

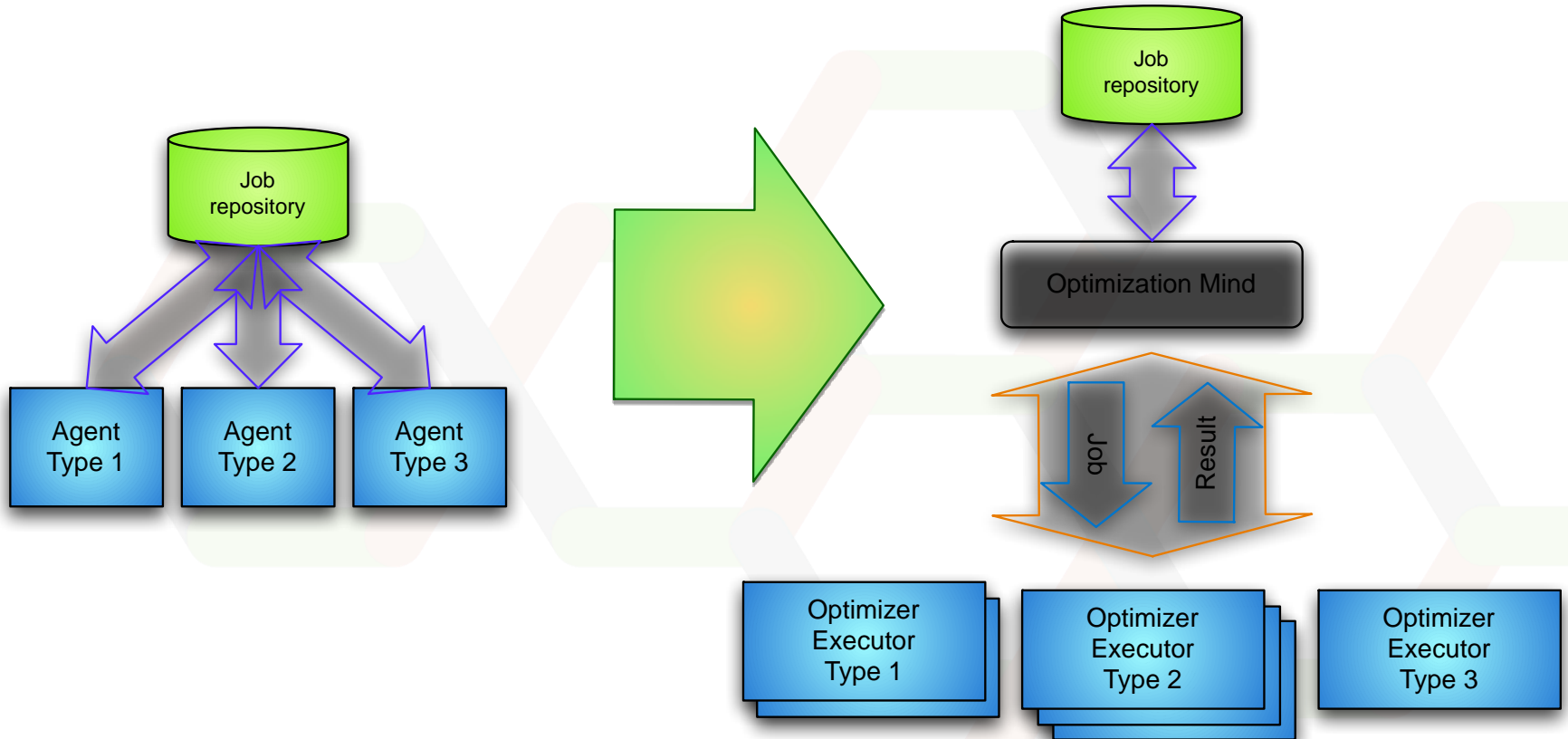
Core, Framework and outlook

4th DIRAC User Workshop
20140526-...29
CERN

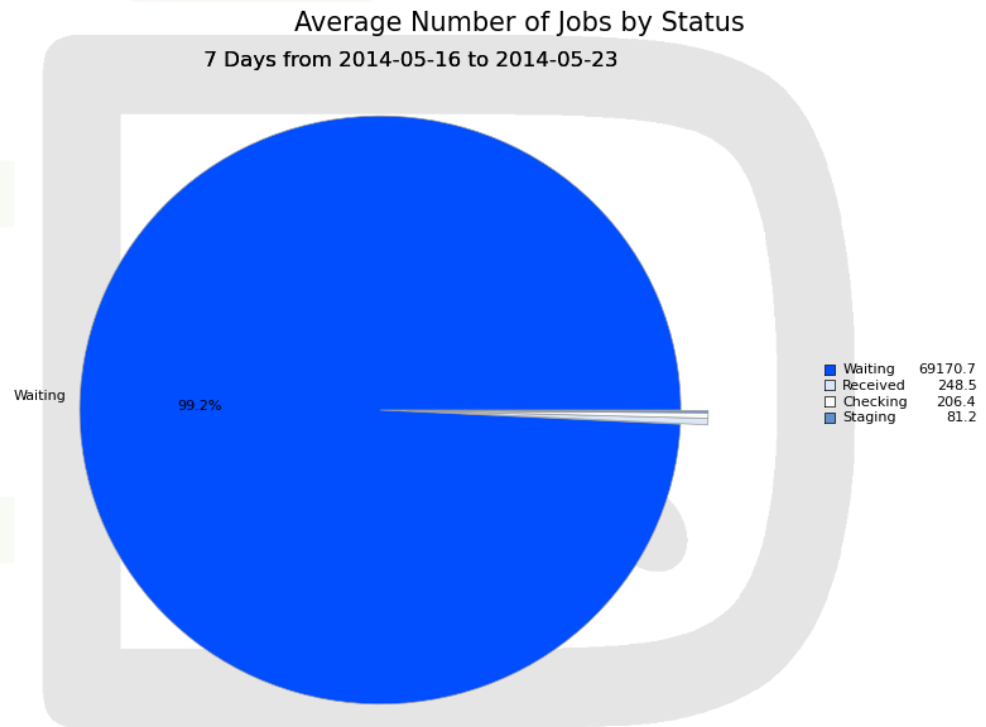
- ▶ WMS stuff
- ▶ Accounting & Monitoring
- ▶ New web framework
- ▶ Bits and pieces

So, as I was saying (last year)...

- ▶ Executors were there but not yet deployed



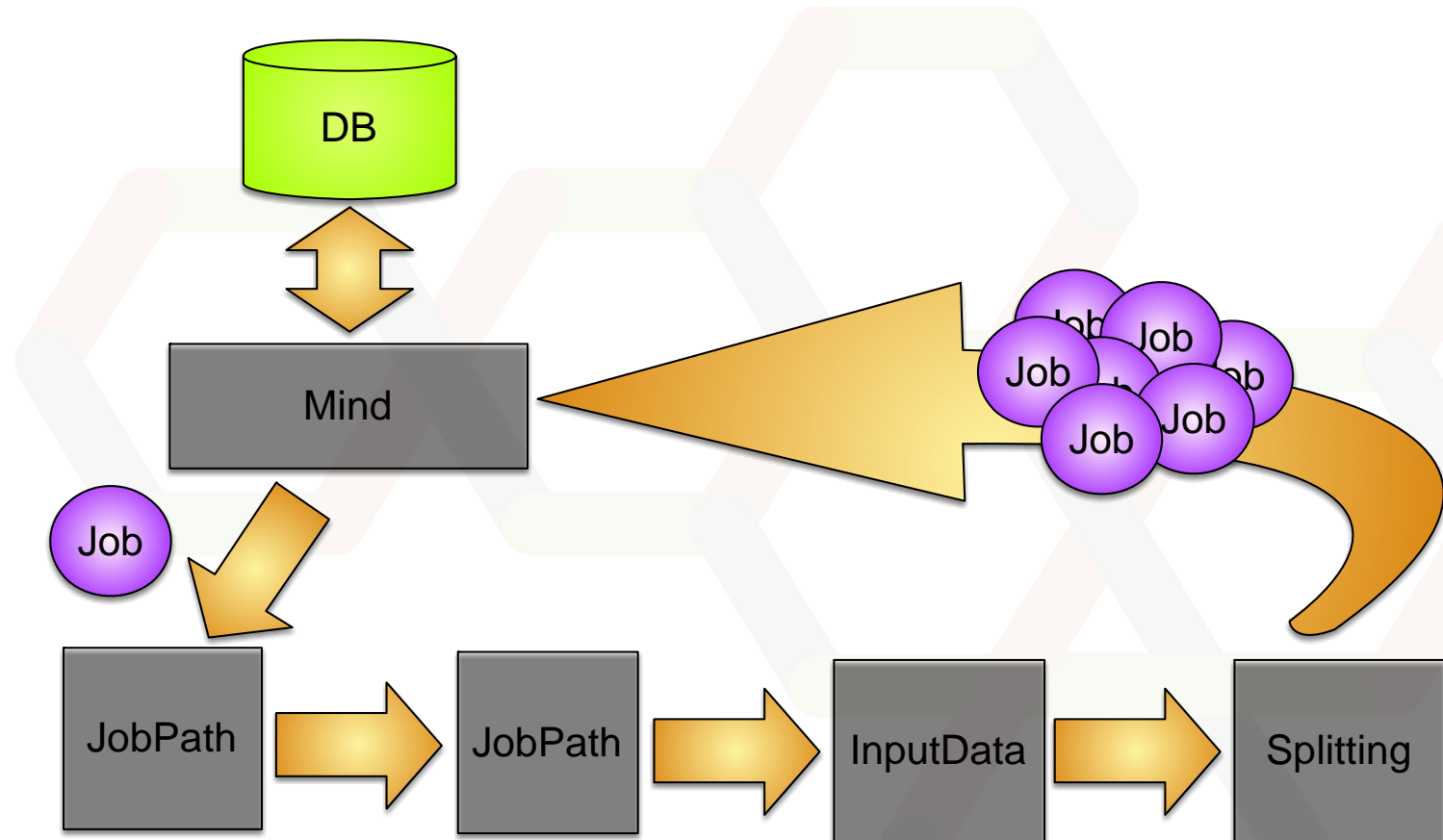
- ▶ LHCb had lots of jobs in Receiving and Checking status
 - ▶ Good Ol' Optimizers bottleneck
- ▶ Now, there are almost no jobs in those states
 - ▶ More jobs ready to run!



- ▶ Currently Job JDL can have special attributes to define how to split the job
- ▶ It is now done at submission time (JobManager)
 - ▶ Don't have info about data/storages/catalogs/site status...
 - ▶ Can only split in a predefined way
 - ▶ Can't let DIRAC optimize the splitting

- ▶ Splitting has to happen when DIRAC has enough information to do it
 - ▶ Probably after checking site, storage, catalog and rss information
- ▶ There's no single way to perform splitting that will match all use cases
 - ▶ We need different splitting modules
- ▶ Jobs that come from a split should be grouped somehow (NOT with the JobGroup)
- ▶ It would also be nice that instead of special %n, %d chars in the JDL we could use something like \$LFN

- ▶ Splitting should be performed when we can optimize the split



Multiple ways to split each job

- ▶ Jobs have to tell DIRAC how they want to be split
 - ▶ By parameter?
 - ▶ ParameterFactor, ParameterStep and so on..
 - ▶ 1 job for each n LFNs?
 - ▶ Make a job for each n LFNs in InputData...
 - ▶ 1 job for each n LFNs that have replicas in the same site?
 - ▶ Make each job with preferably n LFNs that are accessible from one of the selected sites now
 - ▶ Ignore banned sites/SEs?

- ▶ Jobs need to define which splitting module to use
- ▶ For instance in the JDL
 - ▶ `Splitter = InputDataBySE`
- ▶ Each splitter:
 - ▶ Does a split based on different things
 - ▶ Can define when it can run
 - ▶ Will use different attributes from the JDL
 - ▶ `InputDataBySE` will use `InputData` attribute
 - ▶ `Parametric` will use `ParameterFactor`, `ParameterStep...`

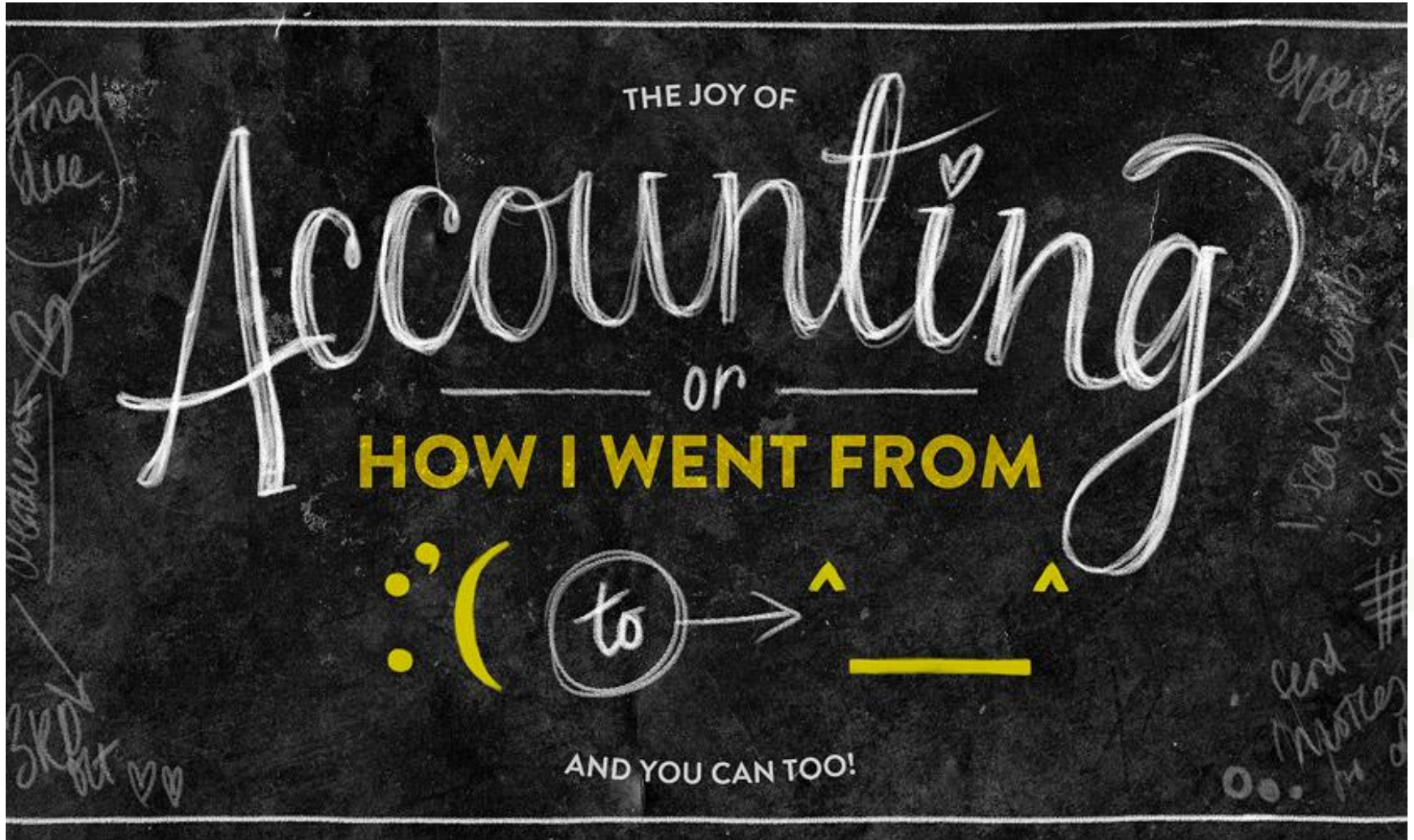
- ▶ First job will be replaced with the first job of the herd so there is no special case
 - ▶ Original state will be preserved in a separate table but a reschedule/reset won't trigger again a split of the same job
- ▶ We already have a MasterJobID in the JobDB. It's time to use it!
- ▶ Jobs with the same MasterJobID as the job originally submitted belong in the same "herd".

- ▶ Instead of using %n, %d... for special values that will be replaced use \$name
 - ▶ %d -> \$SplitID
- ▶ Each splitter module can define it's own variables
 - ▶ InputDataBySE → \$SplitterChosenSE
 - ▶ Parametric → \$Parameter
- ▶ And modify attributes
 - ▶ InputData → Only the LFNs for this job after the split
- ▶ All JDL attributes are variables
 - ▶ StdOutput = "out_ \$JobID"

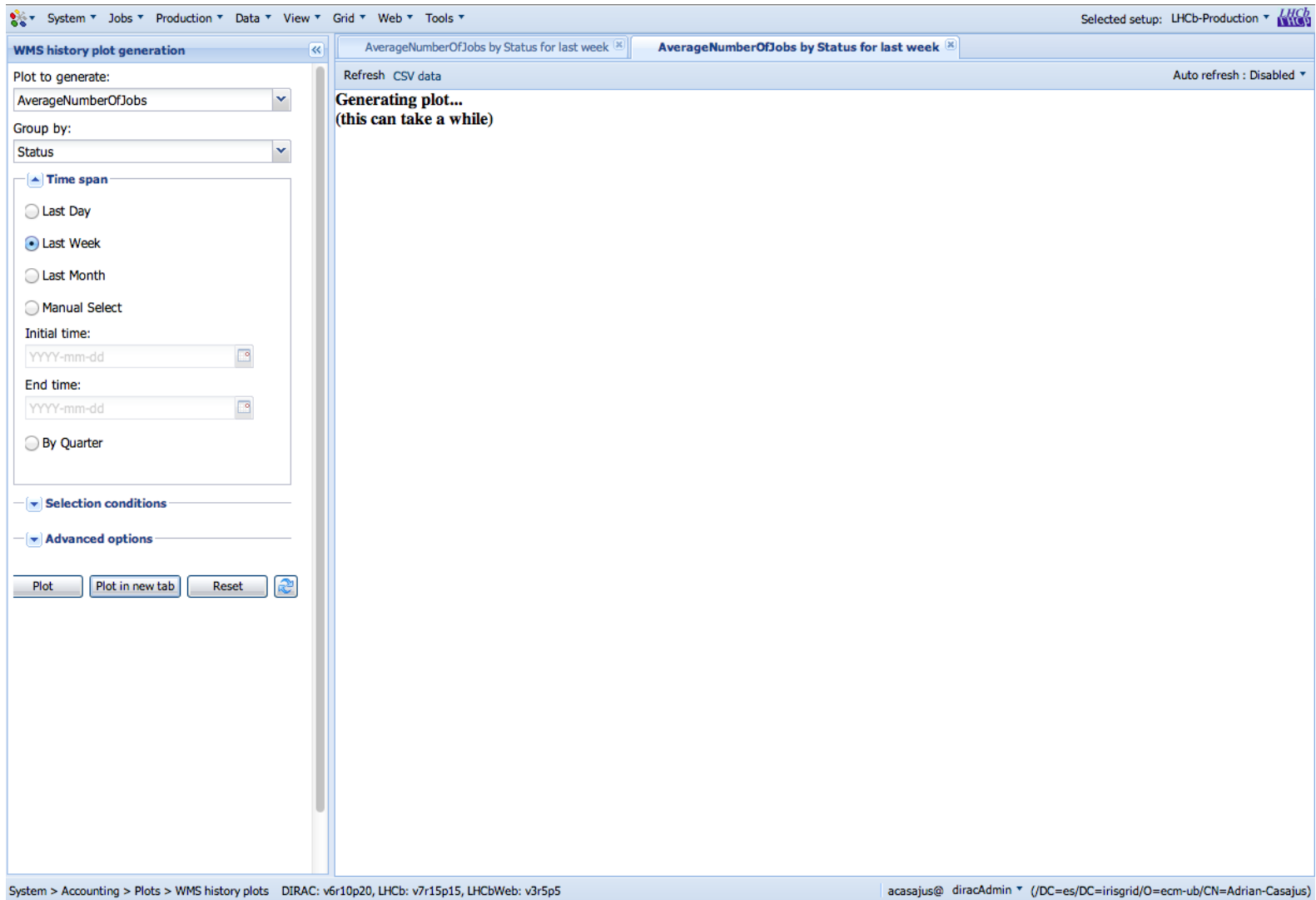
- ▶ It's already in v7

- ▶ Currently the pilot script is monolithic
- ▶ What if a VO wants a slight modification or complete different behavior?
 - ▶ Until now, you need to do your own script
- ▶ What does a pilot script do?
 - ▶ Install DIRAC
 - ▶ Configure DIRAC
 - ▶ Run JobAgent
- ▶ Why not make those configurable parts?

- ▶ Pilot script has to be self-contained prior to job submission to resource
 - ▶ It runs before there's any DIRAC installed!
- ▶ Generate script piece by piece from different functions
- ▶ Can even tailor to type of job!



Now somebody's looking at...



The screenshot shows a web application interface for generating WMS history plots. The main window is titled "WMS history plot generation" and contains several configuration sections:

- Plot to generate:** A dropdown menu set to "AverageNumberOfJobs".
- Group by:** A dropdown menu set to "Status".
- Time span:** Radio buttons for "Last Day", "Last Week" (selected), "Last Month", and "Manual Select".
- Initial time:** A text input field with a date picker icon, containing "YYYY-mm-dd".
- End time:** A text input field with a date picker icon, containing "YYYY-mm-dd".
- By Quarter:** A radio button.
- Selection conditions:** A collapsed section.
- Advanced options:** A collapsed section.

At the bottom of the configuration panel are buttons for "Plot", "Plot in new tab", and "Reset".

The main content area on the right shows two browser tabs, both titled "AverageNumberOfJobs by Status for last week". The active tab displays "Refresh CSV data" and "Auto refresh : Disabled". The main content area is currently blank, with the text "Generating plot... (this can take a while)" centered on the screen.

The status bar at the bottom of the application shows the following information: "System > Accounting > Plots > WMS history plots", "DIRAC: v6r10p20, LHCb: v7r15p15, LHCbWeb: v3r5p5", and the user "acasajus@ diracAdmin" with a path "/DC=es/DC=irisgrid/O=ecm-ub/CN=Adrian-Casajus".

Accounting problems

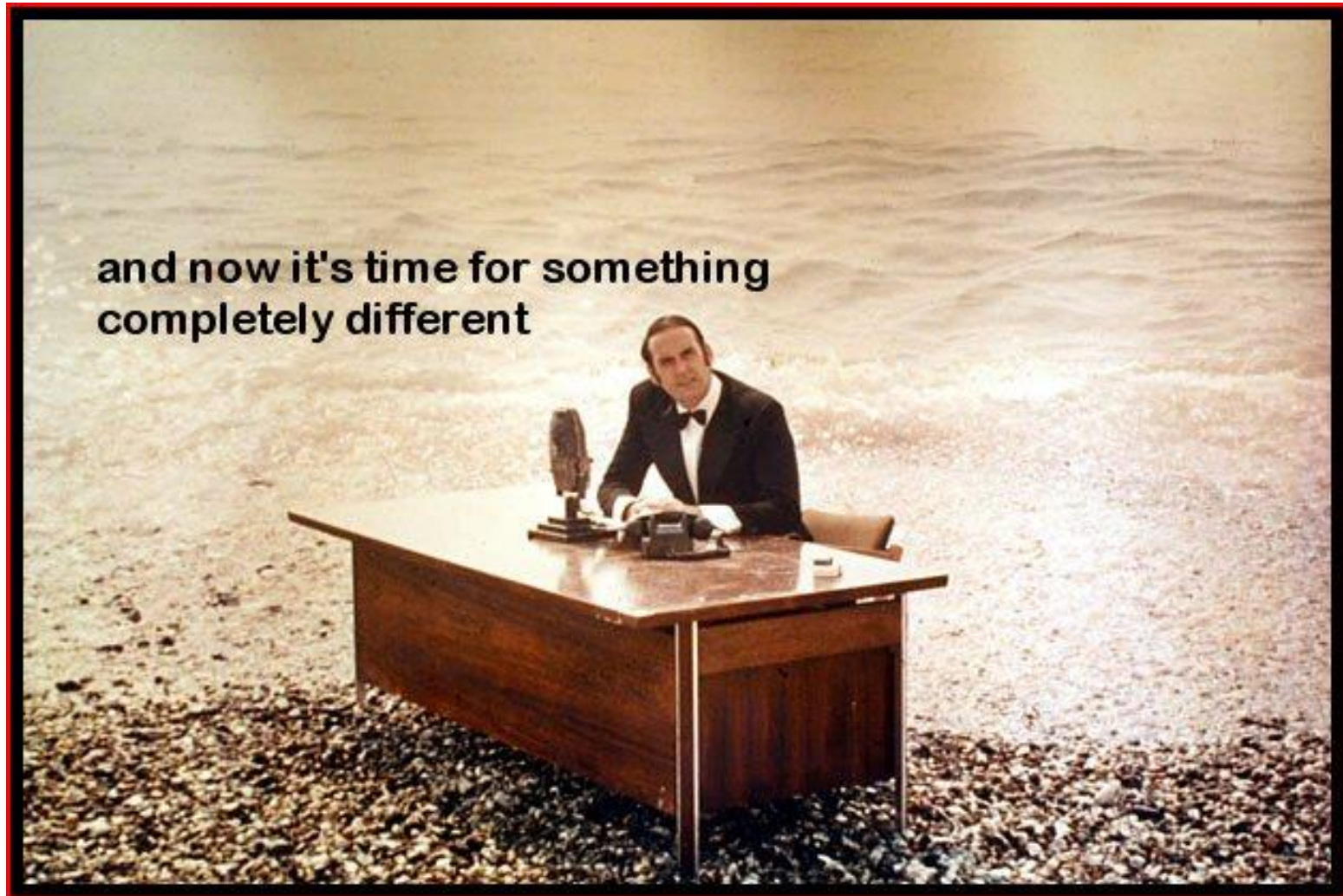
- ▶ The good: You can plot almost anything without restrictions
 - ▶ Show me CPU used by user X at site Y for jobs that failed with state Z last year
- ▶ The ugly: It's fairly static. Modifying an already established type requires some magic
 - ▶ Job now has a new attribute... Ooops...
- ▶ The bad: System is overloaded
 - ▶ WMSHistory is flooding the accounting.
 - ▶ Each attribute for a type increases the DB space and query time

- ▶ Accounting was NOT designed to do monitoring
 - ▶ What happened in the last week? 😊
 - ▶ What's going on at site X now? ☹️
- ▶ Can't keep track of status changes of entities easily
 - ▶ How many jobs went into failed state in the last 10 minutes?
- ▶ Too big granularity for recent events
 - ▶ Lowering it would add even more strain to the system..
- ▶ Difficult to see transient states (Matching state anyone?)

- ▶ Complement the accounting service for monitoring tasks
 - ▶ WMSHistory-like info should go there
- ▶ Focus on recent history, not long-term one
 - ▶ 1 minute resolution would be nice
 - ▶ Keeping last weeks/month of data should be enough
- ▶ More dynamic on attributes definition
 - ▶ Avoid requiring dark arts to change attributes

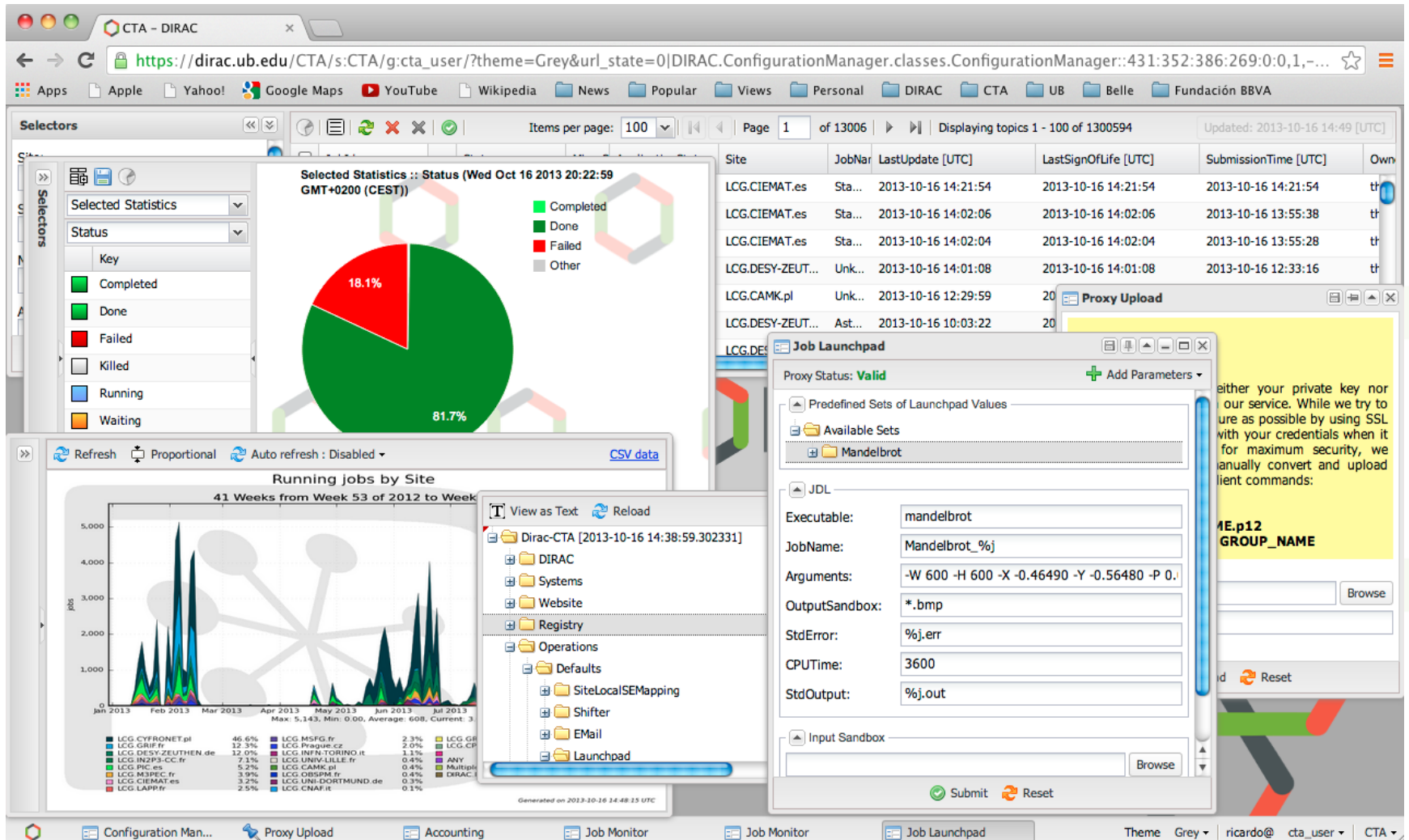
- ▶ There's already an activity accounting
 - ▶ Queries are predefined.
 - ▶ Difficult to retrieve numeric data
- ▶ Should be merged into the new monitoring service
- ▶ Several approaches on how to store the data
 - ▶ MySQL with some clever table structure
 - ▶ NoSQL: Elastic search seems to be a viable candidate
 - ▶ Specialized in storing dated events and search them
 - ▶ Not tested yet though
 - ▶ NoSQL: Hadoop anyone? Others?

- ▶ **Accounting:**
 - ▶ Data is stored after consolidation
 - ▶ Long-term information storage
- ▶ **Monitoring**
 - ▶ Data is stored in raw format (probably)
 - ▶ State changes + full counters
 - ▶ Short-term information storage
- ▶ **Benefits.**
 - ▶ Remove accounting overload
 - ▶ Provide real monitoring information



- ▶ Was nice and did the job
 - ▶ Lacked some functionality but was mostly solving the problem
- ▶ **BUT it was based on:**
 - ▶ ExtJS2 (no longer supported)
 - ▶ Pylons (no longer exists...)
 - ▶ lighttpd (works but there are better alternatives now)
- ▶ Welcome to maintenance hell...

- ▶ New framework
- ▶ Based on
 - ▶ ExtJS4
 - ▶ Tornado
 - ▶ Nginx
- ▶ Faster and has more features



Selected Statistics :: Status (Wed Oct 16 2013 20:22:59 GMT+0200 (CEST))

- Completed: 81.7%
- Done: 18.1%
- Failed: 0%
- Other: 0%
- Killed: 0%
- Running: 0%
- Waiting: 0%

Site	JobName	LastUpdate [UTC]	LastSignOfLife [UTC]	SubmissionTime [UTC]	Own
LCG.CIEMAT.es	Sta...	2013-10-16 14:21:54	2013-10-16 14:21:54	2013-10-16 14:21:54	th
LCG.CIEMAT.es	Sta...	2013-10-16 14:02:06	2013-10-16 14:02:06	2013-10-16 13:55:38	th
LCG.CIEMAT.es	Sta...	2013-10-16 14:02:04	2013-10-16 14:02:04	2013-10-16 13:55:28	th
LCG.DESY-ZEUT...	Unk...	2013-10-16 14:01:08	2013-10-16 14:01:08	2013-10-16 12:33:16	th
LCG.CAMK.pl	Unk...	2013-10-16 12:29:59	2013-10-16 12:29:59		th
LCG.DESY-ZEUT...	Ast...	2013-10-16 10:03:22	2013-10-16 10:03:22		th

Job Launchpad

Proxy Status: **Valid**

Executable: mandelbrot

JobName: Mandelbrot_%j

Arguments: -W 600 -H 600 -X -0.46490 -Y -0.56480 -P 0.

OutputSandbox: *.bmp

StdError: %j.err

CPUTime: 3600

StdOutput: %j.out

Submit Reset

Running jobs by Site

41 Weeks from Week 53 of 2012 to Week 41 of 2013

Max: 5,143, Min: 0.00, Average: 608, Current: 3

Site	Percentage
LCG.CYFRONET.pl	46.6%
LCG.GRIF.fr	12.3%
LCG.DESY-ZEUTHEN.de	12.0%
LCG.IN2P3-CC.fr	7.1%
LCG.PIC.es	5.2%
LCG.M3PEC.fr	3.9%
LCG.CIEMAT.es	3.2%
LCG.LAPPY.fr	2.5%
LCG.MSPG.fr	2.3%
LCG.Prague.cz	2.0%
LCG.RNFT.TORINO.it	1.3%
LCG.UNIV-LILLE.fr	0.4%
LCG.CAMK.pl	0.4%
LCG.OBSIPA.fr	0.4%
LCG.UNI-DORTMUND.de	0.3%
LCG.CNAF.it	0.1%

- ▶ Allow to save any state the web is in and recall it later
- ▶ Allows to share application states between users
- ▶ Almost all the old functionality is there!
- ▶ Still not auto-installable like the old one..
 - ▶ pip install tornado, compile js and install and configure nginx..
 - ▶ working on that...
 - ▶ v1r0 tagged yesterday :P

- ▶ Accessible now for you
 - ▶ <https://dirac.ub.edu>
 - ▶ Need to move it out from ub.edu. Network is crappy..
- ▶ Check out Zoltan's & Nikola session tomorrow

- ▶ The grid will migrate to RFC proxies instead of the current grid proxies at some point in the distant future
 - ▶ We're almost ready. 😞 (same as last year :P)
- ▶ SHA-2 certs are widely used today
 - ▶ No problem! 😊
- ▶ Heartbleed bug hit us (like the rest of the internet)
 - ▶ Took a while to react 😞
 - ▶ Thanks to the guys that recompiled DIRAC externals with the fix
 - ▶ But it's ok now 😊

- ▶ **COMDIRAC:**
 - ▶ Unix-like shorter commands
 - ▶ dinit, dkill, dsub, dput, dget, drm...
- ▶ **DIRAC4Android**
 - ▶ DIRAC interface for android
 - ▶ Based on RESTDIRAC interface
- ▶ **TestDIRAC**
 - ▶ Unit tests for DIRAC
 - ▶ Not very complete... help us! 😊

- ▶ 1 connection per thread
 - ▶ Less connections to MySQL
 - ▶ Still more than 1k...
- ▶ Transaction support
 - ▶ Fabled v7 required
- ▶ All DIRAC DBs have been migrated to InnoDB
 - ▶ Headaches passed and several tweaks required...