21st International Conference on Computing in High Energy and Nuclear Physics **CHEP2015** Okinawa Japan: April 13 - 17, 2015

How the Monte Carlo production of a wide variety of different samples is centrally handled in the LHCb experiment

G. Corti (CERN)

et al.:

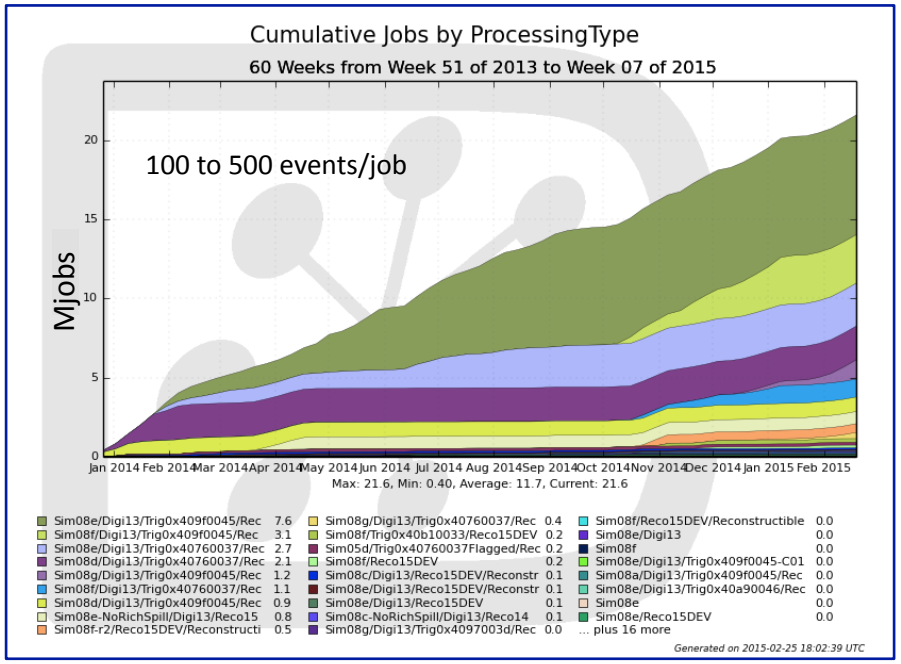
P. Charpentier (CERN), M. Clemencic (CERN), J. Closier (CERN),
B. Couturier (CERN), M. Kreps (University of Warwick), Z. Mathe (CERN),
D. O'Hanlon (University of Warwick), P. Robbe (LAL Orsay),
V. Romanovsky (IHEP Protvino), F. Stagni (CERN),
A. Zhelezov (Universität Heidelberg)

Introduction

- In LHCb a wide variety of Monte Carlo samples need to be produced for the experiment's physics program
- Procedures based on common infrastructures have been setup to handle Monte Carlo productions centrally
  - A numerical **Event Type ID** has been devised to facilitate the configuration of the simulation application
  - Monte Carlo productions are **customized types of productions** centrally handled by the Production Team
  - Deployment of new event types are managed through **standard LHCb distribution software tools**
  - The numerical Event Type ID is also used to transparently **customize Production Requests** and to identify the samples produced
- Conventions allows transparent interplay of different elements

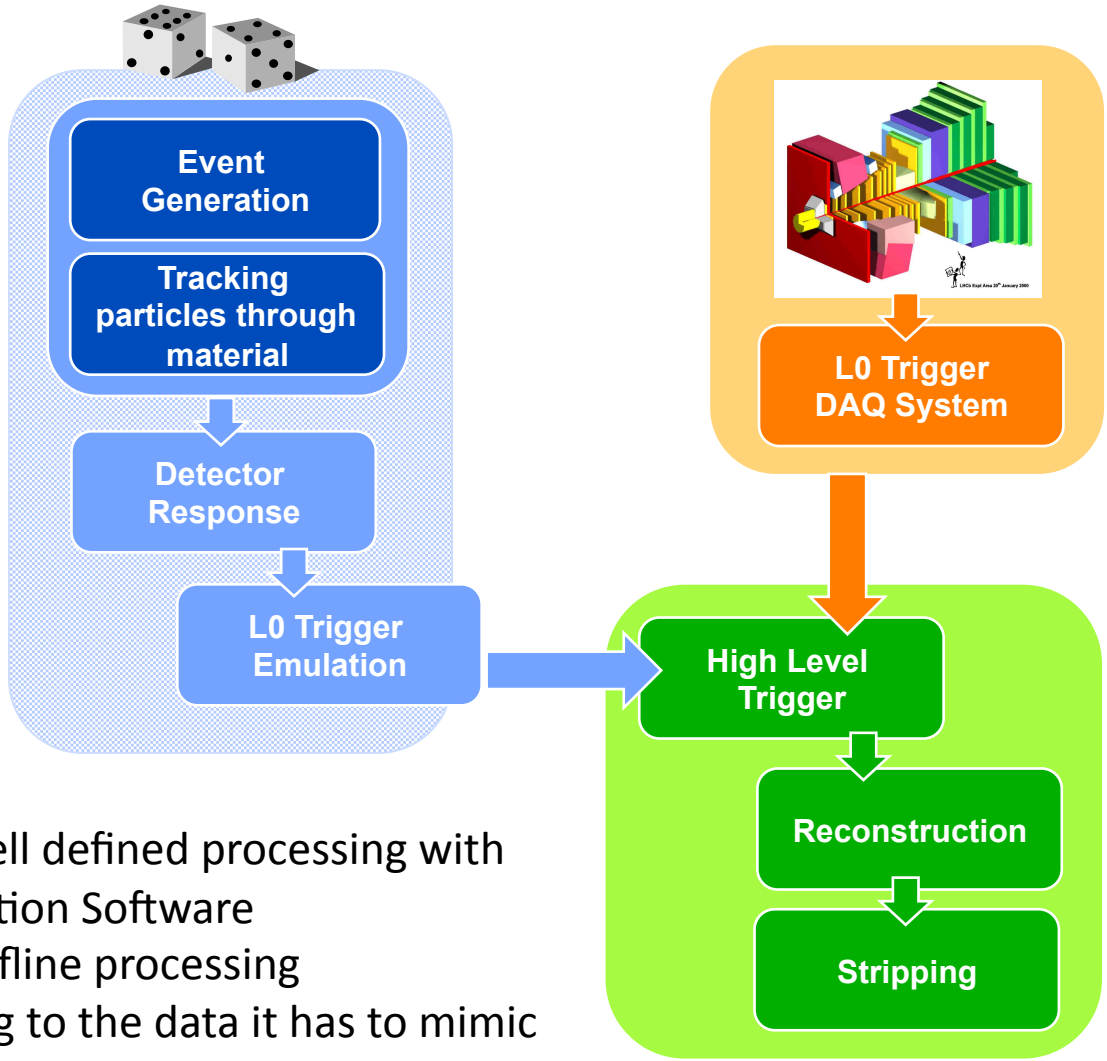
Size of Monte Carlo productions

- Production for analysis of Run 1 data samples ongoing since December 2011
  - Over **9 billions events produced**
  - Two major simulation versions
  - Different running conditions (beam & trigger)
  - Different reconstruction processing to match the data



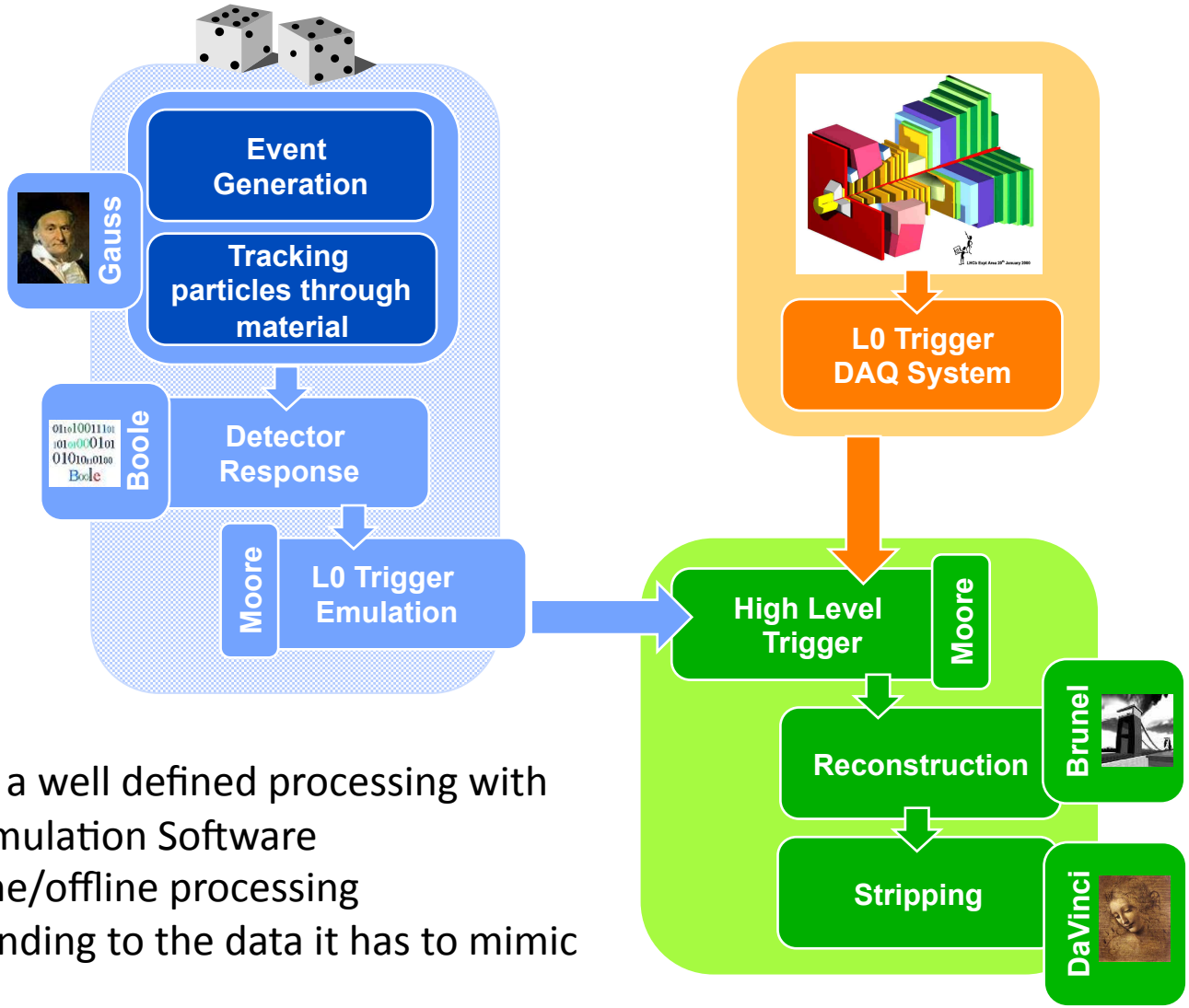
- Wide variety of signal and background samples for different analysis
  - **2260 different event types** up to last week
  - Samples from 50k to 10M events on tape (up to 100M simulated)
- Some samples are shared between physics analysis
  - Essential to keep samples consistent for a given configuration
  - Ensure samples are available to the whole LHCb collaboration

Monte Carlo Productions



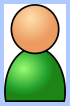
Establish a well defined processing with stable Simulation Software and online/offline processing corresponding to the data it has to mimic

Monte Carlo Productions



Establish a well defined processing with stable Simulation Software and online/offline processing corresponding to the data it has to mimic

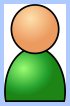
Steps and Players



Physics WG & Simulation software manager

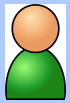
Step 1: Preparation of decay description and configuration for new decay channel(s)

Steps and Players



Physics WG & Simulation software manager

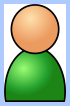
Step 1: Preparation of decay description and configuration for new decay channel(s)



Release manager

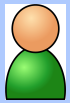
Step 2: Release of new configurations and deployment on the Grid

Steps and Players



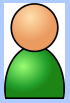
Physics WG & Simulation software manager

Step 1: Preparation of decay description and configuration for new decay channel(s)



Release manager

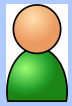
Step 2: Release of new configurations and deployment on the Grid



Simulation software manager

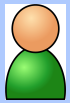
Step 3: Make the production system aware of the new event type(s)

Steps and Players



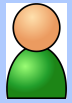
Physics WG & Simulation software manager

Step 1: Preparation of decay description and configuration for new decay channel(s)



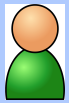
Release manager

Step 2: Release of new configurations and deployment on the Grid



Simulation software manager

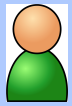
Step 3: Make the production system aware of the new event type(s)



MC Physics WG liaison

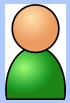
Step 4: Submission of MC request(s)

Steps and Players



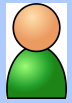
Physics WG & Simulation software manager

Step 1: Preparation of decay description and configuration for new decay channel(s)



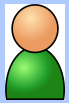
Release manager

Step 2: Release of new configurations and deployment on the Grid



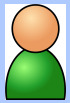
Simulation software manager

Step 3: Make the production system aware of the new event type(s)



MC Physics WG liaison

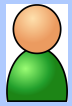
Step 4: Submission of MC request(s)



(MC) Production manager

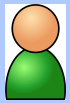
Step 5: Production submission and follow up

Steps and Players



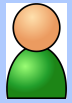
Physics WG & Simulation software manager

Step 1: Preparation of decay description and configuration for new decay channel(s)



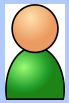
Release manager

Step 2: Release of new configurations and deployment on the Grid



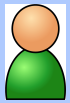
Simulation software manager

Step 3: Make the production system aware of the new event type(s)



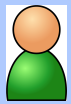
MC Physics WG liaison

Step 4: Submission of MC request(s)



(MC) Production manager

Step 5: Production submission and follow up



Physicists

Step 6: Retrieval of data samples produced for analysis

Event types and configuration

- In LHCb the majority of MC samples are proton-proton collisions with specific decay of b or c hadrons, …but not only.
- Pythia8 and other generators are used for the p-p collisions while EvtGen, an HEP-wide generator(*) is used to model the decay of all particles
  - Default behavior is governed by a general DECAY.DEC table with all known decay modes for all particles
  - User decay files are used to force a specific decay for a signal particle via specific models
- LHCb has an extended version of the user decay files with a steering section to generate the specific configuration of the generators to be used at run time to produce a given sample
- The decays files and their automatically generated options reside in a dedicated data package, DecFiles, linked in at run time by the Gauss simulation application

(*) http://evtgen.warwick.ac.uk

Event types and automatic generation of options

EventType: 8 digits number **"GSDCTNUX" to uniquely identify** each decay file, associated options and samples produced

Based on the nature and topology of the decay

Convention established and documented

Extending to 10 digits: migration to new schema for old samples

Descriptor: Details the decays in the file

Cuts: Generator level cuts. Each is implemented in a C++ class residing in a dedicated package

NickName: Short mnemonic: unique and matching the file name

```
# EventType: 11114005
#
# Descriptor: {[[B0]nos -> mu+ mu- (K*(892)0
-> K+ pi-)]cc, [[B0]os -> mu- mu+ (K*(892)~0
-> K- pi+)]cc}
#
# NickName: Bd_Kstmumu,phsp=DecProdCut,MomCut
#
# Cuts: DaughtersInLHCbAndWithMinP
#
# Documentation: Decay products in acceptance
and minimum momentum cut
# EndDocumentation
#
# PhysicsWG:   RD
# Tested:      Yes
# Responsible: John Doe
# Email:       John.Doe@mail.adrress
# Date:        20110928
```

Documentation: Documentation about the decay file. It will appear with the provenance information on a webdocumentation automatically made at release time

Event types and automatic generation of options

EventType: 8 digits number **"GSDCTNUX" to uniquely identify** each decay file, associated options and samples produced

Based on the nature and topology of the decay

Convention established and documented

Extending to 10 digits: migration to new schema for old samples

Descriptor: Details the decays in the fi

Cuts: Generator level cuts. Each is implemented in a C++ class residing in a dedicated package

NickName: Short mnemonic: unique and matching the file name

```
# EventType: 11114005
#
# Descriptor: {[[B0]nos ->
-> K+ pi-)]cc, [[B0]os
-> K- pi+)]cc}
#
# NickName
#
#
                                  acceptance
                Yes
ible:           John Doe
ail:            John.Doe@mail.adrress
# Date:         20110928
```

All used by a script run on demand or at release time to produce the options to be used by the simulation

Guidelines for all of them!

Documentation: Documentation about the decay file. It will appear with the provenance information on a webdocumentation automatically made at release time

Release of new Event Types

- Decay files for new event types are made continuously by physicists and added to the svn repository of a dedicated package. A notification is also entered in a tag collector.
- DecFiles managers check that rules are respected. Automatic test in nightly builds to verify new event types can be processed.

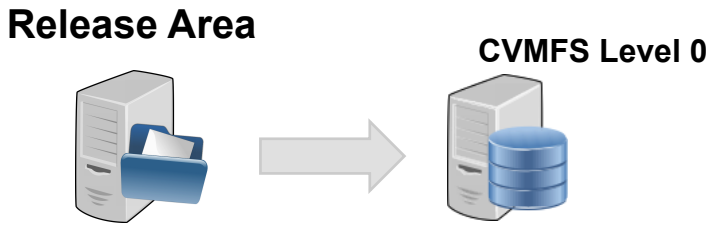
CMT lhcb-decfilestests:

Project	Version	x86_64-slc5-gcc46-opt completed at 0:22:35	
Gauss	v45r9		
DecFilesTests	HEAD	build	tests

- Release of DecFiles package asynchronous from that of the Gauss simulation application
  - Major released version number used to ensure compatibility
  - Version to be used is specified in the production system

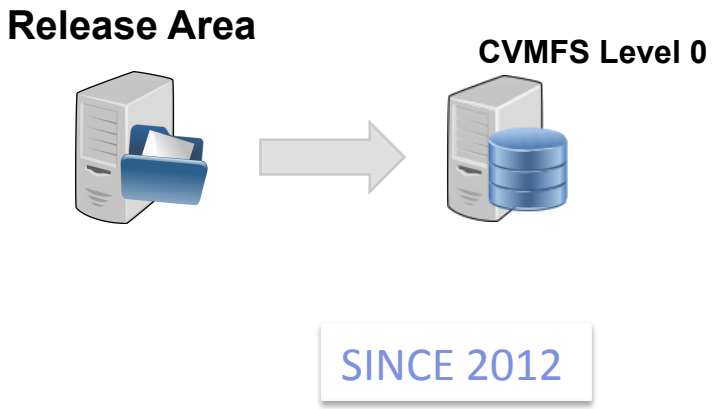
World Wide Deployment

- Release and deployment of DecFiles packages by deployment shifters via common LHCb distribution tools



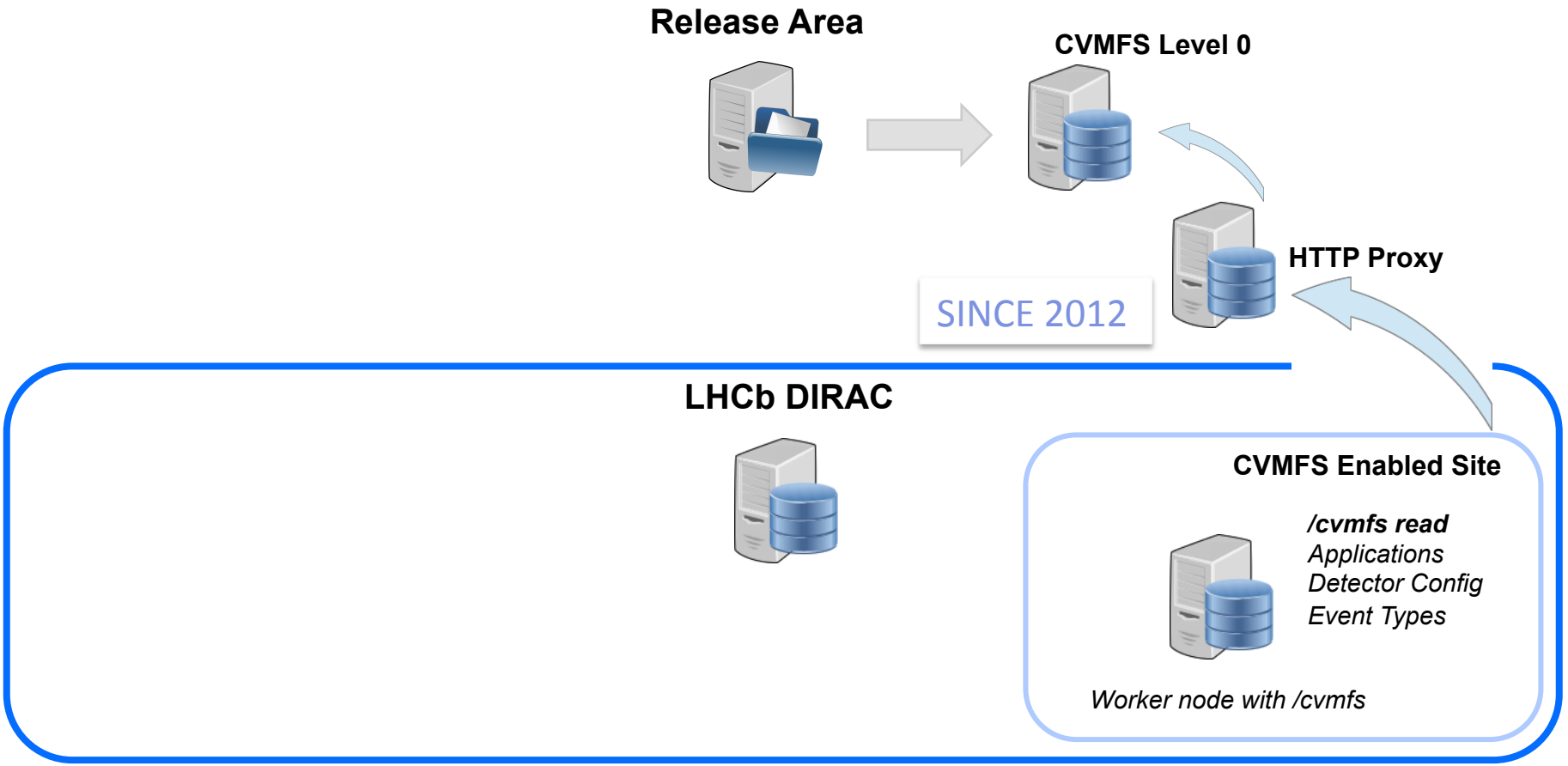
World Wide Deployment

- Release and deployment of DecFiles packages by deployment shifters via common LHCb distribution tools



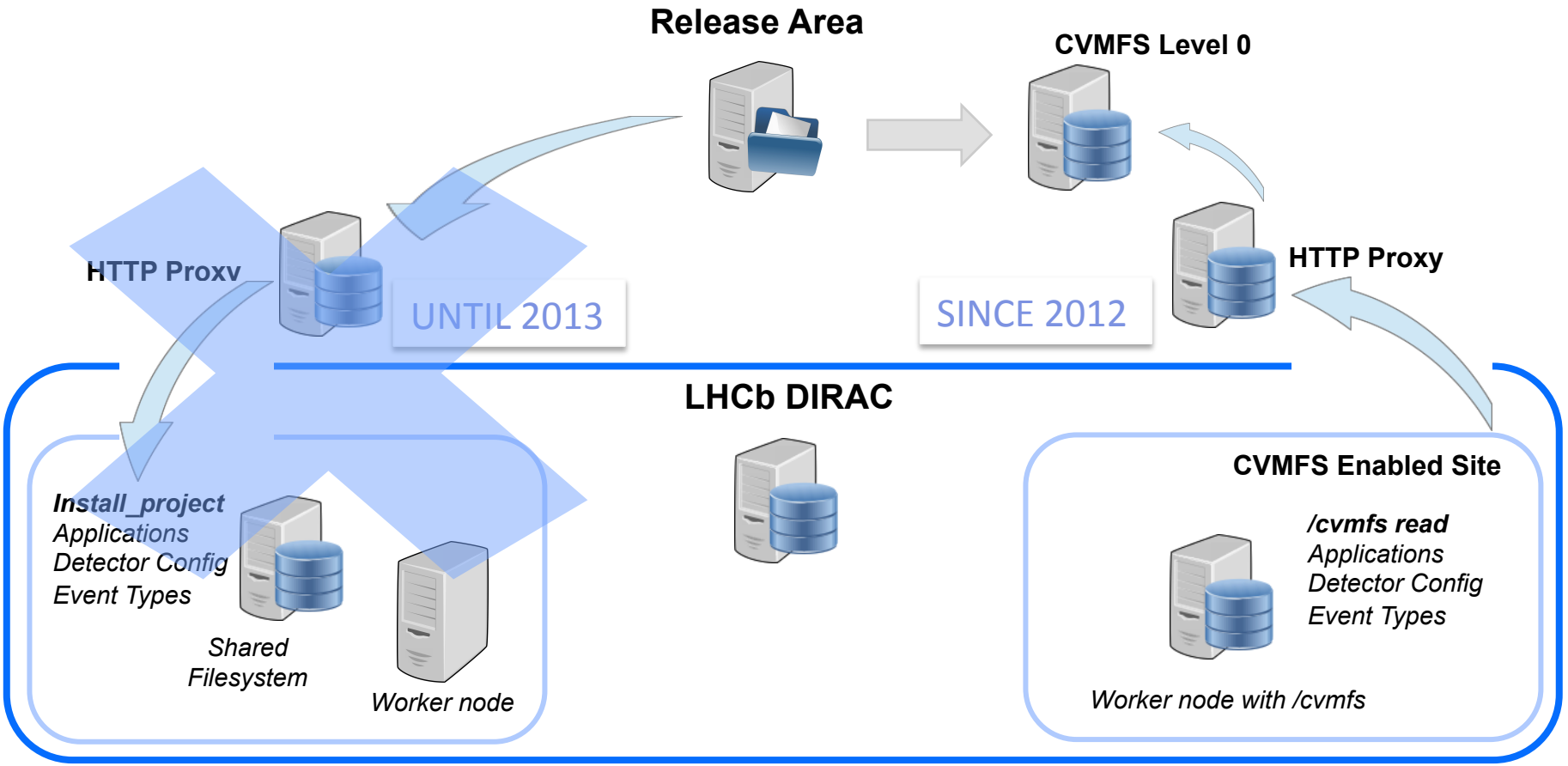
World Wide Deployment

- Release and deployment of DecFiles packages by deployment shifters via common LHCb distribution tools



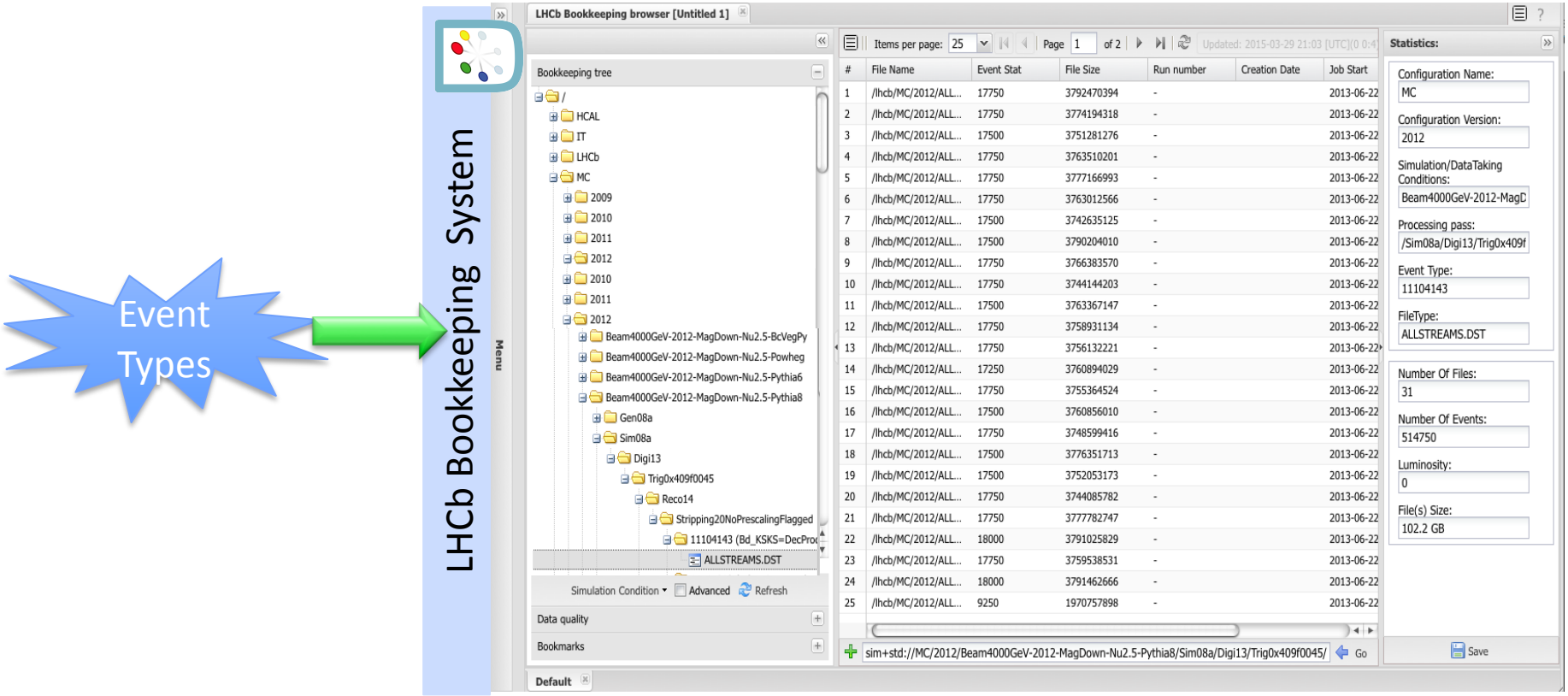
World Wide Deployment

- Release and deployment of DecFiles packages by deployment shifters via common LHCb distribution tools



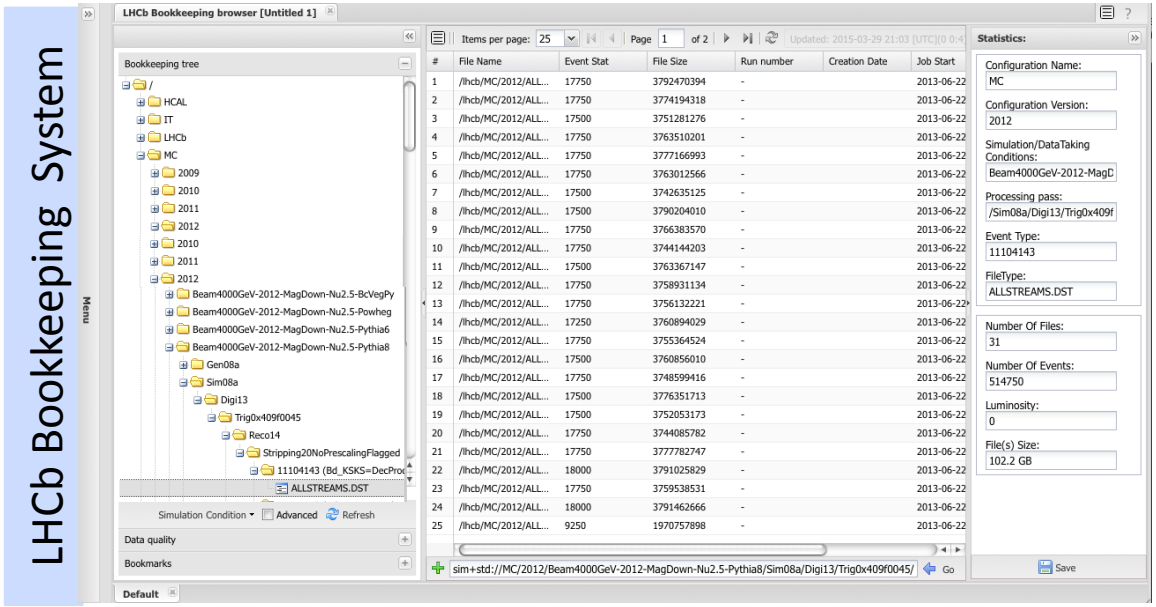
Registration to the Bookkeeping System

A dedicated step in the release procedure is to register the new Event Type into the LHCb Bookkeeping system



Registration to the Bookkeeping System

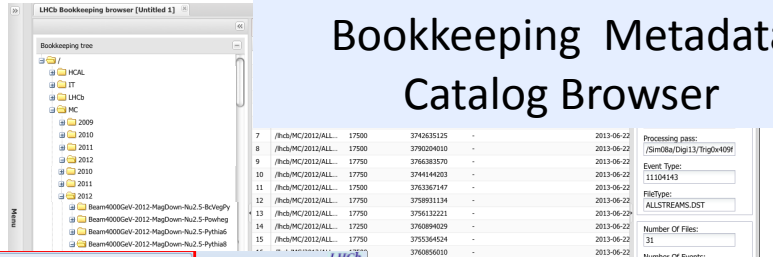
A dedicated step in the release procedure is to register the new Event Type into the LHCb Bookkeeping system



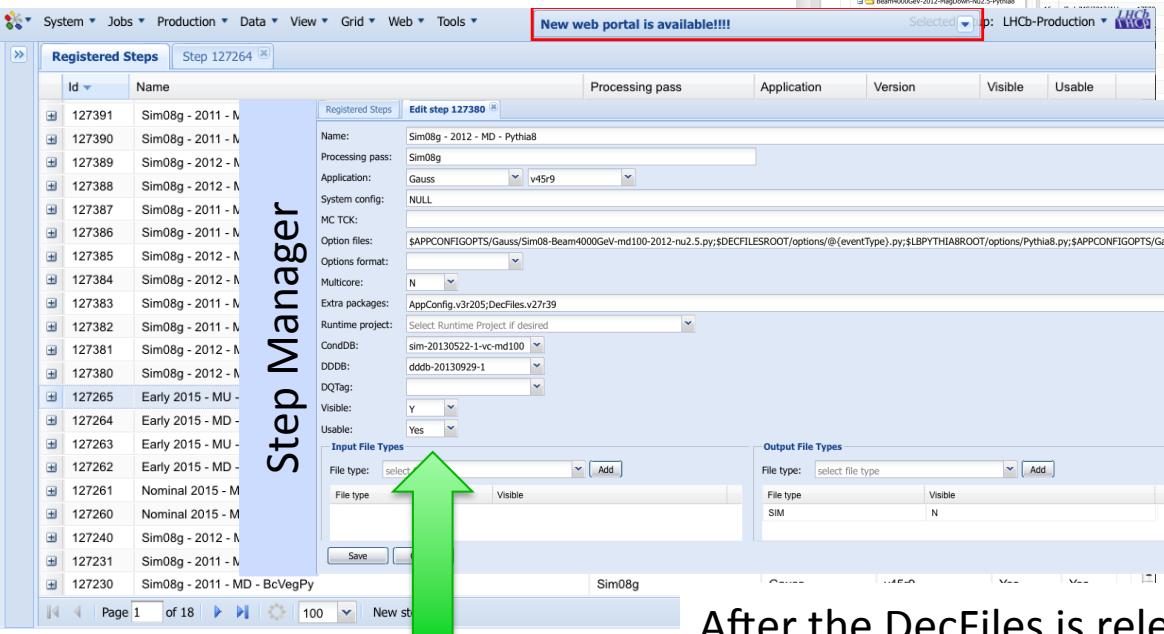
- The LHCb Bookkeeping system is a metadata management system which stores the conditions relative to jobs, files and their metadata, as well as their provenance information in an organized way
  - Data stored in relational model and presented as a hierarchical structure (tree-like format) to the users.
- It has a Web User Interface (WUI), a Graphical User Interface (GUI) and a Command Line Interface (CLI).
- It is based on Oracle RDBMS which has to fulfill the following requirements: rapid response, high throughput, availability and scalability.

Using the new DecFiles version in production

Event Types



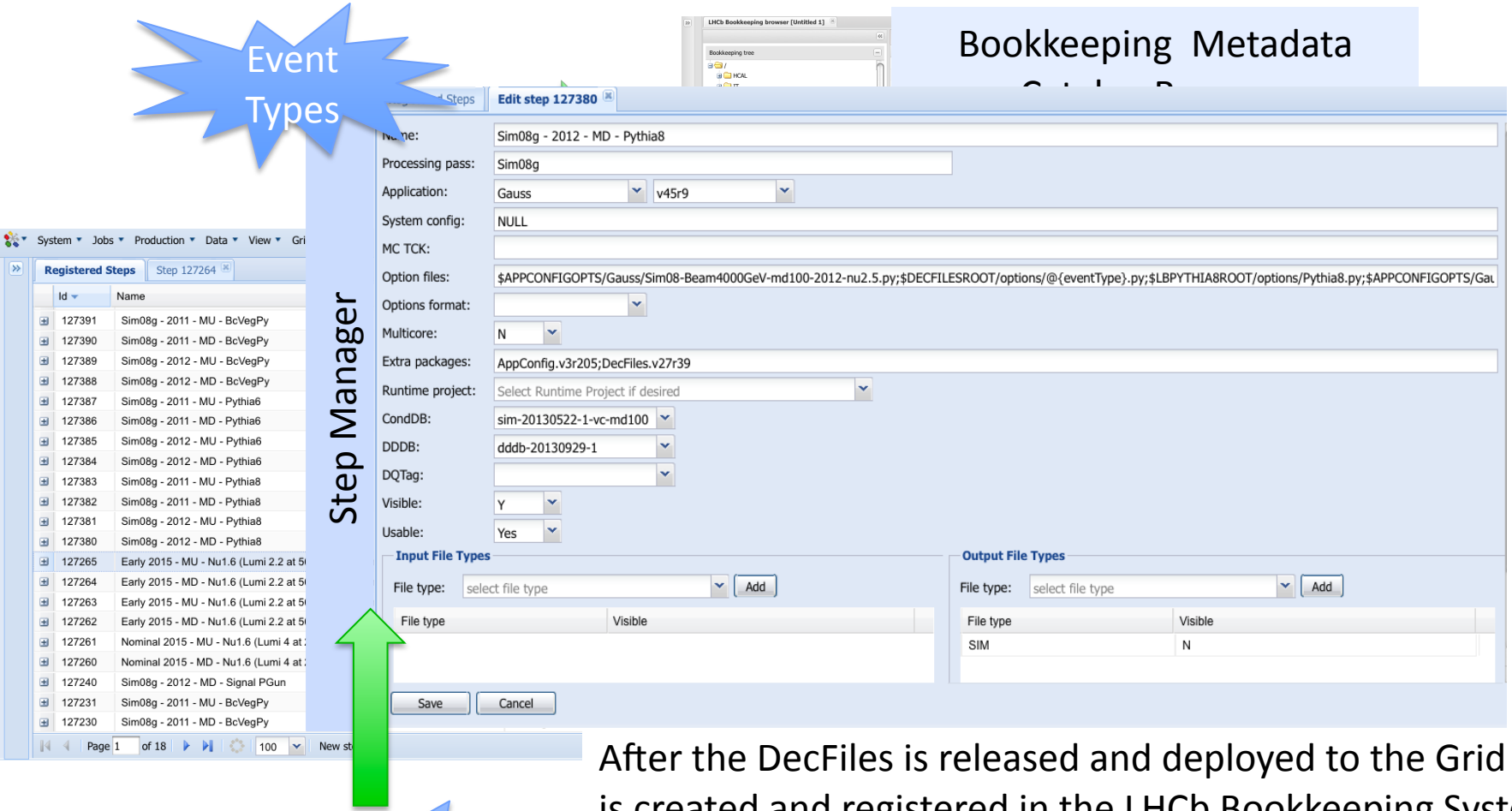
Bookkeeping Metadata Catalog Browser



Application Manager

After the DecFiles is released and deployed to the Grid a Step is created and registered in the LHCb Bookkeeping System. This Steps is an abstraction of an application and its configurations (such as different option files, database tags, etc.) which can be executed during data processing.

Using the new DecFiles version in production

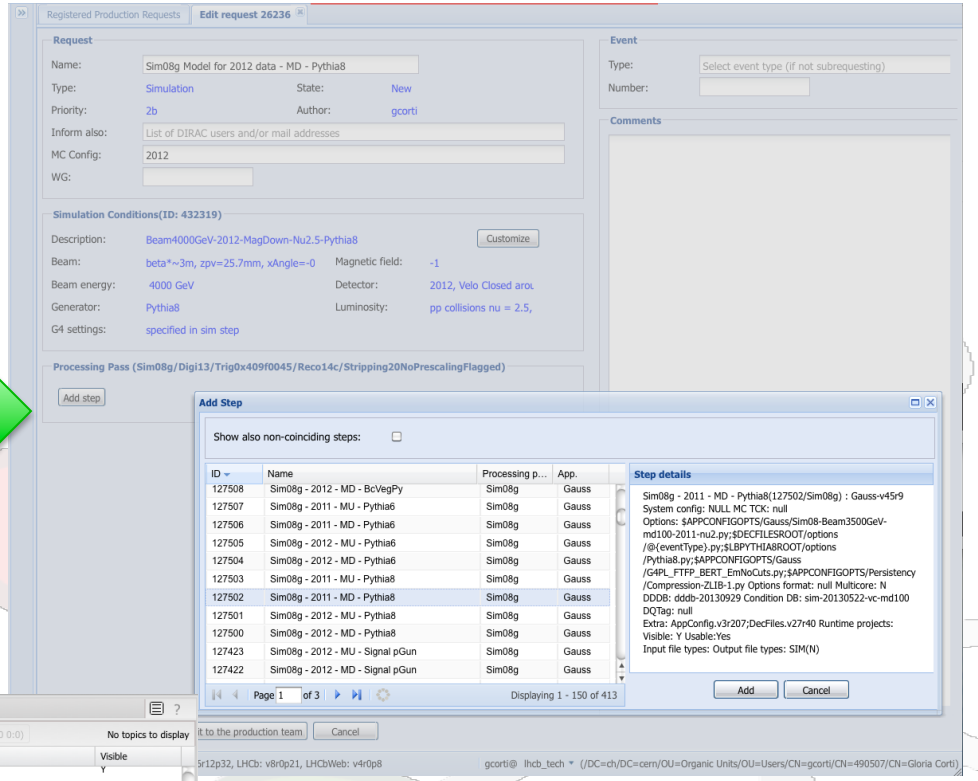


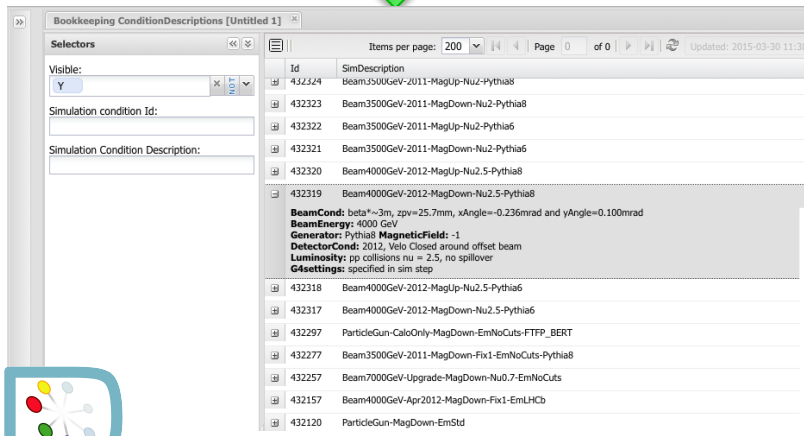
After the DecFiles is released and deployed to the Grid a Step is created and registered in the LHCb Bookkeeping System. This Steps is an abstraction of an application and its configurations (such as different option files, database tags, etc.) which can be executed during data processing.

Requests' configurations

A Request Manager page allows Simulation Managers to link steps with all application to be executed together, creating models for production requests

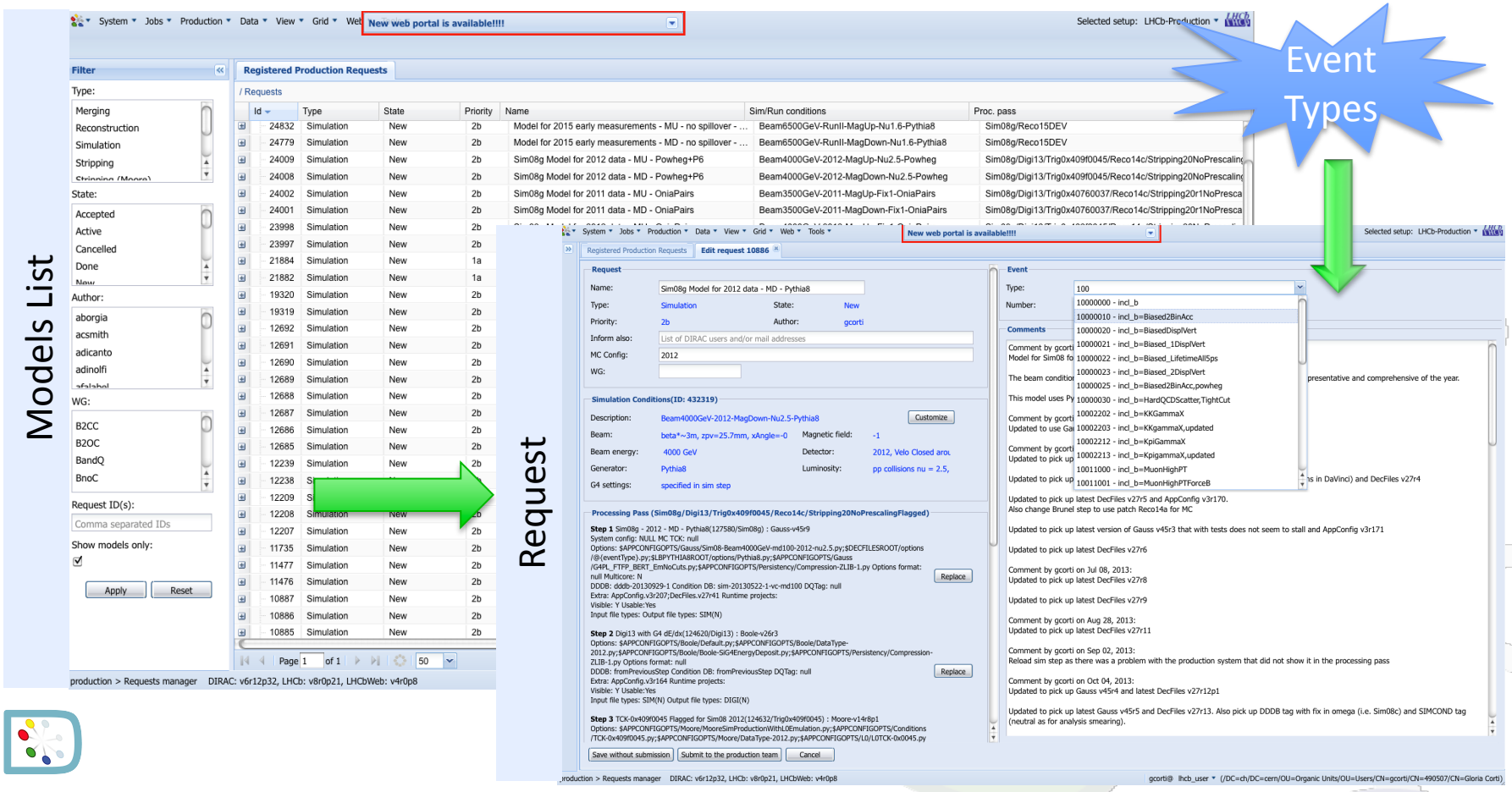
And to define simulation conditions





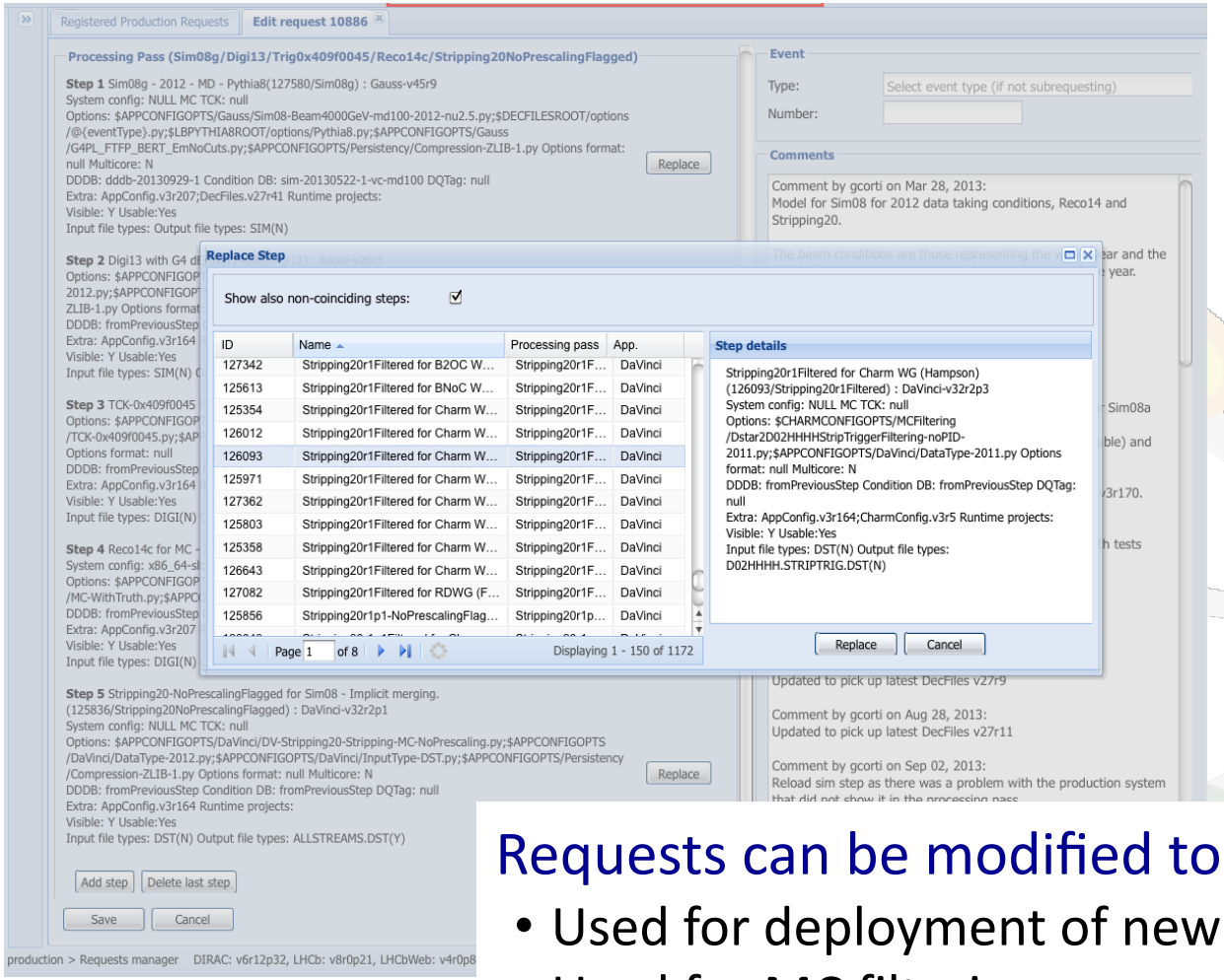
Simulation conditions and processing through all applications are fixed while the Event type is left as a free parameter

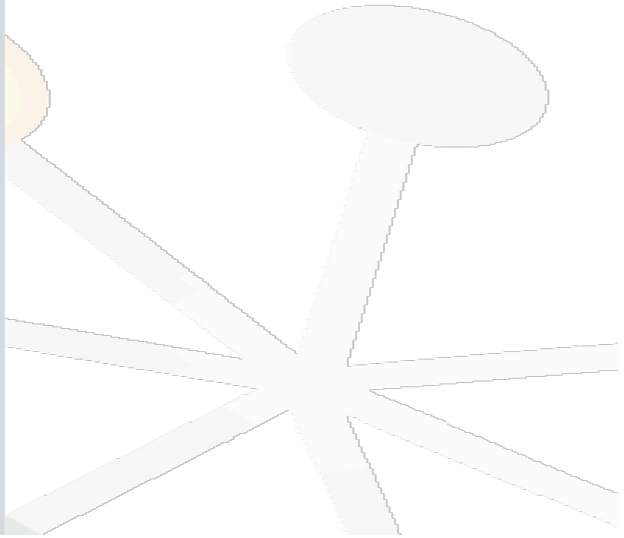
Request models and request



The Request Manager page also allows users to create their simulation requests which will perform on the Grid ‘cloning’ existing models and only specifying the event types and statistics

Request customization and filtering

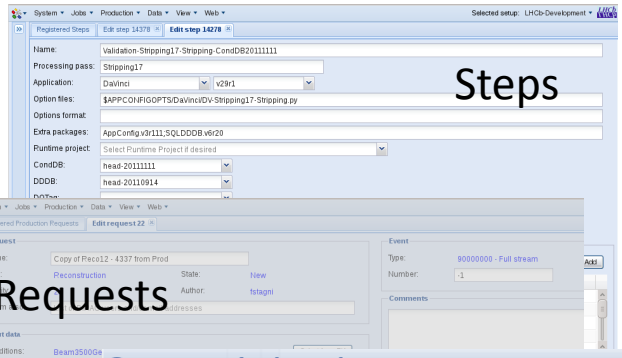


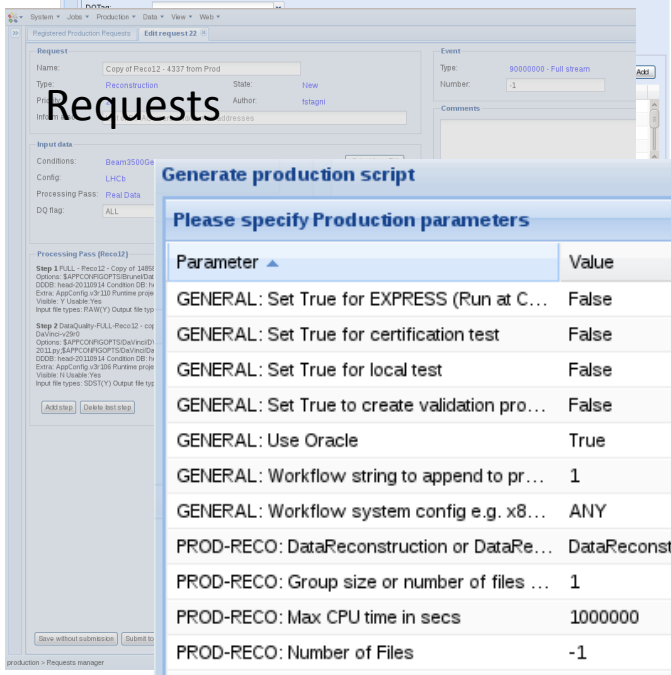


Requests can be modified to replace a single step

- Used for deployment of new decay files versions
- Used for MC filtering on a specific stripping line
  - MC filter Manager sets up different configurations from specific WG packages, e.g. B2OCConfig

LHCbDirac Production System

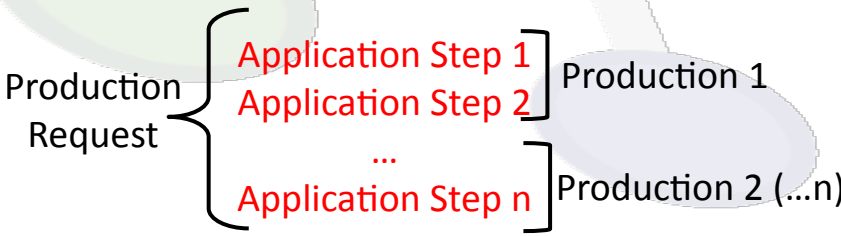




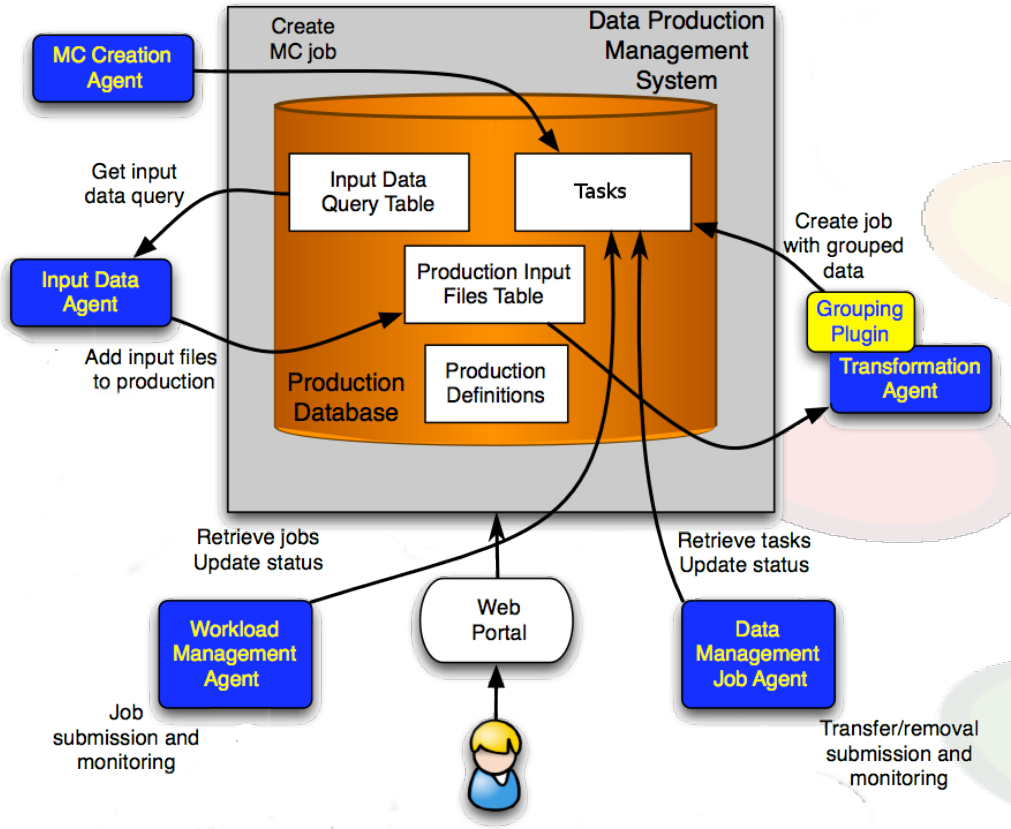
Production Templates

1. Application Managers generate application steps: a job step description
2. Simulation Managers link steps together, creating production requests
3. Physics WG make request for given even types by using a model and selecting the event type

4. Production Managers submit the production requests, monitored by the production team



LHCbDirac Production System cont.



Production requests description becomes workflows, and then DIRAC jobs, executed on the available computing resources.
Productions are extended, and closed automatically when the requested events are produced
(elastic grid jobs)

see F. Stagni, "Jobs masonry with elastic Grid Jobs"
Track #4 (Distributed Computing), Session #2, on Mon. 13th

MC requests workflow

Physics WG Liaison

Collects needs and submit requests

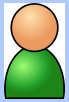
New/Submitted

MC requests workflow

Physics WG Liaison

Collects needs and submit requests

New/Submitted

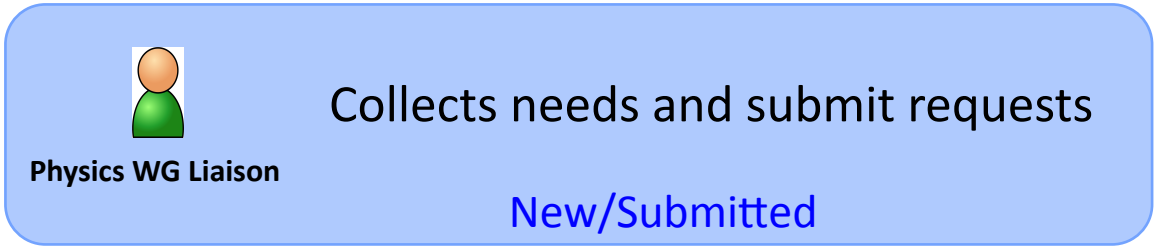


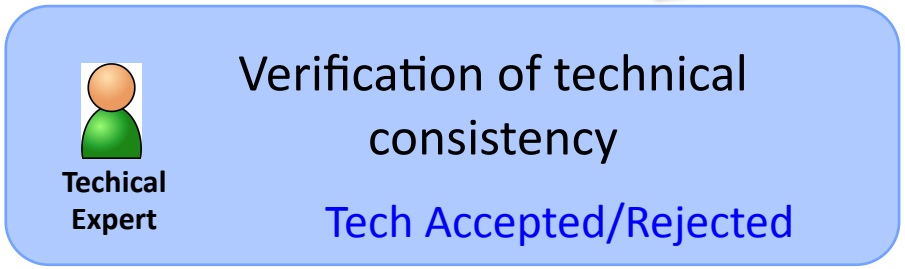
Techical Expert

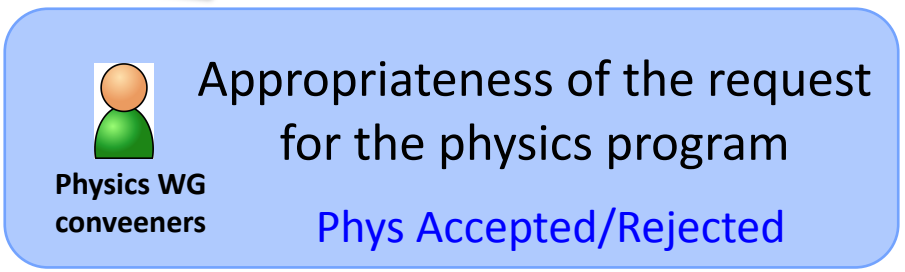
Verification of technical consistency

Tech Accepted/Rejected

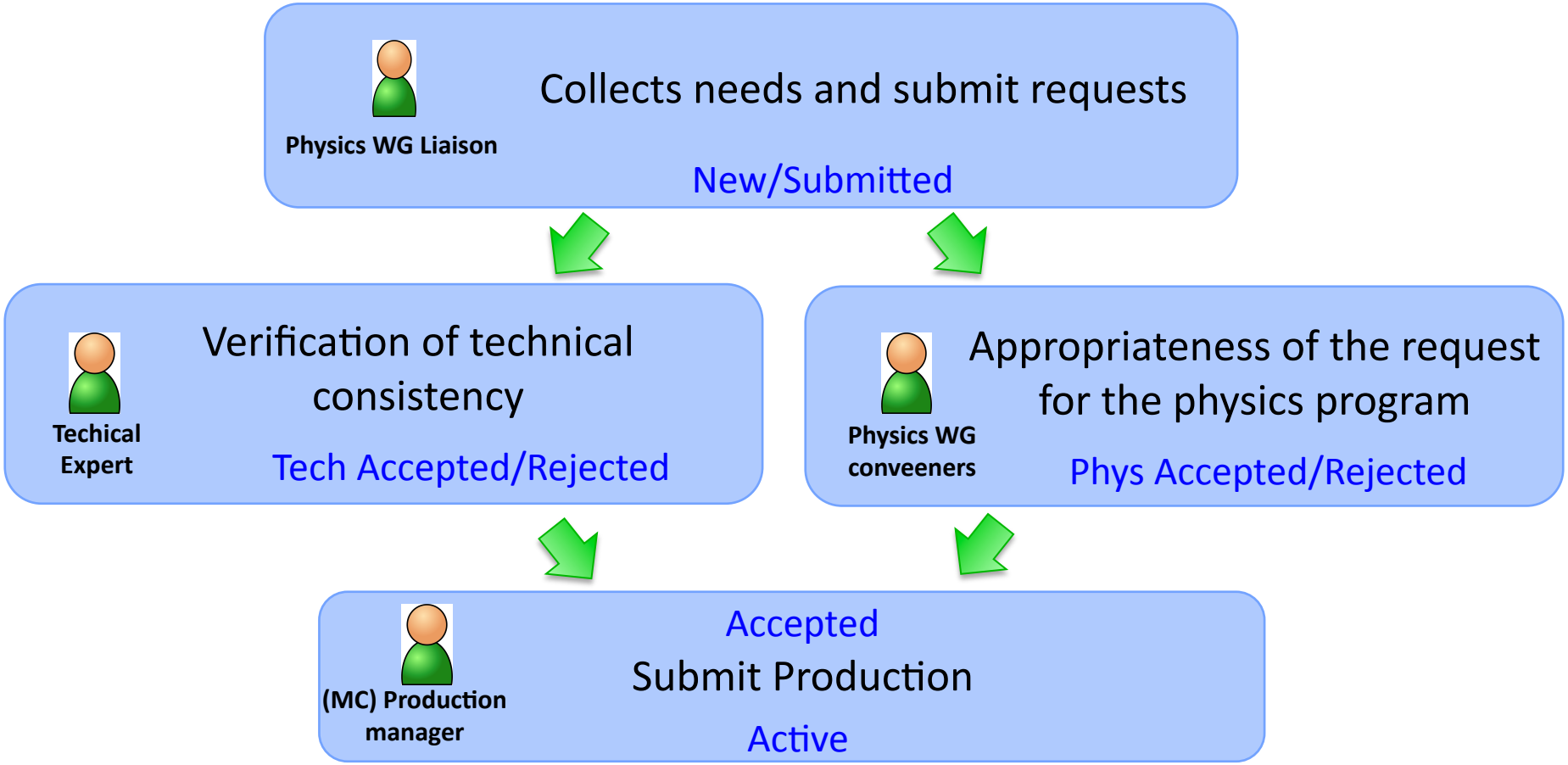
MC requests workflow



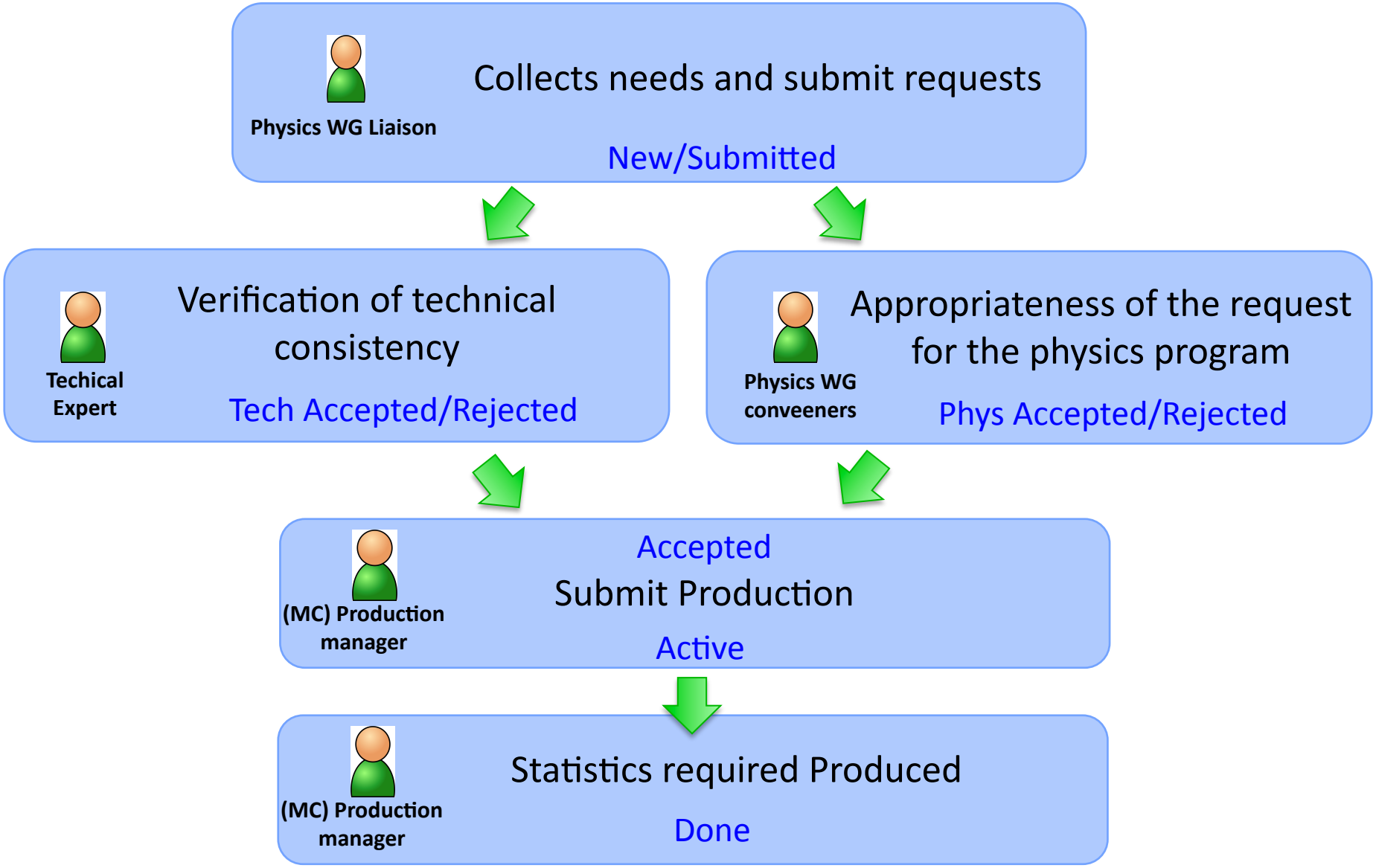




MC requests workflow

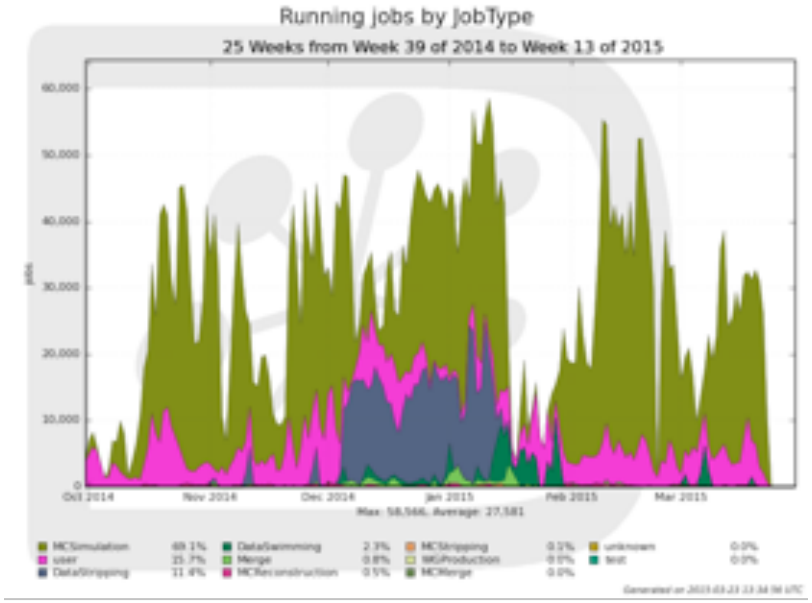


MC requests workflow



Priorities

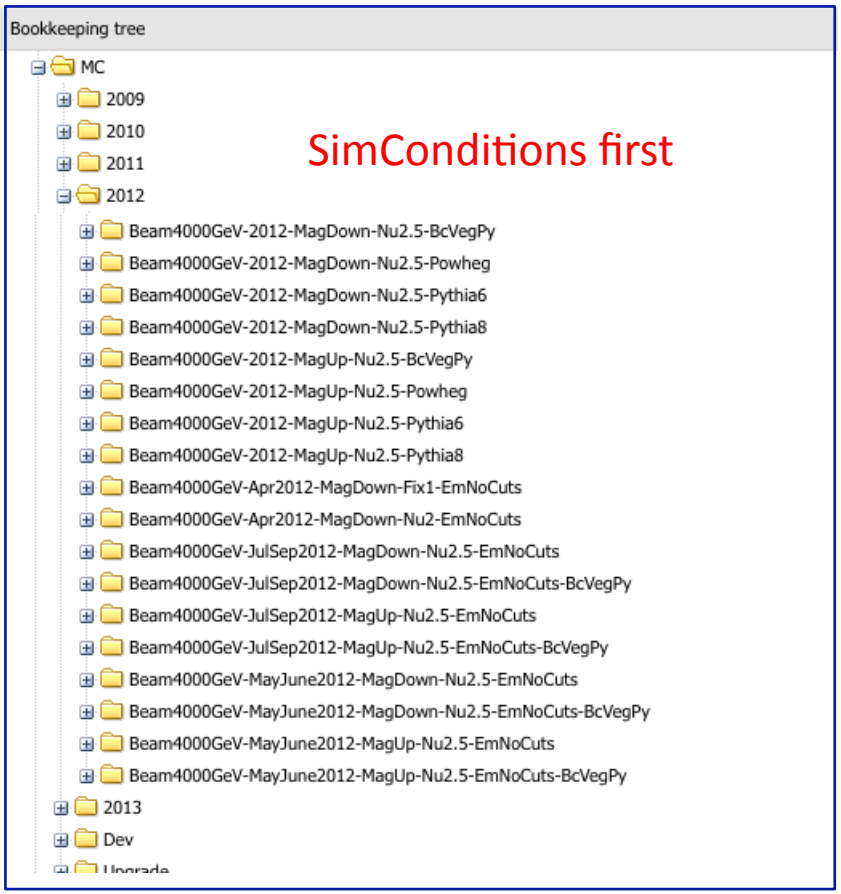
- Relevance and urgency of a production request are evaluated by the Physics WG group conveners
- The MC production manager does final verification with small statistics to verify the request can be processed. The size of the jobs is determined automatically
- With both information the MC production manager checks the computing resources available and gives priority to the various requests for Monte Carlo samples
  - Taking also into account processing activities for real data

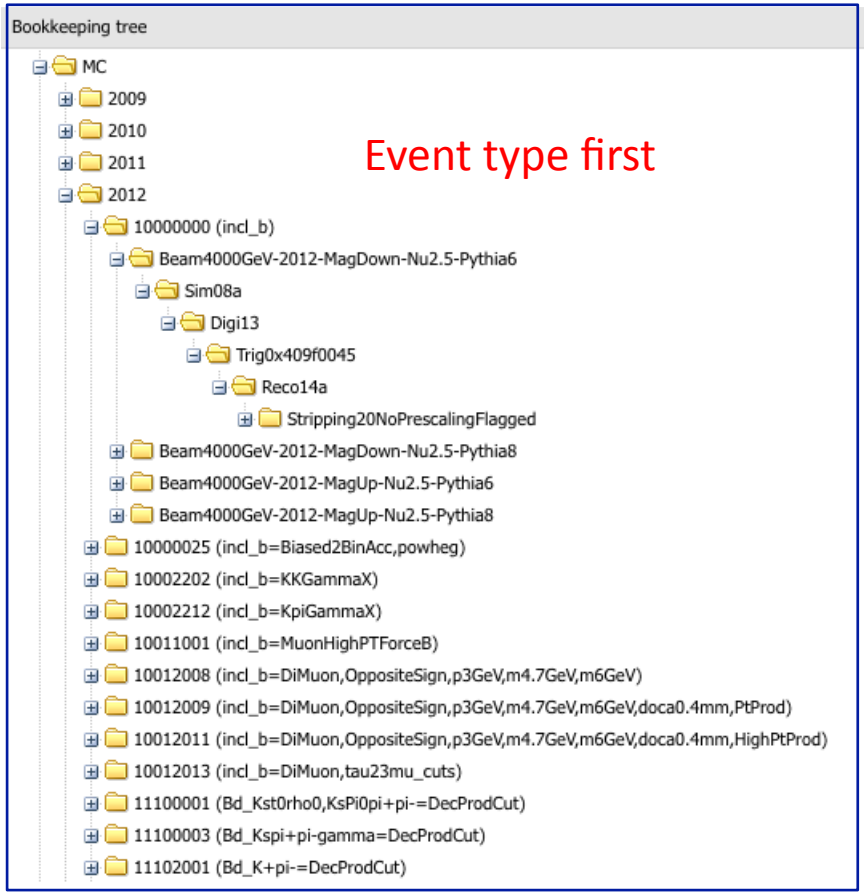


Finding MC datasets

The metadata information of the tasks executed on the Grid will be uploaded to the Bookkeeping Metadata catalog at the end of the job as provenance of the data

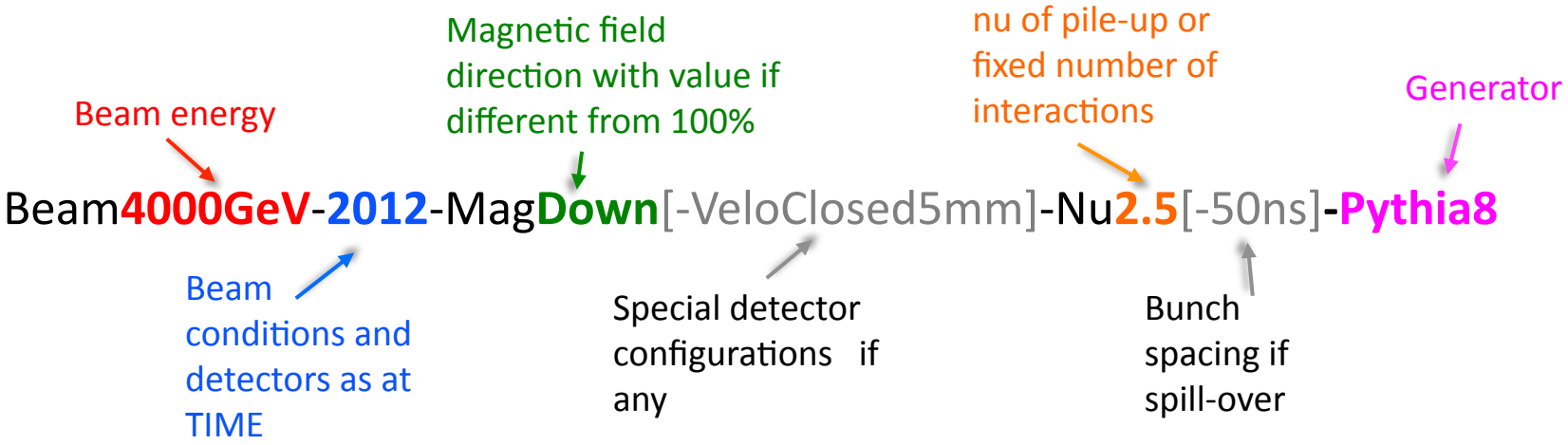
The whole of LHCb has access to all MC samples via the Bookkeeping



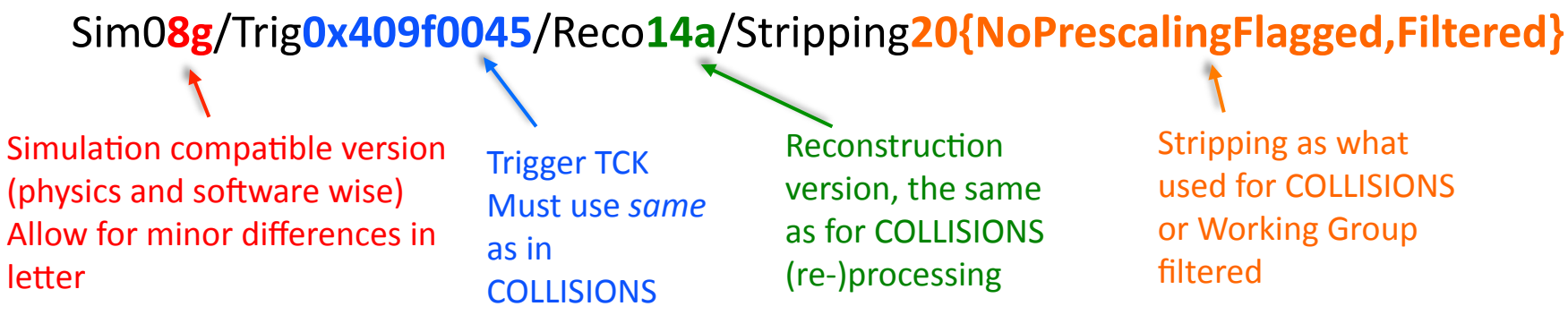


… and their details

- in the Simulation Conditions



- and in the Processing Pass



Conclusions

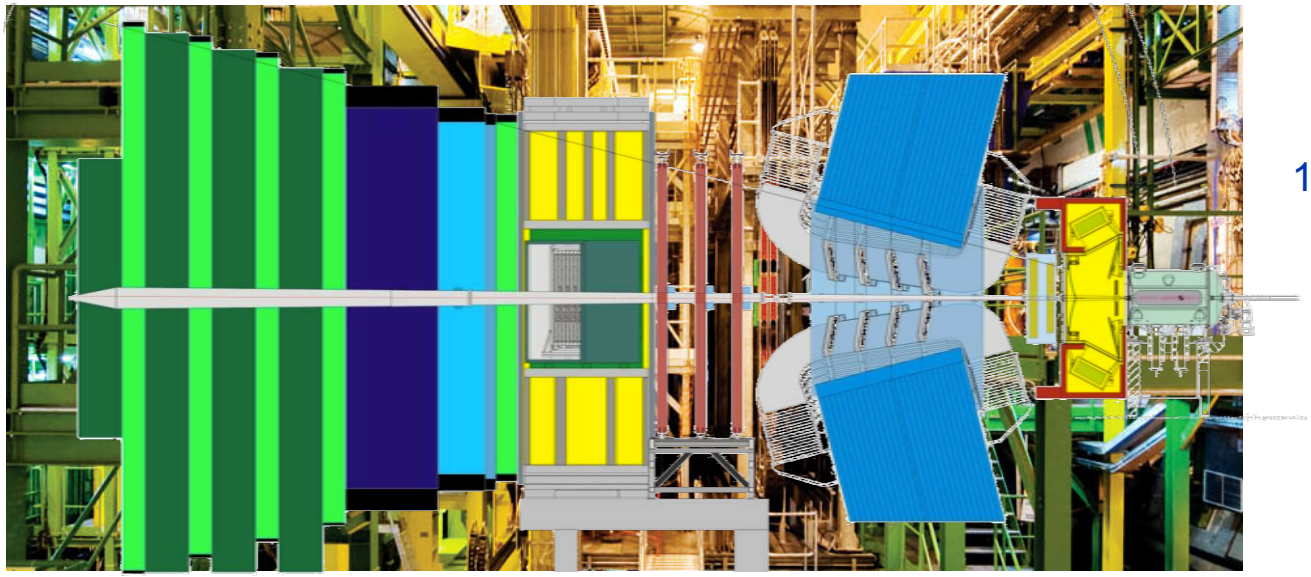
- Conventions and procedures are implemented in LHCb to handle Monte Carlo productions centrally and in a transparent way
  - LHCb standard tools and common computing infrastructures used throughout the whole
  - Automatation and tracking between the steps
  - Unique numerical identifier of Monte Carlo event types through all the steps of software deployment and productions is a major key element
- Experts concentrate on their task
  - Physicist and software simulation experts on configuration of application
  - Production team on job submission and follow-up and data storage
- Allows for **massive** transparent and efficient **production** on a world-wide distributed system **with very little manpower**
  - One part time MC production manager
  - Minimal time of three simulation software experts

BACKUP

LHCb

- Designed to search for New Physics through precision measurement of CP violation and Rare Decays of heavy flavours at the LHC
  - Initial aim of LHCb was b-physics but also doing charm and QCD physics
- Trigger and reconstruct many different decay modes to make independent and complementary measurements

LHCb is a single arm forward spectrometer



Forward production of $b\bar{b}$, correlated

$12 \text{ mrad} < \theta < 300\ (250) \text{ mrad}$

i.e. $2.0 < \eta < 4.9$

EventType and options

- Eight digits number of type **"GSDCTNXU" to uniquely identify** each decay file, associated options and samples produced
- Convention and extensions established and documented in LHCb notes:
  - First six numbers describe the decay and the last two distinguish between similar decays

G:	General event type and production scheme.
S:	Value based on the presence of certain particles.
D:	Number depends on the general features of the decay.
C:	Based on the number of charm hadrons and leptons.
T:	Number of stable charged particles: p, π, K, e and μ.
N:	Number of neutrals : K_S, Λ, K_L, γ, n, π and ν.
X:	Used to distinguish between different decays that share the same first 6 digits
U:	Used to distinguish between the same decay, but different model, cuts, options

  - Need to extend to 10 digits: ensure migration to new extended schema for old samples