International Symposium on Grids and Clouds 2015

15~20 March 2015, Academia Sinica, Taipei, Taiwan

Architecture of the LHCb Distributed Computing System

Federico Stagni, Philippe Charpentier

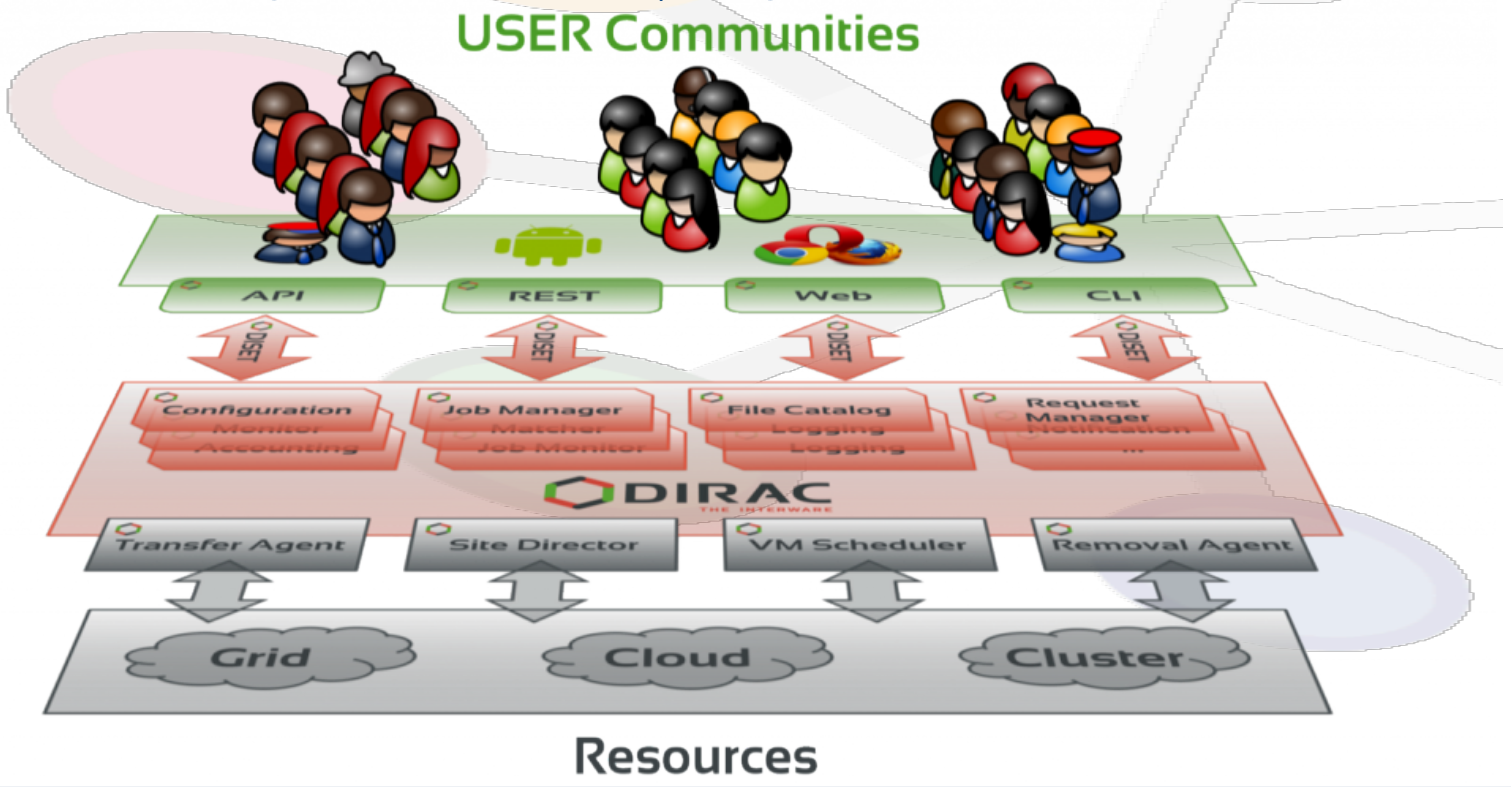
On behalf of the LHCb collaboration

LHCbDirac: What and why

What's DIRAC

The "**DIRAC**" (Distributed Infrastructure with Remote Agent Control) **INTERWARE** is a software framework for distributed computing providing a complete solution to one (or more) **user community** requiring access to **distributed resources**. DIRAC builds a layer between the users and the resources offering a common **interface** to a number of heterogeneous providers, **integrating** them in a seamless manner, providing **interoperability**, at the same time as an optimized, transparent and reliable usage of the resources."

(See A. Tsaregorodtsev 2 days ago)



http://diracgrid.org

What's LHCbDirac (a.k.a. BeautyDirac)

- Dirac consists of a set of collaborating agents and services
  - Using its own secure communication protocol (DiSeT)
- DIRAC grew within LHCb, but now:
  - There's not only LHCb
    - Still, the biggest community
  - Each community can decide to create its own extensions
    - Independent release cycles
  - Beauty/LHCb-Dirac extension: LHCb-specific code

LHCb activities and how to handle them

LHCb distributed computing activities:

(see presentation from C. Haen yesterday)

"Production" (central) activities:

- Reconstruction
- Reprocessing
- Stripping (selection of events)
- Monte-Carlo simulations
- Indexing

Non-"Production" activities:

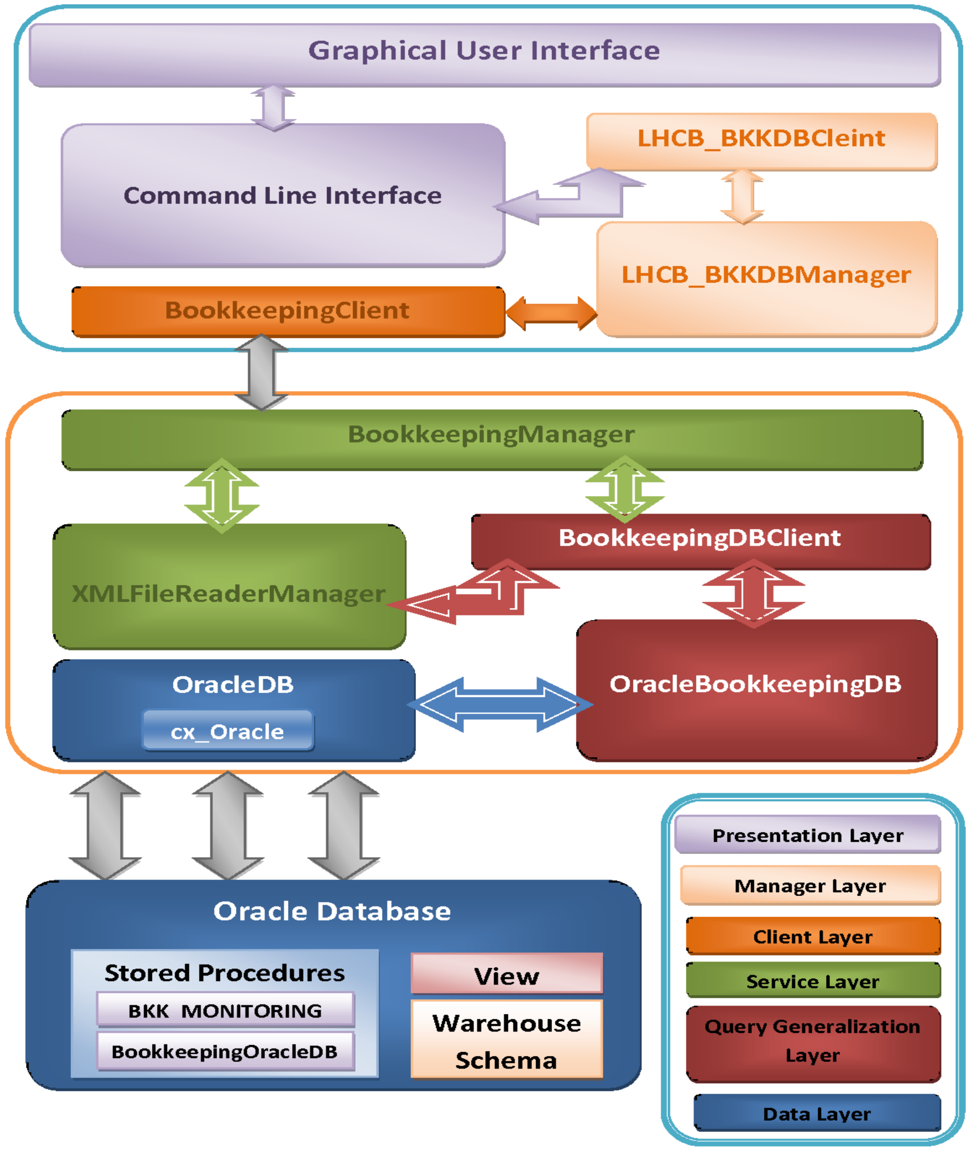
- User analysis
- Monitoring and testing

How to handle:

- Using a Bookkeeping system (BK) for the data provenance and dataset retrieval (see poster on LHCb Data Management)
- Extending the DIRAC Transformation system
  - Using the BK for retreiveing datasets
  - Implementing its own task creation plugins
- Implementing a productions' requests system
- Automating at most production management

(LHCb)Dirac Components and concepts

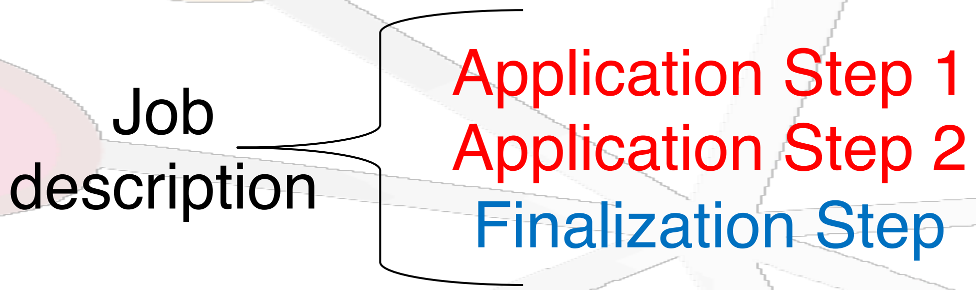
Bookkeeping



- A data provenance catalog
  - Contains all files ever created in LHCb and information about how they were created
    - Raw data as well as derived or simulated data
- Not specifically a tool for distributed computing
- Main tool for retrieving datasets
  - Users (for analysis) and production system
    - Conditions (data taking, simulation)
    - Processing (applications, detector condition parameters)
    - Event type
    - File type
- Fully integrated in LHCbDirac
  - Based on DIRAC services and DiSeT
- Oracle backend
  - High level of optimization (indices, views)

Workflows

- Job description format
- Enables running "complex" jobs
  - e.g. multiple applications, linked together via input/output data
  - I/O chaining
- description in different formats: XML, JDL, python
  - JDL executable: dirac-jobexec
  - Argument: jobDescription.xml (which is in the Input Sandbox)
- A workflow is composed of steps
  - that are made of modules
  - workflow modules are instantiated by python modules
    - ☆ that do the real job
  - parameters at any level



(for users AND production jobs)

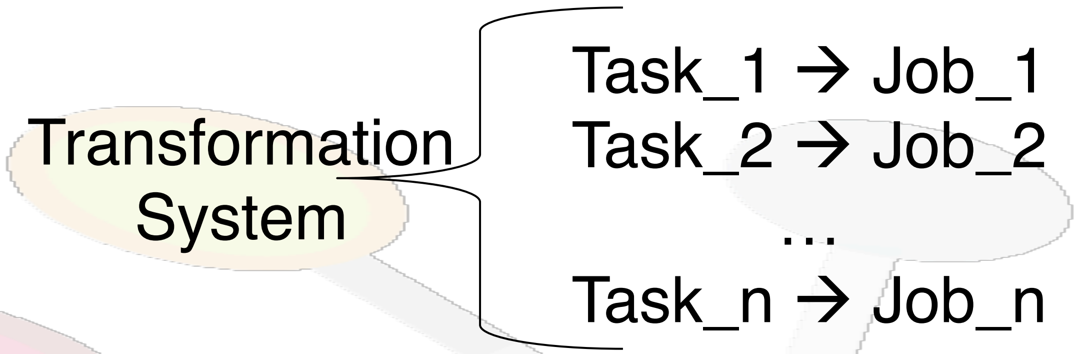
Transformation System

A system for handling "repetitive" work:

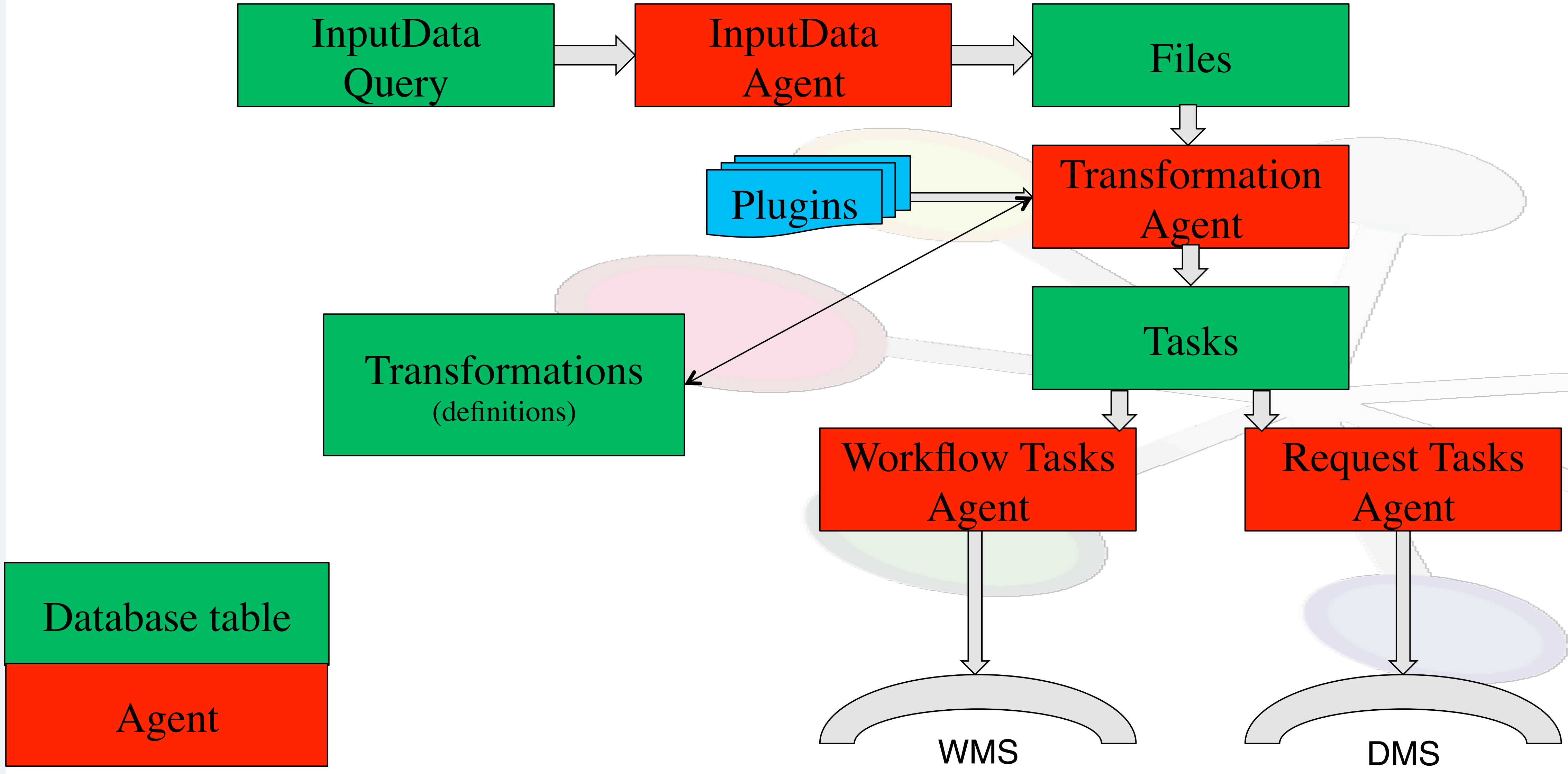
- Handles input datasets (if present)
  - Plugins are grouping input files into tasks according to various criteria
  - Tasks are created

2 main usages:

- Productions: the same job – i.e.the same *workflow* - is executed
  - Client for the Workload Management System
- Data Handling: replications, removals
  - Client for the Data Management System
    - ✫ See Poster #02

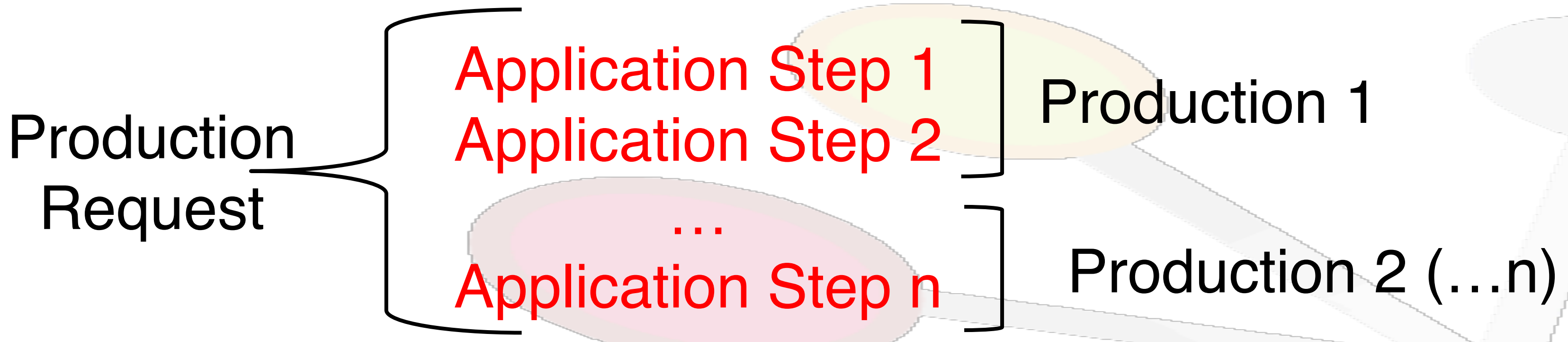


Transformation System architecture



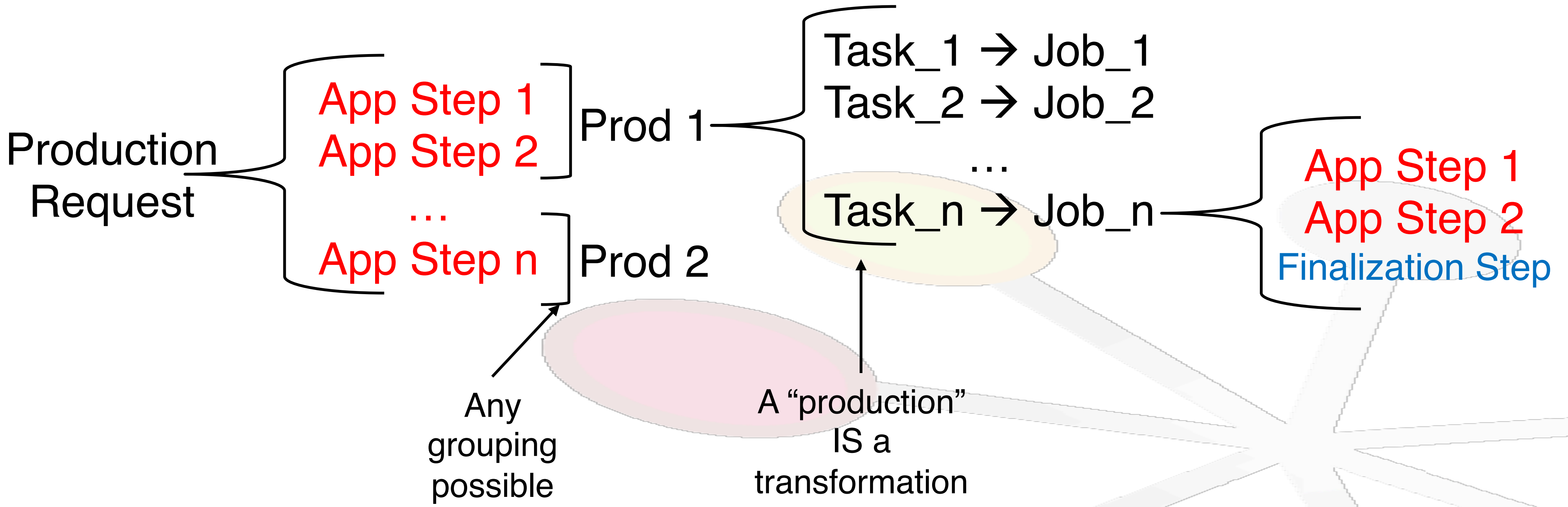
LHCb Production Requests system

- A way for specific users to make requests for data processing or MC simulation
  - Includes an authorization mechanism for physics relevance (and technical acceptance)
- Formalization of the data processing activities



- A list of application steps define a production request (with I/O chained)
- Application steps can be grouped in productions
  - As many productions as convenient
    - Intermediate datasets must be saved (temporarily at least) on storage
    - They can be destroyed by the next production if required

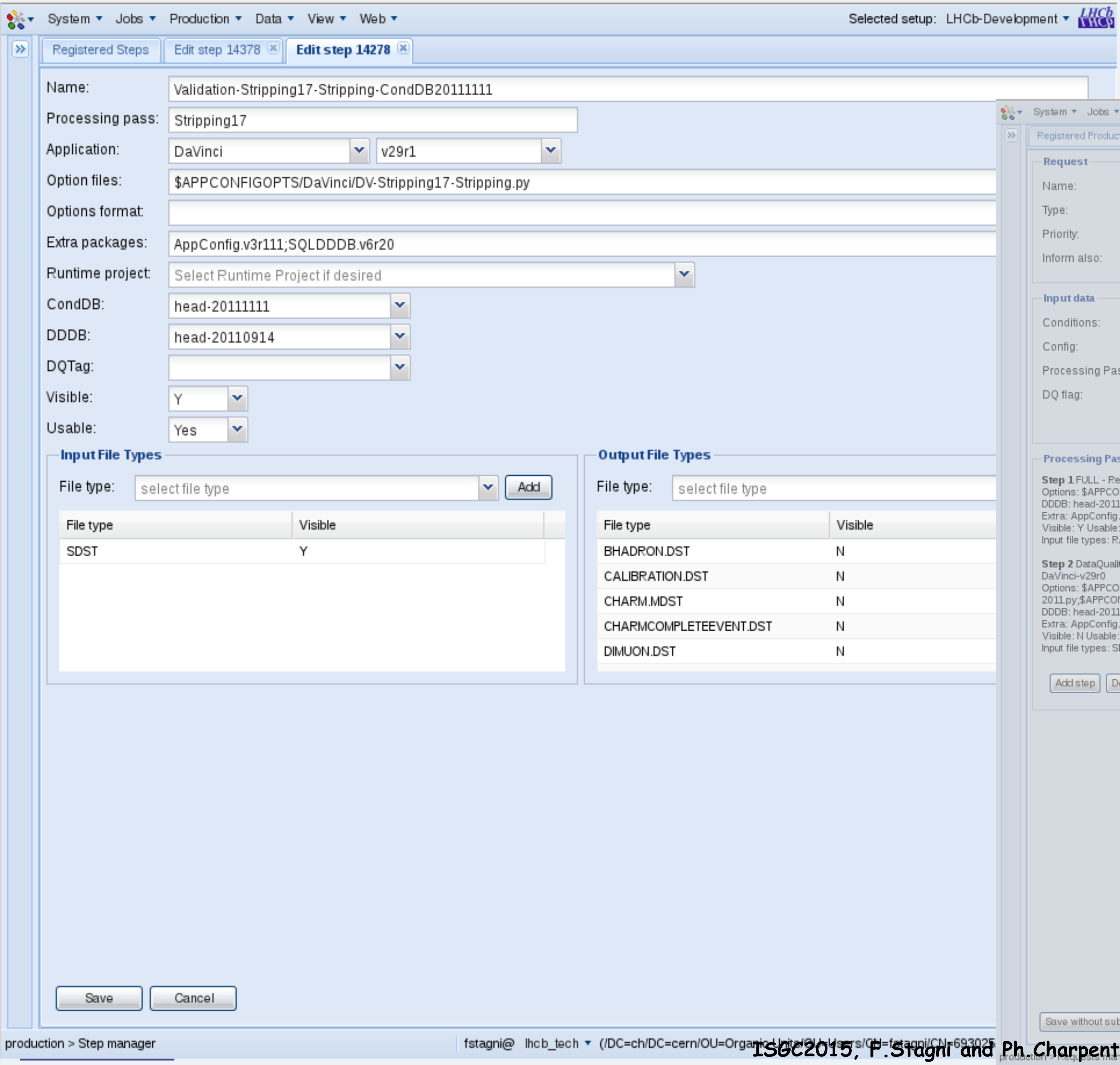
Managing the whole cycle



Operationally:

1. Application managers create steps
2. Production Managers (PMs) or Application Managers create requests
3. PMs launch requests using production templates
4. Productions are followed by shifters, GEOC (Grid Expert On Call) and the same PMs

Production Request System GUI



Name:	Validation-Stripping17-Stripping-CondDB20111111
Processing pass:	Stripping17
Application:	DaVinci v29r1
Option files:	$APPCONFIGOPTS/DaVinci/DV-Stripping17-Stripping.py
Options format:	
Extra packages:	AppConfig.v3r111;SQLDDDB.v6r20
Runtime project:	Select Runtime Project if desired
CondDB:	head-20111111
DDDB:	head-20110914
DQTag:	
Visible:	Y
Usable:	Yes

Input File Types

File type: select file type [Add]

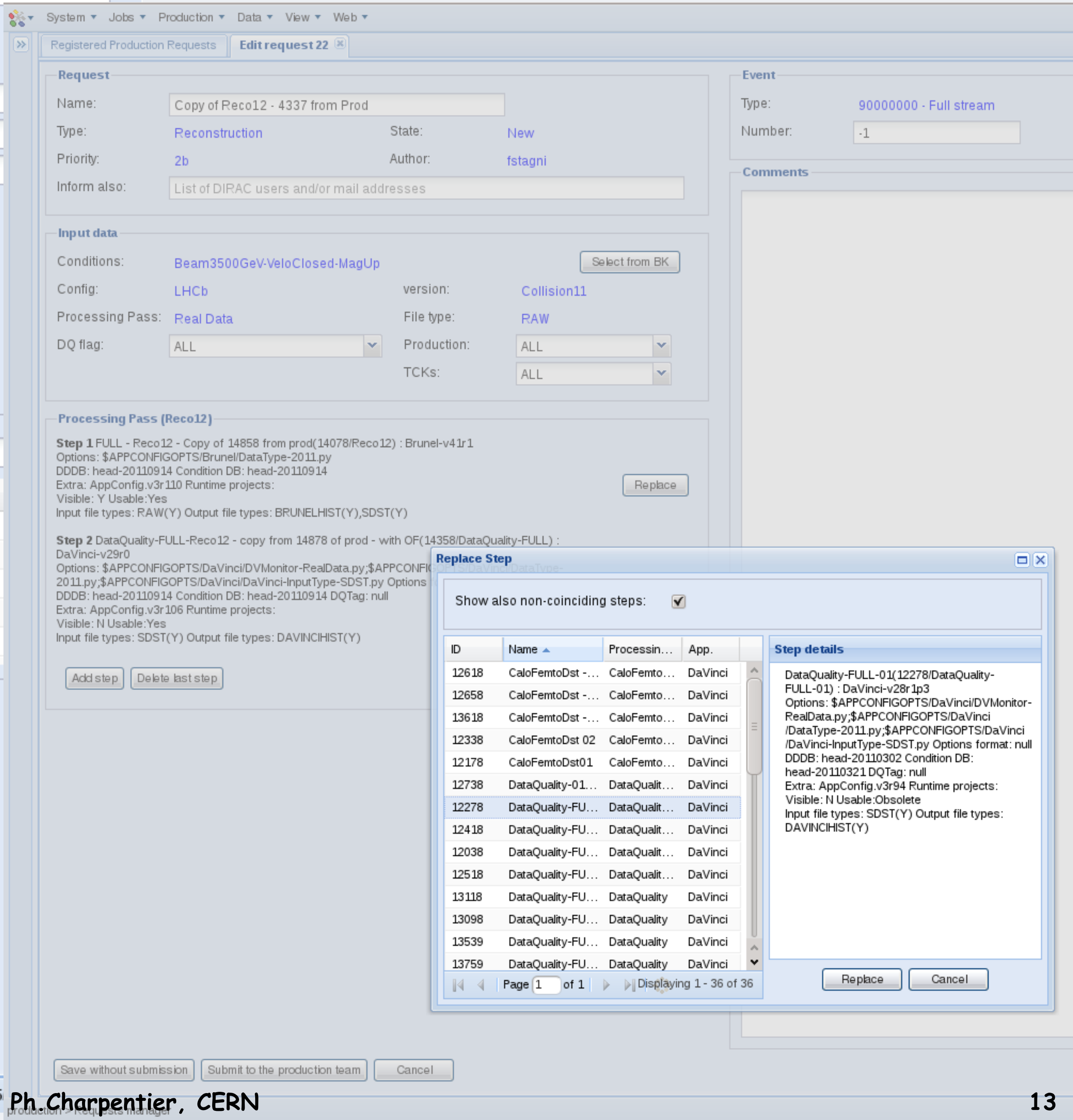
File type	Visible
SDST	Y

Output File Types

File type: select file type

File type	Visible
BHADRON.DST	N
CALIBRATION.DST	N
CHARM.MDST	N
CHARMCOMPLETEEVENT.DST	N
DIMUON.DST	N

Save
Cancel
production > Step manager
fstagni@ lhcb_tech (/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=fstagni/CN=693025


System
Jobs
Production
Data
View
Web
Registered Production Requests
Edit request 22

Request

Name:	Copy of Reco12 - 4337 from Prod		
Type:	Reconstruction	State:	New
Priority:	2b	Author:	fstagni
Inform also:	List of DIRAC users and/or mail addresses		

Event

Type:	90000000 - Full stream
Number:	-1

Comments

Input data

Conditions:	Beam3500GeV-VeloClosed-MagUp		
Config:	LHCb	version:	Collision11
Processing Pass:	Real Data	File type:	RAW
DQ flag:	ALL	Production:	ALL
		TCKs:	ALL

Processing Pass (Reco12)

Step 1 FULL - Reco12 - Copy of 14858 from prod(14078/Reco12) : Brunel-v41r1
Options: $APPCONFIGOPTS/Brunel/DataType-2011.py
DDDB: head-20110914 Condition DB: head-20110914
Extra: AppConfig.v3r110 Runtime projects:
Visible: Y Usable:Yes
Input file types: RAW(Y) Output file types: BRUNELHIST(Y),SDST(Y)

Step 2 DataQuality-FULL-Reco12 - copy from 14878 of prod - with OF(14358/DataQuality-FULL) :
DaVinci-v29r0
Options: $APPCONFIGOPTS/DaVinci/DVMonitor-RealData.py;$APPCONFIGOPTS/DaVinci/DataType-2011.py;$APPCONFIGOPTS/DaVinci/DaVinci-InputType-SDST.py Options
DDDB: head-20110914 Condition DB: head-20110914 DQTag: null
Extra: AppConfig.v3r106 Runtime projects:
Visible: N Usable:Yes
Input file types: SDST(Y) Output file types: DAVINCIHIST(Y)

Replace Step

Show also non-coinciding steps: ✓

ID	Name	Processin...	App.
12618	CaloFemtoDst -...	CaloFemto...	DaVinci
12658	CaloFemtoDst -...	CaloFemto...	DaVinci
13618	CaloFemtoDst -...	CaloFemto...	DaVinci
12338	CaloFemtoDst 02	CaloFemto...	DaVinci
12178	CaloFemtoDst01	CaloFemto...	DaVinci
12738	DataQuality-01...	DataQualit...	DaVinci
12278	DataQuality-FU...	DataQualit...	DaVinci
12418	DataQuality-FU...	DataQualit...	DaVinci
12038	DataQuality-FU...	DataQualit...	DaVinci
12518	DataQuality-FU...	DataQualit...	DaVinci
13118	DataQuality-FU...	DataQuality	DaVinci
13098	DataQuality-FU...	DataQuality	DaVinci
13539	DataQuality-FU...	DataQuality	DaVinci
13759	DataQuality-FU...	DataQuality	DaVinci

Page 1 of 1 Displaying 1 - 36 of 36

Step details

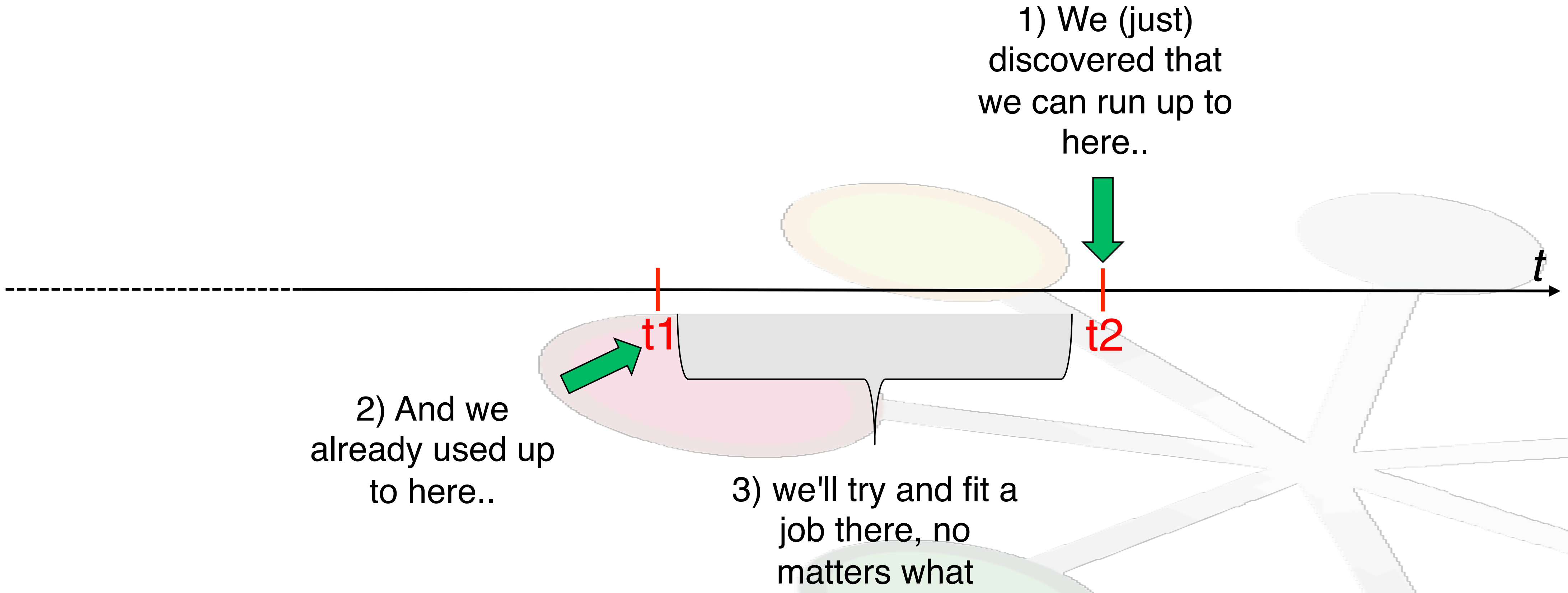
DataQuality-FULL-01(12278/DataQuality-FULL-01) : DaVinci-v28r1p3
Options: $APPCONFIGOPTS/DaVinci/DVMonitor-RealData.py;$APPCONFIGOPTS/DaVinci/DataType-2011.py;$APPCONFIGOPTS/DaVinci/DaVinci-InputType-SDST.py Options format: null
DDDB: head-20110302 Condition DB: head-20110321 DQTag: null
Extra: AppConfig.v3r94 Runtime projects:
Visible: N Usable:Obsolete
Input file types: SDST(Y) Output file types: DAVINCIHIST(Y)

Managing Productions

Automations for Simulation requests - starting a production

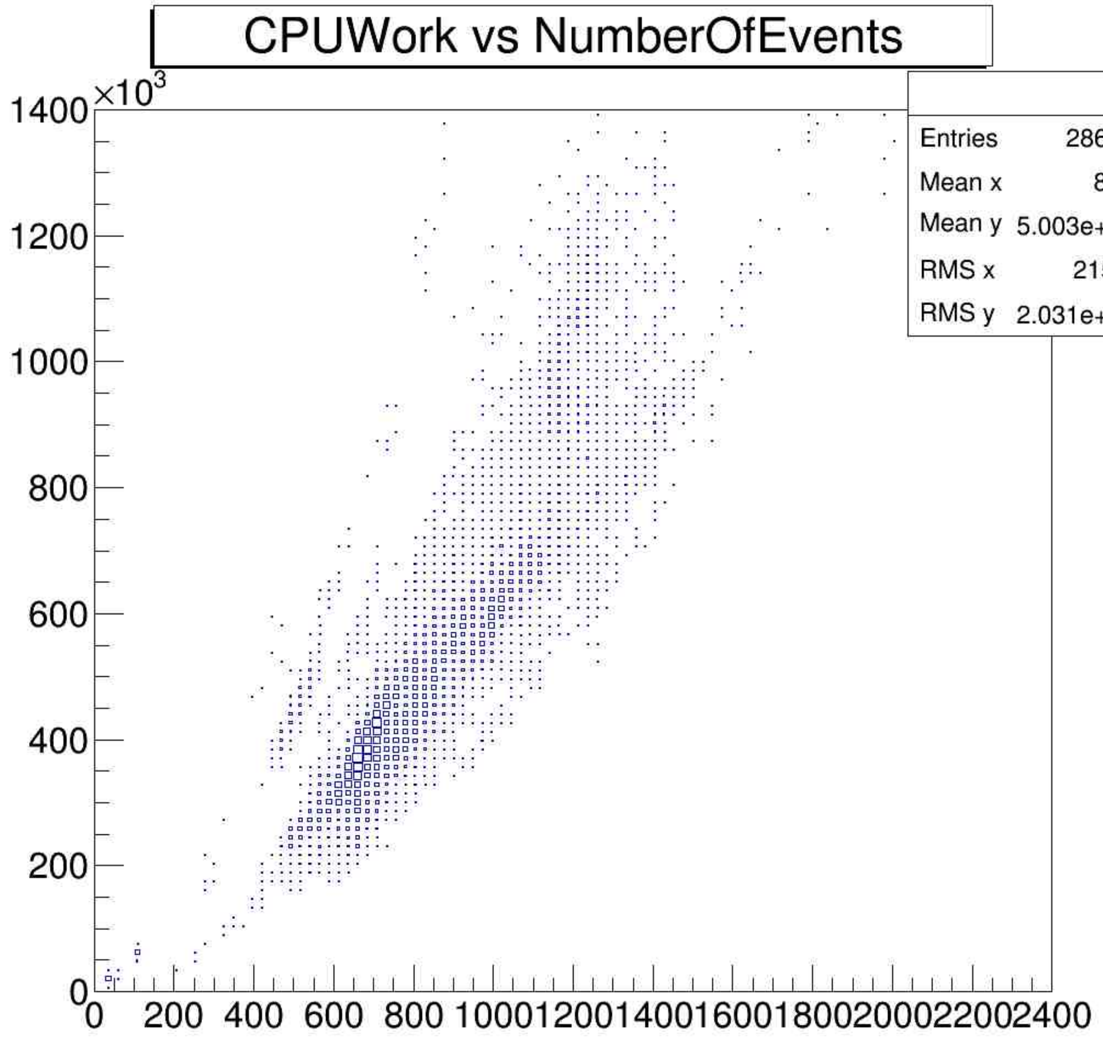
- Simulation productions go through a testing phase before being submitted
- A limited amount of jobs are created and submitted to a site chosen for testing
  - each job produces a fixed amount of events
- Productions undergoing a testing phase are monitored by a dedicated agent
  - when all jobs are finished, an evaluation takes place:
    - If all jobs failed, the production request is rejected
    - If jobs are successful, the following results are evaluated
      - CPU-work (in HS06.s) per event (CPUe) is calculated
      - Job description is modified: CPUe is added, destination is changed…
- Simulation productions have to produce at least a requested amount of events
  - In case not enough events have been produced, simulation productions are automatically extended

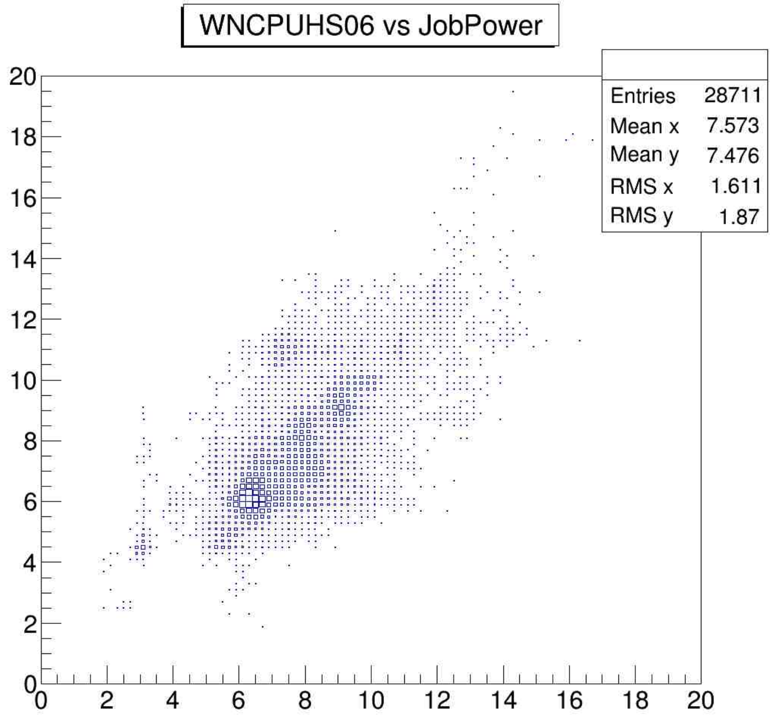
Elastic Simulation jobs

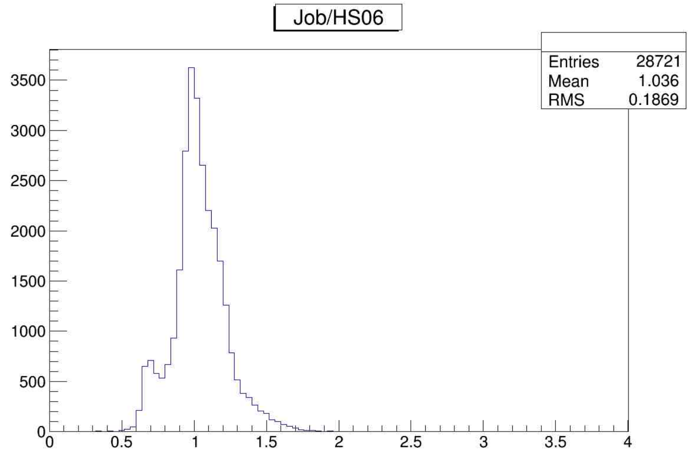

1) We (just) discovered that we can run up to here..
t
t1
t2
2) And we already used up to here..
3) we'll try and fit a job there, no matters what

- Given the CPU-work available and CPUe, compute how many events may be simulated
  - 30% safety margin to cope with uncertainties in CPU power estimate

Elastic MC jobs in practice

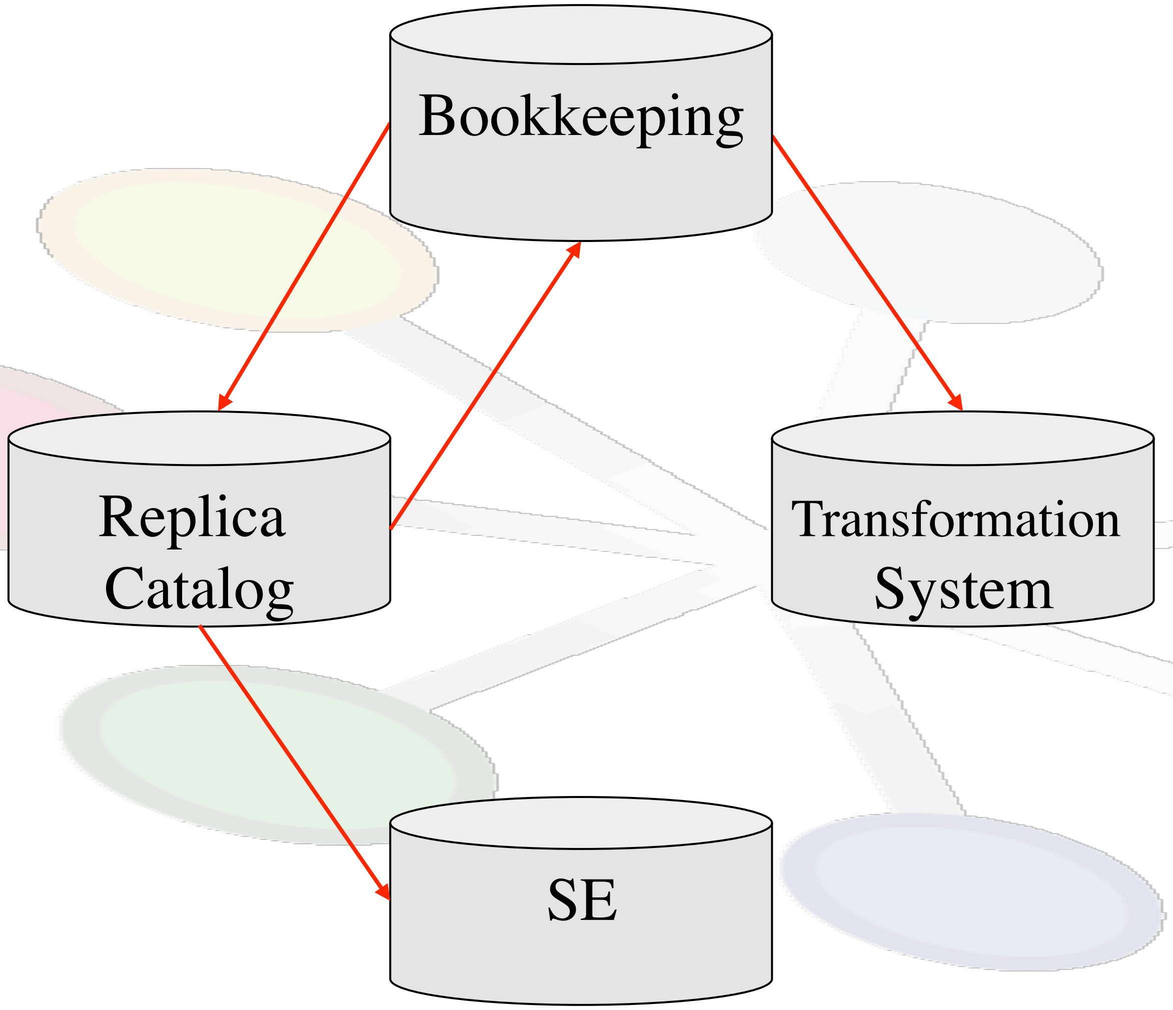






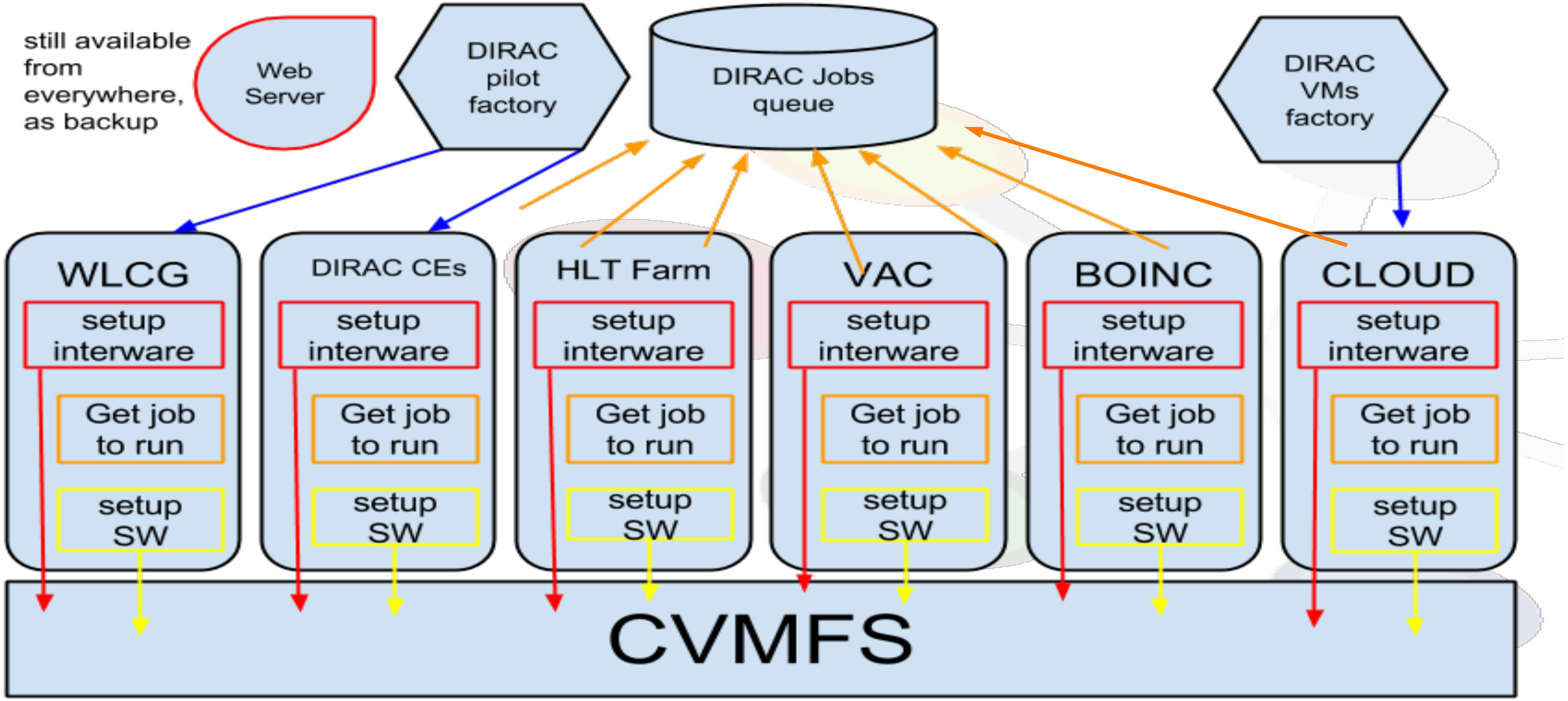
Closing a production

- We can close a production when consistency checks are successful
- Successfully completed productions are archived:
  - Jobs can be removed
  - Transformation tasks can be removed
  - Job logs can be archived



LHCb Pilots

"Pilots to fly in all the skys"



Summary

- LHCbDirac is a DIRAC extension
  - Used for all the distributed computing activities
    - One system to rule them all
- Actively developed
  - 2 or 3 minor releases/year
    - Patch releases as frequently as required (weekly on average)
  - 1 major release every 3 or 4 years
- Spread over ~30 servers:
  - 50 services running (~40 different ones)
  - ~100 agents running (~70 different ones)
  - ~10 Executors
  - ~20 DBs
    - Mostly MySQL
      - ~500 GB of data... with high variance for certain DBs
    - Oracle for the bookkeeping
    - ElasticSearch being experimented for real-time monitoring