0	0	26	0	0
96259	Active	Automatic	Replication	Replication...	656	99.8	0	245	0	0	0	0	0	0	0	0	1	244	0	0
95770	Idle	Automatic	Replication	Replication...	392	100.0	0	255	0	0	0	0	0	0	0	0	0	255	0	0

- Web users' interface
- Frontend: ExtJS6
- Backend: tornado, NGINX

- Each system has its own Web application

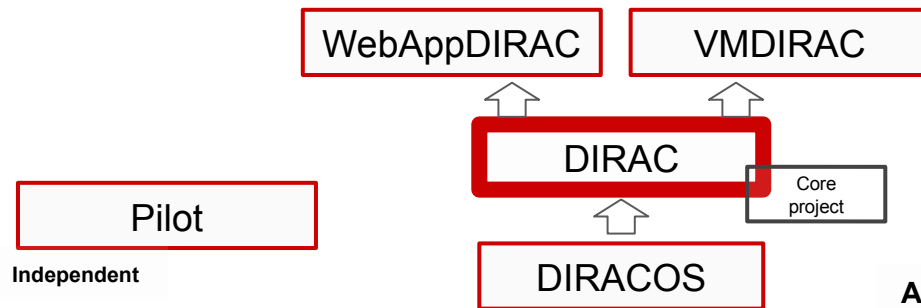
The screenshot displays the DIRAC WebApp interface, which is divided into several panels. The top panel shows a 'Job Monitor' view with a table of jobs, including columns for Name, Country, Site, Status, and Last Update. Below this, there are panels for 'Job Summary' and 'Accounting'. The 'Job Summary' panel shows a list of jobs with filters for Name, Site, and Status. The 'Accounting' panel shows a bar chart of job counts over time and a pie chart of job distribution by site. The interface is designed for monitoring and managing DIRAC jobs.

Experiment agnostic, and extensible

“Horizontal”
extensibility

-

For specific
requirements



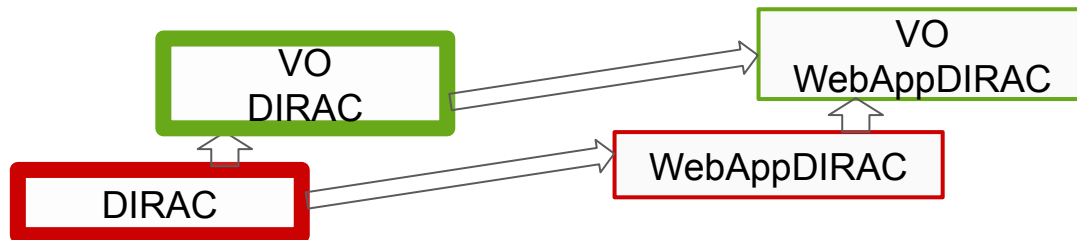
Each project is
independently
versioned

A DIRAC release is composed
by all the projects (strong
dependency)

“Vertical”
extensibility

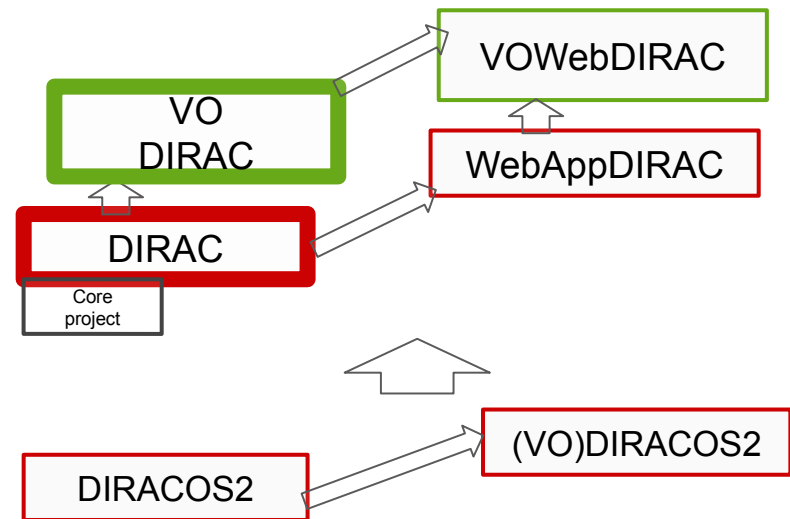
-

Community driven



Recent and ongoing developments

- dips:// → https://
 - dips: DIRAC proprietary protocol for RPC calls (“diset” protocol)
 - http: frameworks already exists in python 2&3 for server-side (tornado) and client side (requests)
 - DIRAC v7r2 will be the first release with the possibility to use https instead of dips → not for all the services, requires activation
- Python 3
 - DIRAC v7r2 will be the first release with a functional Python3 client (default will still be Python 2.7.13)
- DIRAC ↔ Rucio bridge
 - Led by Belle2. Advanced prototype for the Rucio Catalog plugin.



- Will replace DIRACOS
 - Conda-forge based
 - CentOS8 + Python3 support
 - VO-driven extensions would be easy
-
- Server: x86_64 only
 - Possibly support client installations for ARM, POWER

Until “yesterday”: X509 certificates,
DIRAC groups, proxies, VOMS

- DIRAC can delegate AuthN to an external server
 - ensure provisioning of X509 certificate proxies
- Focus: OAuth/OIDC as “industry standards”
 - Use case: [EGI Check-in](#) SSO hub





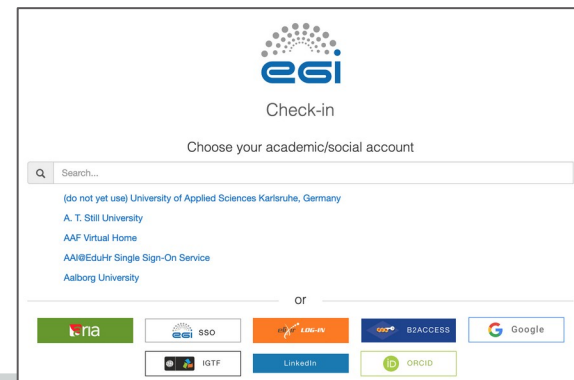
Web portal authentication

CLI authentication

```
[dirac@ca-emi pro]$ python DIRAC/FrameworkSystem/scripts/dirac-proxy-init.py -O CheckIn -g training_user -q
OAuth authentication from CheckIn.
Use link to authentication..
https://ce-emi.bltp.kiev.ua:9943/oauth2/oauth?getlink=MZ7Xn04lyMYTz9Vw2wkpBbHrn30z8F

Waiting 3.0 minutes when you authenticated.. ... [3-]

Proxy generated:
subject      : /DC=org/DC=ugrid/O=people/O=B1TP/CN=Andrey Litovchenko/CN=3461819742
issuer       : /DC=org/DC=ugrid/O=people/O=B1TP/CN=Andrey Litovchenko
identity     : /DC=org/DC=ugrid/O=people/O=B1TP/CN=Andrey Litovchenko
timeleft     : 23:59:59
DIRAC group  : training_user
rfc          : True
path         : /tmp/x509up_u3318
username     : alltov
```



Development and testing

```

Integration tests / Integration (5.7, sl6)
successful 2 hours ago in 16m 7s

- [x] Set up job
- [x] Run actions checks on CI
- [x] Clean up GitHub actions environment
- [x] Prepare environment
- [x] Install server
- [x] Install client
- [x] Server tests

ServerInstallDIRAC/tests/Integration/ServerInstallDIRAC.py: ElasticTestChain: test_querySimple PASSED [100%]

##### 13 passed in 44.92 seconds #####
*** Thu Oct 17 08:49:09 UTC 2019 *** FRAMEWORK TESTS (partially skipped) ***

WARNING: assertions not in test modules or plugins will be ignored because assert statements are not executed by the underlying Python interpreter (are you using python -d?)
#####
platform linux2 -- python 2.7.13, pytest-4.6.6, py-1.8.6, pluggy-0.13.6 -- /home/DIRAC/ServerInstallDIRAC/DiracCore/bin/python
cachedir: .pytest_cache
hypothesis profile 'default' -- database: directorybasedexampledatabase('/home/DIRAC/hypothesis/examples')
rootdir: /home/DIRAC/ServerInstallDIRAC/tests, inifile: pytest.ini
plugins: mock-1.11.1, cov-2.8.1, hypothesis-4.11.1
collecting ... collected 4 items

ServerInstallDIRAC/tests/Integration/Framework
Test_InstallComponentDB.py: ComponentMonitoringClientChain: testComponents PASSED [ 25%]
ServerInstallDIRAC/tests/Integration/Framework
Test_InstallComponentDB.py: ComponentMonitoringClientChain: testMostLogging PASSED [ 50%]
ServerInstallDIRAC/tests/Integration/Framework
Test_InstallComponentDB.py: ComponentMonitoringClientChain: testMost PASSED [ 75%]
ServerInstallDIRAC/tests/Integration/Framework
Test_InstallComponentDB.py: ComponentMonitoringClientChain: testInstallations PASSED [100%]

##### 4 passed in 2.22 seconds #####

- [x] Client tests

ClientInstallDIRAC/tests/Integration/RequestManagementSystem
Test_ClientReq.py: ReqClientMgt: testFullChain PASSED [ 50%]
ClientInstallDIRAC/tests/Integration/RequestManagementSystem
Test_ClientReq.py: ReqClientMgt: testAuthorization PASSED [100%]

##### 2 passed in 8.71 seconds #####
*** Thu Oct 17 08:52:09 UTC 2019 *** RDS TESTS ***

WARNING: assertions not in test modules or plugins will be ignored because assert statements are not executed by the underlying Python interpreter (are you using python -d?)
#####
platform linux2 -- python 2.7.13, pytest-4.6.6, py-1.8.6, pluggy-0.13.6 -- /home/DIRAC/ClientInstallDIRAC/DiracCore/bin/python
cachedir: .pytest_cache
hypothesis profile 'default' -- database: directorybasedexampledatabase('/home/DIRAC/hypothesis/examples')
rootdir: /home/DIRAC/ClientInstallDIRAC/tests, inifile: pytest.ini
plugins: mock-1.10.4, cov-2.7.1, hypothesis-4.30.2
collecting ... collected 8 items

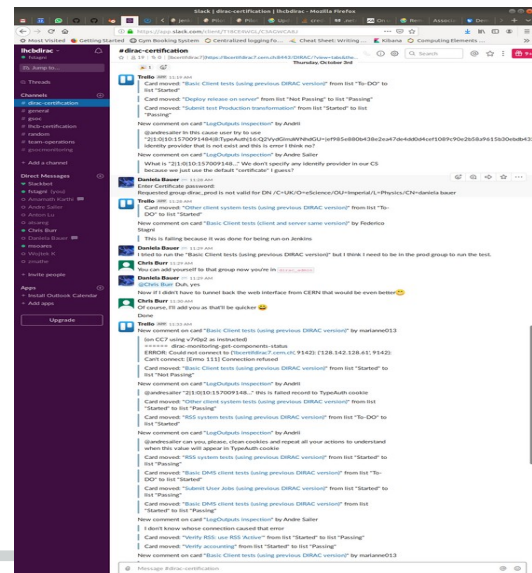
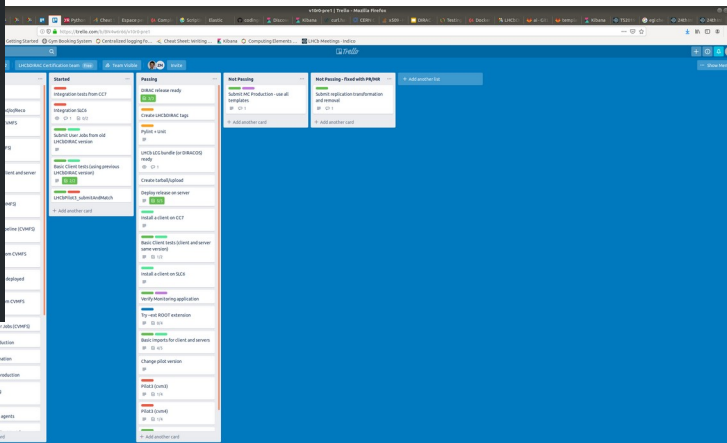
ClientInstallDIRAC/tests/Integration/ResourceStatusSystem
Test_ResourceManagement.py: ResourceManagementClientChain: test_AccountingCache PASSED [ 12%]
ClientInstallDIRAC/tests/Integration/ResourceStatusSystem
Test_ResourceManagement.py: ResourceManagementClientChain: test_DownloadCache PASSED [ 25%]
ClientInstallDIRAC/tests/Integration/ResourceStatusSystem
Test_ResourceManagement.py: ResourceManagementClientChain: test_SGUSTestCache PASSED [ 37%]
ClientInstallDIRAC/tests/Integration/ResourceStatusSystem
Test_ResourceManagement.py: ResourceManagementClientChain: test_JobCache PASSED [ 50%]

- [x] Check test slowness
- [x] Complete job
  
```

~5/6 FTE as core developers, a dozen contributing developers

Tests, certification, integration process is a daily work.

- We use GitHub Actions, GitLab CI/CD, Jenkins...
- We run certification hackathons

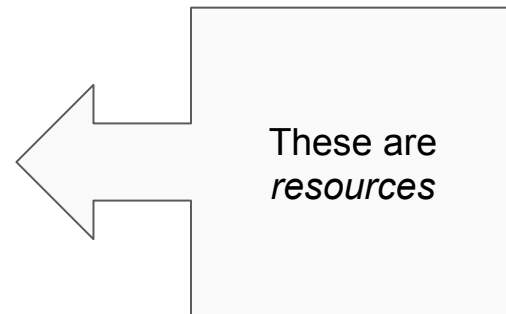


- diracgrid.org
- dirac.readthedocs.io
 - including [code documentation](#)
- Ops and general questions: Google [forum](#)
- Dev and DevOps issues: on [github](#)
- Bi-weekly developers meetings (and/or hackathons):
[BILD](#)

?

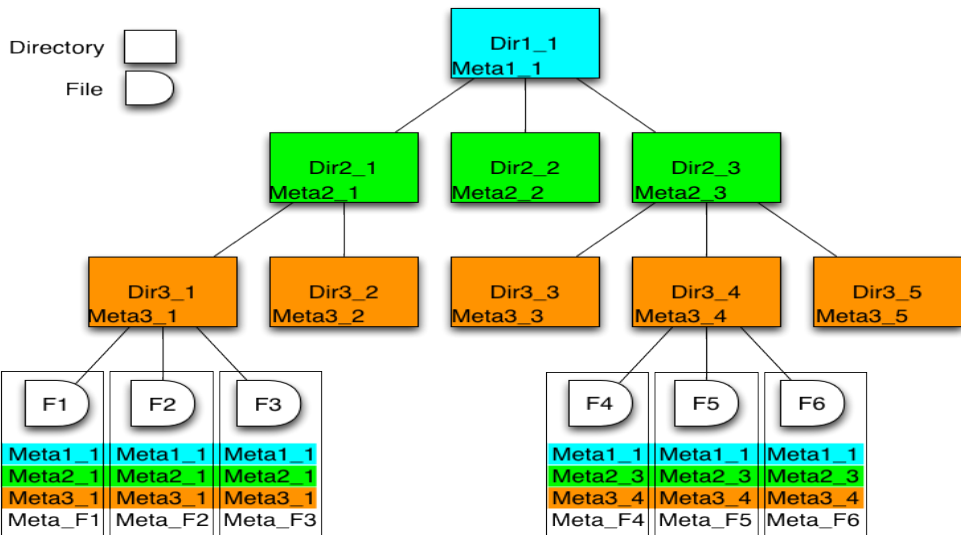
... a few examples of what DIRAC can be used for

- sending jobs to “the Grid”
 - the obvious one...
 - interfacing with different *sites*
 - with different *computing elements*
 - and *batch systems*
 - with different *storage elements*
 - interfacing with different *information systems*
 - interfacing with different *catalogs*
 - interfacing with different *MQs, DBs*
 - authenticate through different *providers*
-
- managing “productions” (e.g. reconstruction, simulation...)
 - managing dataset transfers
 - and removals...
 - providing a failover system
 - your jobs won’t fail because a certain SE is down, nor because of central service are down
 - transfer data from the experiment to a Grid SE
 - monitor your resources with a policy-based system
 - ... and more

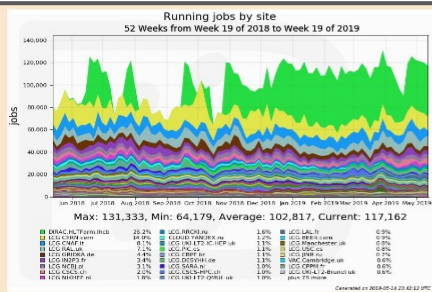


- **Computing**
 - **CEs**: ARC, CREAM, HTCondor, “SSH” for standalone BS, ...
 - **Batch**: LSF, BQS, SGE, PBS/Torque, SLURM, Condor,...
 - Clouds, BOINC, HPC, “desktops”
- **Storage**
 - SRM2, GSIFTP, XRoot, http, DIPs, ...
 - EOS, Castor, DPM, dCache, StoRM, ECHO, CTA, ...
- **Catalog**
 - DIRAC FC, LFC, (Rucio), [LHCb Bookkeeping], ...
- **Information services**
 - BDII, GOCDB, (CRIC)...
- **IdProviders**
- **ProxyProviders**
 - VOMS, PUSP, (OAuth2), ...
- **DBs, MQs, LogBackends**
 - MySQL, ElasticSearch, [Oracle]
 - stomp → ActiveMQ, RabbitMQ
 - file, MQ, ES
 - and logs centralization is easy to set up

- 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

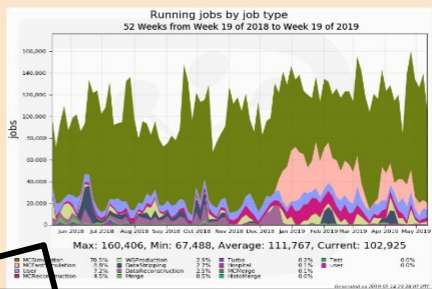
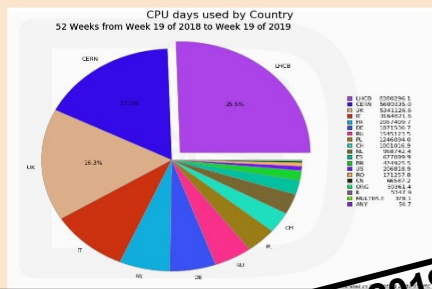


- Example query:
 - `find /lhcb/mcdata LastAccess < 01-01-2012 GaussVersion=v1,v2 SE=IN2P3,CERN Name=*.raw`
- Result of file search is a precise list of corresponding files
 - Unlike Google index



LHCb activities in the last year

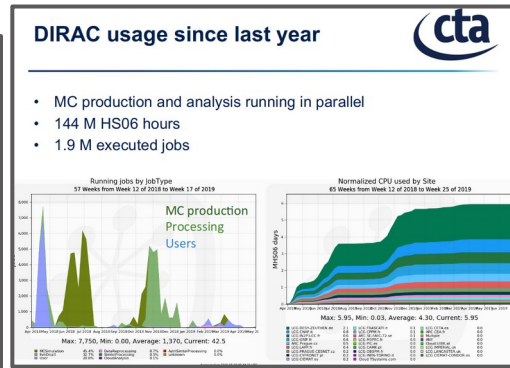
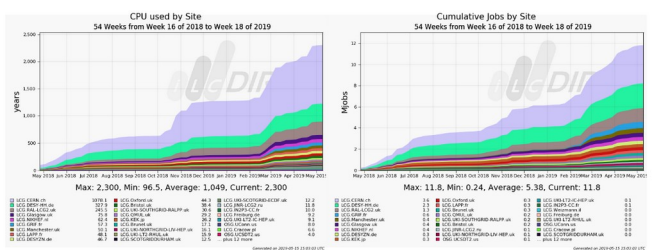
- HLTfarm works even during data taking
- LHCb 25%
- CERN
- UK, IT, FR, DE
- RU, PL, CH,



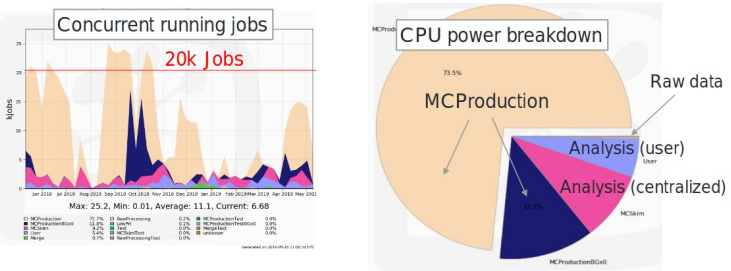
- MCSimulation
- MCFastSimulation
- User
-

From 2019 DIRAC users' workshop

Usage

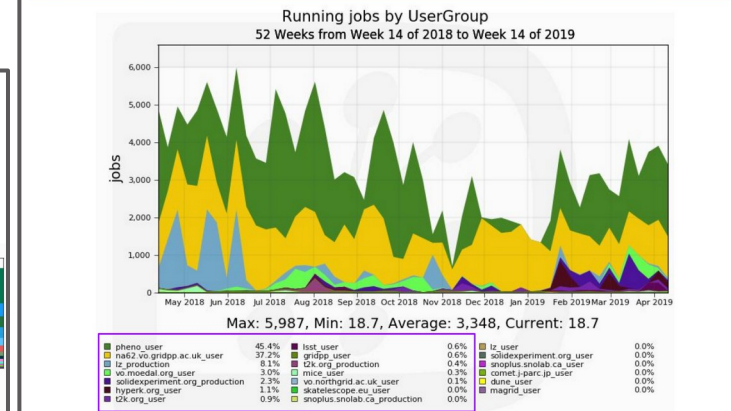


Belle II computing performance in a year



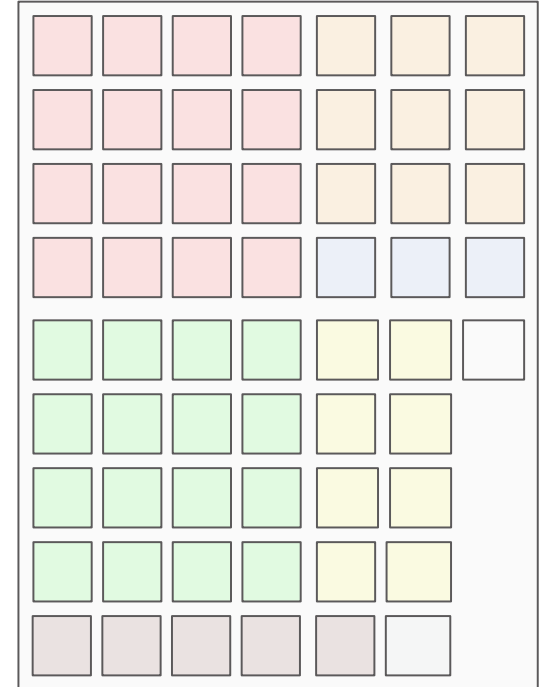
- Consumed CPU power is not so different from last year
- Resource usage is getting practical
 - Increasing analysis jobs
 - Increasing raw (beam and cosmic ray) data processing

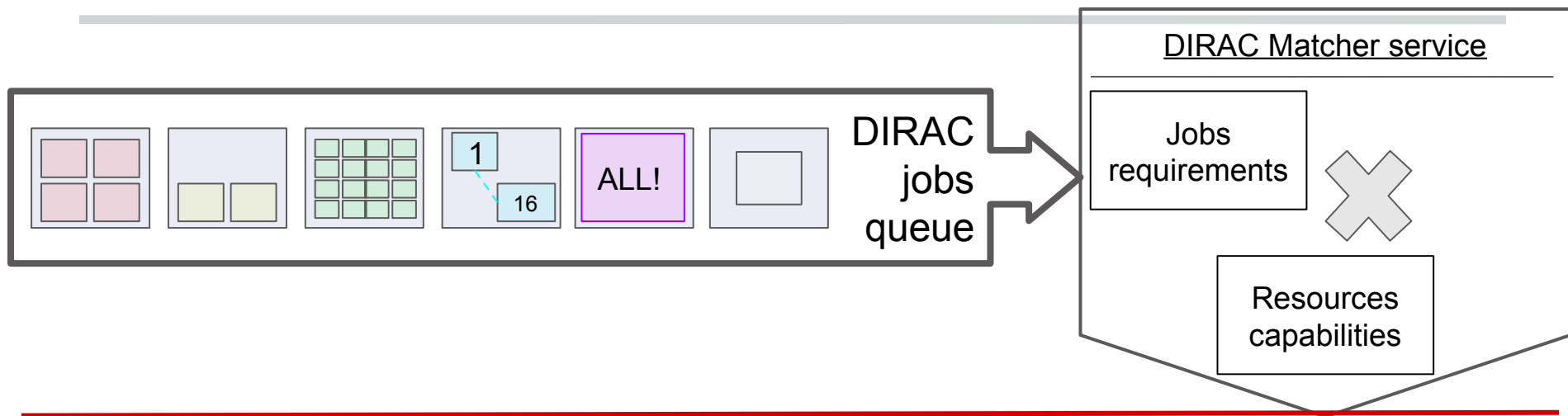
Business as usual - just more of it (Success!)



The case of fat nodes

- Exploiting many-core architectures
 - [LHCb has a case for running on nodes with 272 logical processors (CINECA)]
- DIRAC needs to “partition” the node for optimal memory and throughput (and maybe only use a subset of the logical processors)
 - Use DIRAC “Pool”, an “[inner Computing Element](#)”
 - Parallel jobs matching





Resources (1 pilot per box)

