Software & Analysis in CMS

L. Silvestris

I.N.F.N. – Bari

CERN School Thailand 2010

ChulalongKorn University

Bangkok Thailand

12-13 October 2010

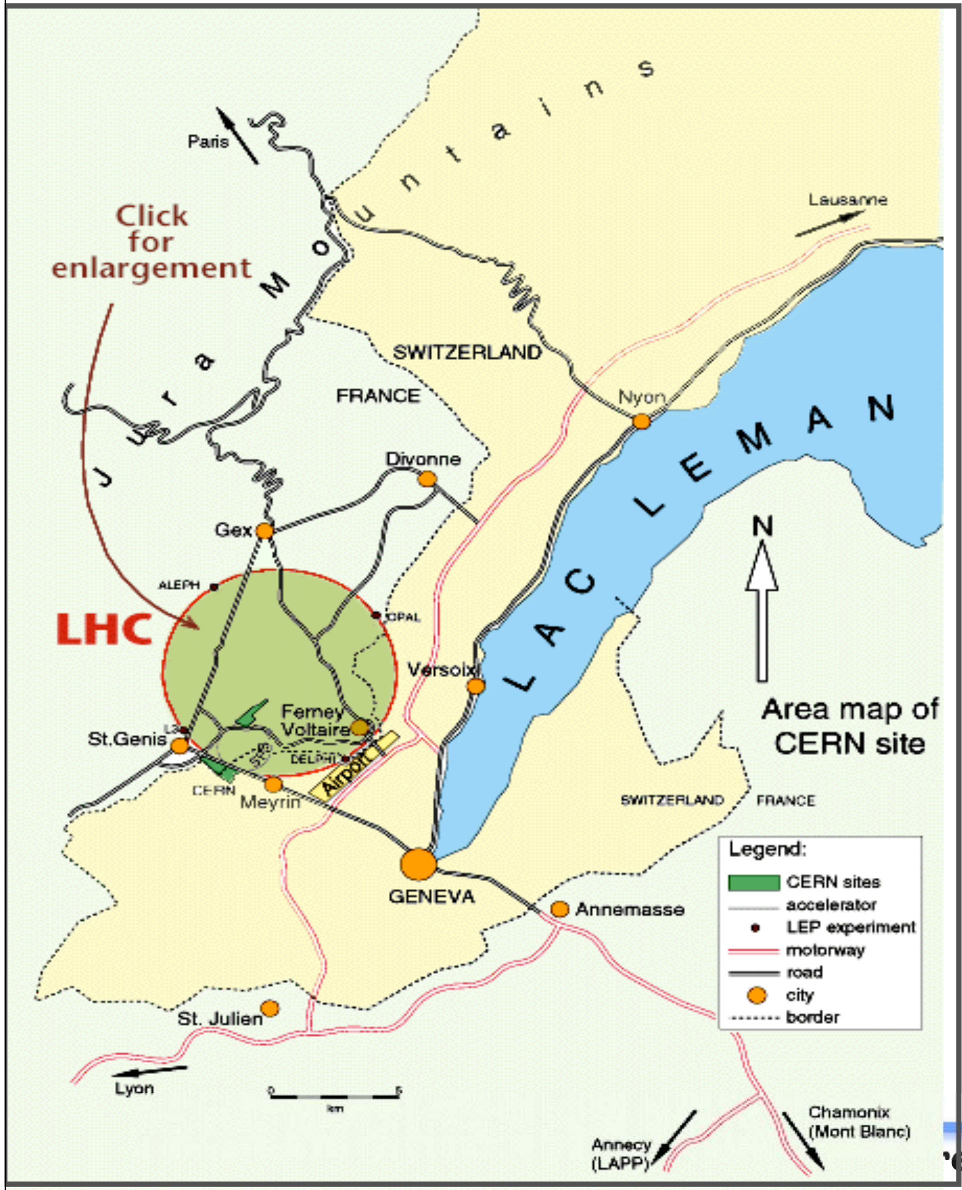
Outline For These Lectures

- ★ Introduction to LHC & CMS Experiment
- ★ LHC Common Software
- ★ CMS Software
  - ★ CMS Framework and Event Data Model
  - ★ Calibration and Alignment: Non Event Data Model
  - ★ Data Format
  - ★ FWLite and cmsRun
  - ★ Software Development Tools and Releases
  - ★ Geometry and Simulation
  - ★ Event Visualization
  - ★ Trigger and Reconstruction
- ★ DCMS Analysis
- ★ Data Flow, Offline & Computing Operations

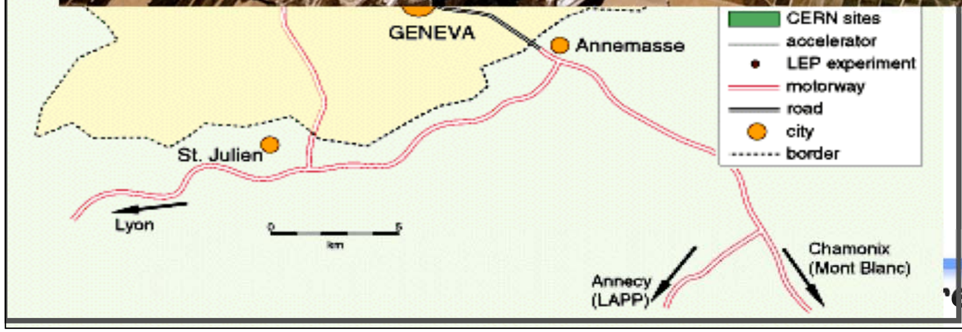
Large Hadron Collider &

CMS Experiment

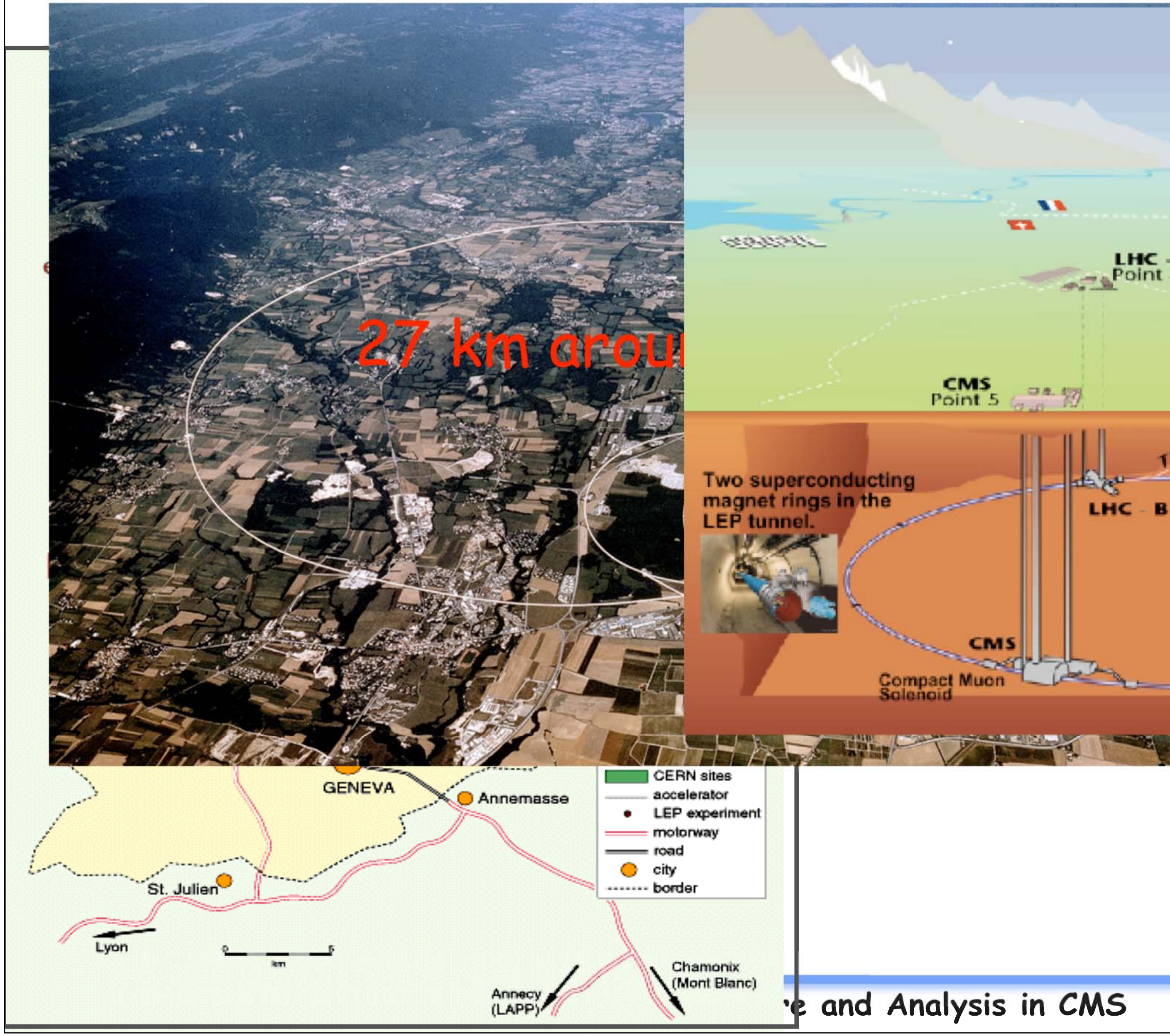
Large Hadron Collider

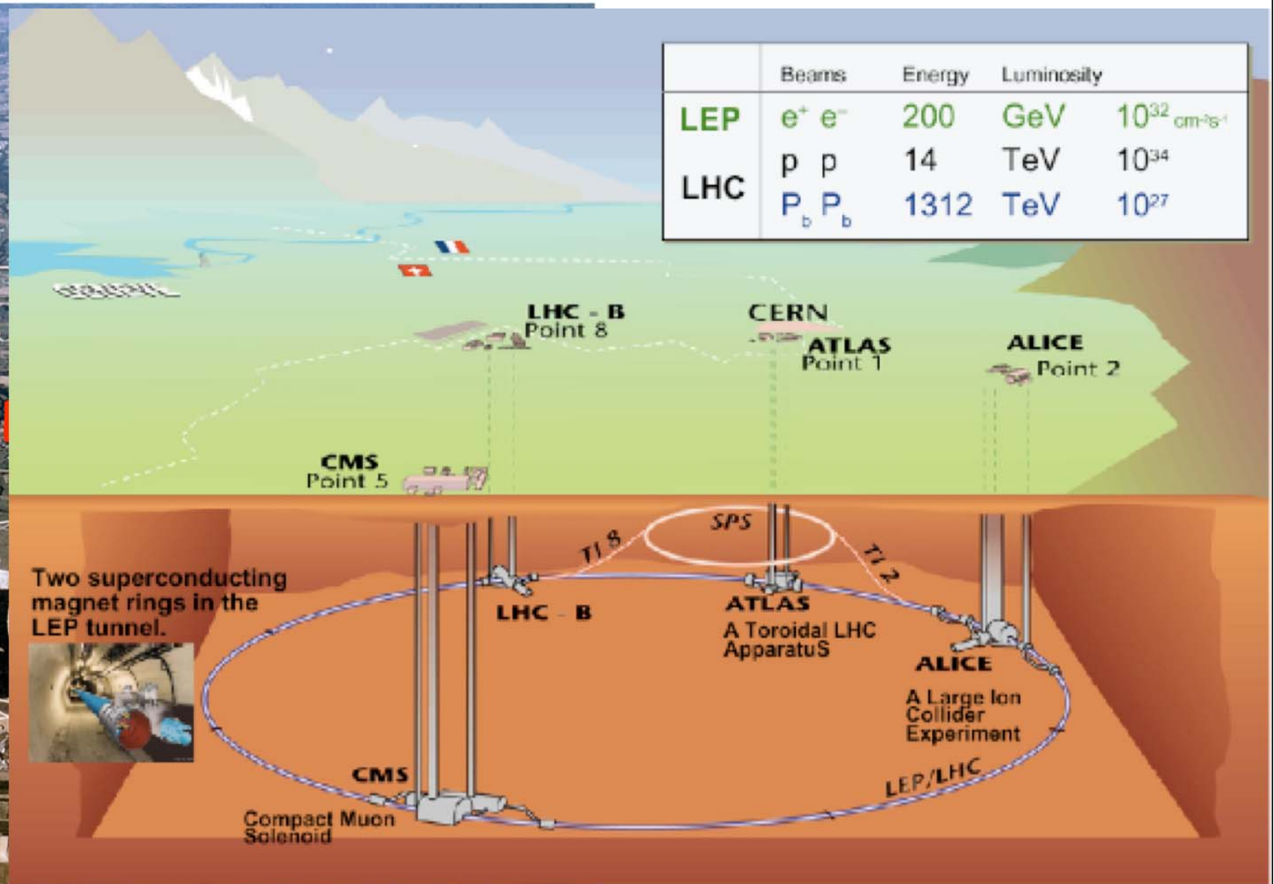


Large Hadron Collider

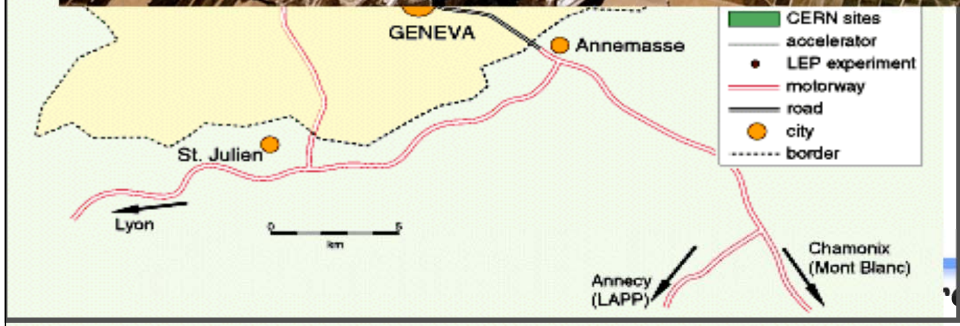


Large Hadron Collider

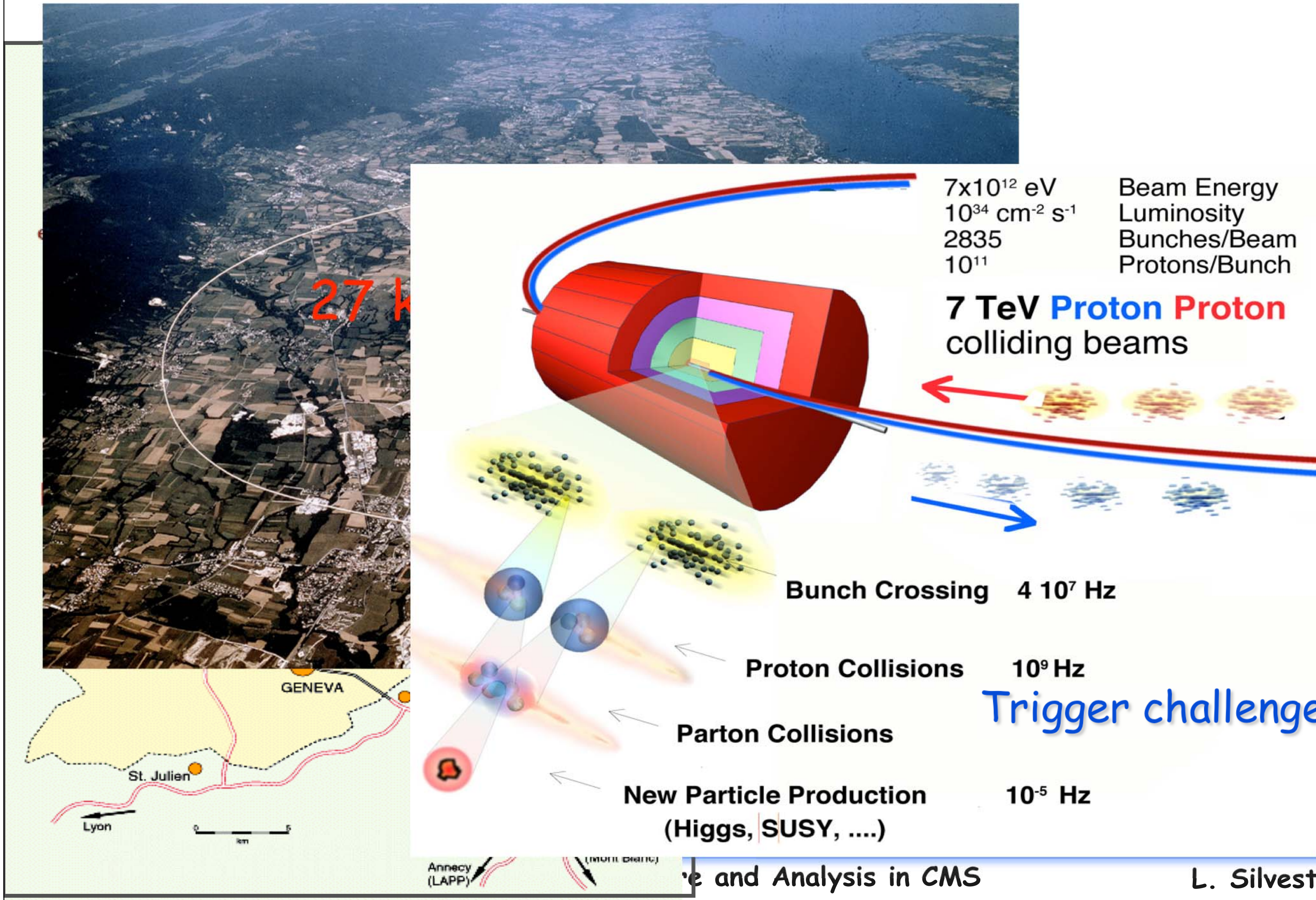




Large Hadron Collider



Large Hadron Collider



Trigger challenge task !!

GOOD and BAD at LHC

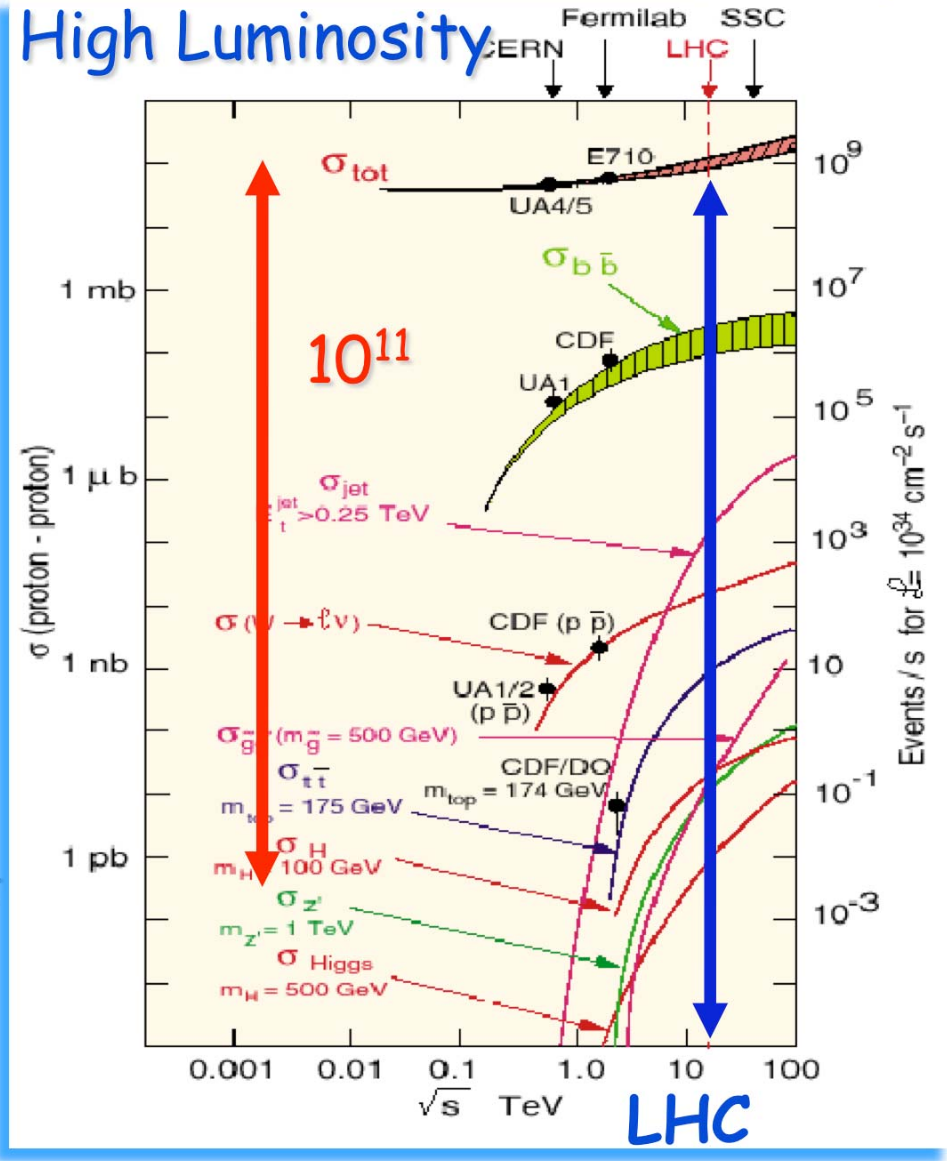
Cross-sections of physics processes vary over many orders of magnitude:

- inelastic: 10^9 Hz
- b b production: 10^6-10^7 Hz
- $W \to l \nu$: 10^2 Hz
- t t production: 10 Hz
- Higgs (100 GeV/c^2): 0.1 Hz
- Higgs (600 GeV/c^2): 10^{-2} Hz

Only 100 ev/sec on tape for ALL interesting events

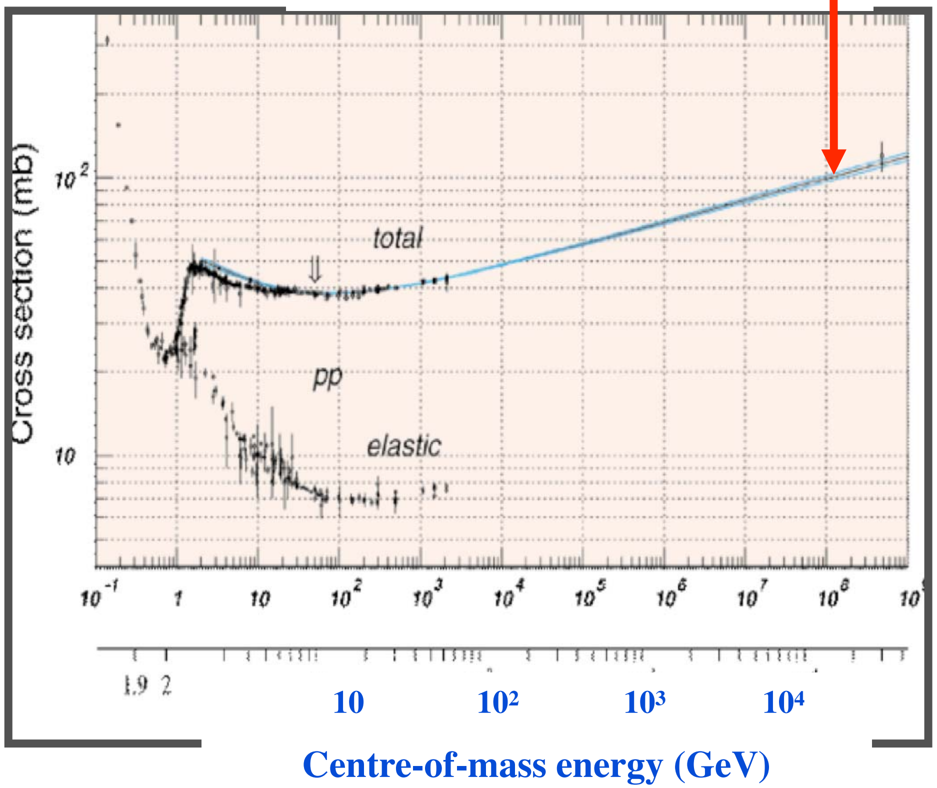
→ Selection needed: $1:10^{10-11}$

Trigger is a challenging task at LHC



Minimum bias e pileup per bunch

$\sigma_{tot}(pp) \approx 100$ mb



σ_{tot} (pp) and $\sigma_{inel} = \sigma_{tot} - \sigma_{el} - \sigma_{diff}$

@ LHC $\sigma_{inel} \approx 70$ mb

Pileup:

$\langle n \rangle = \sigma_{inel} \times L \times \Delta t$

$= 70 \text{ mb} \times 10^{34} \text{ cm}^{-2}\text{s}^{-1} \times 25 \text{ ns}$

$\cong 20$ interactions/BC

Big change with respect to previous machines:

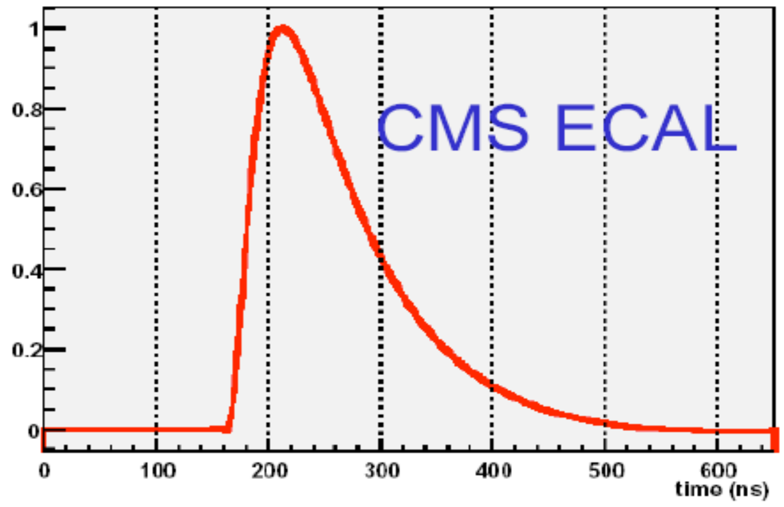
LEP:	$\Delta t = 22\ \mu s$	$\langle n \rangle \ll 1$
SppS:	$\Delta t = 3.3\ \mu s$	$\langle n \rangle \approx 3$
HERA:	$\Delta t = 96$ ns	$\langle n \rangle \ll 1$
Tevatron :	$\Delta t = 3.5\ \mu s$	$\langle n \rangle \ll 1$
Tev RunII:	$\Delta t = 0.4\ \mu s$	$\langle n \rangle \approx 2$

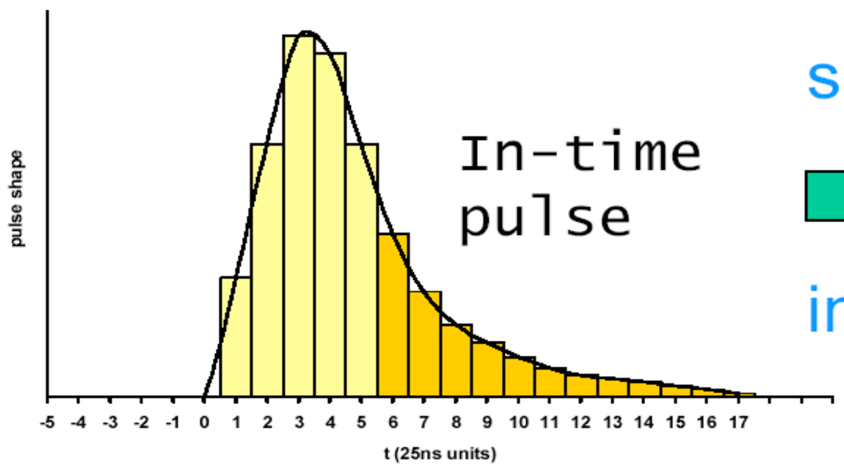
Pile-up & Electronics

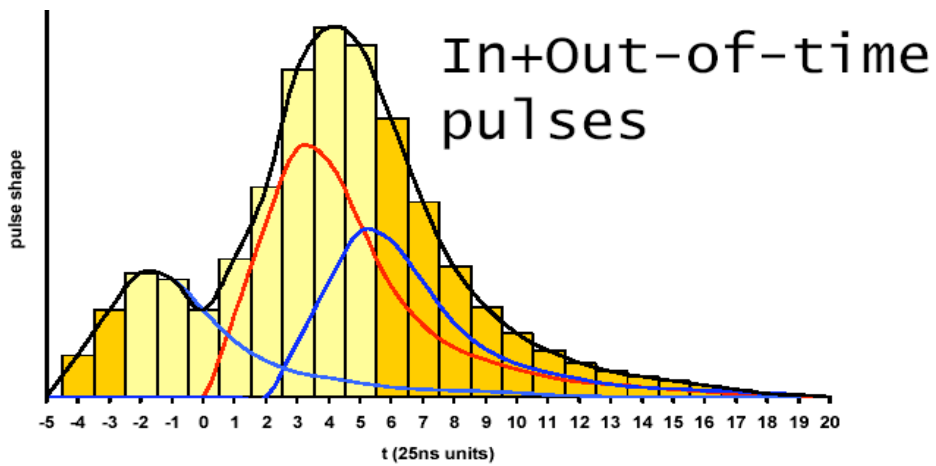
"In-time" pile-up: particles from the same crossing but from a different pp interaction

Long detector response/pulse shapes:

- "Out-of-time" pile-up: left-over signals from interactions in previous crossings
- Need "bunch-crossing identification"







Impact on detector design

LHC detectors must have fast response

- Otherwise will integrate over many bunch crossings → large "pile-up"
- Typical response time : 20-50 ns

  → integrate over 1-2 bunch crossings → pile-up of 25-50 min-bias

  → very challenging readout electronics

LHC detectors must be highly granular

- Minimize probability that pile-up particles be in the same detector element as interesting object (e.g. γ from $H \rightarrow \gamma\gamma$ decays)

  → large number of electronic channels

  → high cost

LHC detectors must be radiation resistant:

- high flux of particles from pp collisions → high radiation environment e.g. in forward calorimeters:
  - up to 10^{17} n/cm^2 in 10 years of LHC operation
  - up to 10^7 Gy (1 Gy = unit of absorbed energy = 1 Joule/Kg)

CMS (LHC) detectors Requirements

Basic principle: need "general-purpose" experiments covering as such of the solid angle as possible ("4π") since we don't know how New Physics will manifest itself

→ detectors must be able to detect as many particles and signatures as possible: e, μ, τ, ν, γ, jets, b-quarks, ….

Momentum / charge of tracks and secondary vertices (e.g. from b-quark decays) are measured in central tracker.

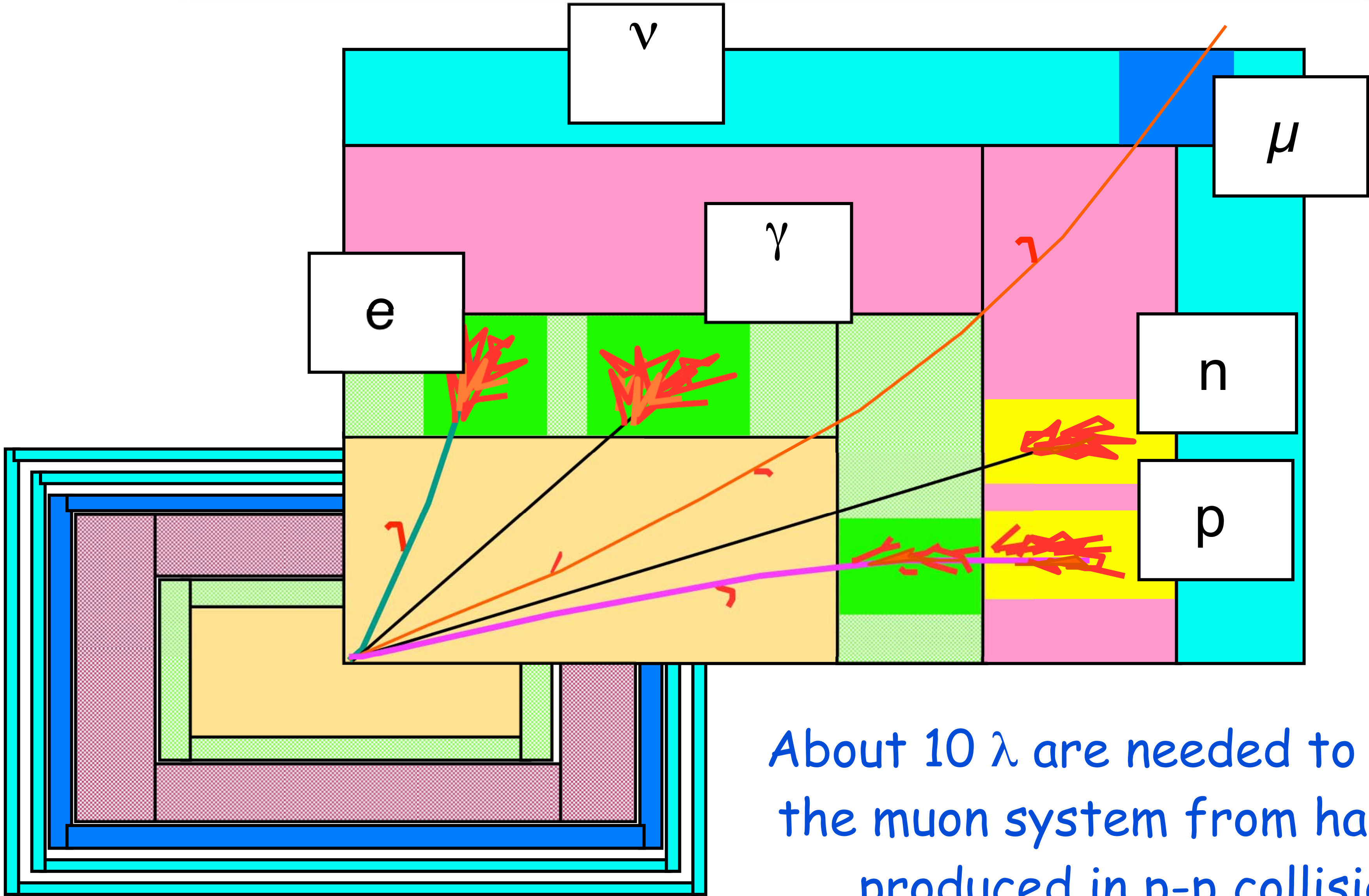
Energy and positions of electrons and photons measured in electromagnetic calorimeters.

Energy and position of hadrons and jets measured mainly in hadronic calorimeters.

Muons identified and momentum measured in external muon spectrometer (+central tracker).

Neutrinos "detected and measured" through measurement of missing transverse energy (E_T^{miss}) in calorimeters.

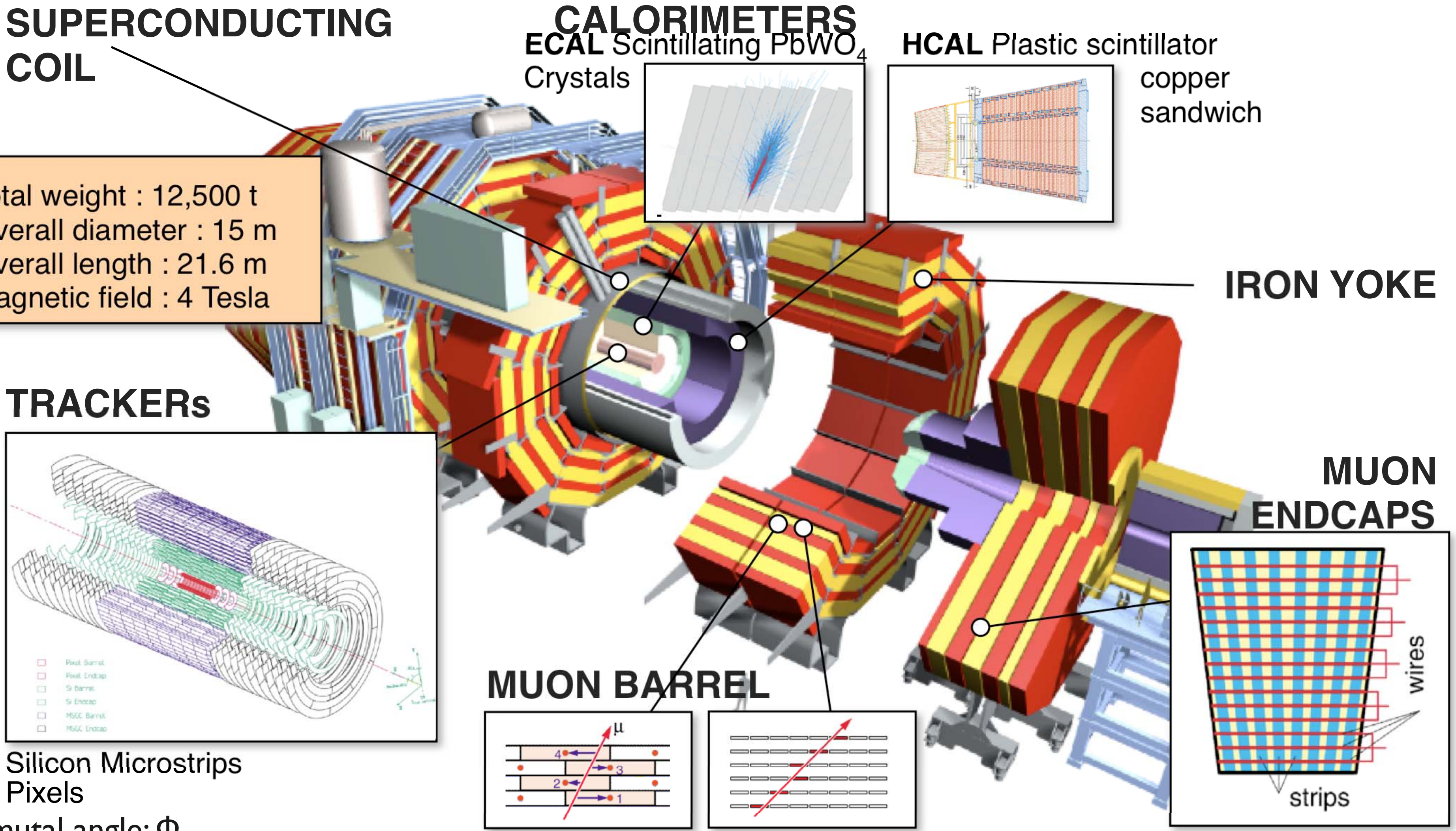
A Generic Multipurpose LHC Detector



About 10 λ are needed to shield the muon system from hadrons produced in p-p collision

The experiments: CMS

Total weight : 12,500 t
Overall diameter : 15 m
Overall length : 21.6 m
Magnetic field : 4 Tesla



Silicon Microstrips
Pixels

Drift Tube Chambers (**DT**)

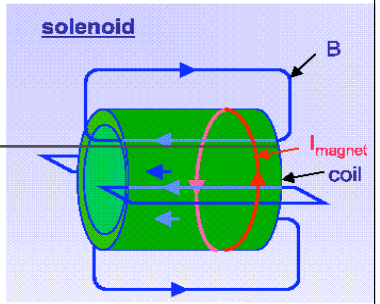
Resistive Plate Chambers (**RPC**)

Cathode Strip Chambers (**CSC**)
Resistive Plate Chambers (**RPC**)

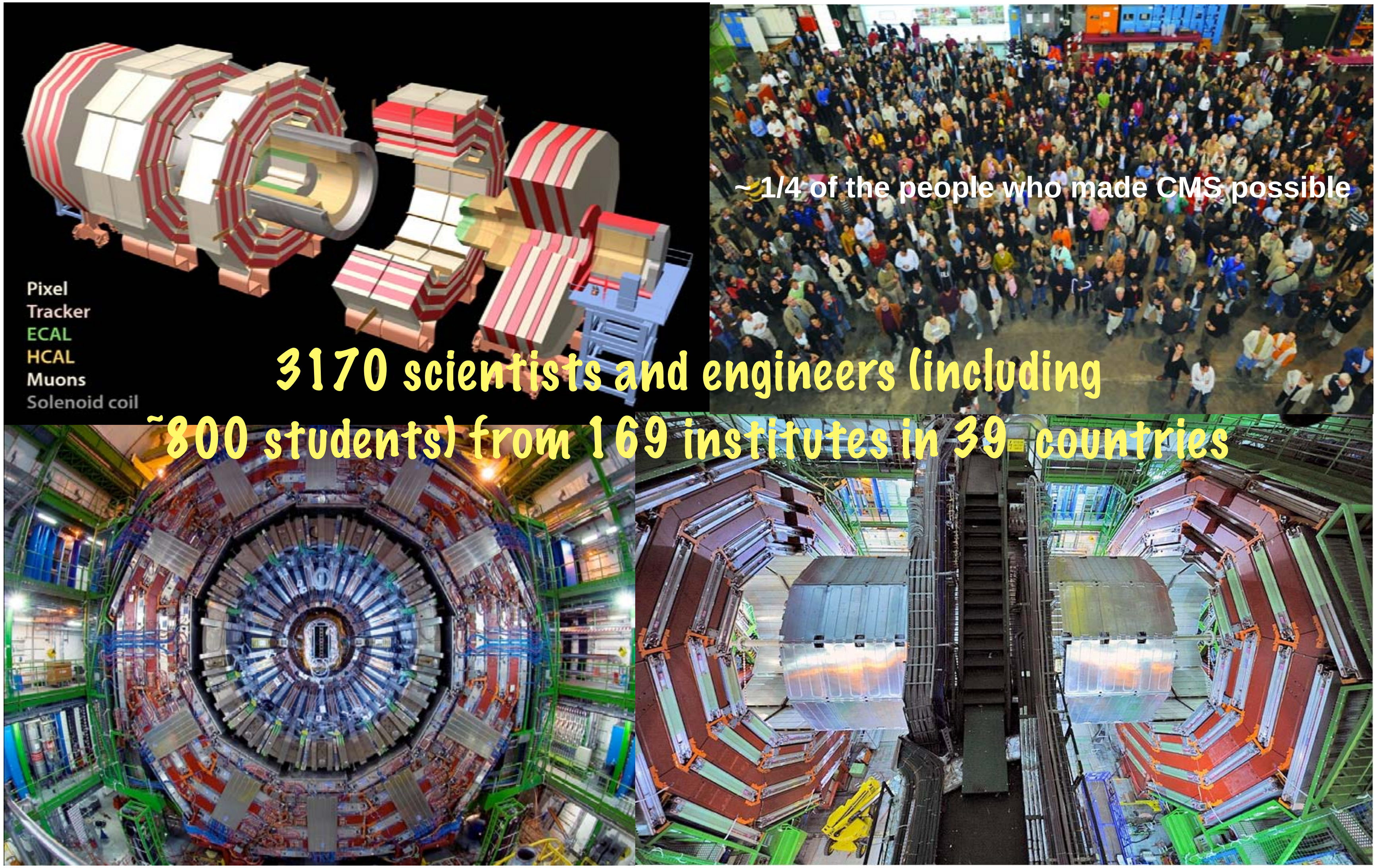
Azimutal angle: Φ
Polar Angle: θ
Pseudorapidity: $\eta = -\ln \tan(\theta/2)$

Design: CMS Detector performance

	CMS (Compact Muon Solenoid)
INNER TRACKER	Silicon pixels + strips No particle identification B=4T $\sigma/p_T \sim 1.5\times10^{-4}\, p_T \oplus 0.005$
EM CAL.	$PbWO_4$ crystals $\sigma/E \sim 2\text{-}5\%/\sqrt{E}$ no longitudinal segmentation.
HAD CAL.	Cu-scint. (> 5.8 λ +catcher) $\sigma/E \sim 100\%/\sqrt{E} \oplus 0.05$
MUON	Fe → $\sigma/p_T \sim 5\%$ at 1 TeV combining with tracker



CMS Collaboration: Many years of hard work and excitement...



3170 scientists and engineers (including ~800 students) from 169 institutes in 39 countries

Startup plan and Software

Turn-on is fast

- LOTS of physics

Necessary Steps:

- Commission detector and readout
- Commission trigger systems
- Calibrate/align detector(s)
- Commission computing and software systems
- Rediscover the Standard Model

Simulation

Reconstruction and Trigger

Monitoring

Calibration/Alignment

- calculation
- application

User-level data objects

- selection

Analysis

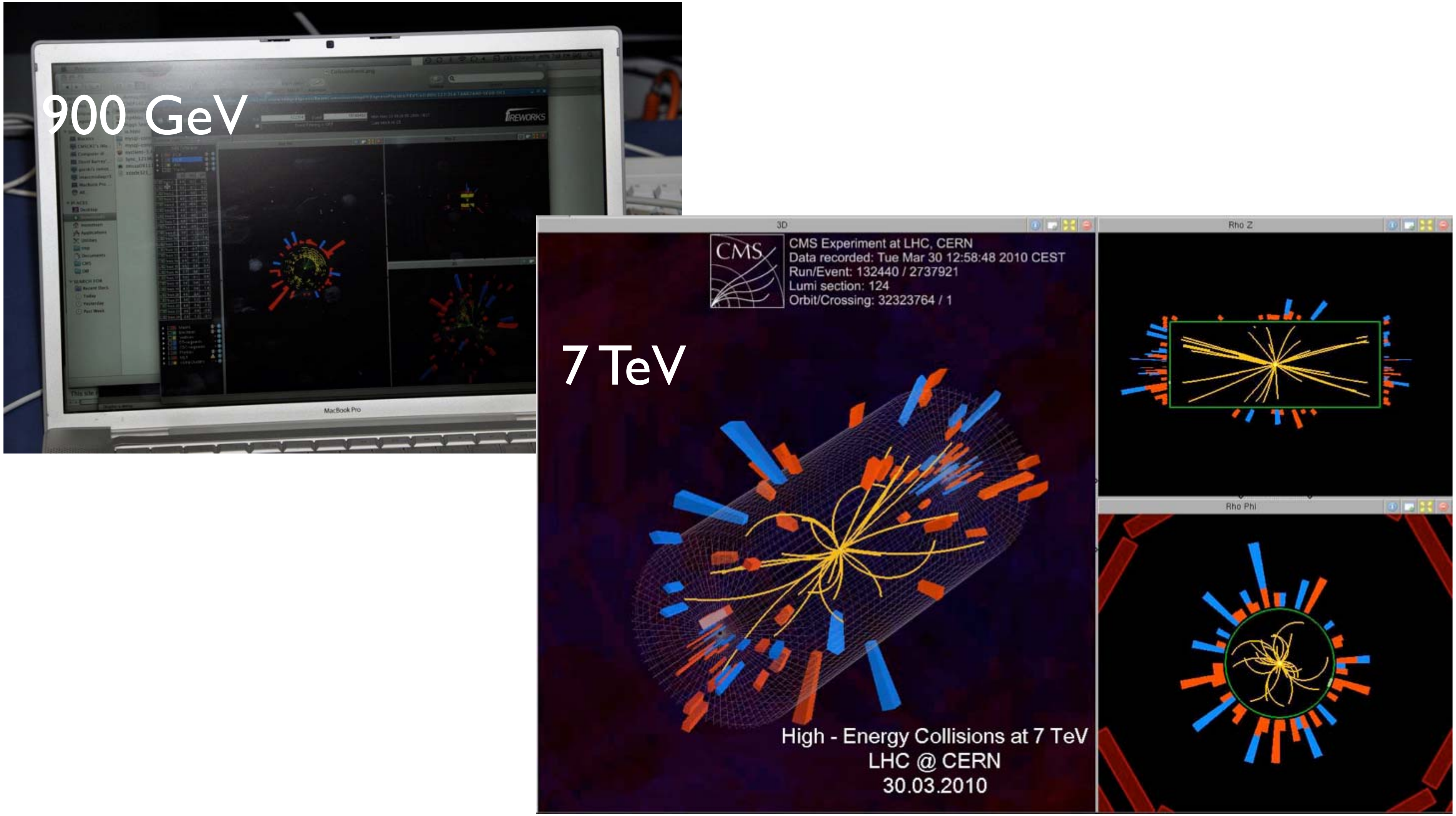
Visualization

SW Development Tools

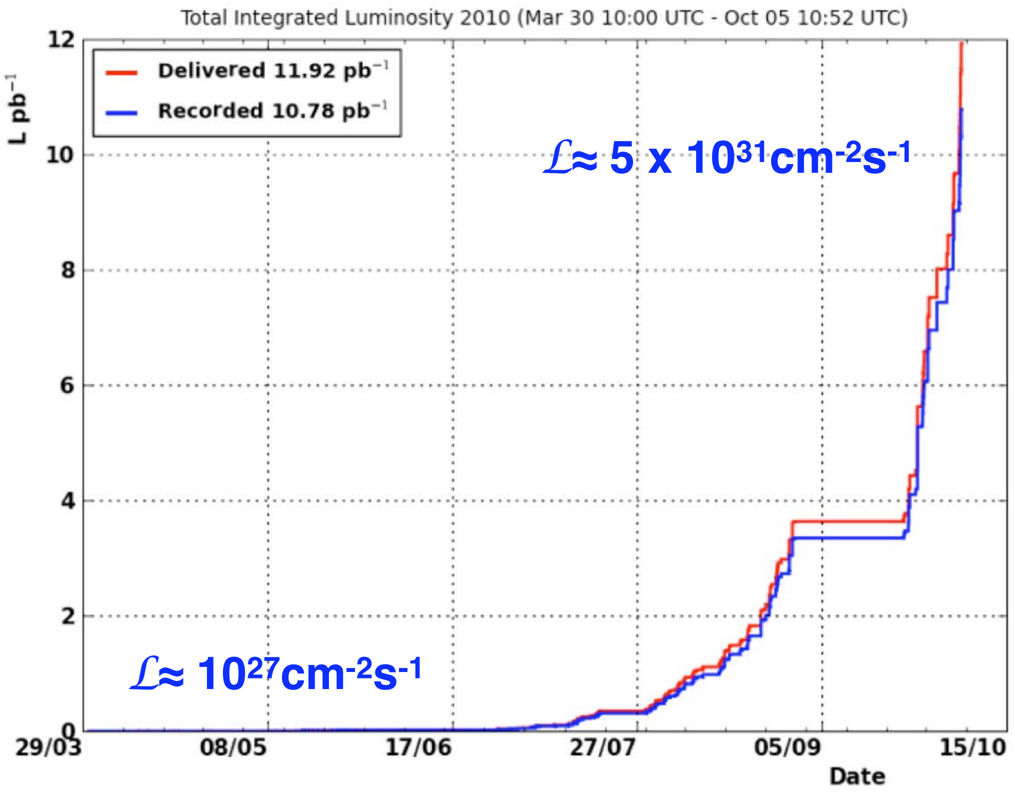
Documentation

Collision data

Then, collisions came



Very successfully data taking: 7 TeV operations since March 30…



About 11.92pb^{-1} delivered by LHC and 10.78 pb^{-1} of data collected by CMS (stable beam only)

Overall data taking efficiency ~91 % with full detector on

Good performance of CMS in coping with ~5 orders of magnitude increase in instantaneous luminosity.

Recorded luminosity increase on a day by day basis. Now we are moving at 200 bunches operations

LHC Common Software

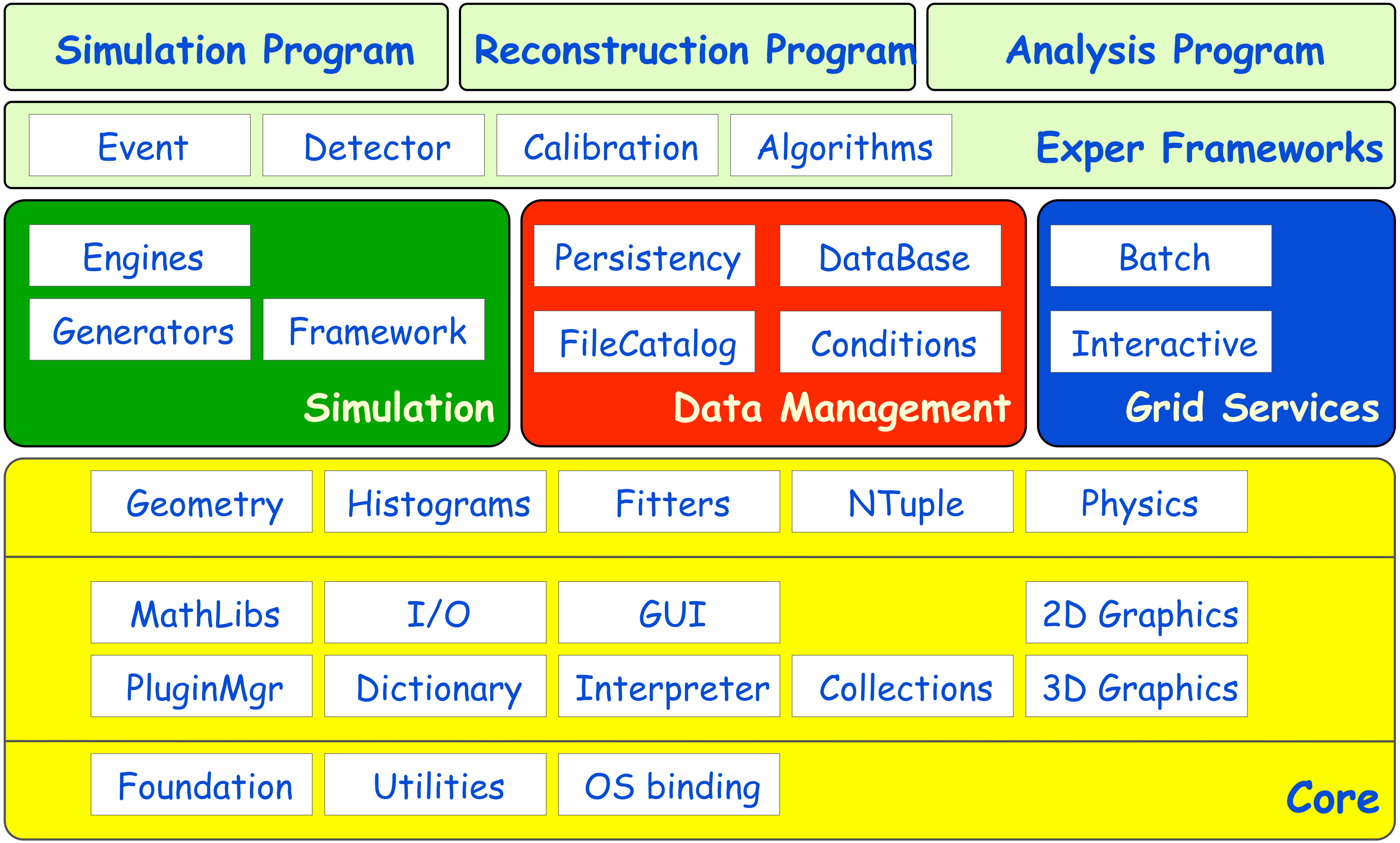
LCG Application Area

Deliver the common physics applications software for the LHC experiments (http://lcgapp.cern.ch/)

Organized to ensure focus on real experiment needs

- Experiment-driven requirements and monitoring
- Architects in management and execution
- Open information flow and decision making
- Participation of experiment developers
- Frequent releases enabling iterative feedback

Software Domain Decomposition



Simplified Software Decomposition

Experiment SW

- Applications — Applications are built on top of frameworks and implementing the required algorithms
- Exp. Framework — Every experiment has a framework for basic services and various specialized frameworks: event model, detector description, visualization, persistency, interactivity, simulation, etc.
- Simulation | Data Mgt. | Distrib. Analysis — Specialized domains that are common among the experiments
- Core Libraries — Core libraries and services that are widely used and provide basic functionality
- non-HEP specific software packages — Many non-HEP libraries widely used

Common SW

Application Area Projects

ROOT – Core Libraries and Services

- Foundation class libraries, math libraries, framework services, dictionaries, scripting, GUI, graphics, etc.

POOL – Persistency Framework

- Storage manager, file catalogs, event collections, relational access layer, conditions database, etc.

SIMU - Simulation project

- Simulation framework, physics validation studies, MC event generators, Garfield, participation in Geant4 and Fluka.

SPI – Software Process Infrastructure

- Software and development services: external libraries, savannah, software distribution, support for build, test, QA, etc.

ROOT: Core Library and services

ROOT is now at the "root" of the software for all the LHC experiments

Current work packages (SW Components)

- BASE: Foundation and system classes, documentation and releases
- DICT: Reflexion system, meta classes, CINT and Python interpreters
- I/O: Basic I/O, Trees, queries
- PROOF: parallel ROOT facility, xrootd
- MATH: Mathematical libraries, histogramming, fitting
- GUI: Graphical User interfaces and Object editors
- GRAPHICS: 2-D and 3-D graphics
- GEOM: Geometry system

Web Page: http://root.cern.ch/

Data Management

FILES - based on ROOT I/O

- Targeted for complex data structure: event data, analysis data
- Based on Reflex object dictionaries
- Management of object relationships: file catalogues
- Interface to Grid file catalogs and Grid file access

Relational Databases – Oracle, MySQL, SQLite

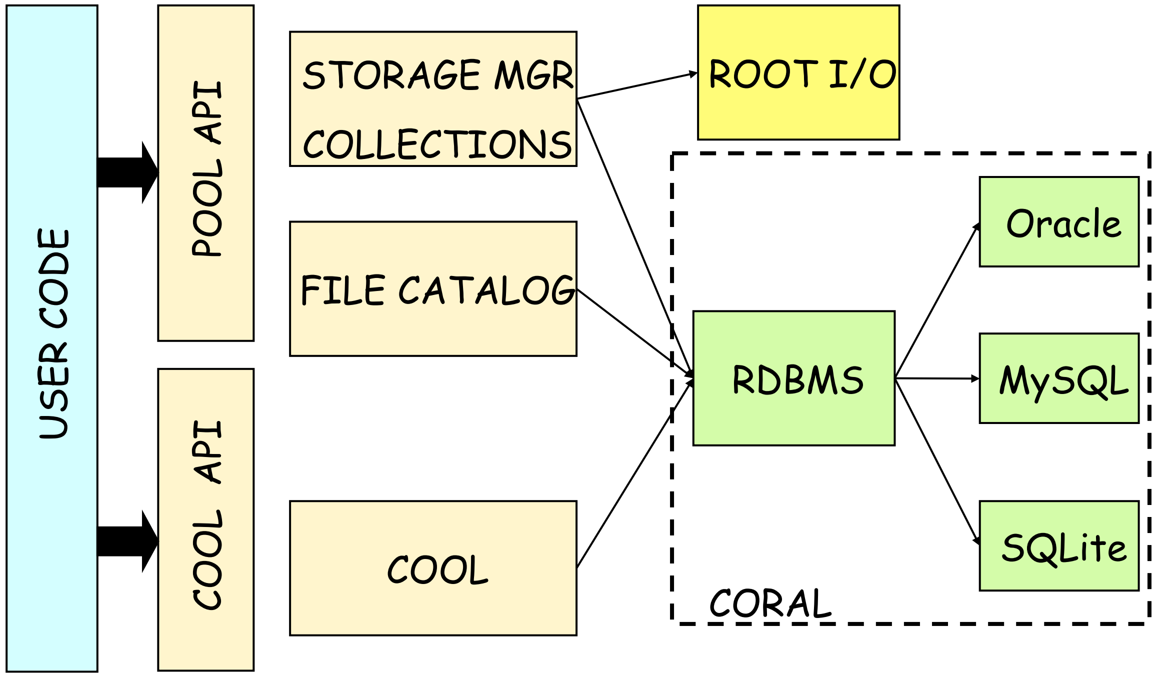
- Suitable for conditions, calibration, alignment, detector description data - possibly produced by online systems
- Complex use cases and requirements, multiple 'environments' – difficult to be satisfied by a single solution
- Isolating applications from the database implementations with a standardized relational database interface
  - facilitate the life of the application developers
  - no change in the application to run in different environments
  - encode "good practices" once for all

Persistency framework

The AA/POOL project is delivering a number of "products"

- POOL – Object and references persistency framework
- CORAL – Generic database access interface
- ORA – Mapping C++ objects into relational database
- COOL – Detector conditions database

http://pool.cern.ch/



Simulation

MC generators

- MC generators specialized on different physics domains, developed by different authors
- Needed to guarantee support for the LHC experiments and collaboration with the authors.

Simulation engines

- Geant4 and Fluka are well established products

Common additional utilities required by the experiments

- Interoperability between MC generators and simulation engines
- Interactivity, visualization and analysis facilities
- Geometry and Event data persistency
- Comparison and validation (between engines and real data)

http://lcgapp.cern.ch/project/simu

Simulation framework utilities

HepMC: C++ Event Record for Monte Carlo Generators

GDML: Geometry description markup language

- Geometry interchange format or geometry source
- GDML writer and readers exists for Geant4 and ROOT

Geant4 Geometry persistency

- Saving/retrieving Geant4 geometries with ROOT I/O

FLUGG: using Geant4 geometry from FLUKA

- Framework for comparing simulations
- Example applications have been developed

Python interface to Geant4

- Provide Python bindings to G4 classes
- Steering Geant4 applications from Python scripts

Utilities for MC truth handling

Application Area Highlights - SPI

SPI is concentrating on the following areas:

- Savannah service (bug tracking, task management, etc.)
  - >160 hosted projects, >1350 registered users (doubled in one year)
  - Web Page: http://savannah.cern.ch/
- Software services (installation and distribution of software)
  - >90 external packages installed in the external service
- Software development service
  - Tools for development, testing, profiling, QA
- Web, Hypernews, Documentation

SPI Web Page http://lcgapp.cern.ch/project/spi/

SPI - Software Configuration

An LCG configuration is a combination of packages and versions which are coherent and compatible

Configurations are given names like "LCG_40"

Experiments build their application software based on a given LCG configuration

- Interfaces to the experiments configuratio systems are provided (SCRAM, CMT)
- Concurrent configurations are everyday situation

Configurations are decided in the AF

Configuration of LCG software: LCG_40

Package: external

Version: 40

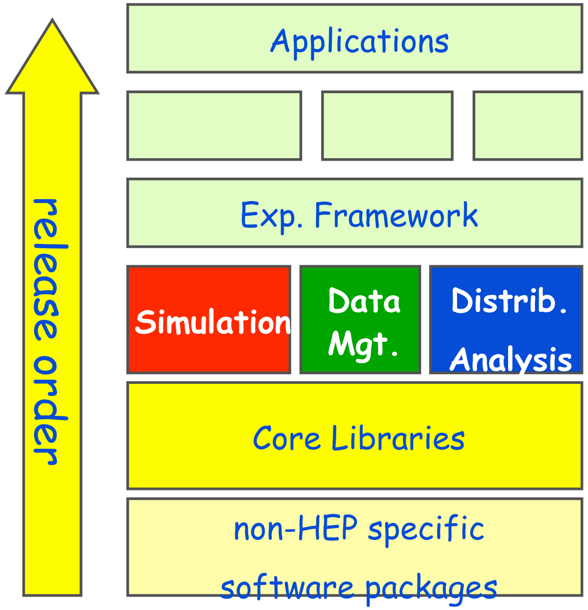
Platform: slc3_ia32_gcc323

Listing of configuration for LCG_lcg40

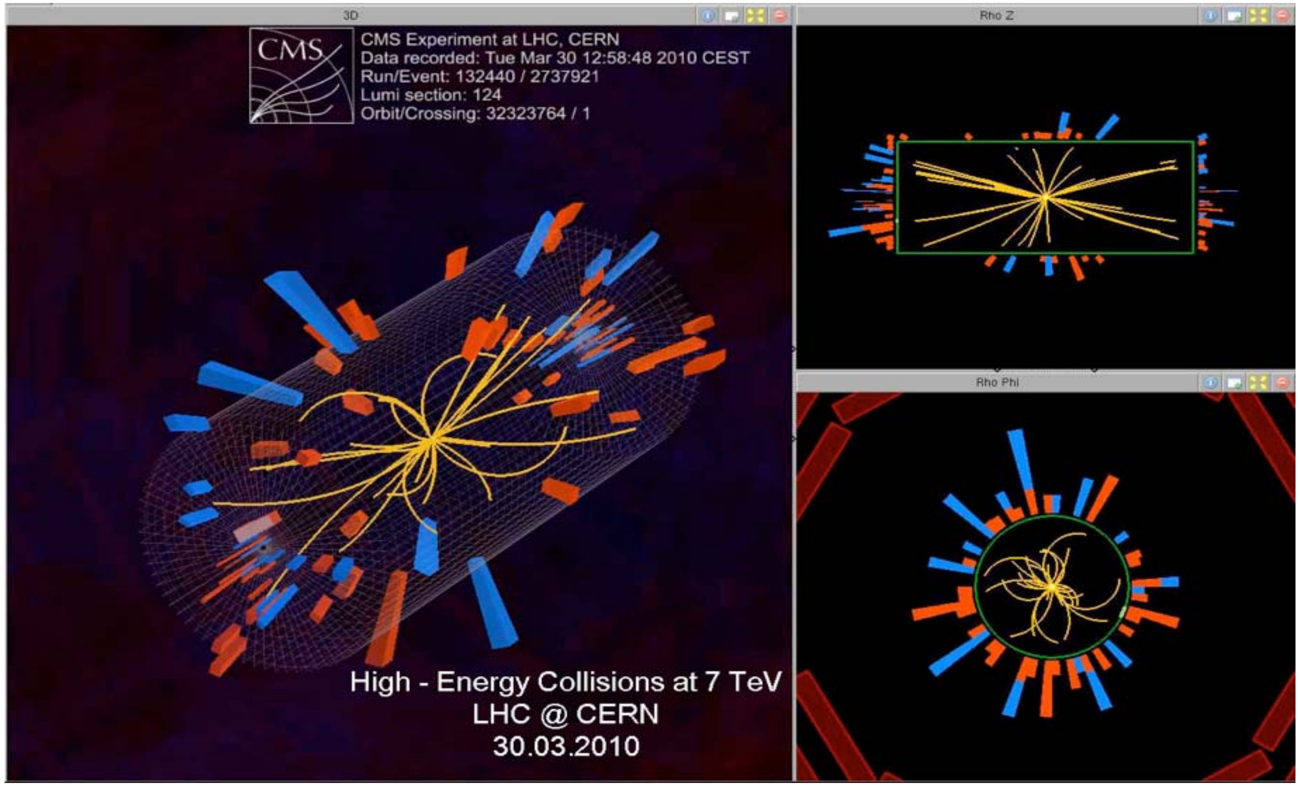
package	version
external	lcg40
gcc3	3.2.3
uuid	1.38
gccxml	0.6.0_patch3
CMake	1.8.3
boost	1.32.0_python242
bjam	3.1.10
python	2.4.2
clhep	1.9.2.2

SPI - Software Releases

- The AA/Experiments software stack is quite large and complex
  - Many steps and many teams are involved
- Only 2-3 production quality releases per year is affordable
  - Complete documentation, complete platform set, complete regression tests, test coverage, etc.
- Feedback is required before the production release is made
  - No clear solution on how to achieve this
  - Currently under discussion
- As often as needed bug fix releases
  - Quick reaction time and minimal time to release

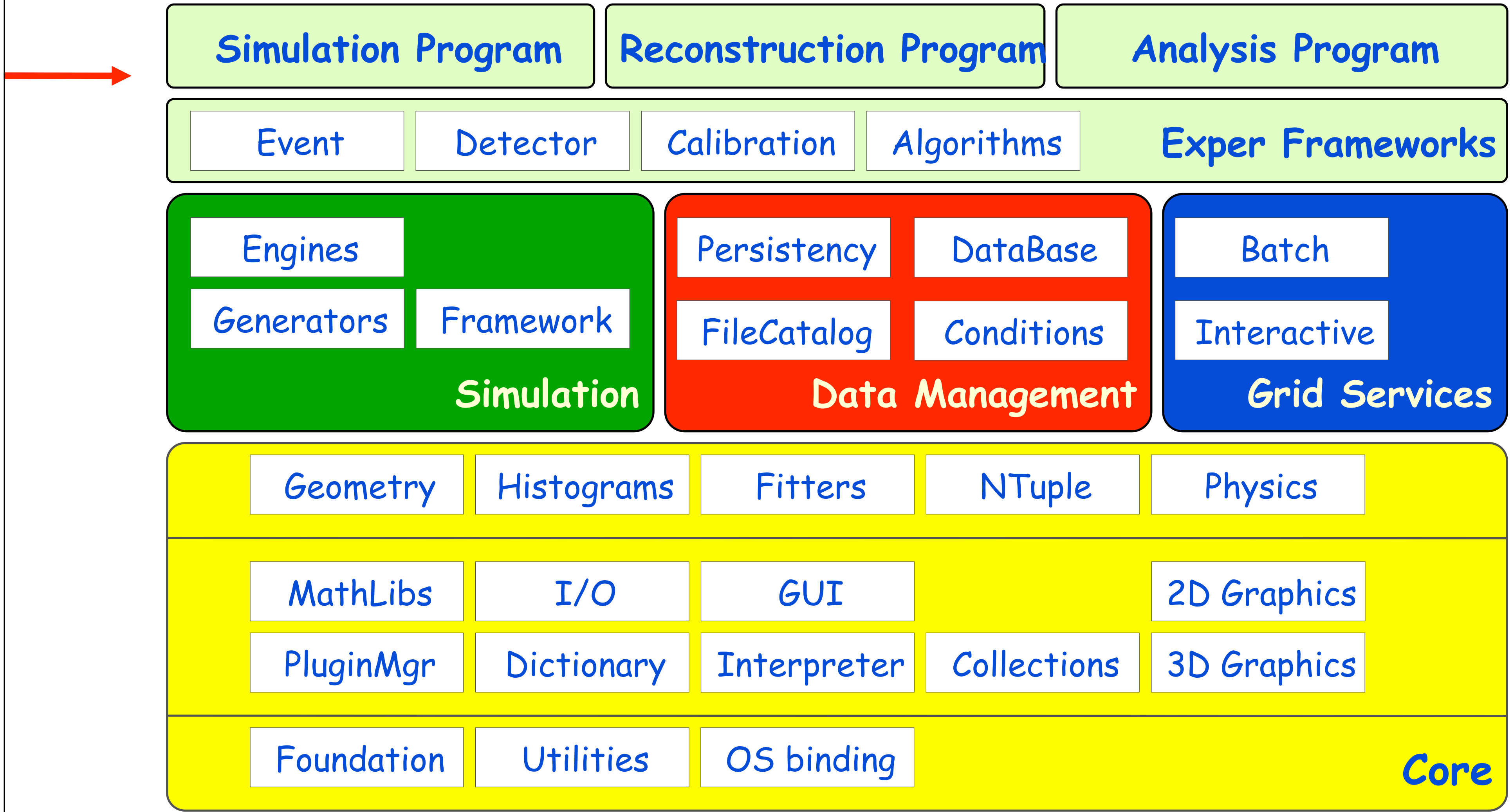


7 TeV 30 March 2010



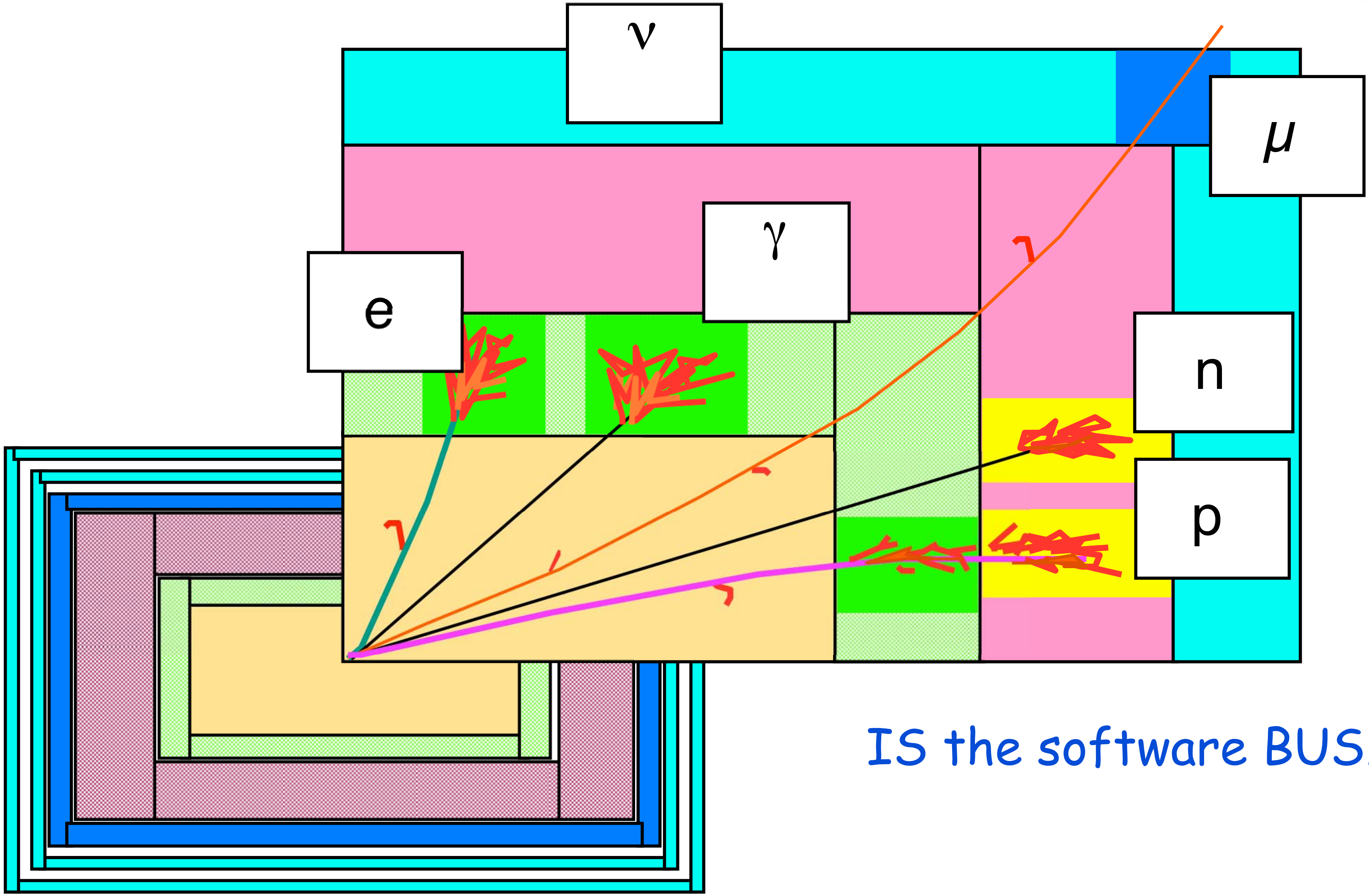
CMS Software

Software Domain Decomposition



The CMS Framework and the Event Data Model

THE EVENT....



IS the software BUS...

Framework: Processing Model

Source creates the Event

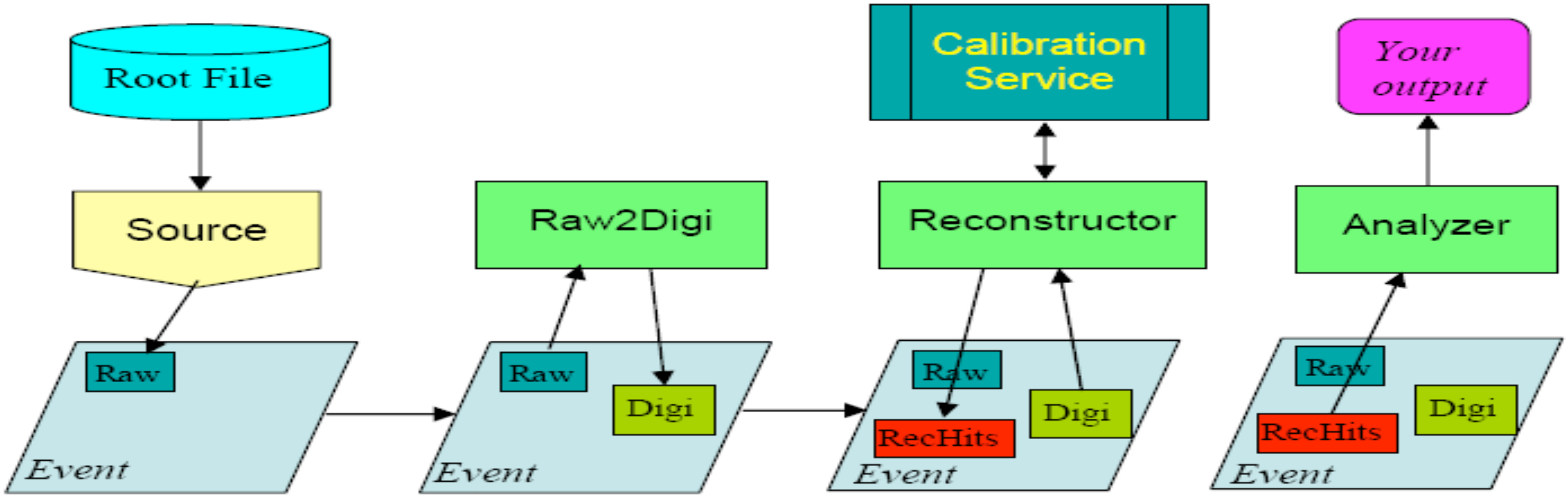
The Event is passed to execution paths

Path is an ordered list of Producer/Filter/Analyzer modules

Producers add data to the Event

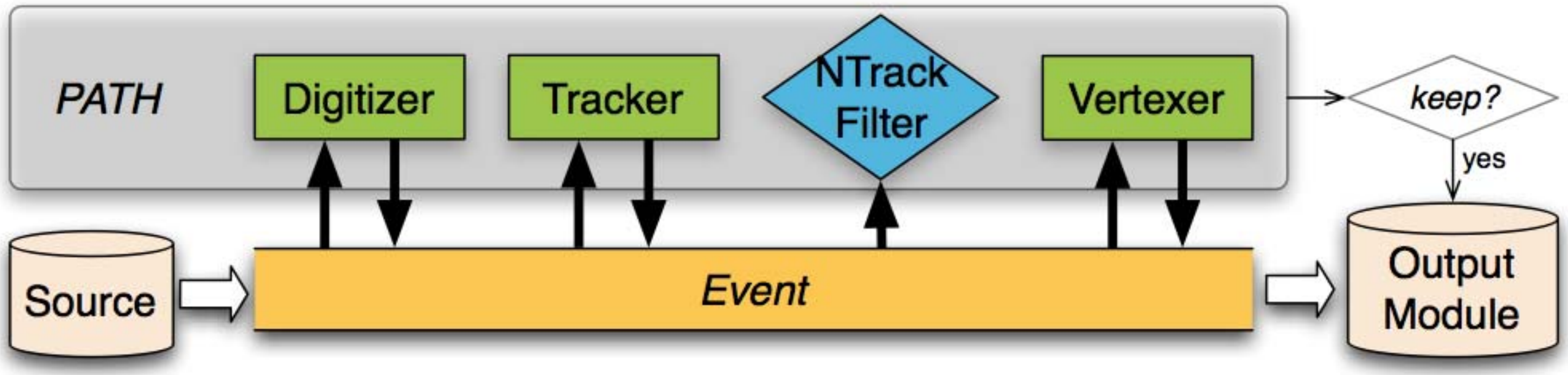
OutputModule given Event if certain Paths run to completion

https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookCMSSWFramework



Modules are configurable and communicate via the Event

Flow of the Data



Modules are configurable and communicate via the Event

https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookCMSSWFramework

Framework: Component Architecture

Five types of dynamically loadable processing components

- Source
  - Provides the Event to be processed (read the event)
- OutputModule
  - Stores the data from the Event. Can use filter decisions
- EDProducer (read/write)
  - Creates new data to be placed in the Event
- EDFilter (read/write)
  - Decides if processing should continue for an Event
- EDAnalyzer (read)
  - Studies properties of the Event
  - Creating histograms

Components only communicate via the Event
Components are configured at the start of a job using a ParameterSet
Steered via Python job configuration

CMSSW Job configuration

Data processing is steered via configuration file written using Python script language:

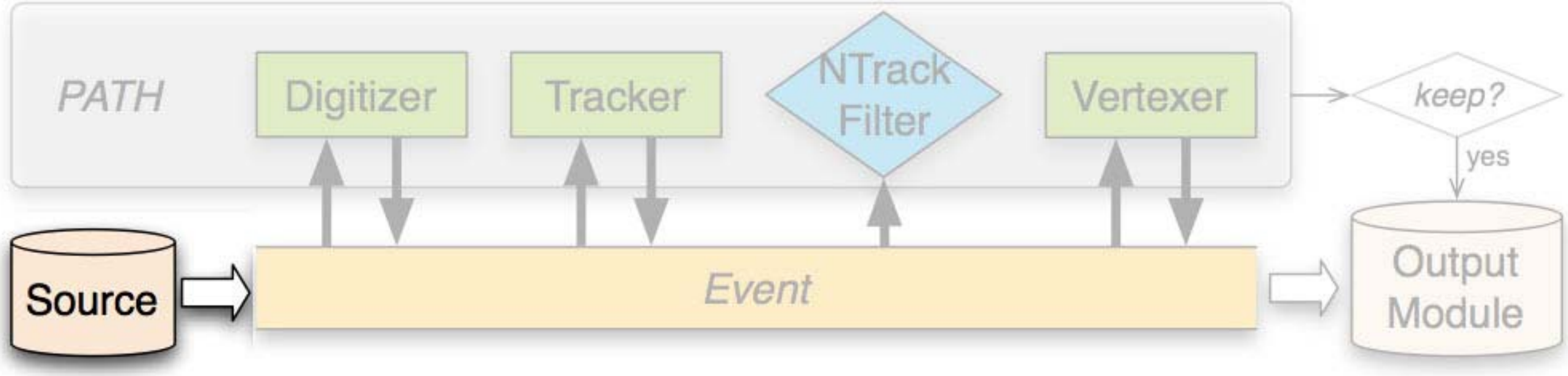
https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookConfigFileIntro

```
import FWCore.ParameterSet.Config as cms
process = cms.Process("EXAMPLE")
process.source = cms.Source("EmptySource")
process.maxEvents = cms.untracked.PSet( input =
                                          cms.untracked.int32(100)
                                        )
process.int = cms.EDProducer("IntProducer",
                              ivalue = cms.int32(2)
                              )
process.test = cms.EDAnalyzer("IntTestAnalyzer",
                               valueMustMatch = cms.untracked.int32(2)
                               )
process.Tracer = cms.Service("Tracer")
process.path = cms.Path( process.int * process.test)
```

```
$ cmsRun example_cfg.py
```

One executable

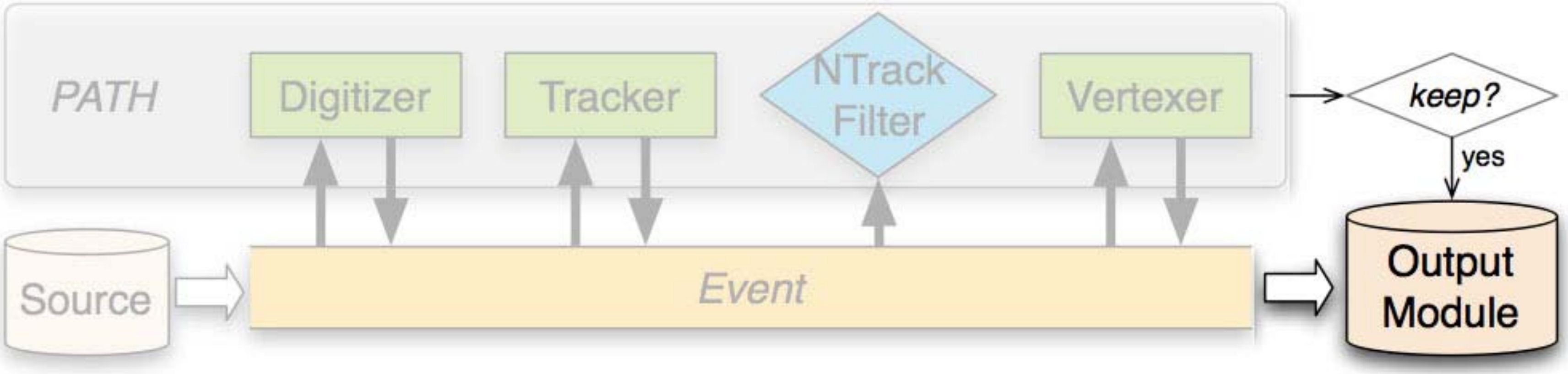
Input - Source




```
process.source = cms.Source("PoolSource”,
        fileNames = cms.untracked.vstring("file:input.root”)
)
```

INPUT

Output - OutputModule




```
process.out = cms.OutputModule("PoolOutputModule",
        fileName = cms.untracked.string("output.root")
)
```

OUTPUT

```
Process.out.outputCommands = cms.untracked.vstring(
        "keep *",
        "drop *_*_*_HLT",
        "keep FEDRawDataCollection_*_*_*"
)
```

Configurable

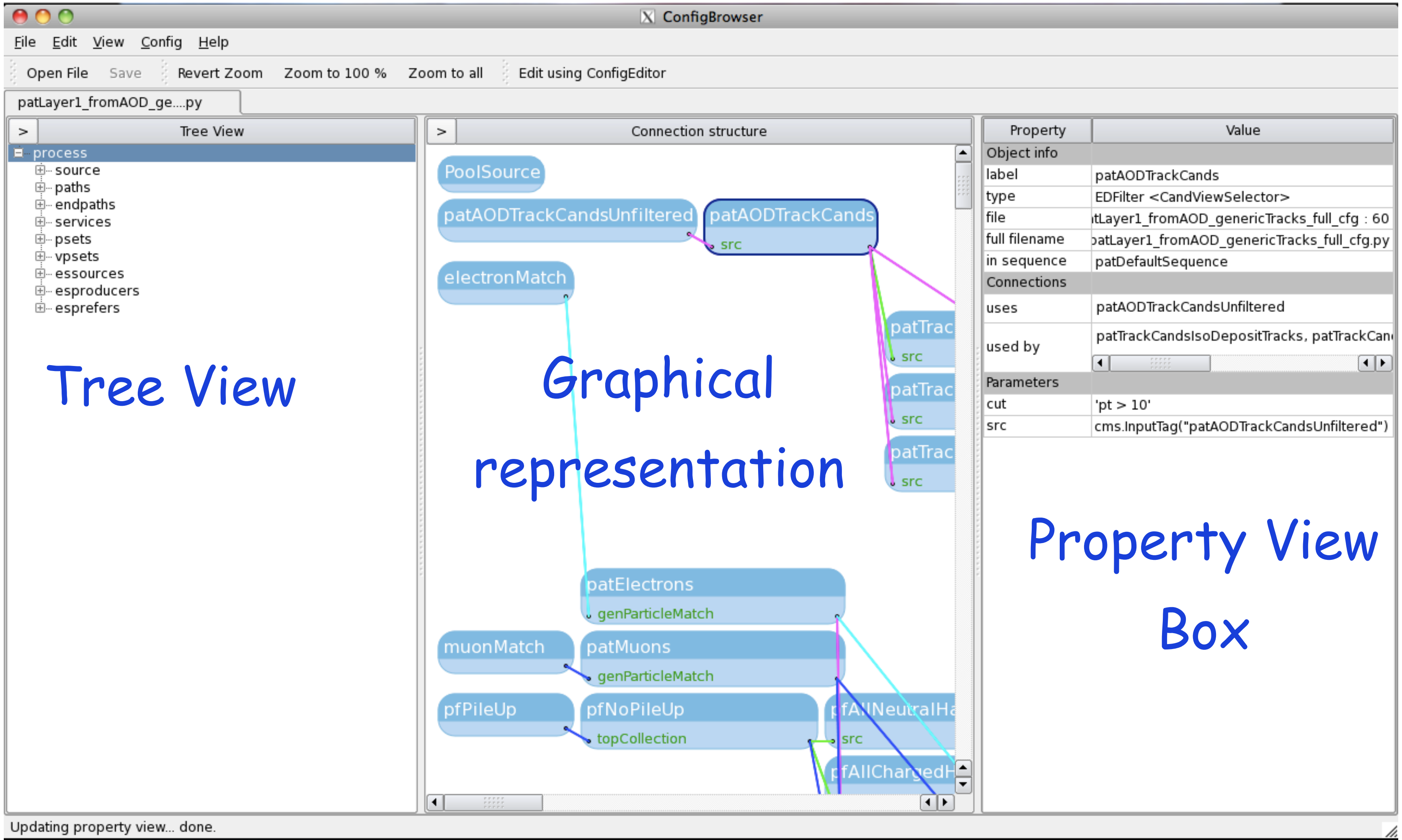
Event Content:

What to drop?

What to keep?

Inspect your configuration: the ConfigBrowser

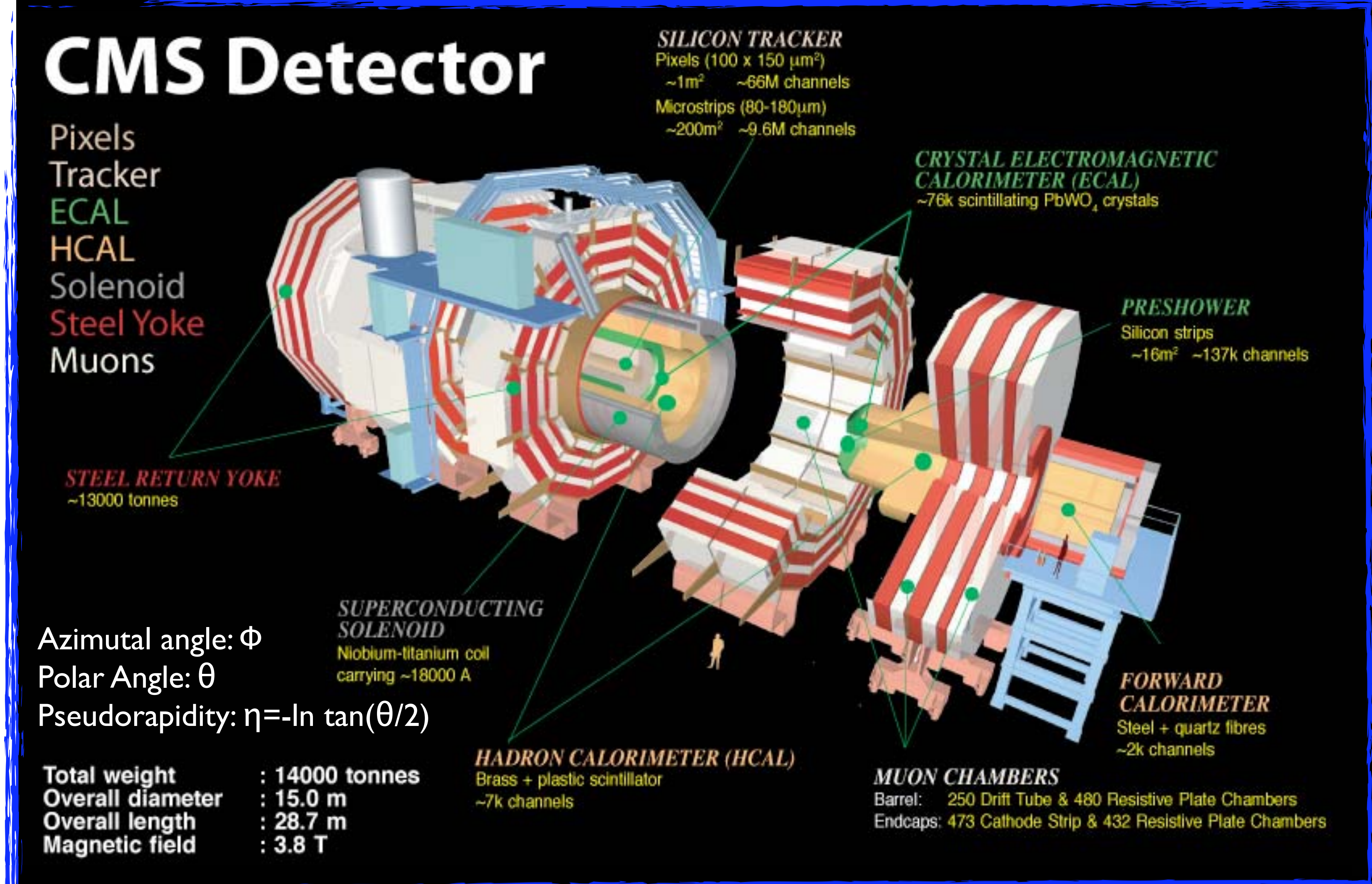
You can inspect your config file using a graphical tool as well: the **ConfigBrowser**



https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideConfigBrowser

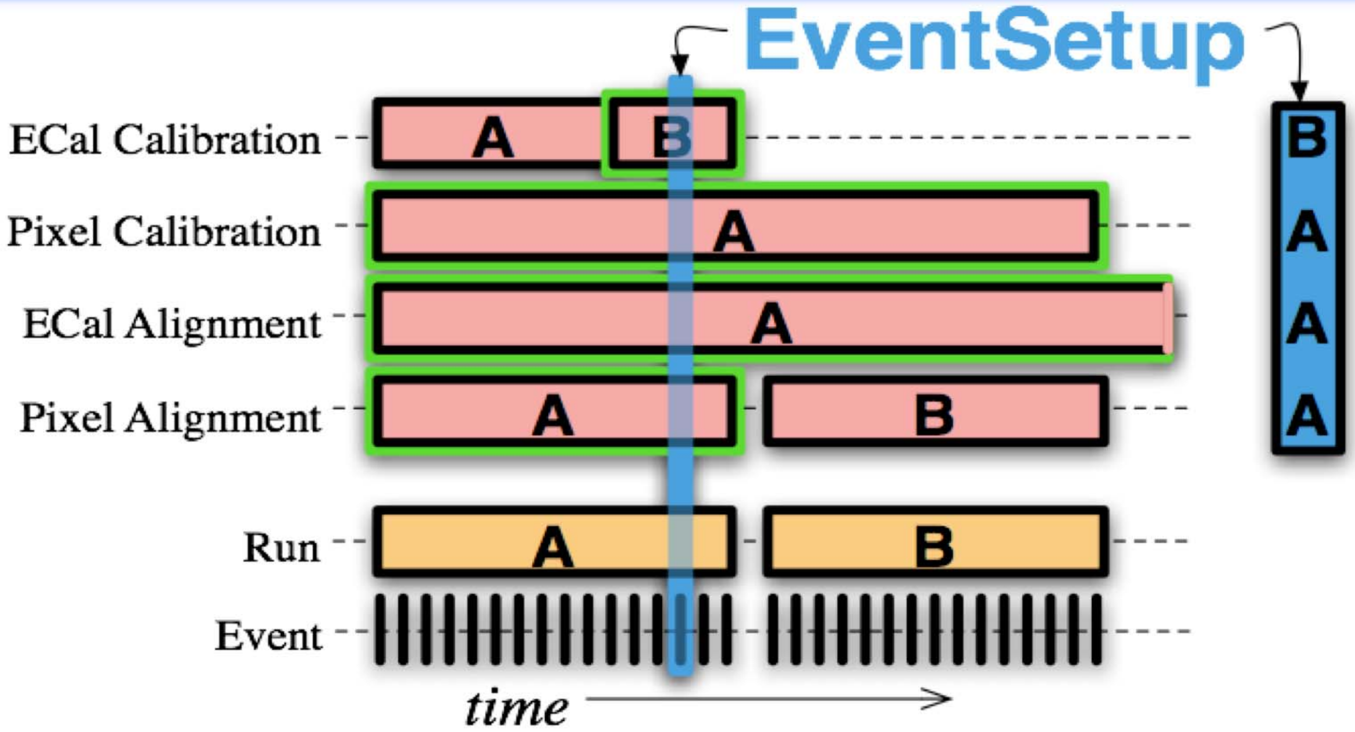
Calibration & Alignment
Non-Event Data Model

CMS Detector



Azimutal angle: Φ
Polar Angle: θ
Pseudorapidity: $\eta = -\ln \tan(\theta/2)$

Calibration & Alignment: The Model



Provides a unified access mechanism for non-Event data

Record:	holds data with same interval of validity
EventSetup	"snapshot" of detector at an instant in time

Calibration & Alignment Event SetUp Components

Components do the work of actually creating/reading the data

The EventSetup supports two types of dynamically loaded components:

–ESSource

- reads data from disk
- sets the 'interval of validity' for data in a Record e.g., read calibration information from a database for a particular run range

–ESProducer

- creates data by running an algorithm
- obtains data needed by the algorithm from Records in the EventSetup e.g., create tracking geometry by combining alignment shifts and perfect positioning of material

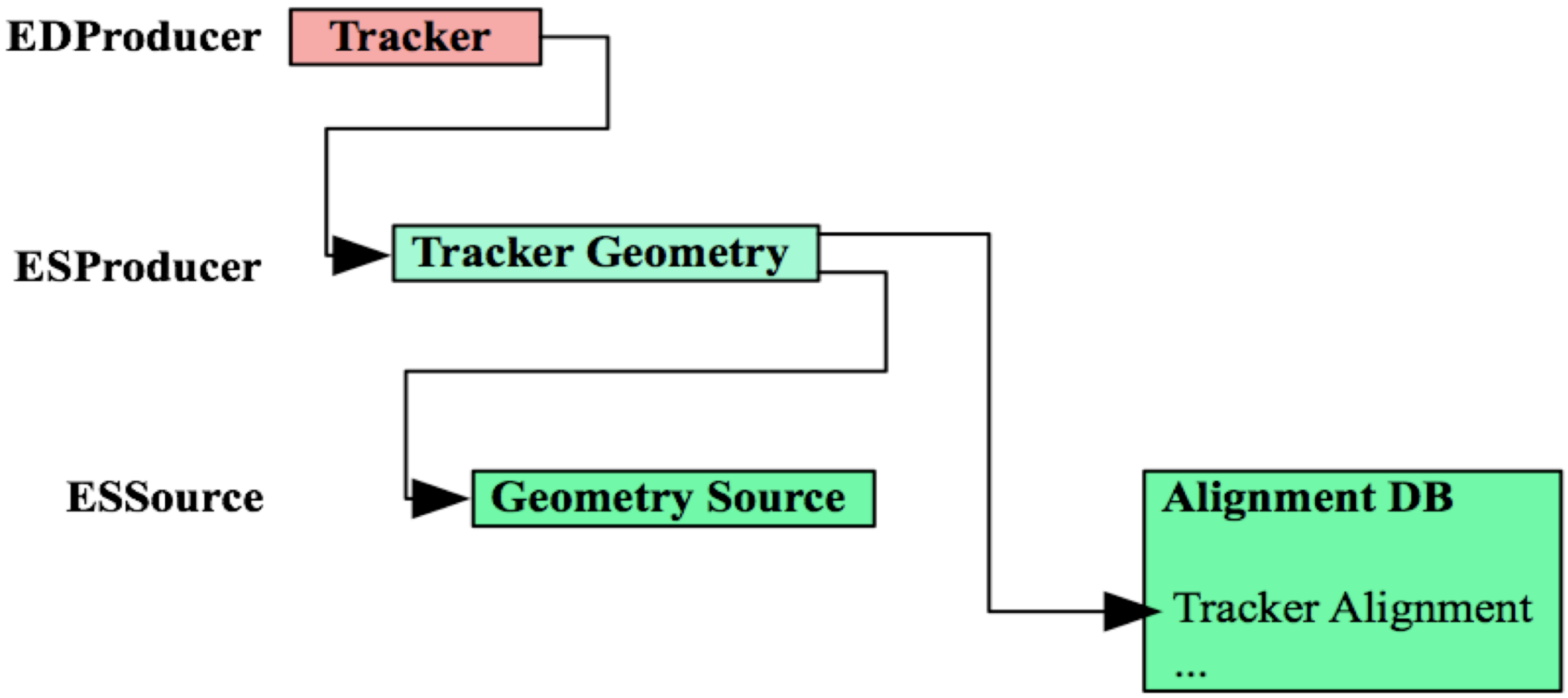
Examples of Framework Services and setups are:

-Geometry, Calibration, MessageLogger

Calibration & Alignment Event SetUp: Data Retrieval

To a user, EventSetup appears to have all its data loaded

To avoid unnecessary computation, data is retrieved on the first request



NOTE: an EDProducer is an Event module

Record Interdependency

ESProducer's may need to get data from other Records

If data in Record A depends on data in Record B then when Record B's validity interval changes Record A's validity must also change

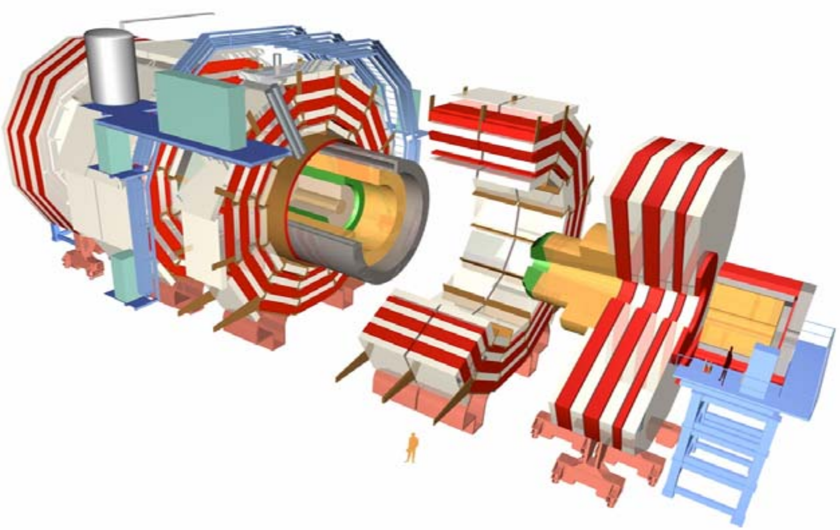
System automatically handles validity dependencies

Record interdependencies set and checked at compile time

```
class TrackerGeometryRecord :
  public DependentRecord<mpl::vector<GeometryRecord,
                                     TrackerAlignmentRecord>
                        >
{ };
```

Data Formats in edm files

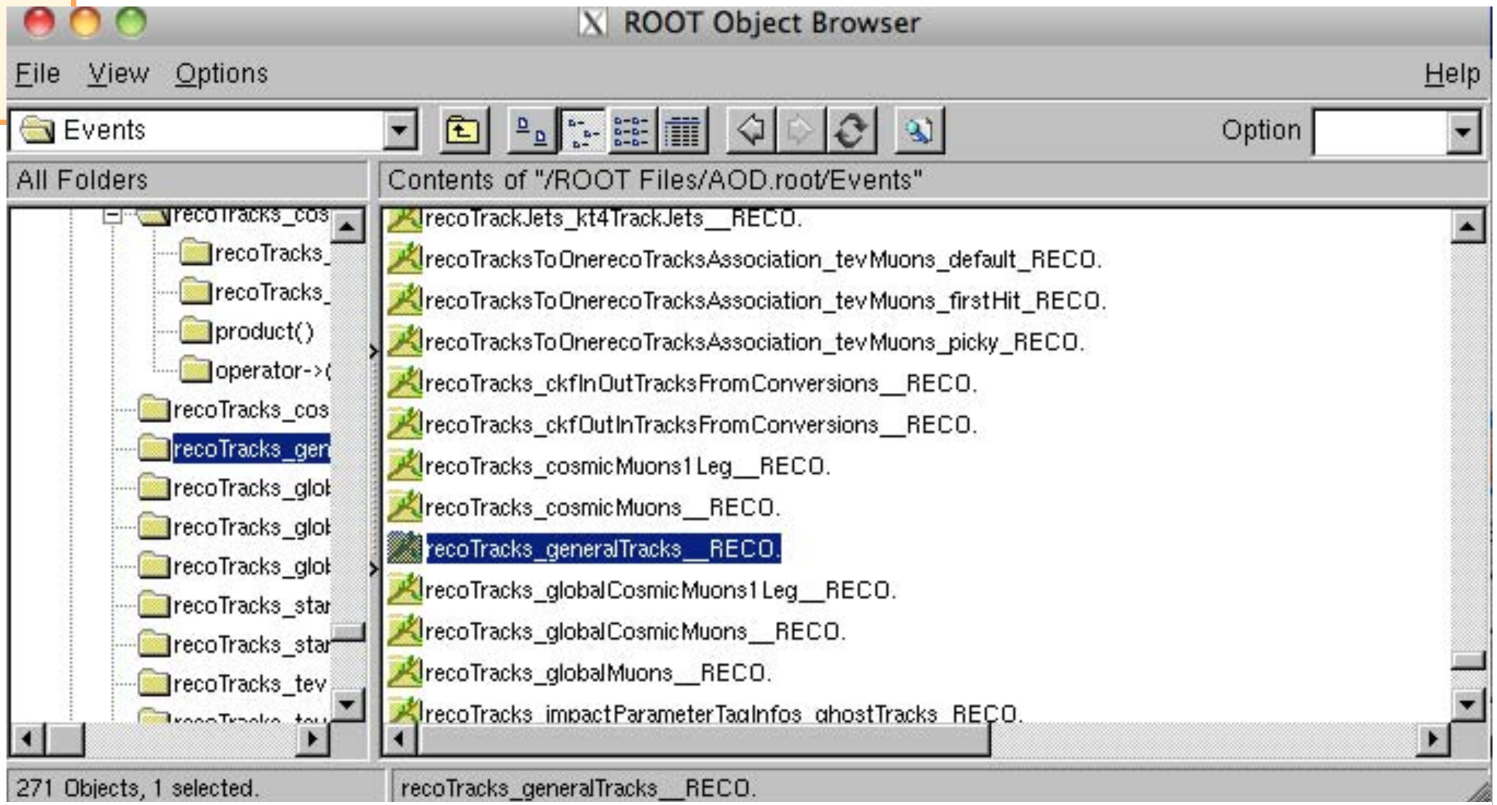
What is stored in the event files?



- RAW:
  - Data like they come from the detector
- RECO (Reconstruction):
  - Output of the event reconstruction
- AOD (Analysis Object Data):
  - Subset of data needed for standard analysis
- RAWSIM, RECOSIM, AODSIM:
  - with additional simulation information

Files can be inspected with ROOT

```
root -l
[] TFile f("AOD.root")
[] new TBrowser()
```

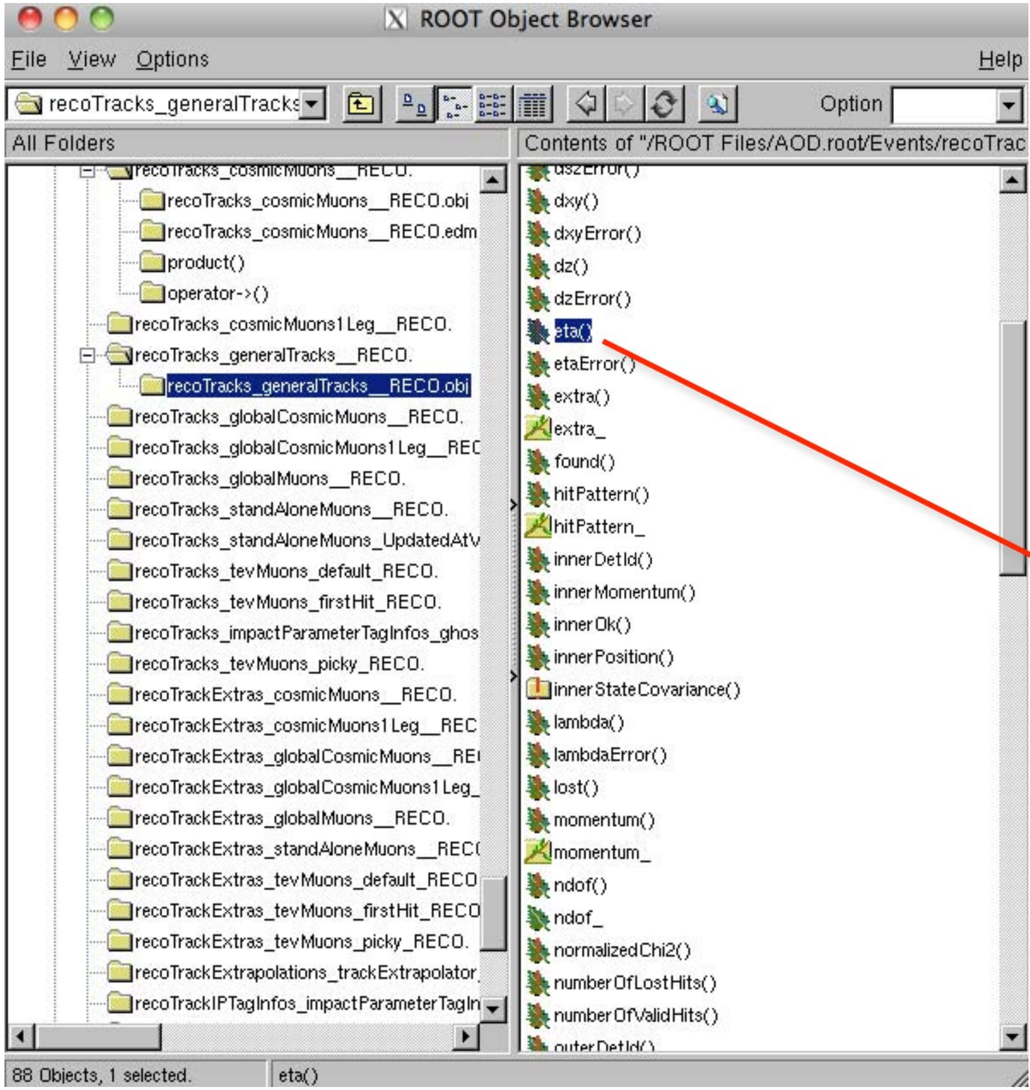


Data inside the event are called "Product"

moduleLabel : productInstanceLabel : processName

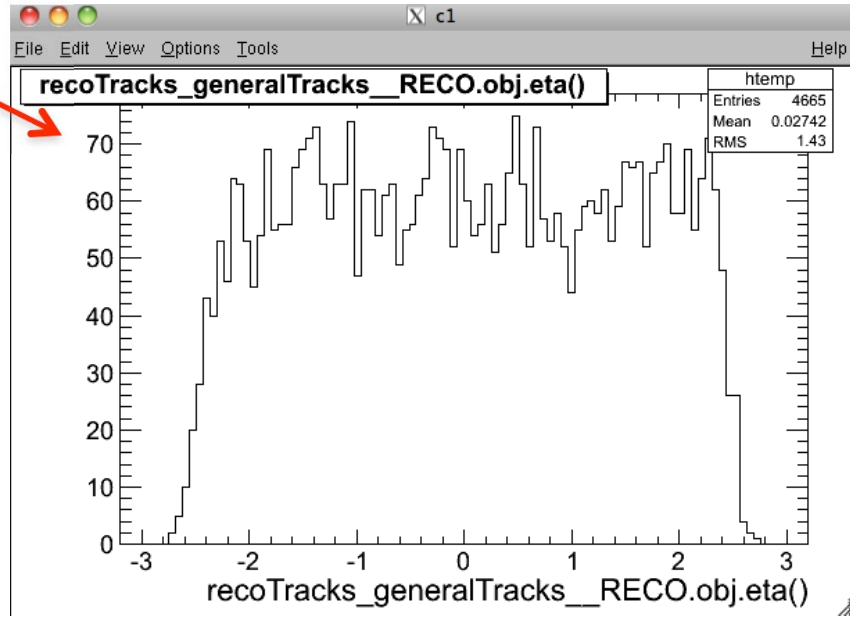
Example: recoTracks_generalTracks_RECO

FWLite gives access to classes



Automatic library loading

- gSystem->Load("libFWCoreFWLite")
- AutoLibraryLoader::enable()
- new TBrowser()



What are the stored products?

edmDumpEventContent <filename>

C++ class type	product alias	label	process name
vector<reco::MET>	"tcMet"	""	"RECO."
vector<reco::Muon>	"muons"	""	"RECO."
vector<reco::Muon>	"muonsFromCosmics"	""	"RECO."
vector<reco::Muon>	"muonsFromCosmics1Leg"	""	"RECO."
vector<reco::PFCandidate>	"particleFlow"	""	"RECO."
vector<reco::PFCandidate>	"particleFlow"	"electrons"	"RECO."
vector<reco::PFJet>	"ak5PFJets"	""	"RECO."

```
Handle<reco::MuonCollection> muons;
Event.getByLabel("muons",muons );
```

Access the single Product in the framework module

reco::MuonCollection is a typedef for vector<reco::Muon>

Provenance Tracking

The history of each single product in the event is stored in the “provenance”

```
...

Module: caloTowers Rec
 PSet id:e03ccfff88a2fd4ed3c2b9bd8261000b
 products: {
  recoCandidatesOwned_caloTowers__Rec.
}
 parameters: {
  @module_label: string tracked  = 'caloTowers'
  @module_type: string tracked  = 'CaloTowerCandidateCreator'
  minimumE: double tracked  = -1
  minimumEt: double tracked  = -1
  src: InputTag tracked  = towerMaker::
}

...
```

edmProvDump <filename>

Accessing Event Data

You can access the products in the module using the Handle

```
# by module and default product label
Handle<reco::MuonCollection> muons;
iEvent.getByLabel(”muons", muons );

# by module and product label
Handle<vector<reco::PFCandidate> > particleFlow;
iEvent.getByLabel("particleFlow", ”electrons" ,
particleFlow_electrons );
```

Framework modules are written in C++ , you can find a basic C++ guide at:

https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookBasicCPlusPlus

FWLite

FWLite basics

CMSSW provides a 'batch' (cmsRun) and an 'interactive' access (FWLite) to event data

ROOT + CMS data formats + helper classes = FWLite

Working in FWLite on data objects, you have

- simplicity of working in ROOT on TTrees
- however enhanced with usefulness of CMS data model

You can install FWLite on your laptop

native on several linux flavours & MacOS

via a virtual machine

FWLite code

```
{
   gSystem->Load("libFWCoreFWLite.so");
   AutoLibraryLoader::enable();
   gSystem->Load("libDataFormatsFWLite.so");

   #include "DataFormats/FWLite/interface/Handle.h"
   vector<string> fileNames;
   fileNames.push_back("....root");

   fwlite::ChainEvent ev(fileNames);

   for (ev.toBegin(); ! ev.atEnd(); ++ev) {
      edm::EventBase const & event = ev;

        // This snippet can be used in EITHER FWLite or the Full Framework
        edm::Handle<vector<reco::Vertex> > vertices;
        event.getByLabel( edm::InputTag("offlinePrimaryVertices"), vertices);
   }
}
```

https://twiki.cern.ch/twiki/bin/viewauth/CMS/WorkBookFWLiteEventLoop

Software Development Tools and Releases

Release Process

Main problem: large number of developers and geographical diversity

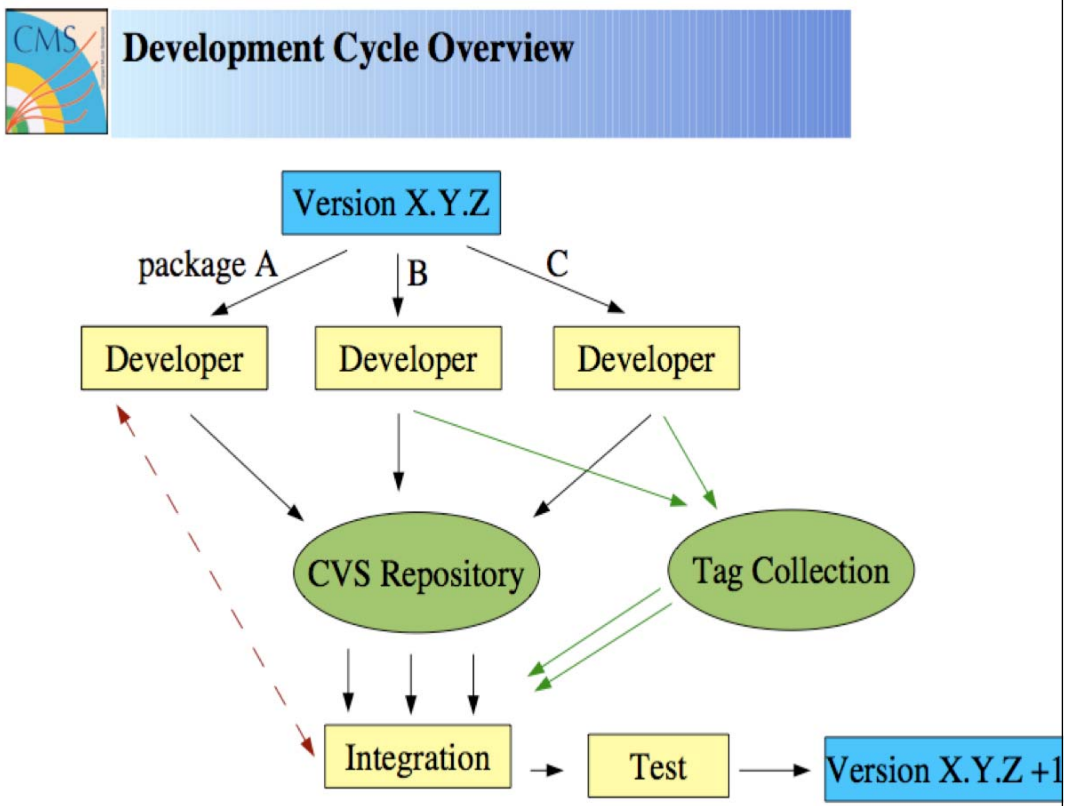
- Use different tools for configuration mgt and build
- Quite some commonality in process (and (some) tools)
  - Nightlies
  - collecting/controlling tags

Software Development Tools

Release Process

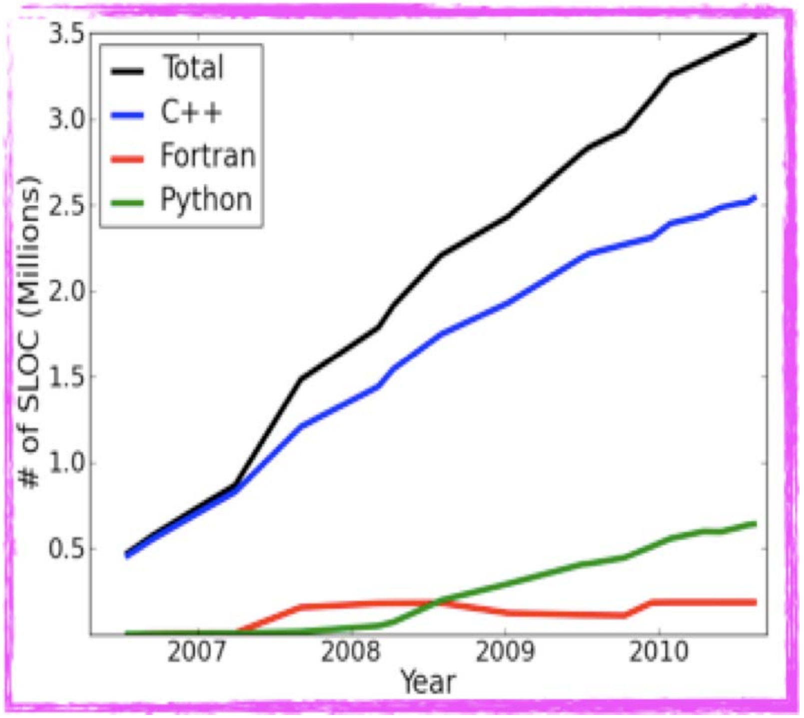
Main problem: large number of developers and geographical diversity

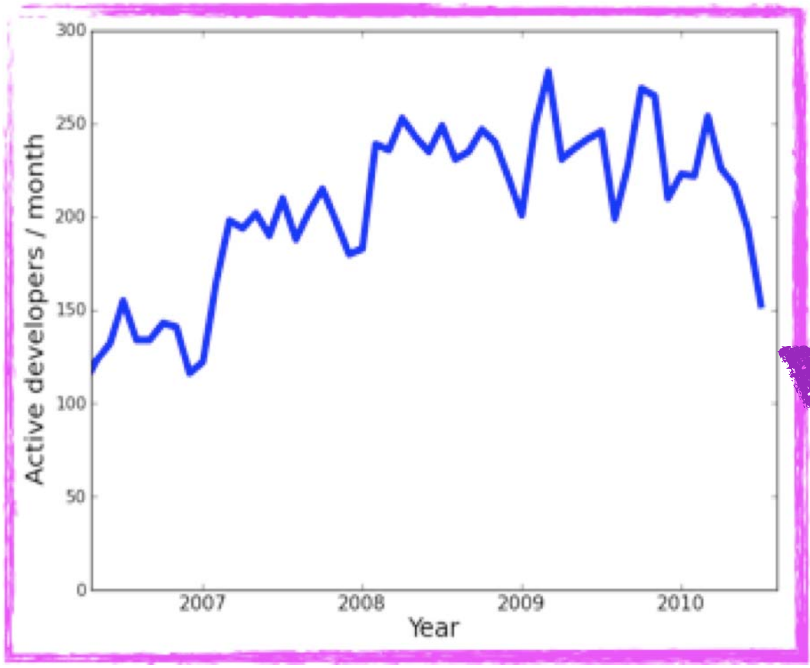
- Use different tools for configuration mgt and build
- Quite some commonality in process (and (some) tools)
  - Nightlies
  - collecting/controlling tags



Offline: code and developers

Huge enterprise to provide stable software while incorporating latest developments. Hundreds of code developer

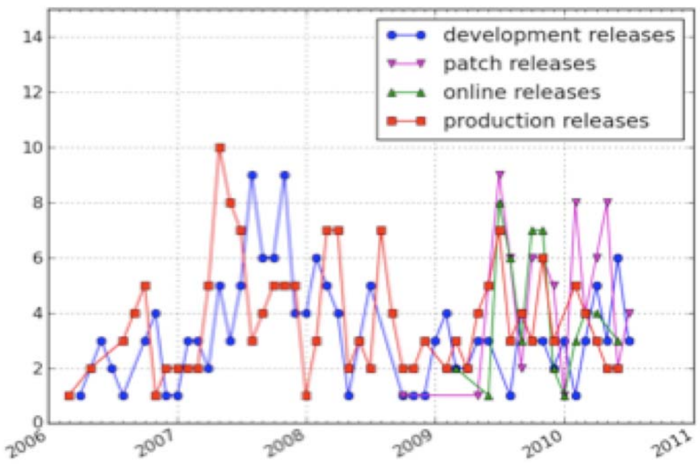




Summer vacations

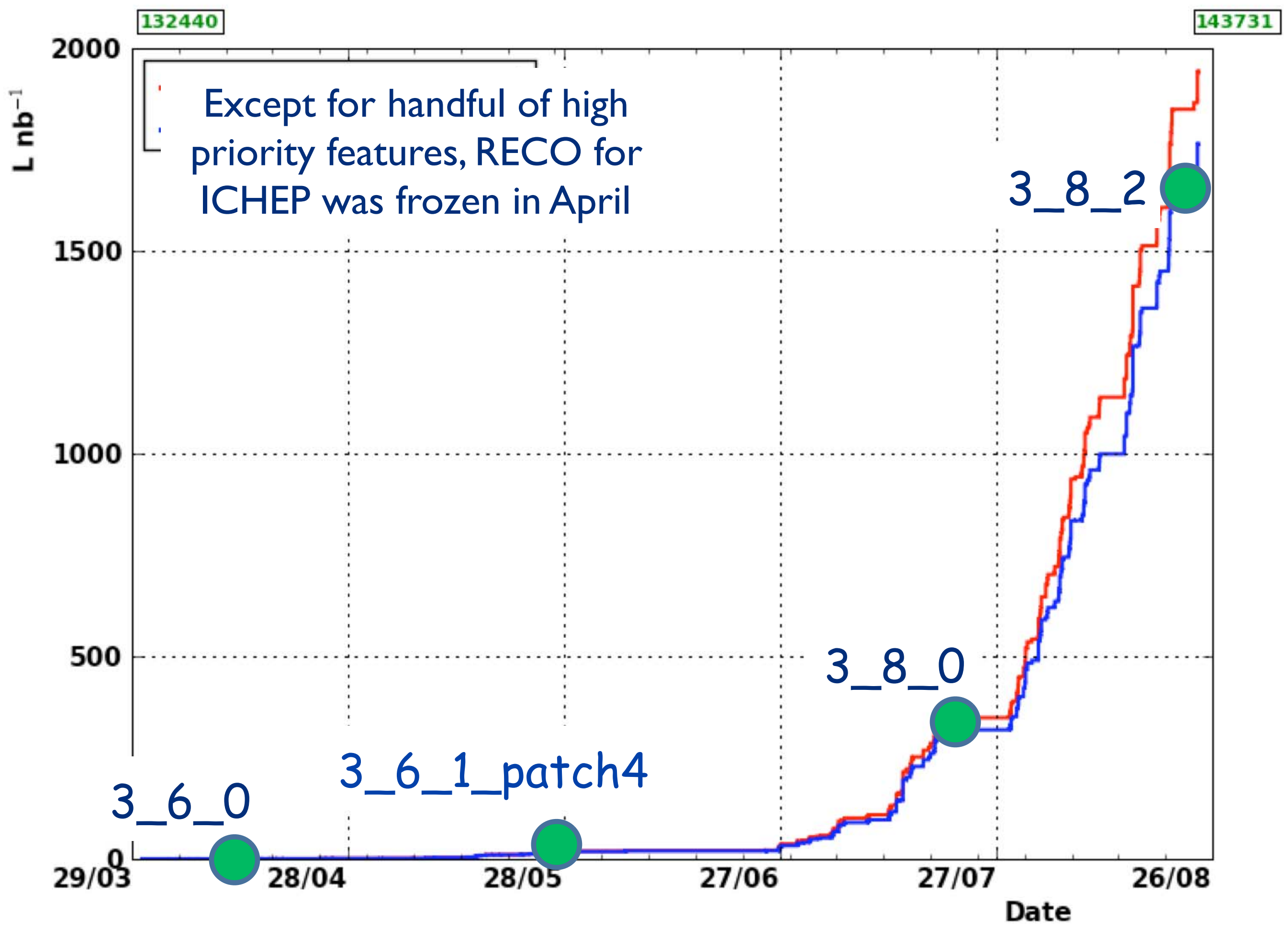
Offline:Releases and performances

- Release cycles
  - Patch release mechanism deployed
  - Deployed train model for release cycle
  - Detailed and frequent monitoring of software quality and performance



- Software performance
  - Extensive optimization program
  - Looking into multi-core usage
  - Reconstruction of collision data (MinBias)
    - 0.6 seconds per event
    - 400 kB RECO, 150 kB AOD
    - 900 MB memory
  - Simulation of Monte Carlo (ttbar)
    - 90 s/evt (50 s for low-pT QCD)
    - 1400 kB RAW SIM
    - 980 MB memory

Recent release cycles have consolidated lessons learned from ICHEP operations and analysis



End Lecture 1

Where we are ??

- ★ Introduction to LHC & CMS Experiment
- ★ LHC Common Software
- ★ CMS Software
  - ★ CMS Framework and Event Data Model
  - ★ Calibration and Alignment: Non Event Data Model
  - ★ Data Format
  - ★ FWLite and cmsRun
  - ★ Software Development Tools and Releases
  - ★ Geometry and Simulation ←
  - ★ Event Visualization
  - ★ Trigger and Reconstruction
- ★ CMS Analysis
- ★ Data Flow, Offline & Computing Operations

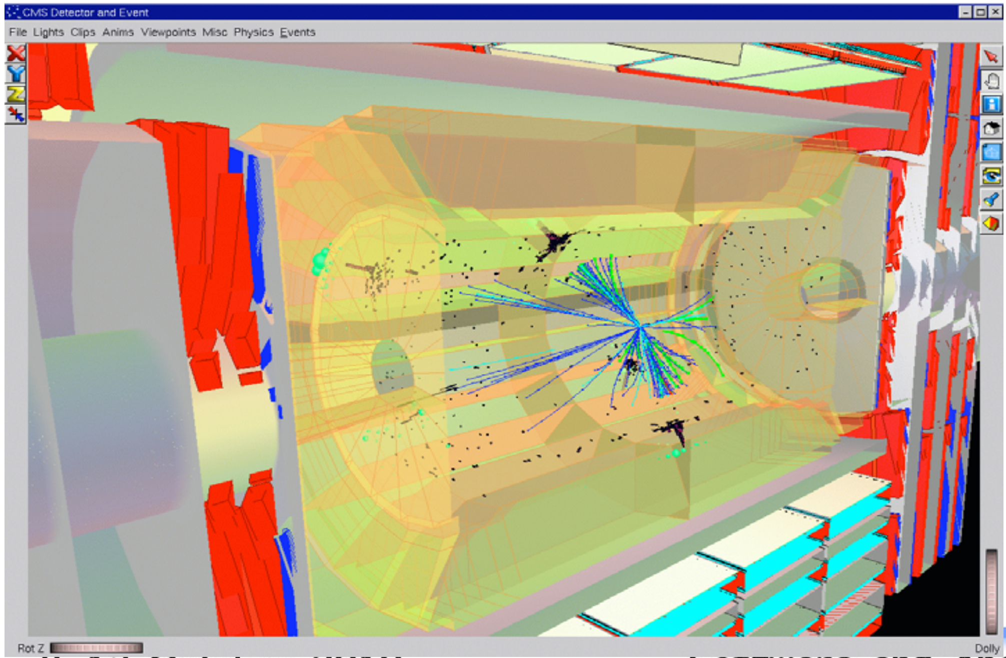
Geometry & Simulation

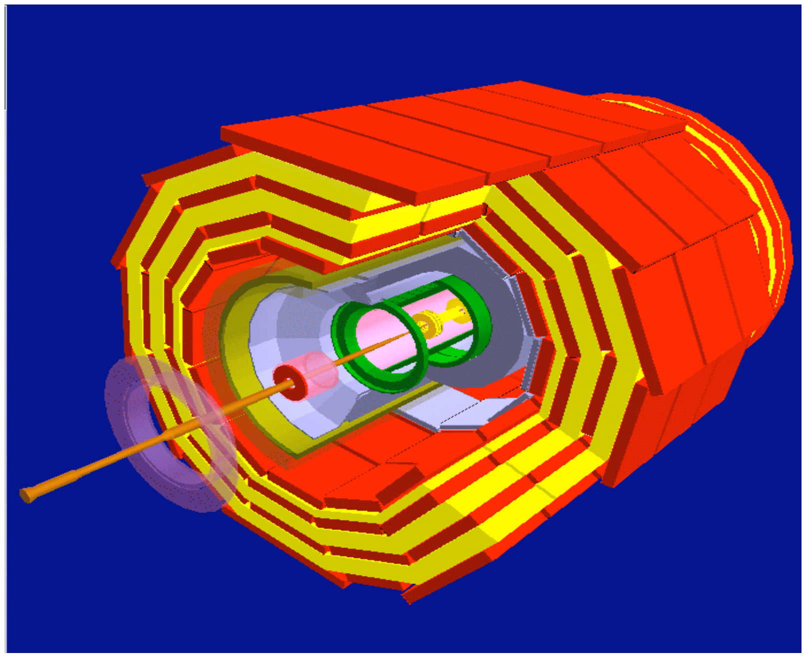
Geometry: DDD detector description system

The CMS detector description system (DDD) provides an application-independent way to describe the geometry

- Simulation, Reconstruction, Event Display etc. use the same basic geometr but with different views.

Geometry data are stored in a database with a Hierarchical Versioning SystemAlignment corrections are applied with reference to a given baseline geometry





Simulation

Event generator framework interfaces multiple packages

- including the Genser distribution provided by LCG-AA

Simulation with Geant4 since end 2003

- several Billions of events fully simulated up to now since mid-2005

Digitization tested and tuned with Test Beam, cosmics and first data

MC truth collection include info from particle gun or physics generator about vertices and particles. Stored in HepMC format.

Hit collection.

Hit object with timing, position, energy loss info.

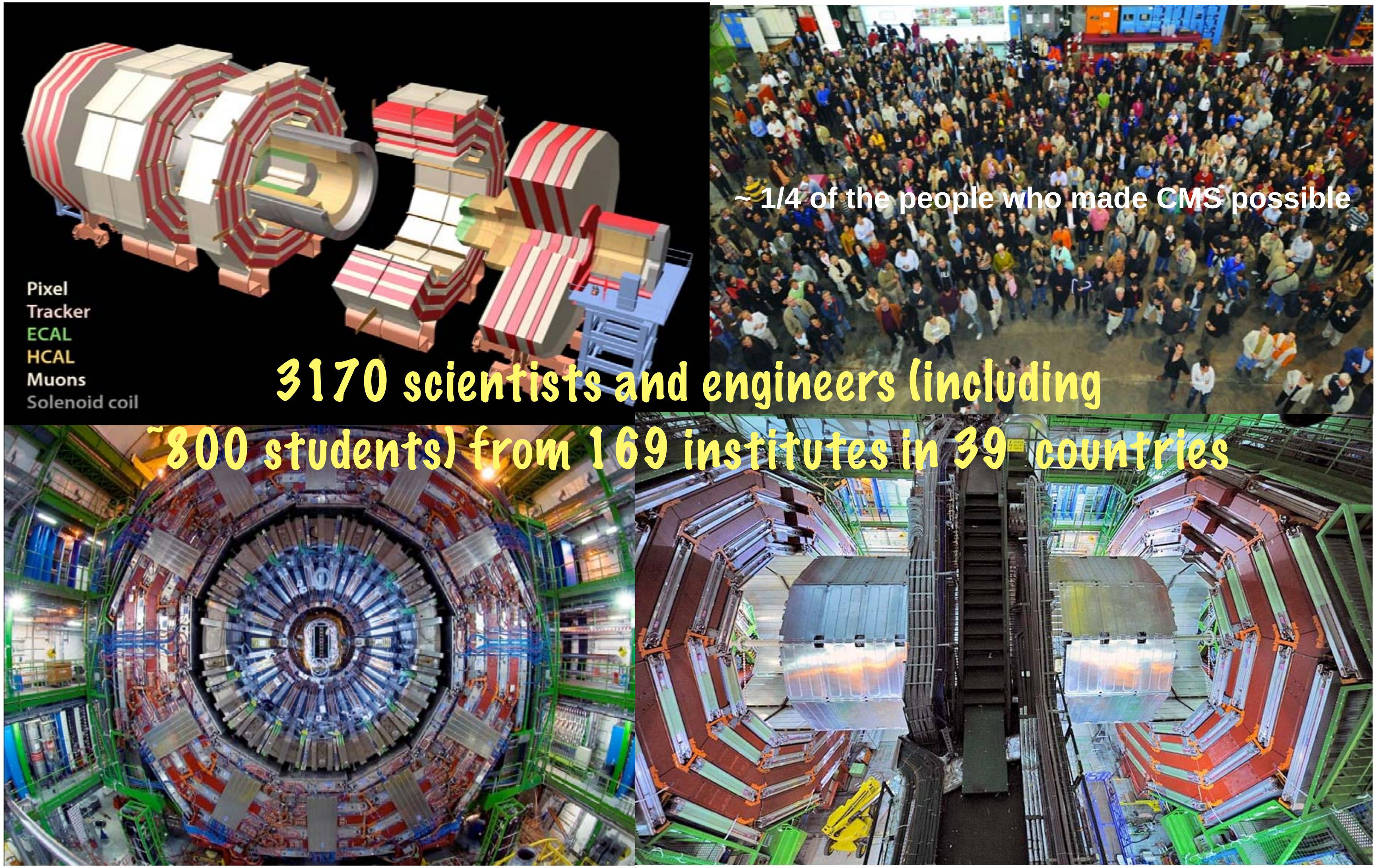
Based on Geant4

Digi Collection

Digi objects which include realistic modeling of electronic signal.

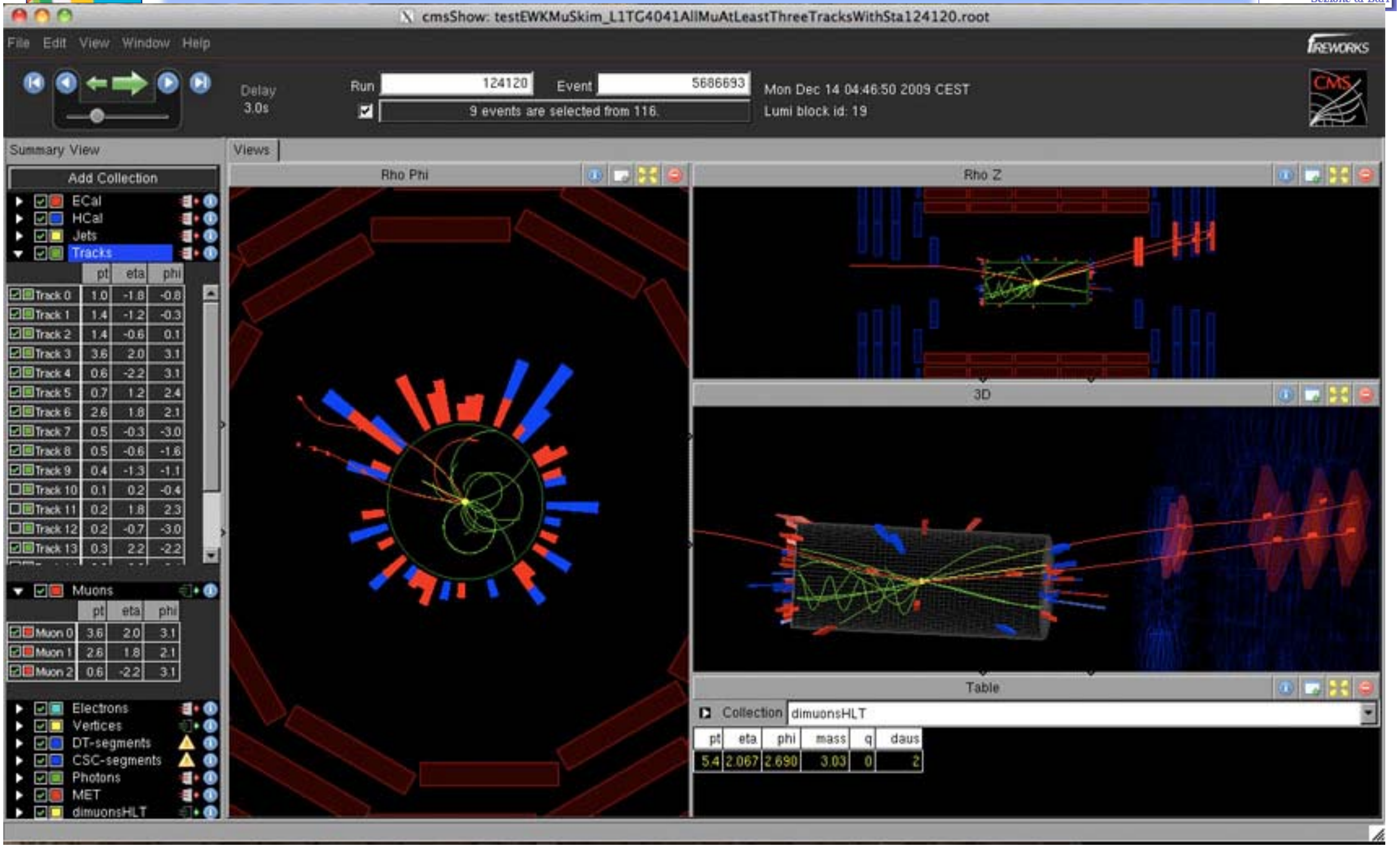
Event Visualization

CMS Collaboration: Many years of hard work and excitement...



3170 scientists and engineers (including ~800 students) from 169 institutes in 39 countries

Fireworks


cmsShow: testEWKMuSkim_L1TG4041AllMuAtLeastThreeTracksWithSta124120.root
File Edit View Window Help
FIREWORKS
Delay
3.0s
Run 124120
Event 5686693
Mon Dec 14 04:46:50 2009 CEST
9 events are selected from 116.
Lumi block id: 19
Summary View
Add Collection
ECal
HCal
Jets
Tracks
pt eta phi
Track 0 1.0 -1.8 -0.8
Track 1 1.4 -1.2 -0.3
Track 2 1.4 -0.6 0.1
Track 3 3.6 2.0 3.1
Track 4 0.6 -2.2 3.1
Track 5 0.7 1.2 2.4
Track 6 2.6 1.8 2.1
Track 7 0.5 -0.3 -3.0
Track 8 0.5 -0.6 -1.6
Track 9 0.4 -1.3 -1.1
Track 10 0.1 0.2 -0.4
Track 11 0.2 1.8 2.3
Track 12 0.2 -0.7 -3.0
Track 13 0.3 2.2 -2.2
Muons
pt eta phi
Muon 0 3.6 2.0 3.1
Muon 1 2.6 1.8 2.1
Muon 2 0.6 -2.2 3.1
Electrons
Vertices
DT-segments
CSC-segments
Photons
MET
dimuonsHLT
Views
Rho Phi
Rho Z
3D
Table
Collection dimuonsHLT
pt eta phi mass q daus
5.4 2.067 2.690 3.03 0 2

Visualization Tool

Fireworks is the light weight event display for analysis. It can be installed on your laptop. You can find it at:

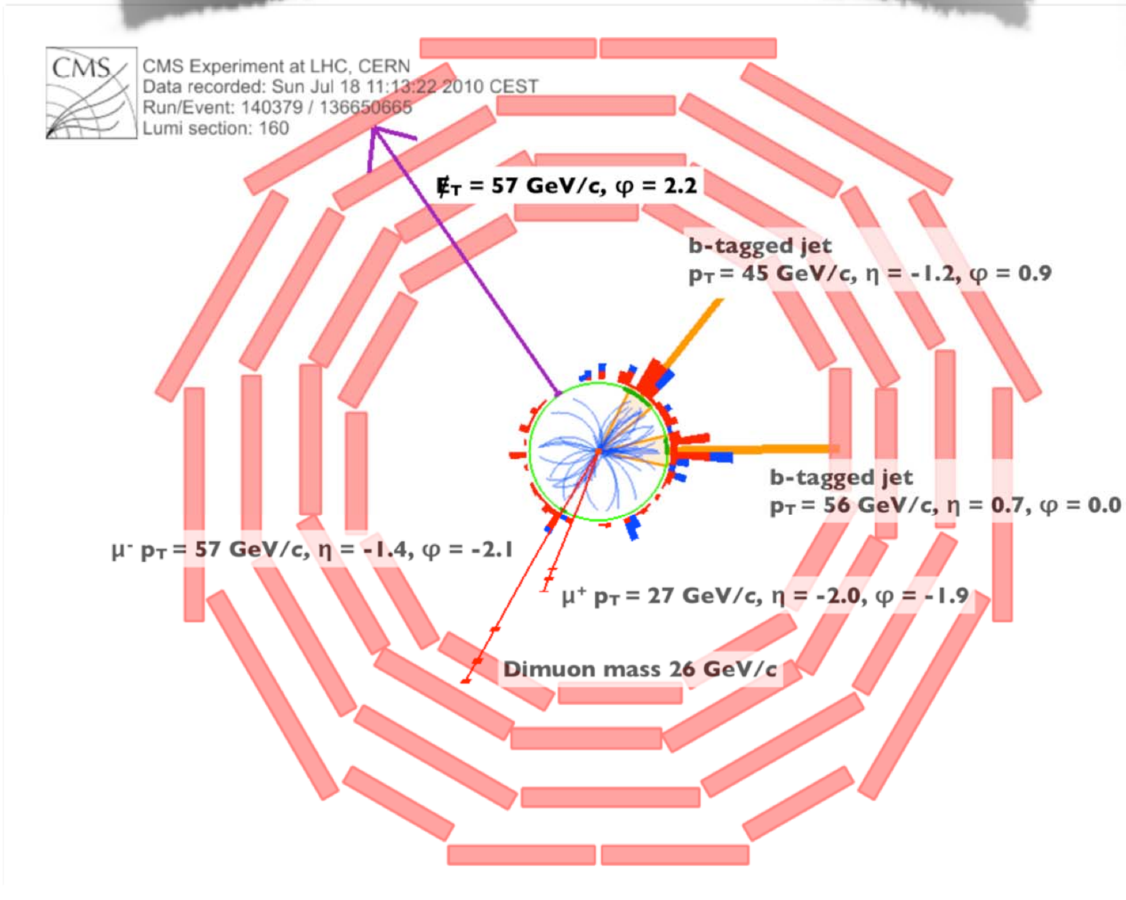
https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookFireworks

Try out the video tutorial!

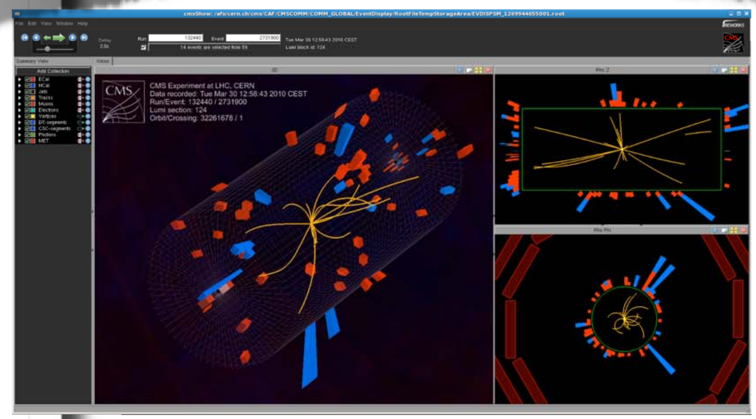
http://cern.ch/cms-sdt/fireworks/demo.mov

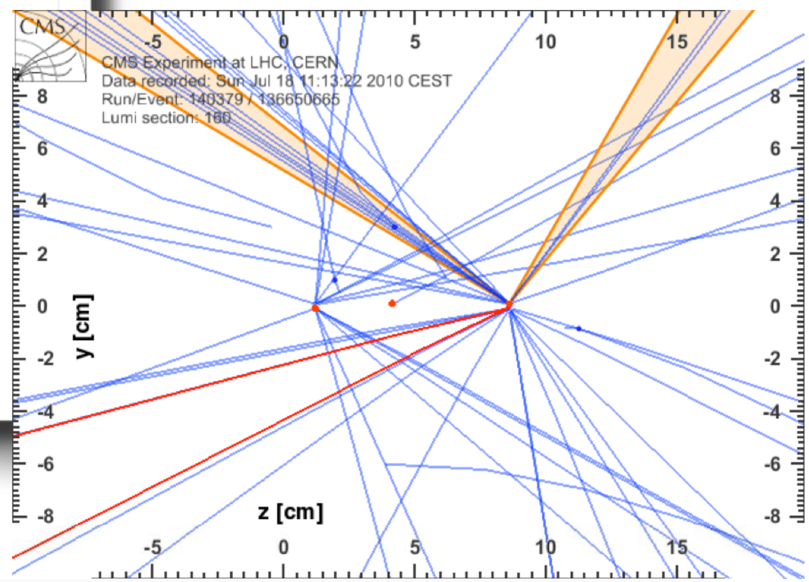
Event Display: Just works !!

18 July ttbar dilepton candidate



30 March first media event





Multiple primary vertices
multiple *pp* collisions ("pile-up").
Jets & muons originate from same vertex.

Reconstruction and Trigger

Reconstruction and Trigger

General feature:

- Multi-threading is necessary for online environment
- Most Algorithms & Tools are common with offline

Two big versions:

- Full reconstruction
- "seeded", or "partial", or "reconstruction inside a region of interest"
  - This one used in HLT

Online monitoring and event displays

- "Spying" on Trigger/DAQ data online (online DQM)
- But also in express analysis on CAF (Offline DQM)
- Online and Offline event display based on a "dedicated" express line

Trigger performance

LEVEL 1

reduces rate from 40 MHz -> 100 kHz

hardware based fast decision logic

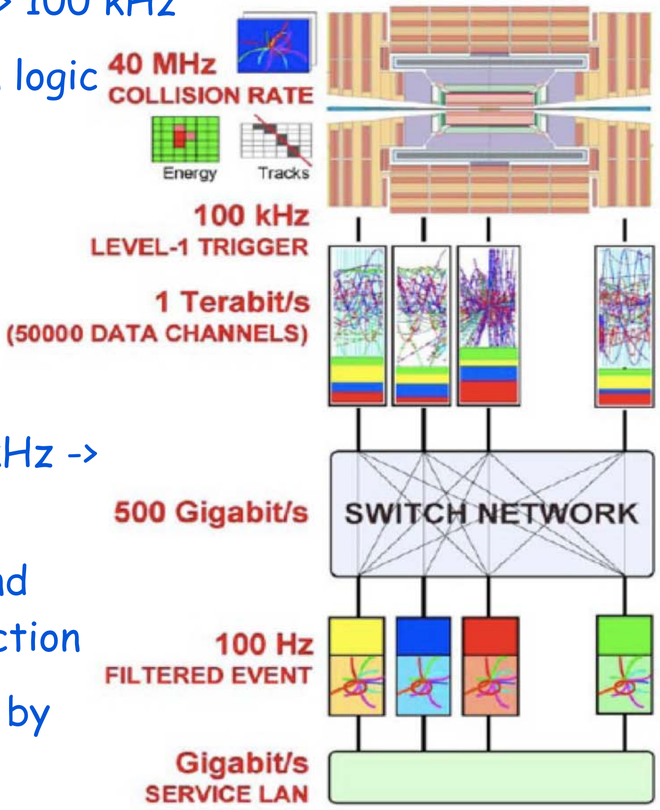
MUON SYSTEMS and

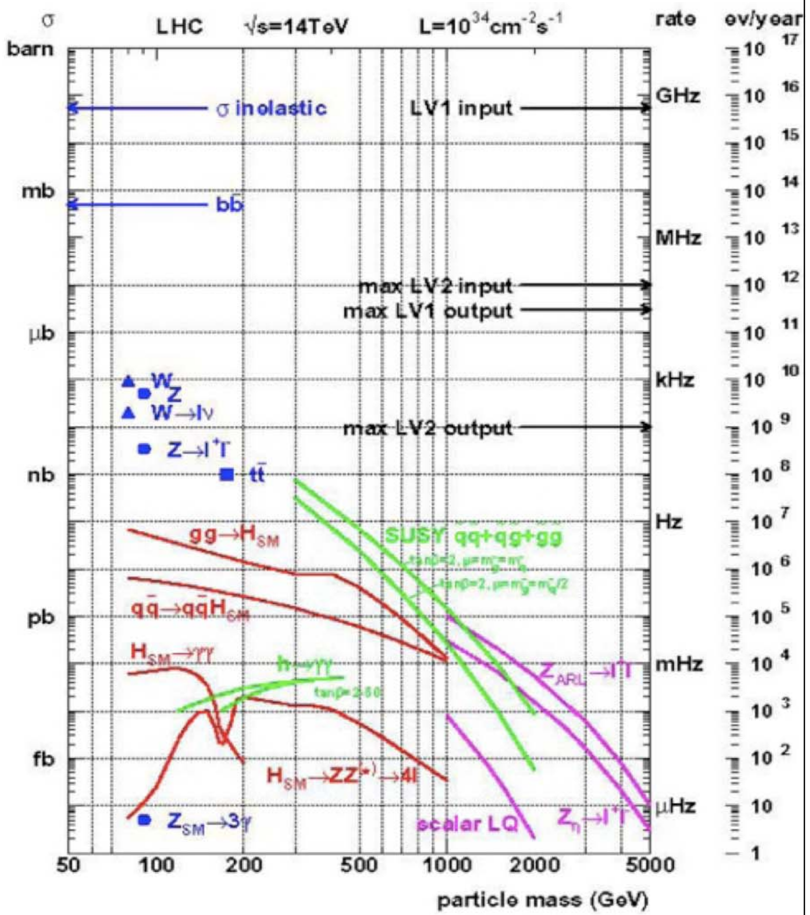
CALORIMETERS info used

- pipelined
- maximum latency: 4 μs

HLT reduces rate from 100 kHz -> O(100 Hz)

- uses full detector data and close to offline reconstruction
- HLT trigger paths seeded by L1 trigger objects





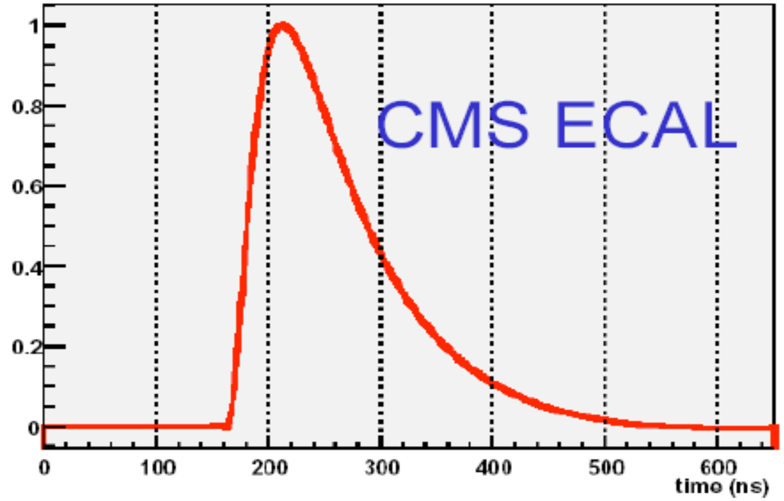
- Processing is done in a huge PC farm
- Events are classified and shipped out according to groups of trigger paths (datasets) for physics studies, Maximum accepted latency ~40ms/ev

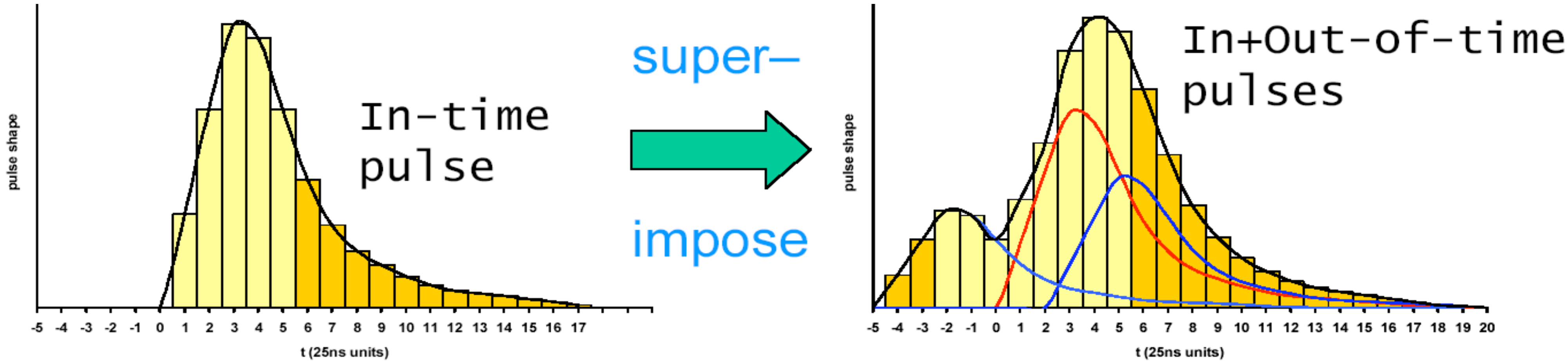
Pile-up & Electronics

"In-time" pile-up: particles from the same crossing but from a different pp interaction

Long detector response/pulse shapes:

- "Out-of-time" pile-up: left-over signals from interactions in previous crossings
- Need "bunch-crossing identification"

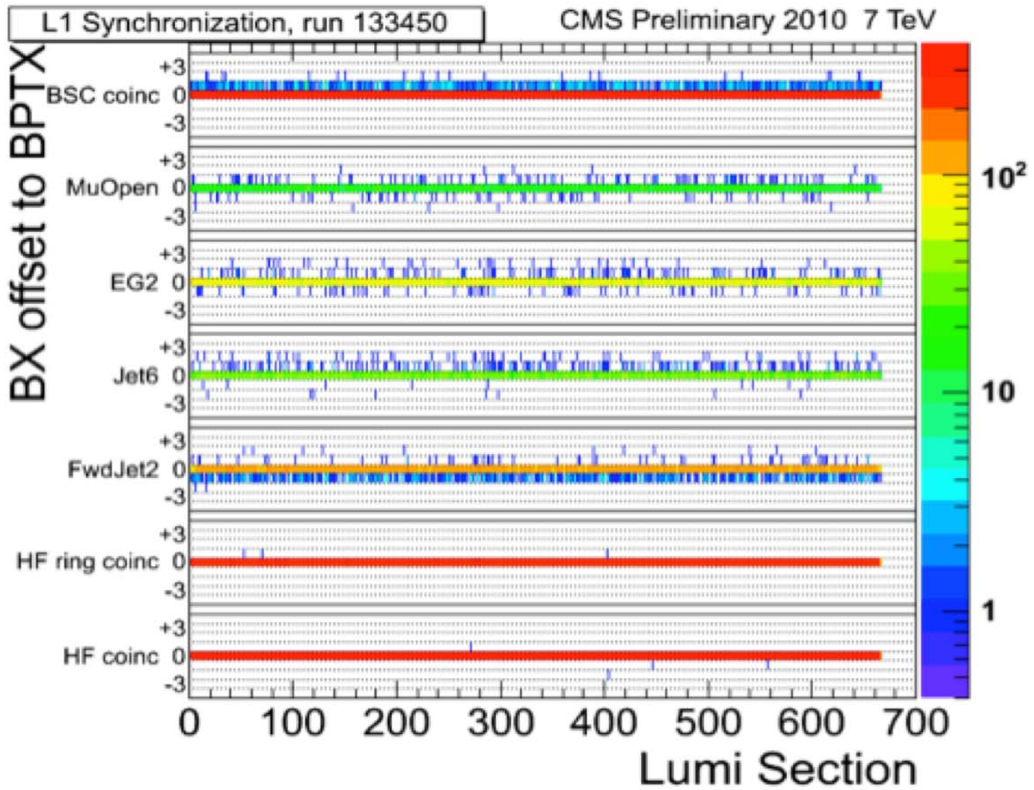


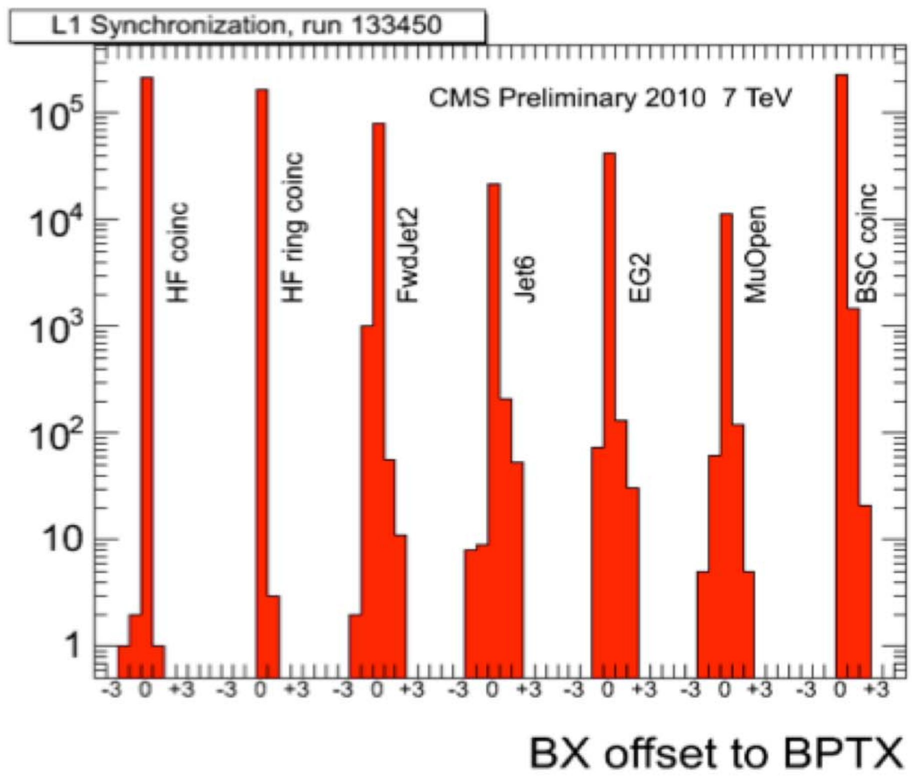


Trigger performance: L1

First essential activity after 30 March was the trigger and read-out synchronisation. Time scans we performed in all detectors and optimal point (maximum efficiency) were set. In most cases only minimal adjustments needed w.r.t. to cosmics and splash studies.

MinBias event sample

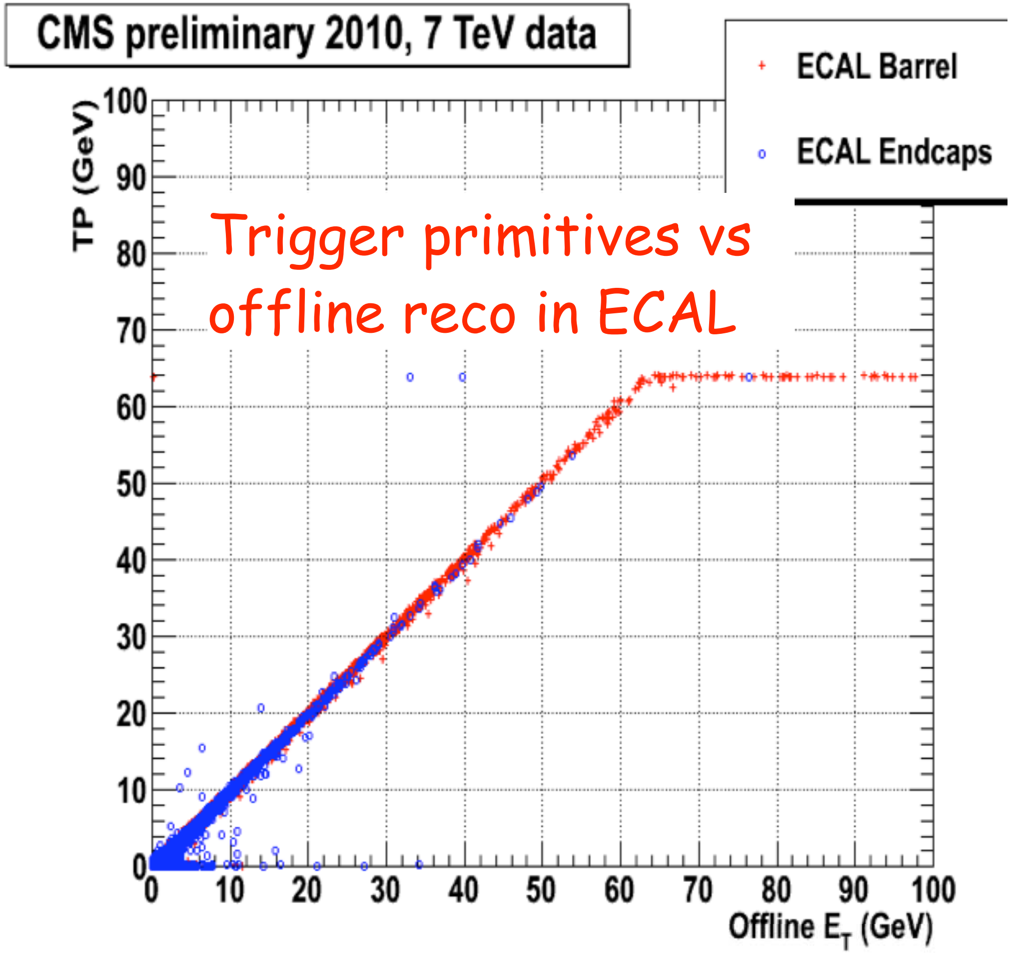


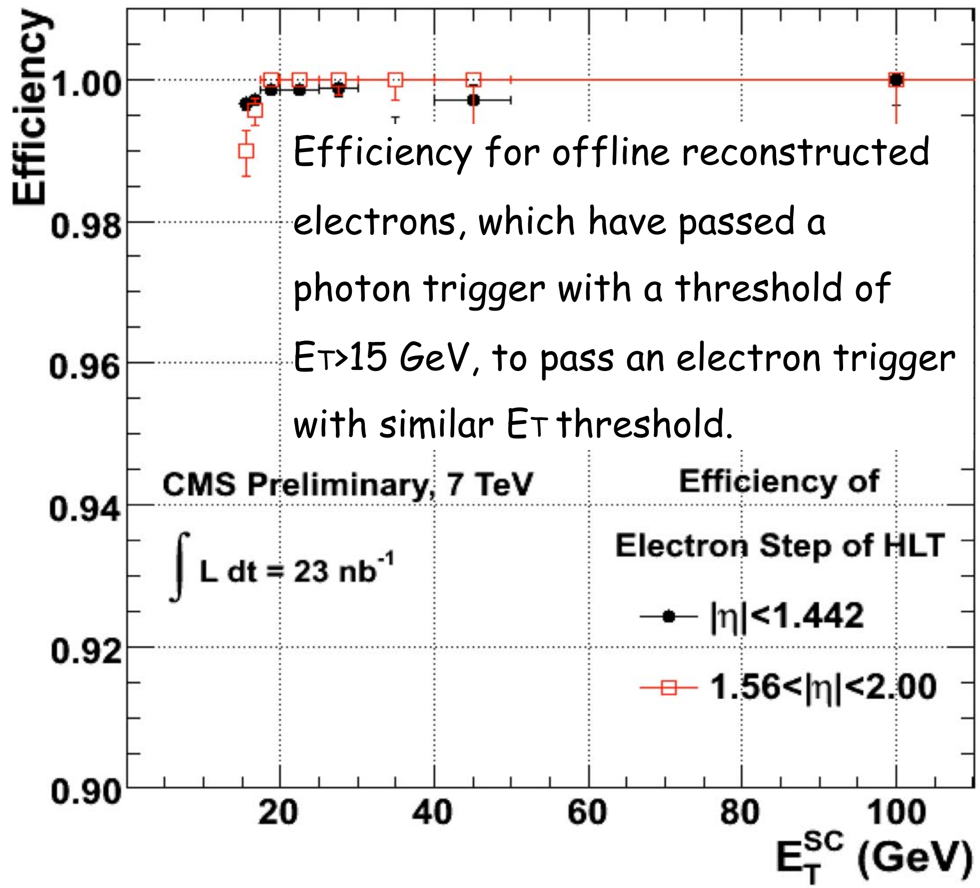


Trigger performance: L1+HLT

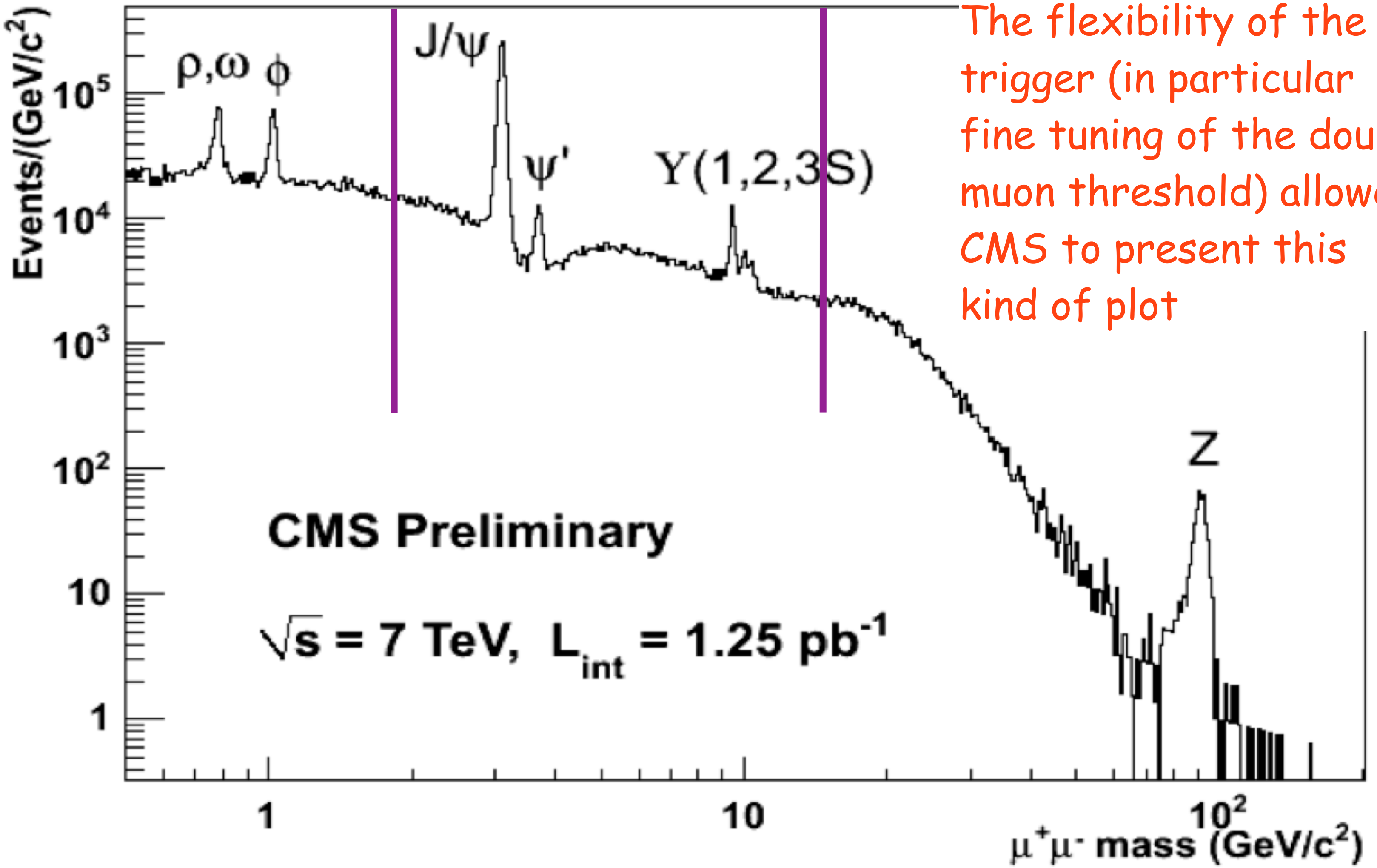
Systematically checked all turn-on curves and linearity of L1 vs offline full reco.

Just one example given here.



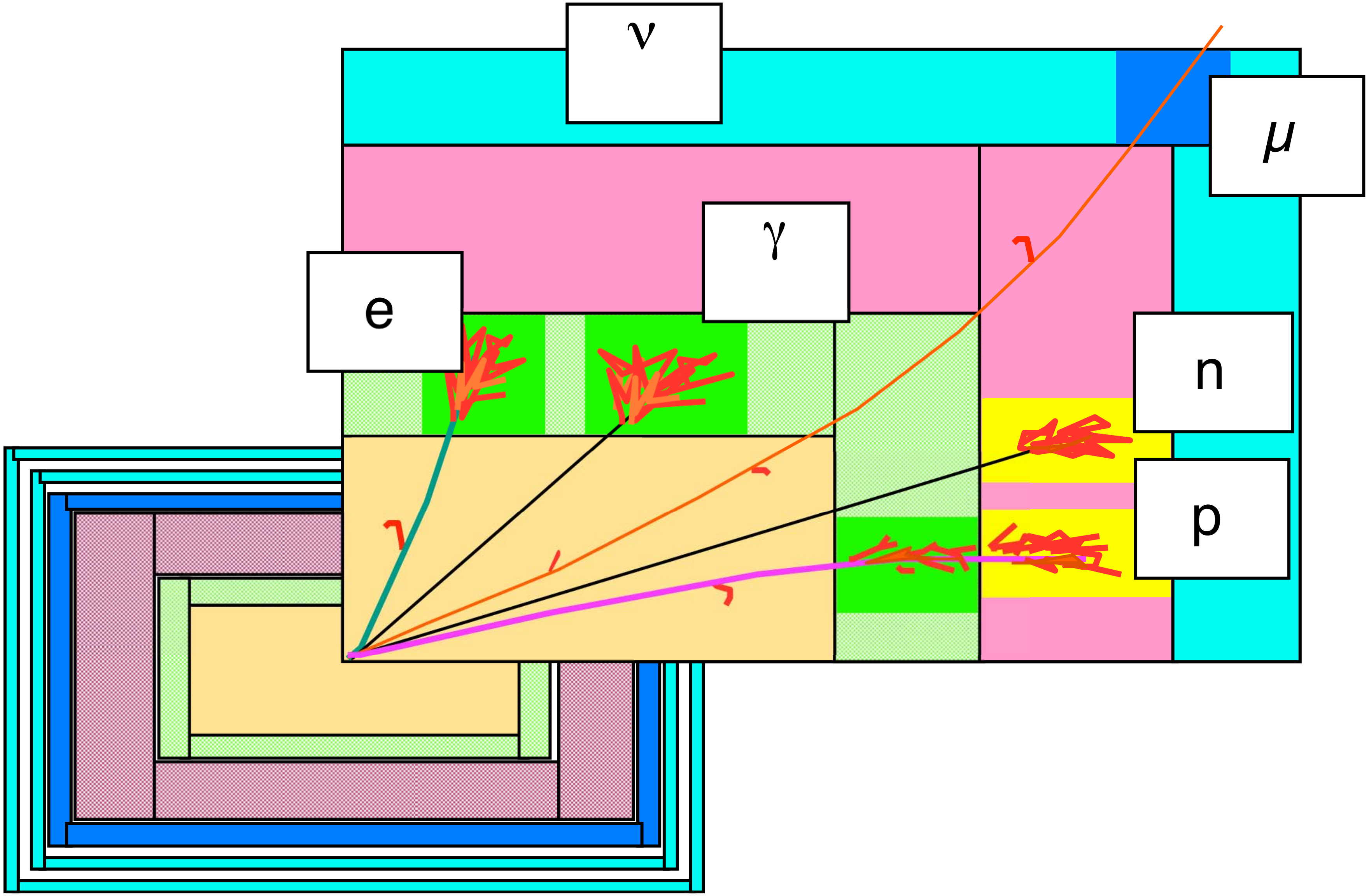


Trigger performance



The flexibility of the trigger (in particular fine tuning of the double muon threshold) allowed CMS to present this kind of plot

Let's concentrate on Inner Tracking



CMS Inner Tracker ...

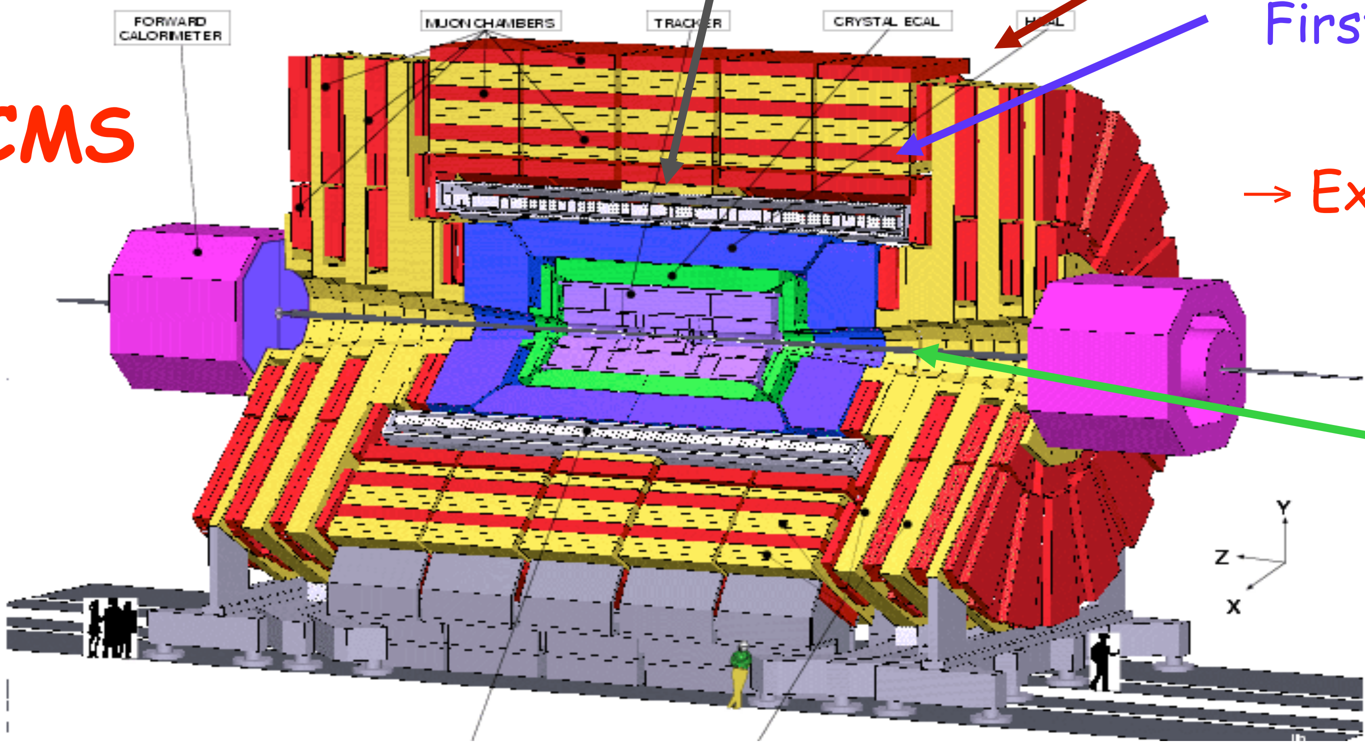
CMS Inner Tracker ...

13m x 6m Solenoid: 4 Tesla Field

→ Tracking up to η ~ 2.4

Muon system in return yoke

CMS



First muon chamber just after solenoid

→ Extended lever arm for p_t measurement

ECAL & HCAL

Inside solenoid

22m Long, 15m Diameter, 14'000 Ton Detector

CMS Inner Tracker ...

CMS Inner Tracker ...

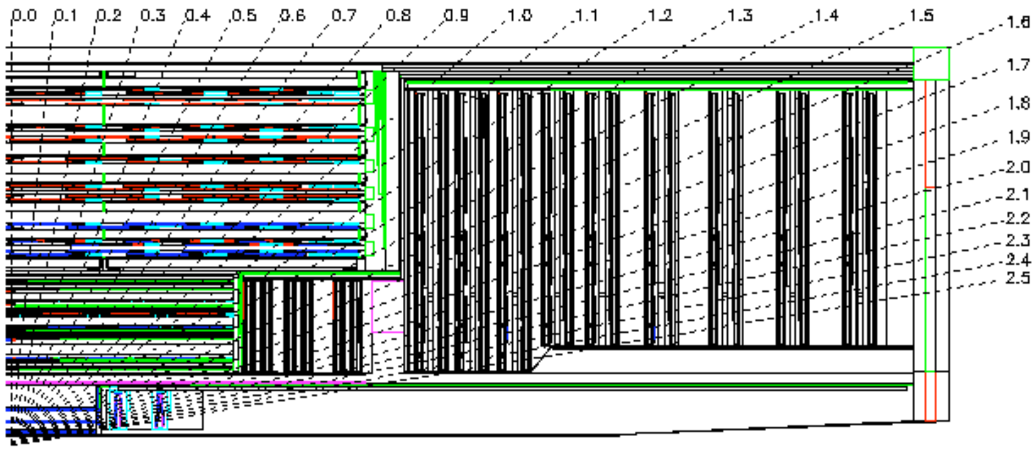
CMS has chosen an all-silicon configuration

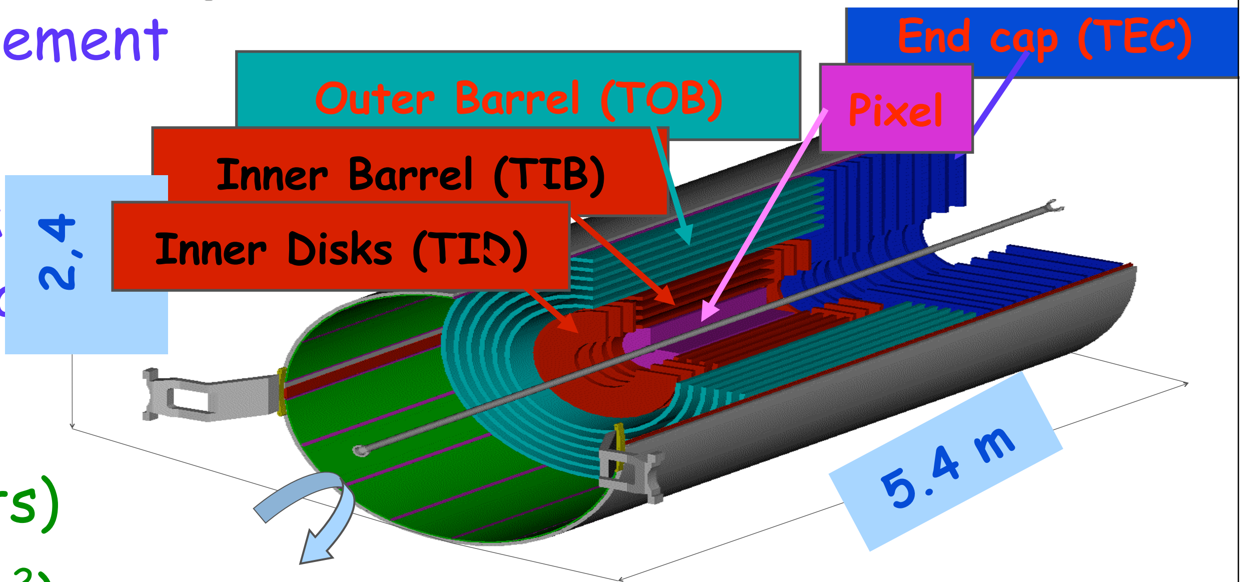
CMS Tracker

Inside 4T solenoid field

Tracking rely on "few" measurement layers, each able

to provide robust (clean) a
precise coordinate determin





Precision Tracking:

- Pixel detector (2-3 points)
- Silicon Strip Tracker (220 m^2) SST (10 – 14 points)

volume 24.4 m^3

running temperature – 10 ^0C

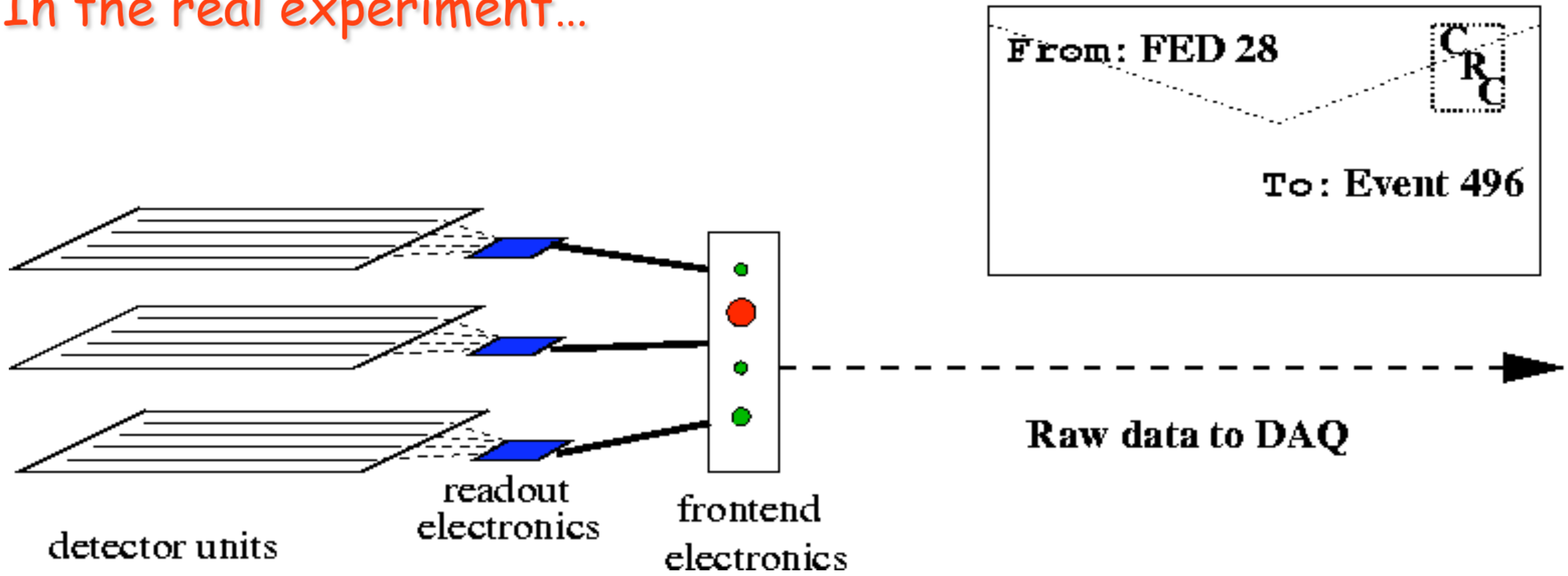
Raw data (Digis) coming from detectors..

Raw data formation is not reconstruction

For the purpose of on-line reconstruction

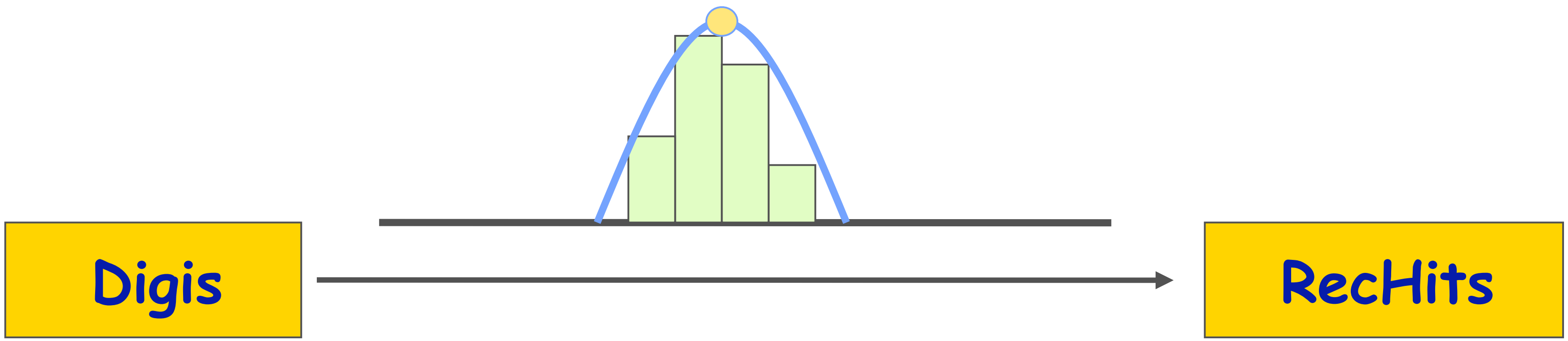
DAQ is like the post: the front ends send packets...

In the real experiment...



From Digi to Local Reconstruction: Clusterization

This is the process that, given a set of Digis, recreates the cluster, with its position and estimated error.



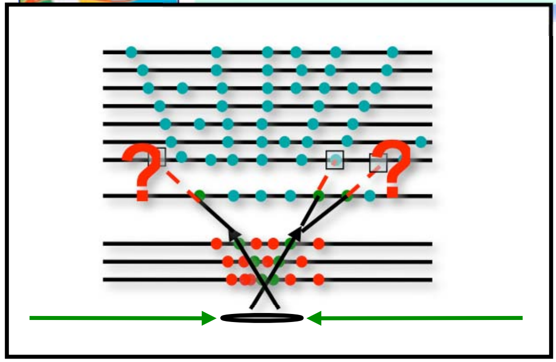
Important quantities are:

position: the cluster position must be as close as possible to the Simulated hit position, not to bias the reconstruction

error: important for the tracking, to estimate how far a Reconstructed hit is from the expected track intersection with the detector surface

CMS Tracking in a nutshell

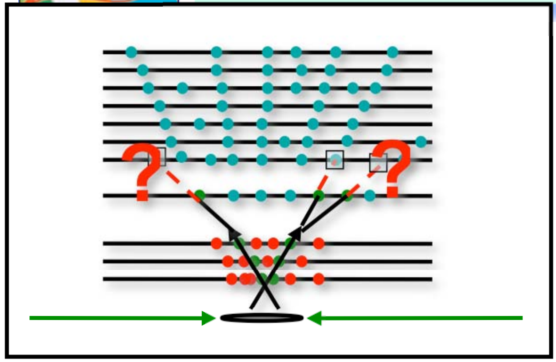
CMS Tracking in a nutshell



Seeding starts from innermost pixel layers.

Inside-out trajectory building

CMS Tracking in a nutshell

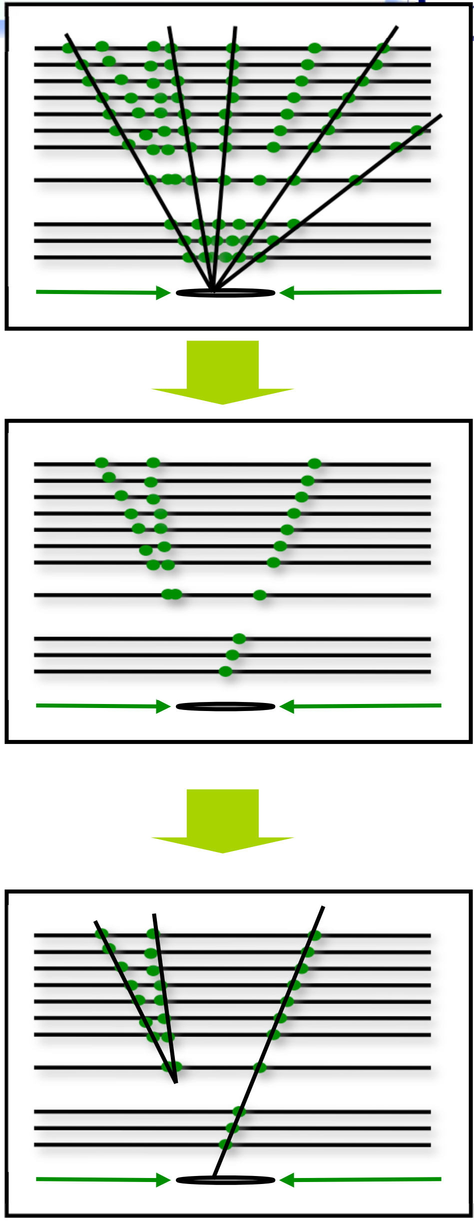


Seeding starts from innermost pixel layers.

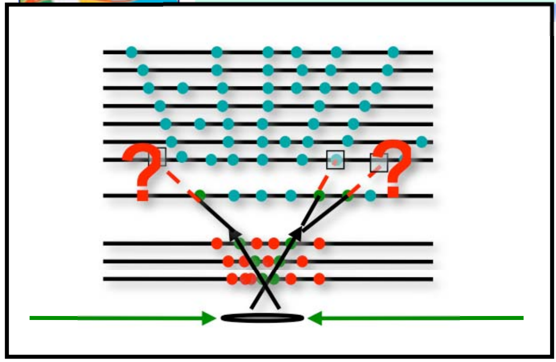
Inside-out trajectory building

Iterative tracking with hits-removal

(6 iterations like this)



CMS Tracking in a nutshell

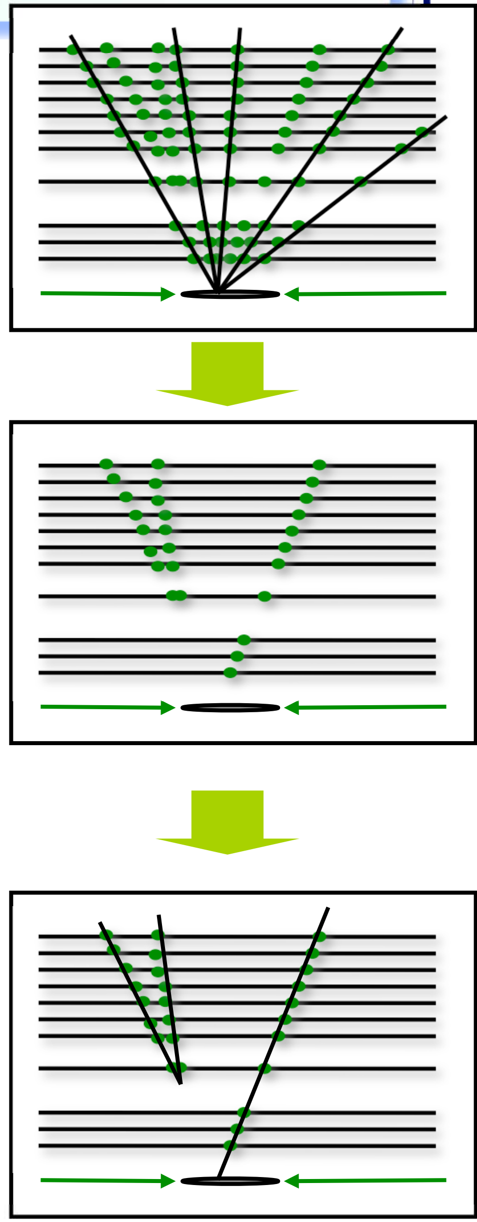


Seeding starts from innermost pixel layers.

Inside-out trajectory building

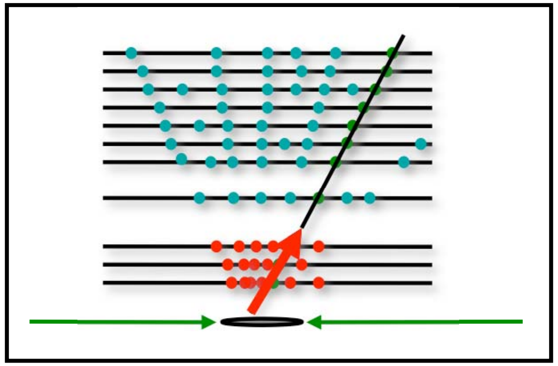
Iterative tracking with hits-removal

(6 iterations like this)



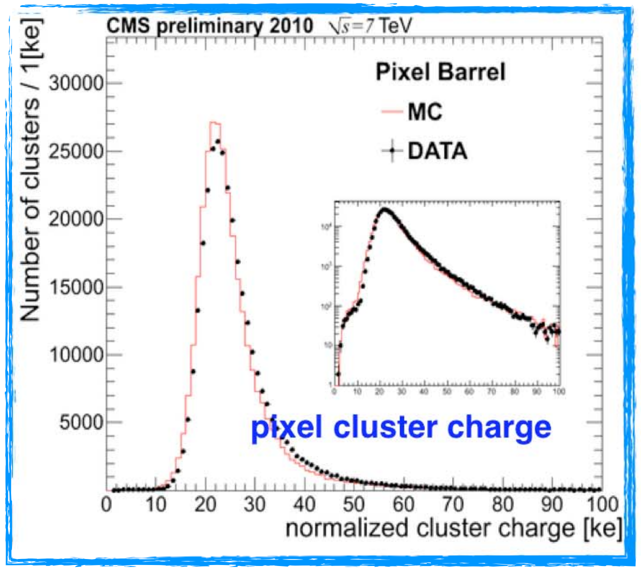
Final fit using **Kalman Filter/ Smoother**.

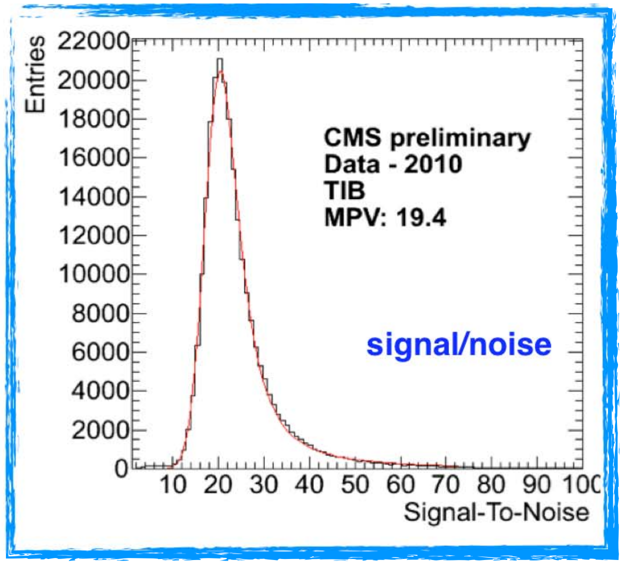
Parameters propagated through magnetic field inhomogeneities using **Runge-Kutta propagator**

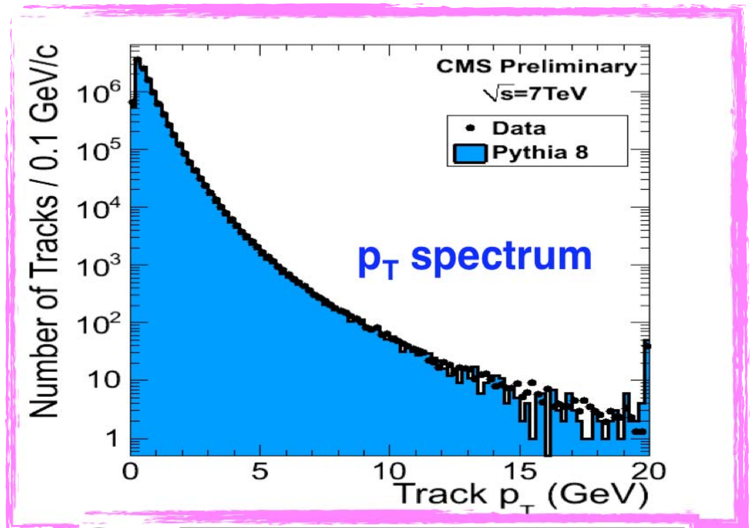


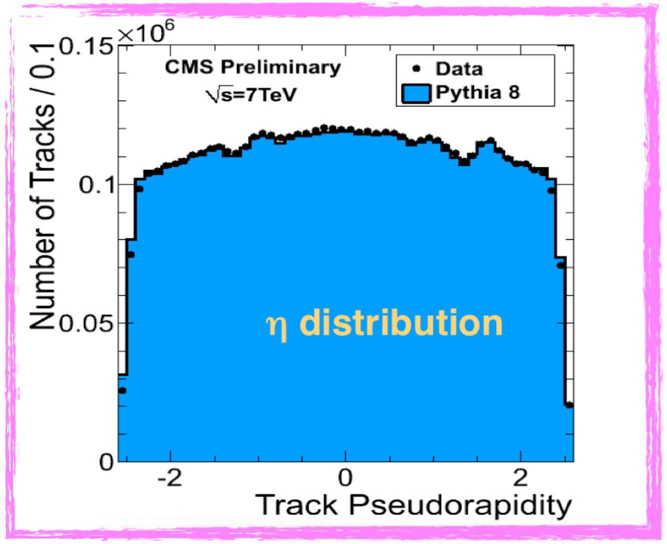
Track Parameters (q/p,eta,phi,dz,d0)

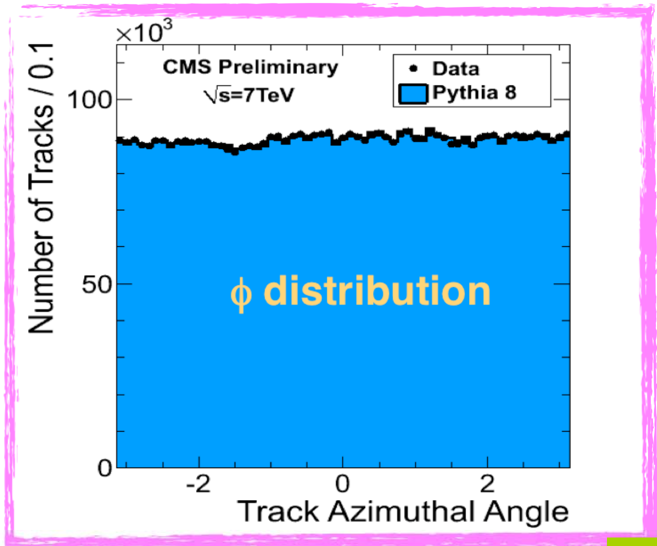
Tracker and Tracking Performance

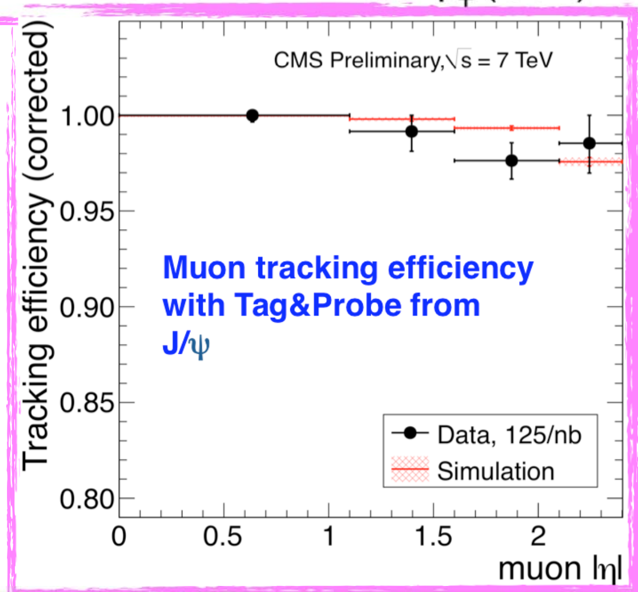








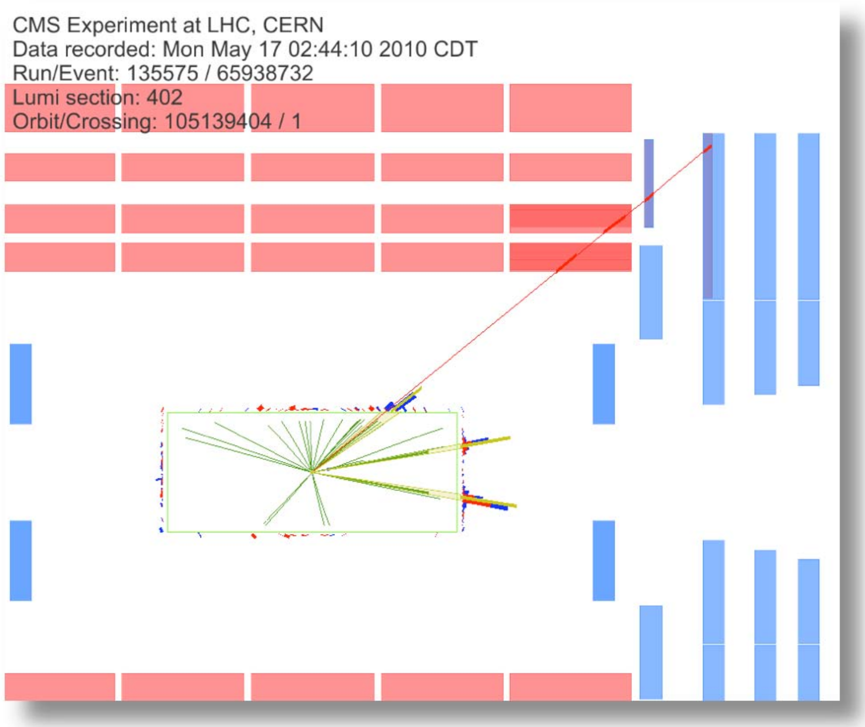




Measured tracking efficiency close to 99% and compatible with simulation

Muon identification

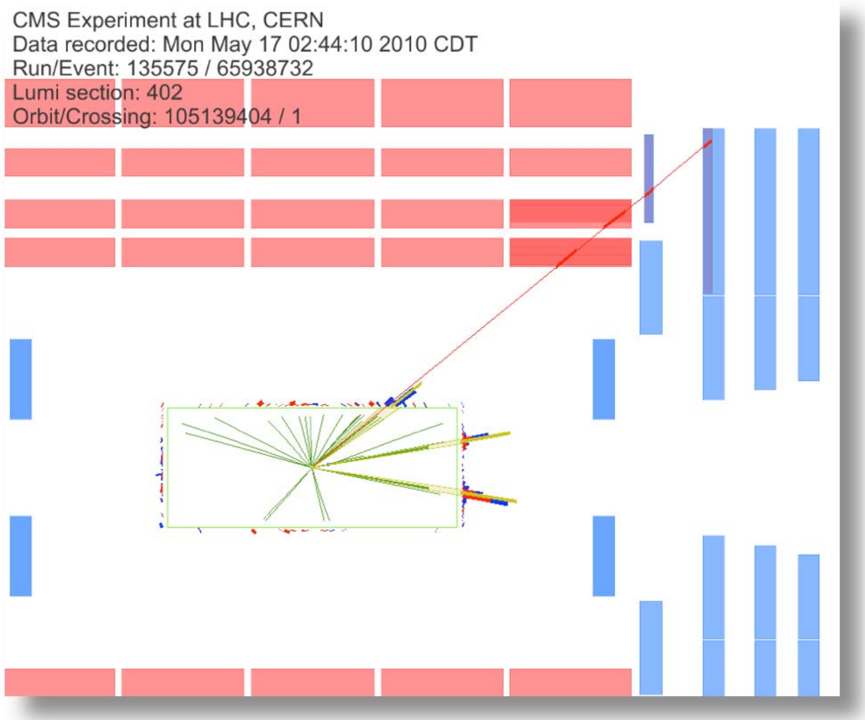
Different algorithms are available.



Muon identification

Different algorithms are available.

- **Soft muon:** a tracker track matched to at least one CSC or DT stub, to collect muons down to p_T about 500 MeV in the endcaps (e.g. for J/Ψ)

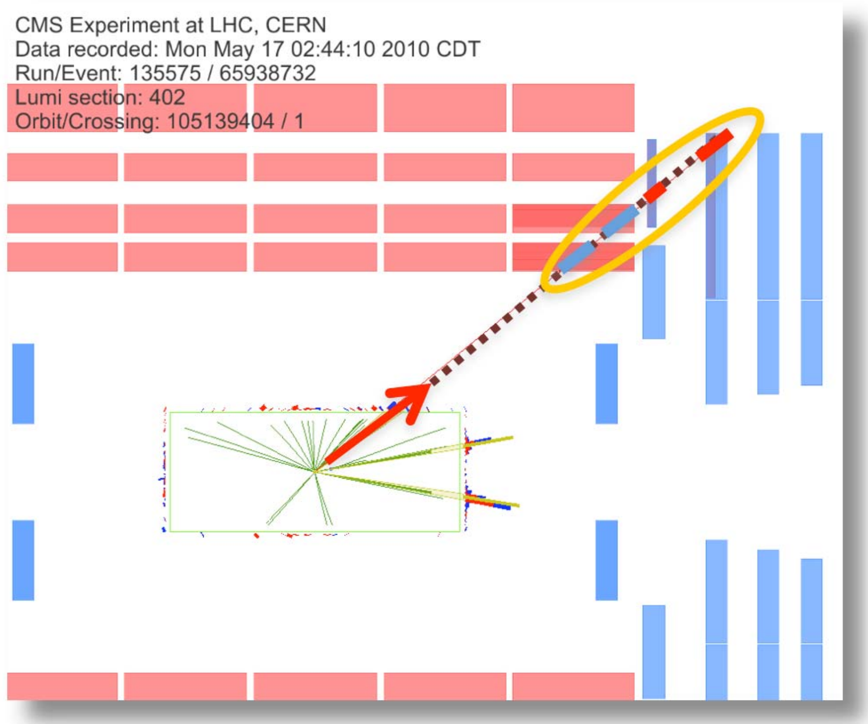


Muon identification

Different algorithms are available.

- **Soft muon:** a tracker track matched to at least one CSC or DT stub, to collect muons down to p_T about 500 MeV in the endcaps (e.g. for J/Ψ)

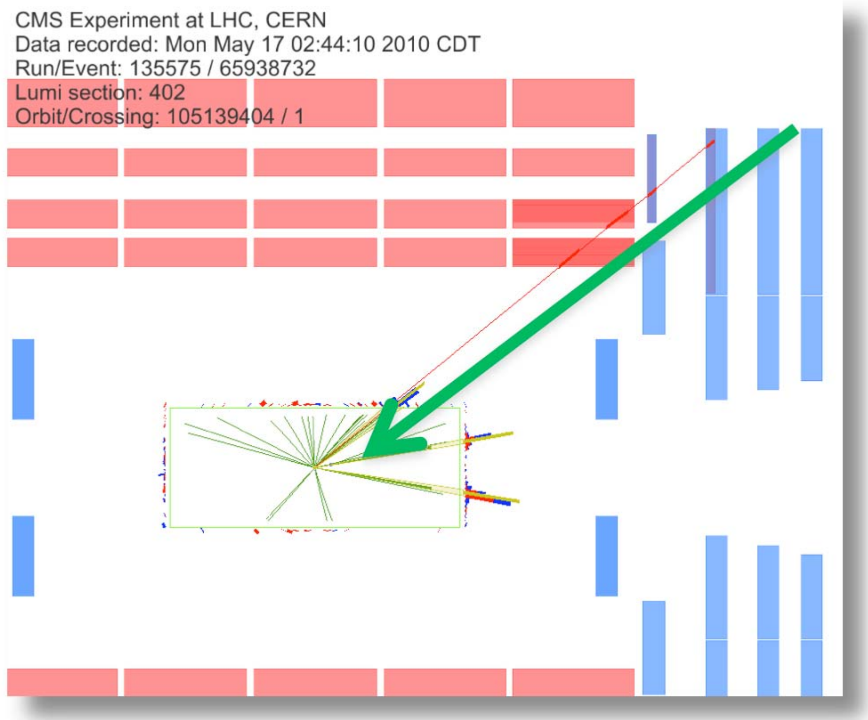
- **Tight muon:** a good quality track from a combined fit of the hits in the tracker and muon system, requiring signal in at least two muon stations to improve purity (e.g. for W, Z).



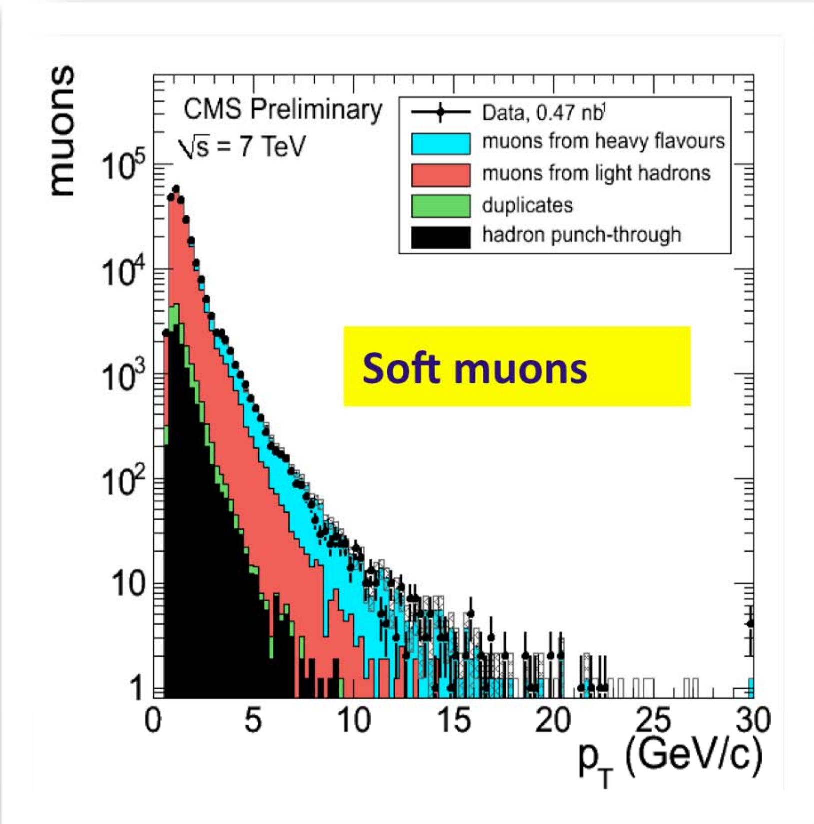
Muon identification

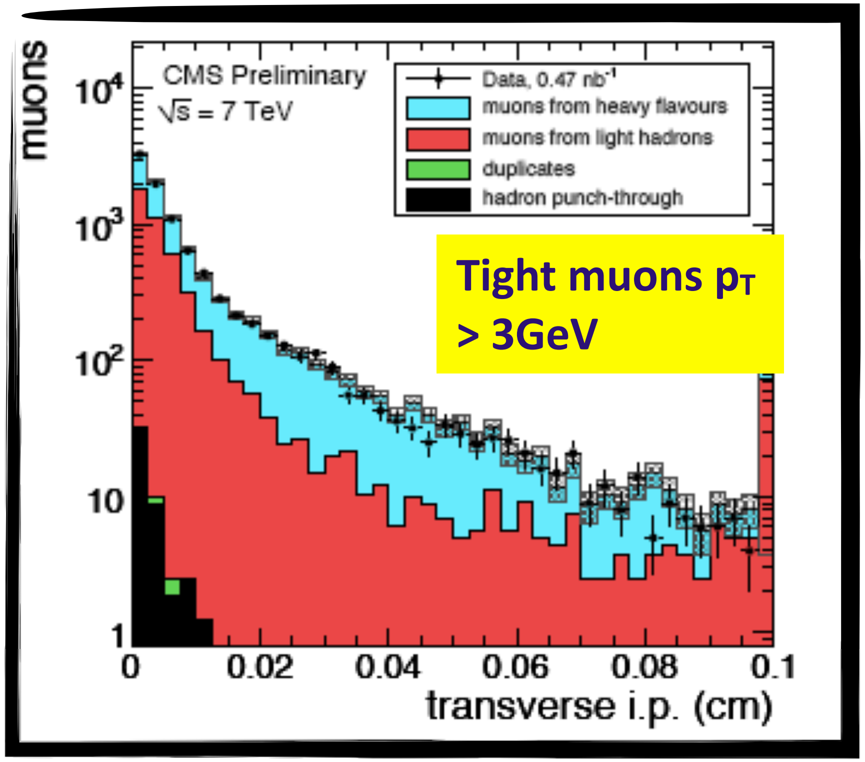
Different algorithms are available.

- **Soft muon:** a tracker track matched to at least one CSC or DT stub, to collect muons down to p_T about 500 MeV in the endcaps (e.g. for J/Ψ)
- **Tight muon:** a good quality track from a combined fit of the hits in the tracker and muon system, requiring signal in at least two muon stations to improve purity (e.g. for W, Z).



Muon Performance in minimum bias Trigger





84% π/k decays

9% b/c decays

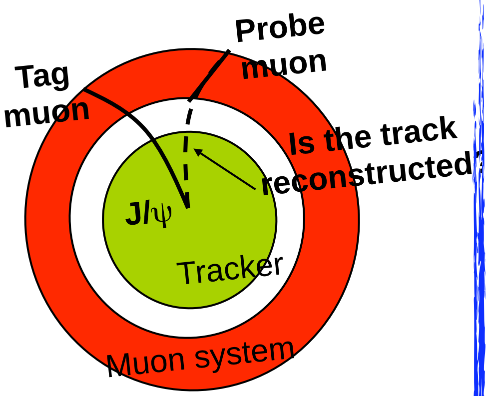
4.4% hadron punch-through

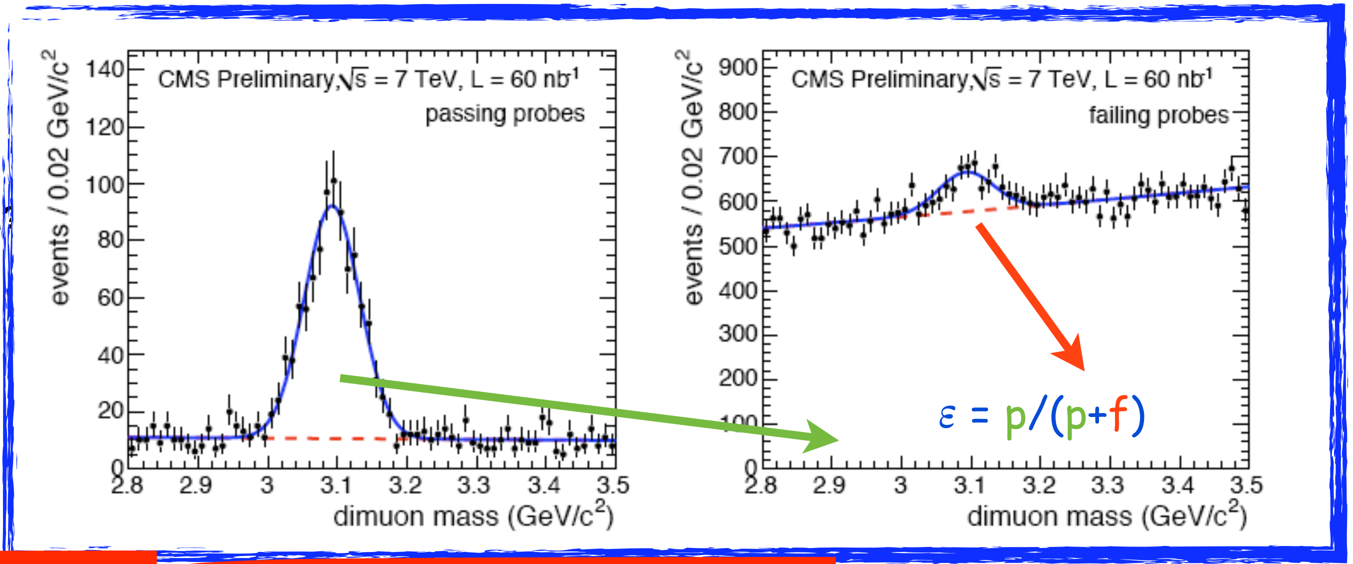
47,5% π/k decays

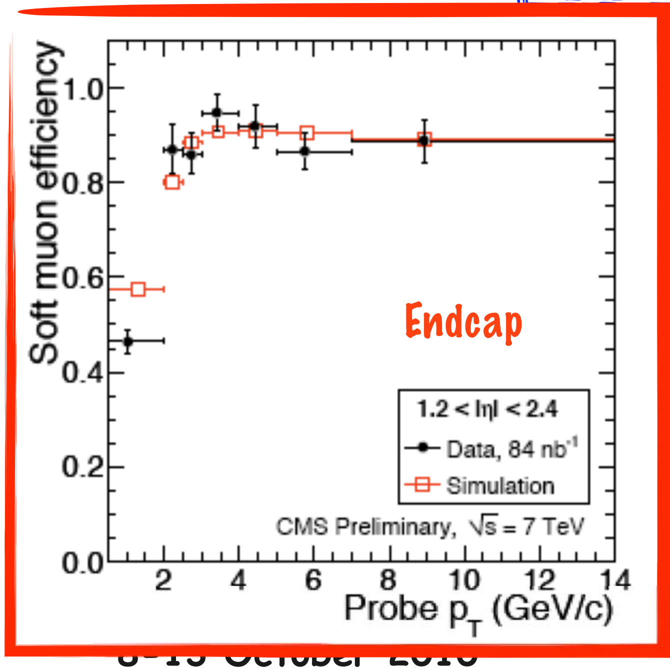
52,0% b/c decays

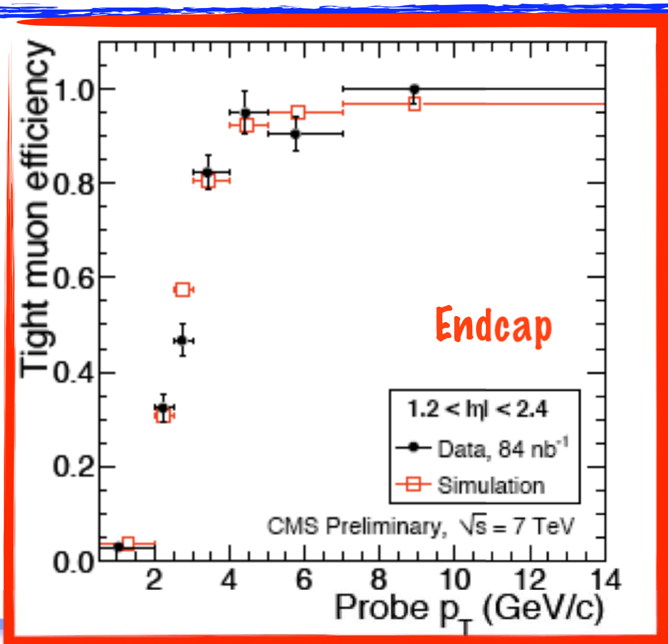
0.5% hadron punch-through

Muon Efficiency from J/Ψ (Low P_T)







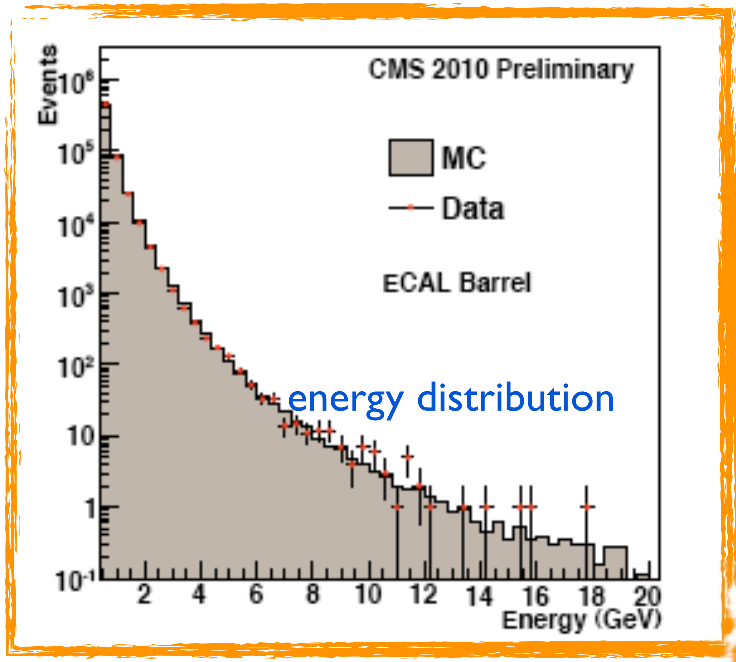


Agreement Data MonteCarlo

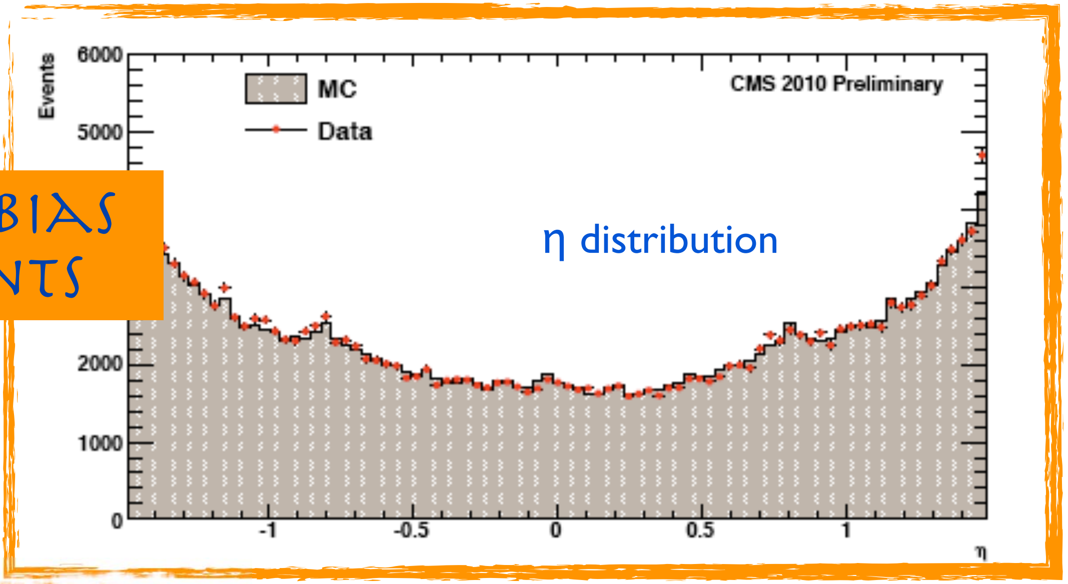
at 5-10% level

.... just few months after the startup

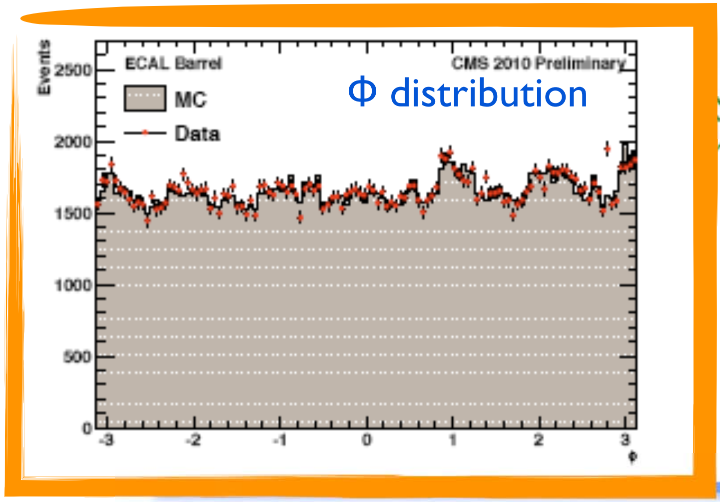
ECAL (Electron and Photons)

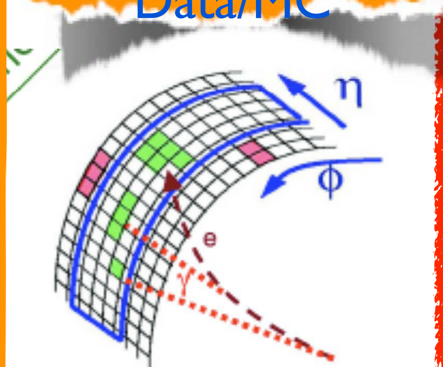


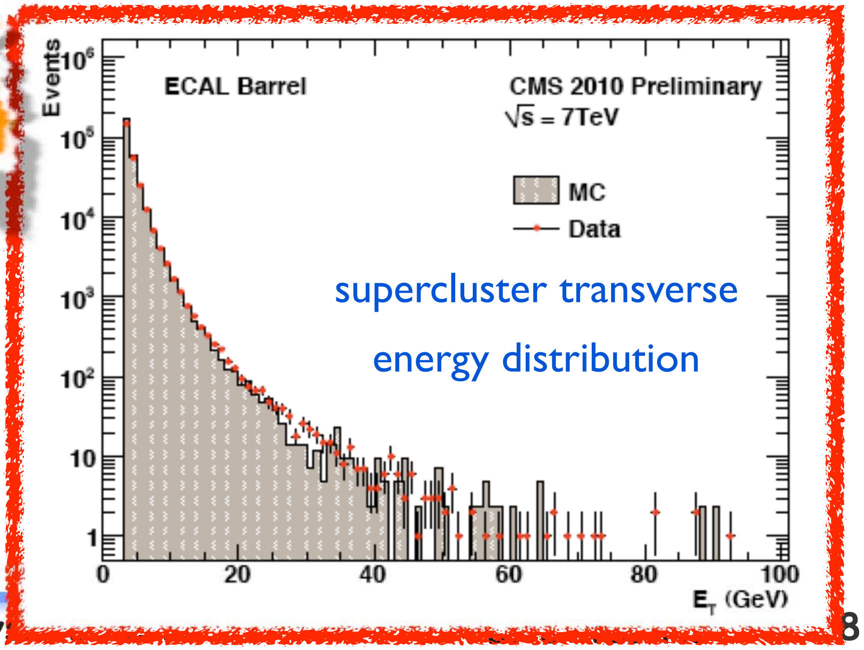
MINBIAS EVENTS



Very good agreement Data/MC







Electron Reconstruction

Energy clustering to recover bremsstrahlung

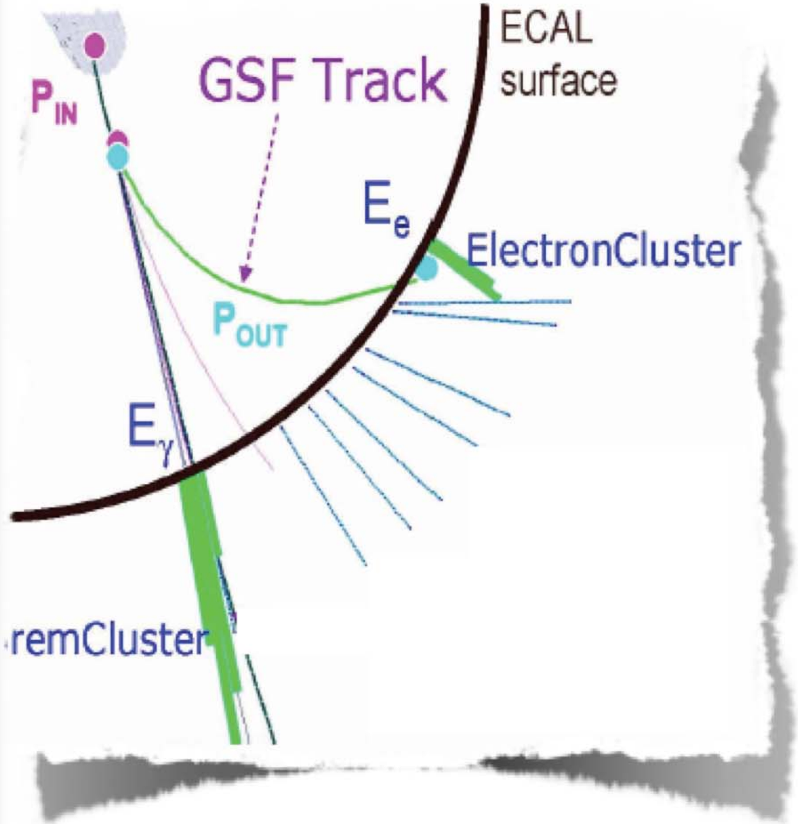
Superclusters are built by collecting cluster of crystal within Φ window

Electron seeding two complementary algorithms:

- Start from ECAL superclusters and search for compatible hits in the tracker inner layers (ECAL driven)
- Start from tracks (Tracker driven)

Electrons tracking

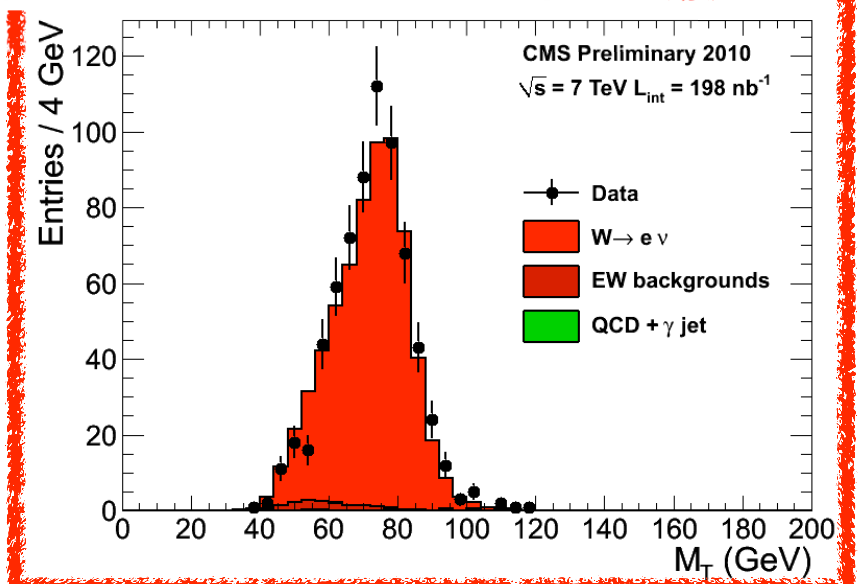
Bremsstrahlung energy loss modeled with a mixute of Gaussians (Gaussian Sum Filter)

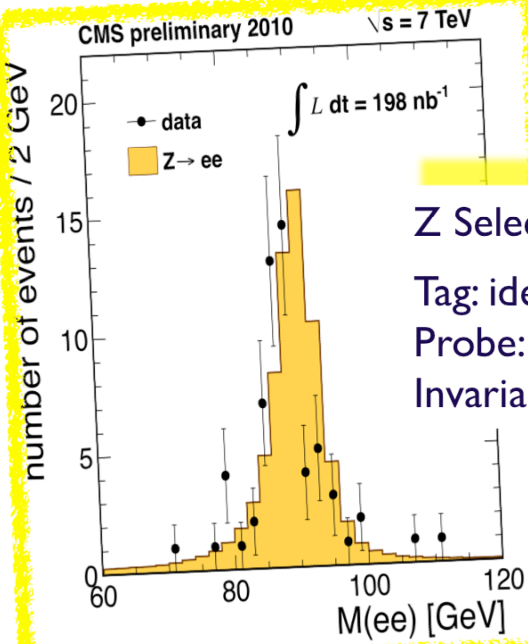


Electron Efficiency from W&Z (high p_T)

W Selection:

high MET, 1 high energy ECAL supercluster, little hadronic activity

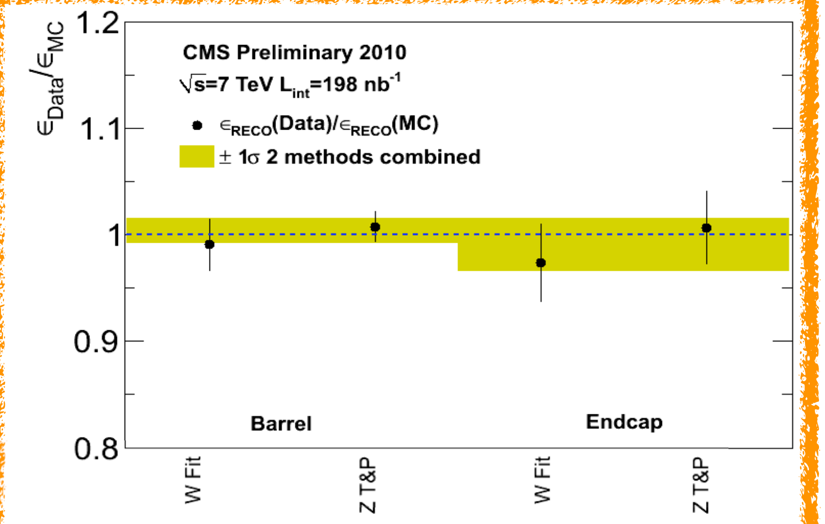




Z Selection:

Tag: identified/isolated electron, Probe: 1 ECAL supercluster, Invariant Mass

Z Tag & Probe	Measured efficiency	Error (stat. + syst)	MC efficiency
Reco Eff Barrel	**99.3%**	**1.4%**	**98.5%**
Reco Eff Endcap	**96.8%**	**3.4%**	**96.1%**



Low Mass di-photons: π^0/η

$1.46M$ of $\pi^0 \to \gamma\gamma$

$P_T(\gamma) > 0.4$ GeV,

$P_T(pair) > 1$ GeV

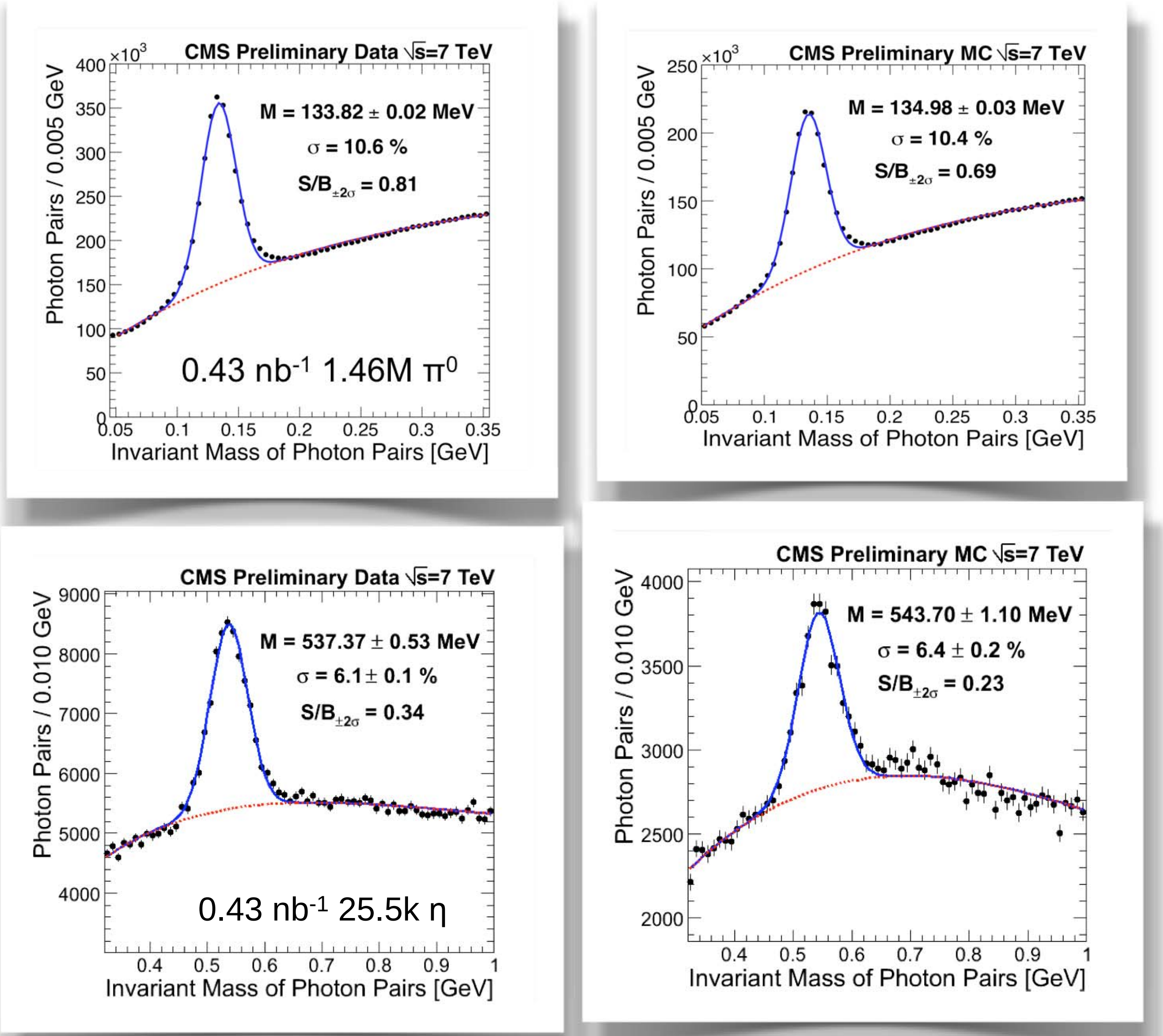
$25.5K$ $\eta \to \gamma\gamma$

$P_T(\gamma) > 0.5$ GeV,

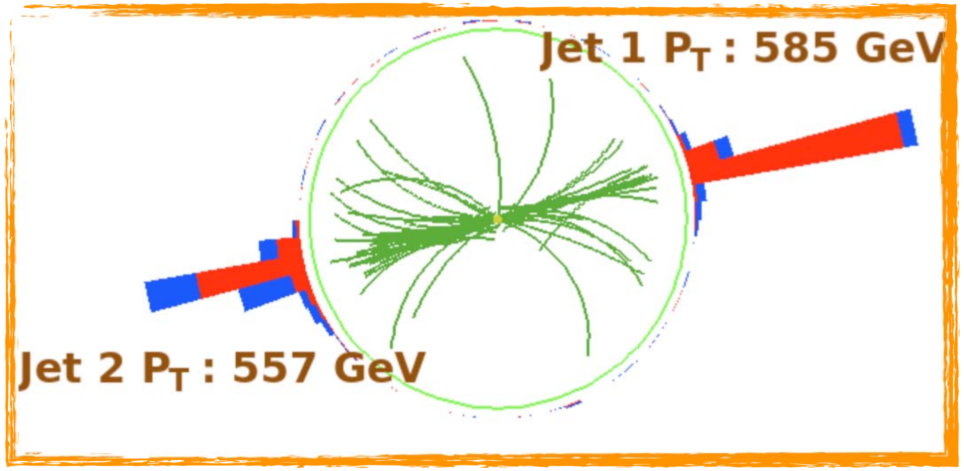
$P_T(pair) > 2.5$ GeV

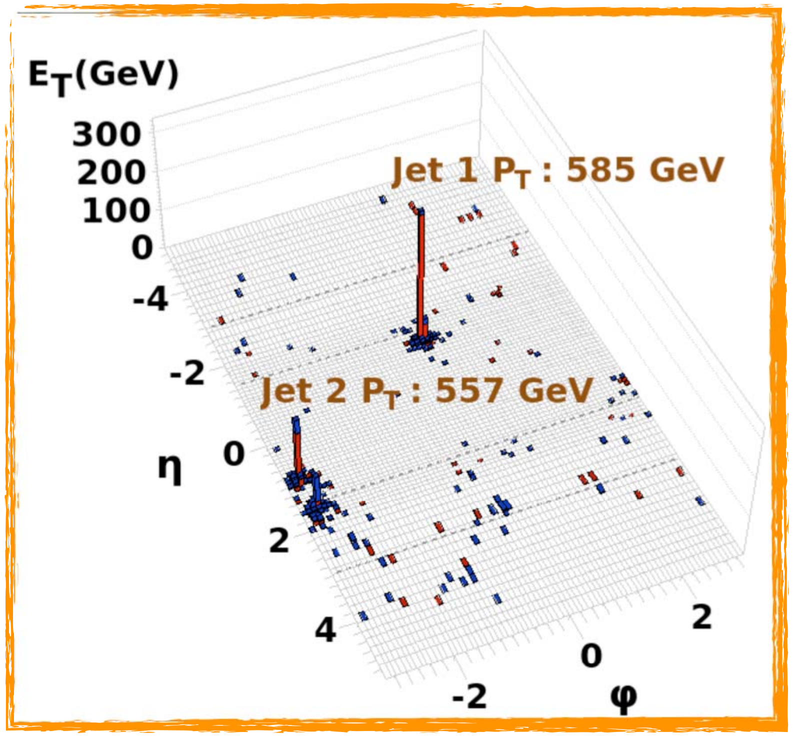
Numbers refer to a few % of the currently available statistics.

Very useful tool to inter-calibrate the crystals.



Di-Jets Events

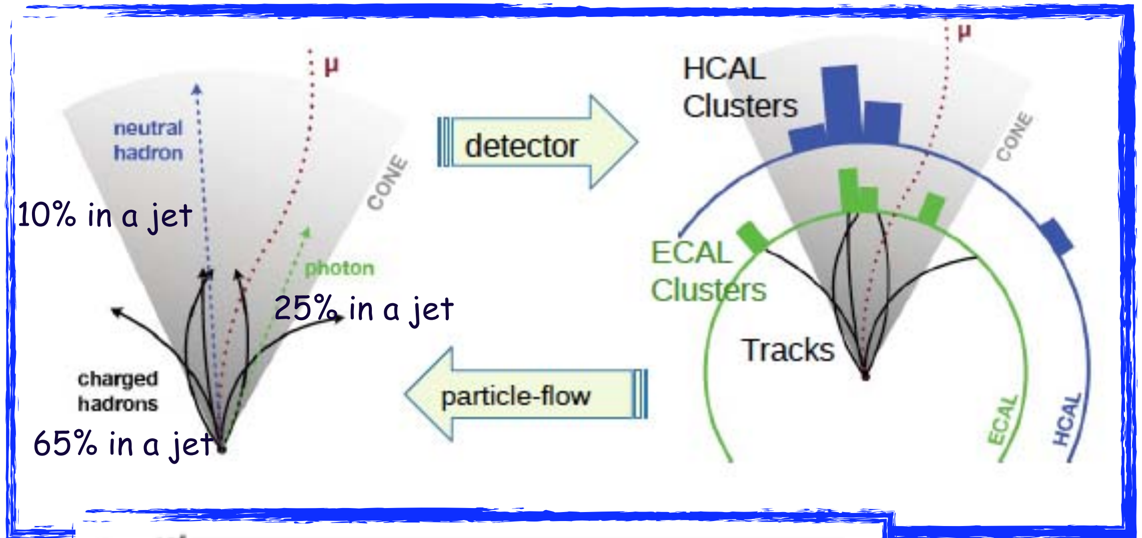




Run : 138919
Event : 32253996
Dijet Mass : 2.130 TeV

The highest mass dijet event in the first 120nb^{-1} of data

Particle Flow: Missing E_T and $\sum E_T$

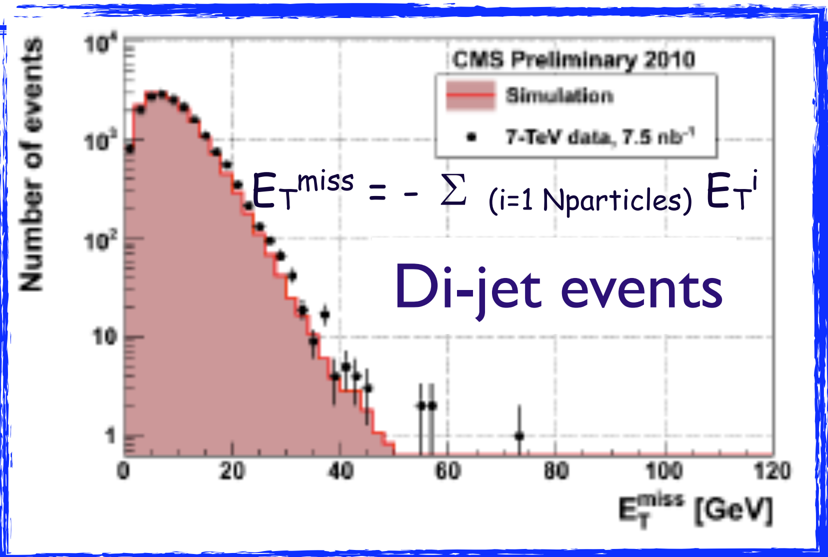


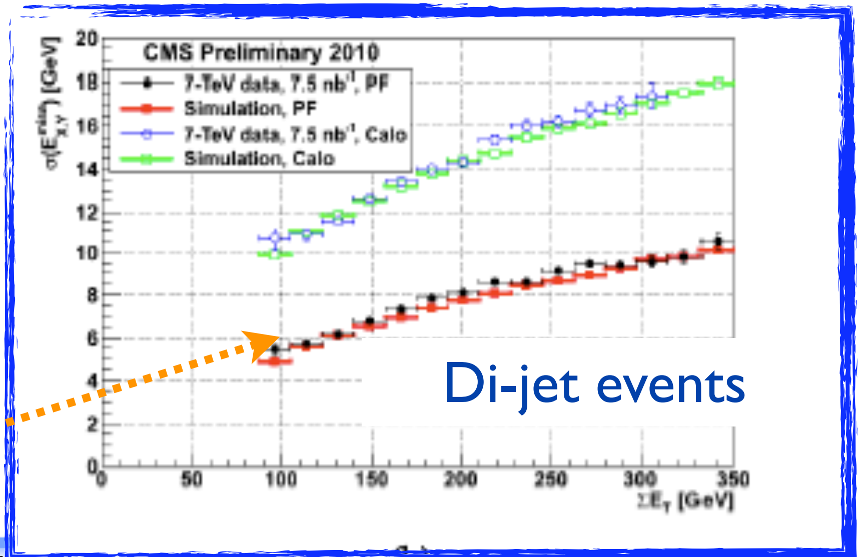
~90% of the jet energy is carried by charged hadrons and photons. Use tracking information whenever is possible.

remarkable agreement for E_T^{miss} known to be challenging to reproduce at hadron colliders

1) robustness of the algorithms

2) a precise detector simulation

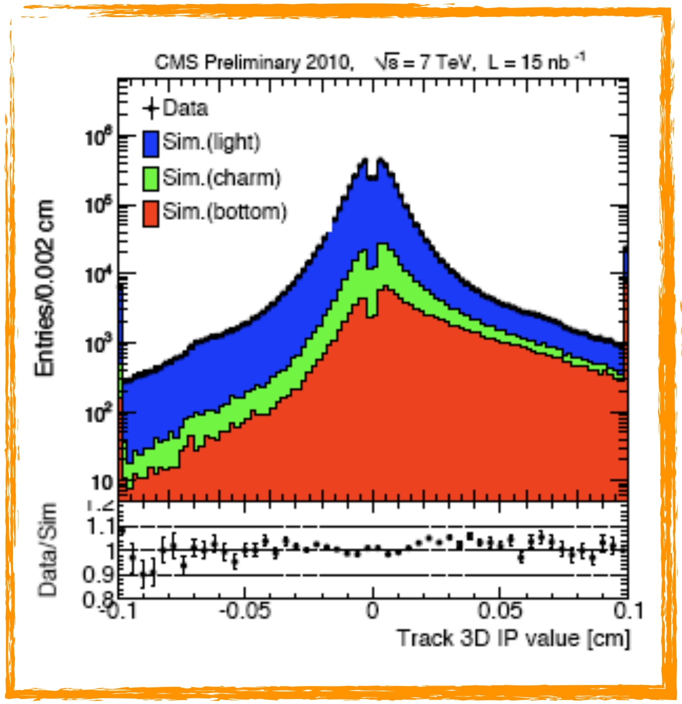




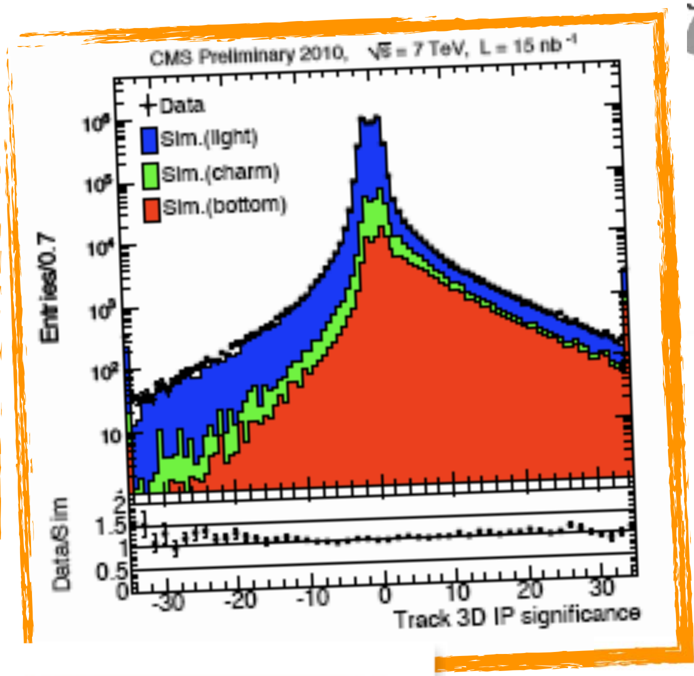
E_T miss resolution is improved a factor ~ 2 respect to calorimeter based one

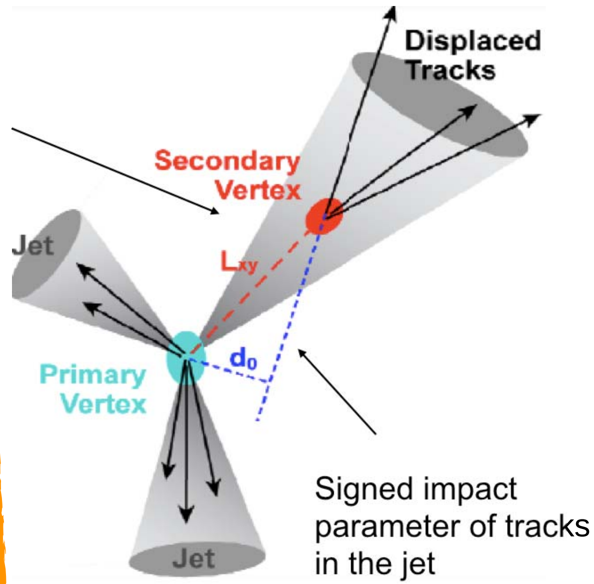
B Tagging Reconstruction

Excellent alignment and tracking performance



Signed decay length of secondary vertexes

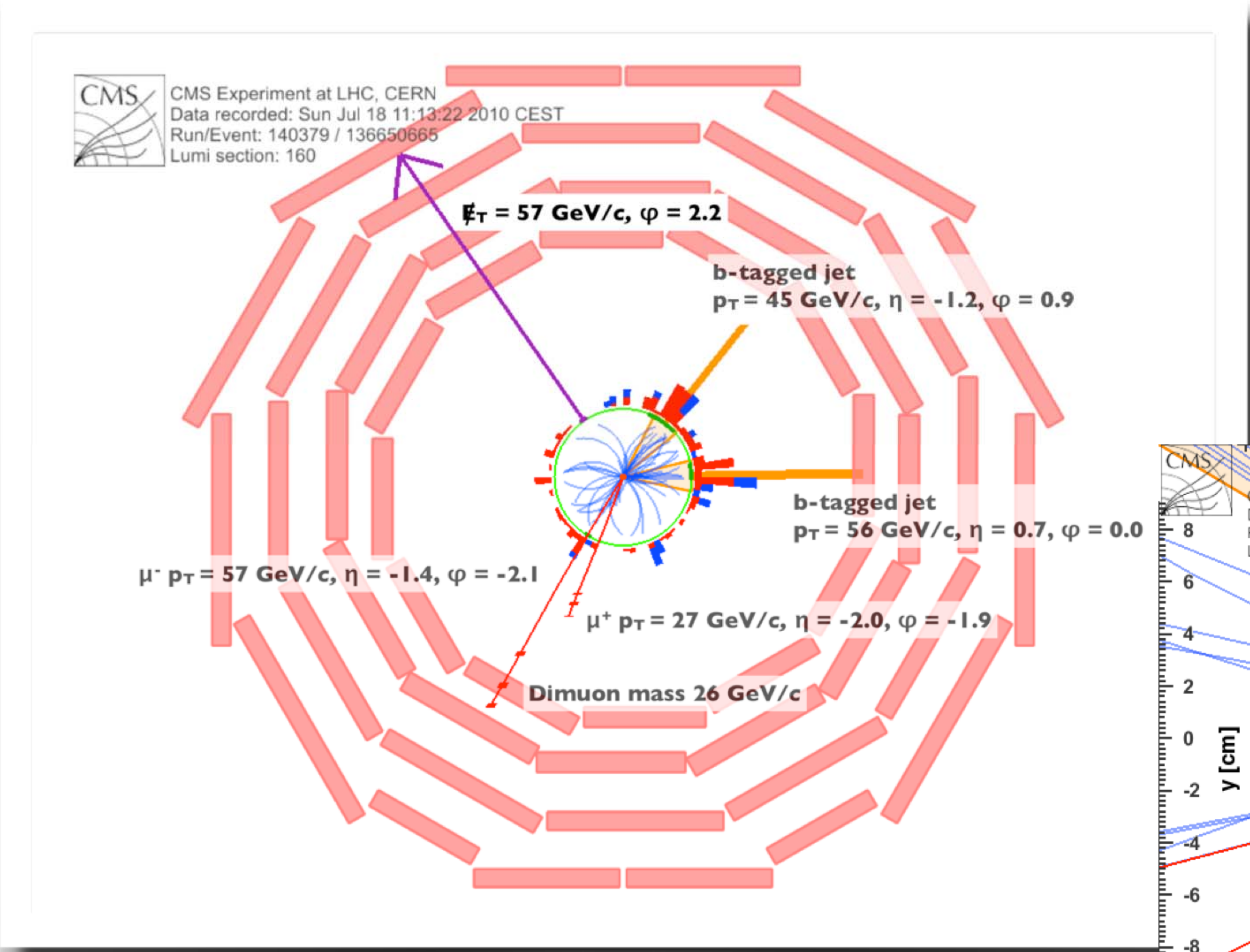




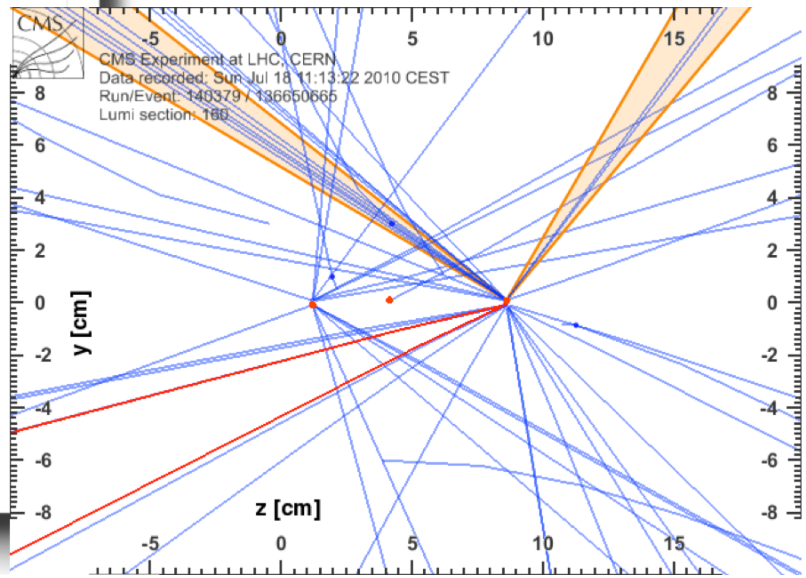
3D impact parameter and significance for all tracks with P_T>1GeV belonging to jets with p_T > 40 GeV and |η| < 1.5 (PFlow Jets anti-k_T R=0.5).

B tagging ready for physics since the beginning!!

Physics objects at work in Top events



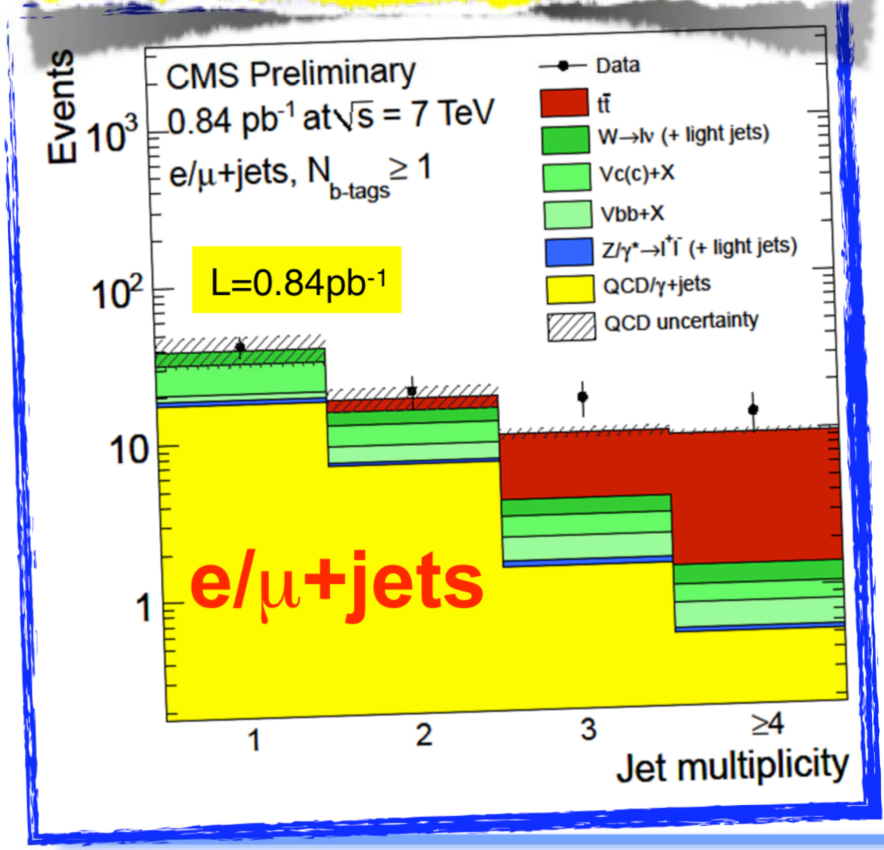
Multiple primary vertices multiple *pp* collisions (“pile-up”). Jets & muons originate from same vertex.



ttbar dilepton candidate on Jul 18th

Physics Objects at work in Top events

Good agreement demonstrates the reliability of the different physics objects reconstruction!!



Using the full statistics currently available (0.84 pb^{-1}) and requiring **at least 1 jet b tagged** (secondary vertex tagger with at least 2 tracks associated with the jet)

For N(jets)≥3 we count 30 signal candidates over a predicted background of 5.3

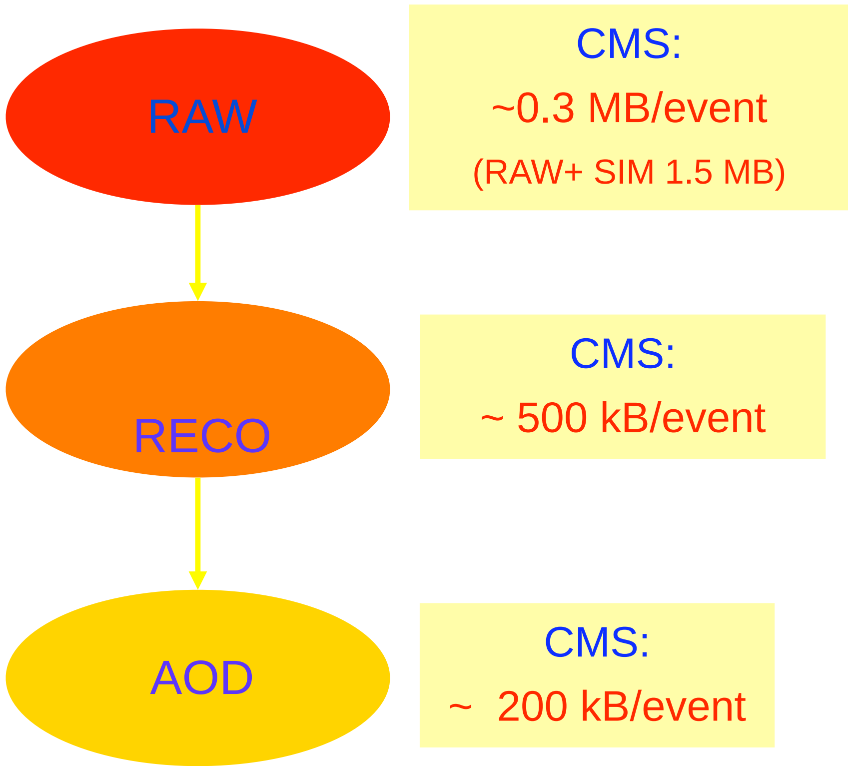
t-tbar events are observed in CMS at a rate consistent with NLO cross section, considering experimental (JES, b-tagging) and theoretical (scale, PDF, HF modelling, …) uncertainties.

CMS Analysis

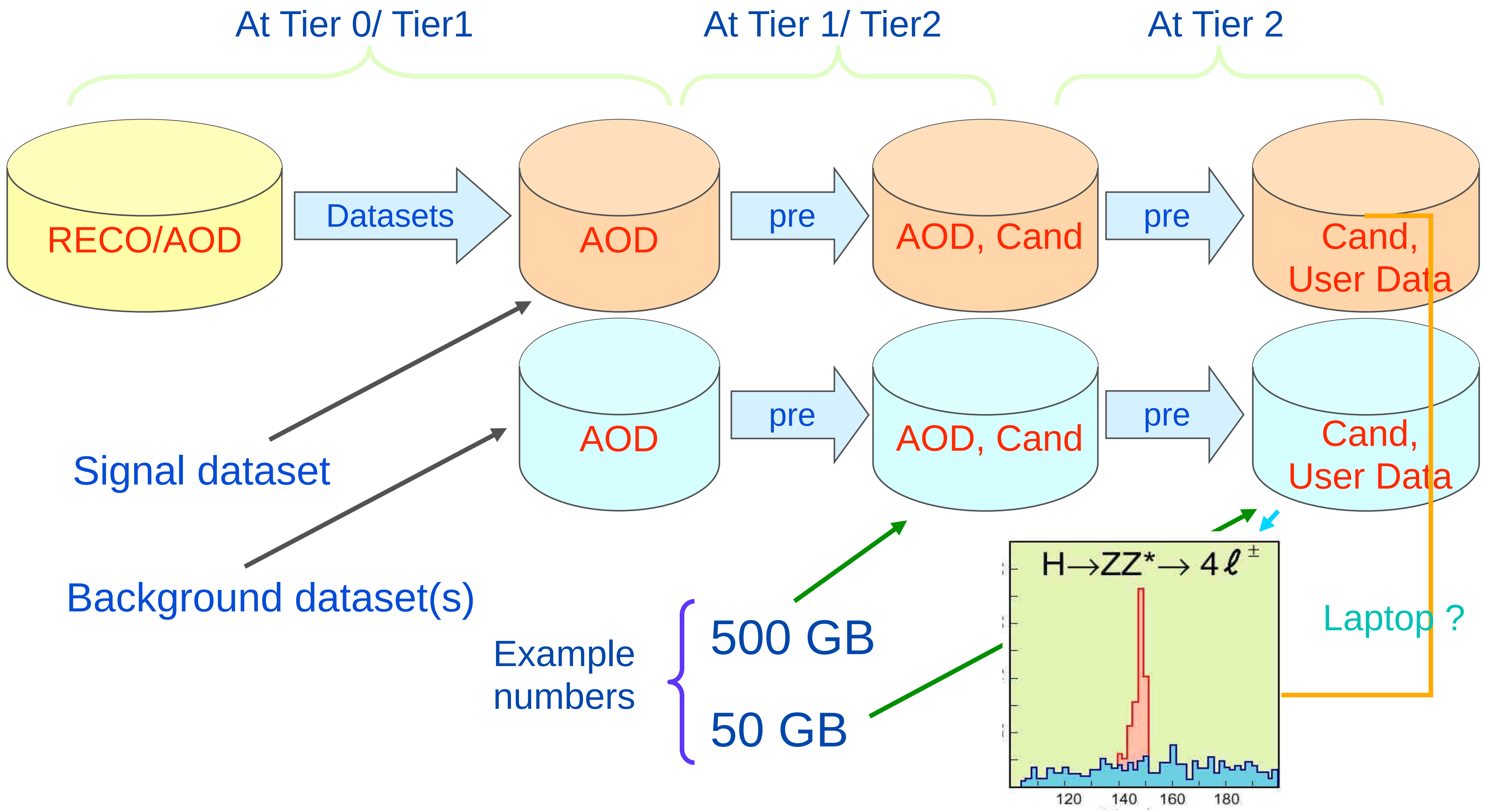
Analysis: Data Tiers

CMS plans to implement a hierarchy of Data Tiers

- Raw Data: as from the Detector
- RECO: contains the objects created by Reconstruction
- Full Event: contains the previous RAW+RECO
- AOD: again a subset of the previous, sufficient for the large majority of "standard" physics analyses
  - Contains tracks, vertices etc and in general enough info to (for example) apply a different b-tagging
  - Can contain very partial hit level information



Analysis "flow": an example



PAT

PAT is a toolkit as part of the CMSSW framework aimed at performing analysis

It provides:

- data format
- common modules

- It serves as well tested and supported common ground for group and user analyses.
- It facilitates reproducibility and comprehensibility of analyses,
- It is an interface between the sometimes complicated EDM and the simple mind of the common user.
- You can view it as a common language between CMS analysts:
- If another CMS analyst describes you a PAT analysis you can easily know what he/ she is talking about

PAT provides a very quick start for beginners

PAT - DATA Formats

Representation of reconstructed physics particles

pat::Candidate (pat::Jet, pat::Photon, pat::Muon,etc..)

There is a base class common to all kind of “Particles”: the reco::Candidate

It provides access:

- kinematics (pt, mass, eta, phi, etc.)
- underlying componenents (link to track, superclaster, etc.)
- navigation among the daughters (to access the daughter particles and their attributes)

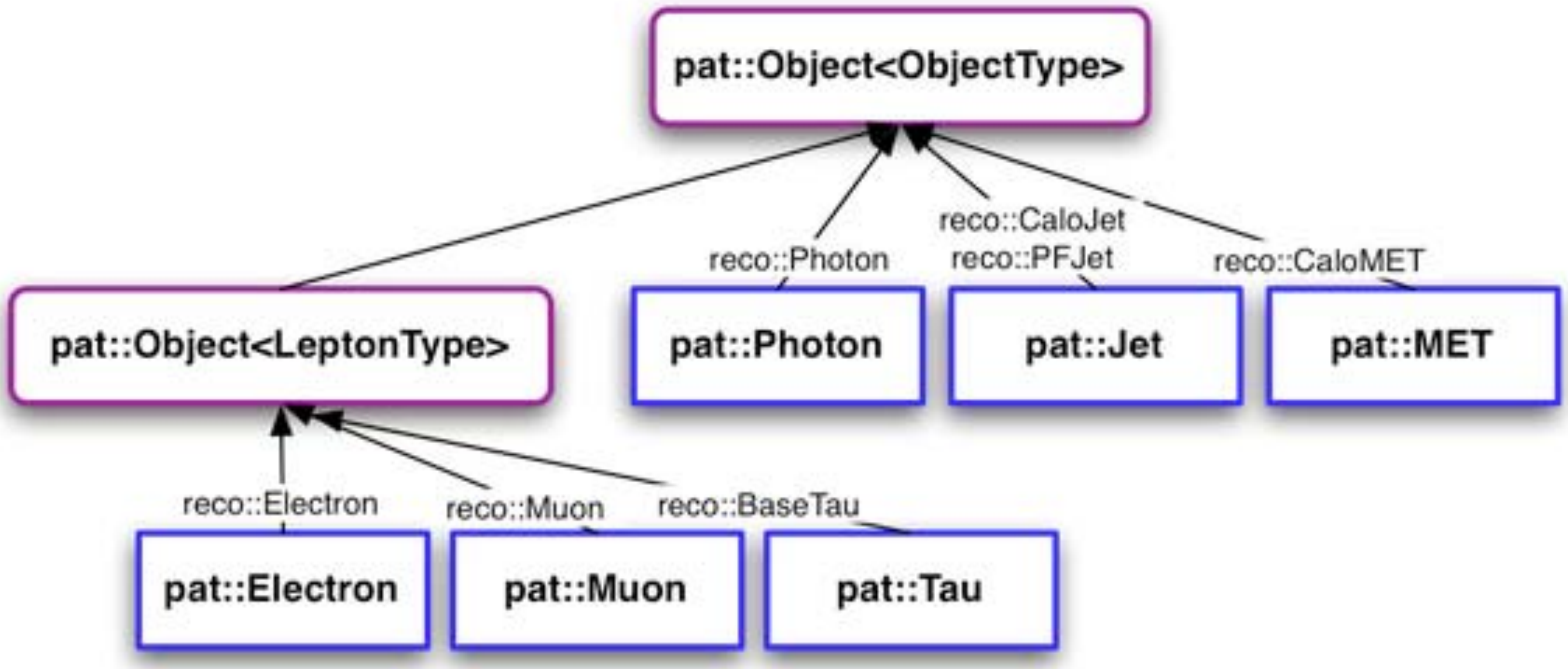
The pat::Object inherits from the reco:Candidate

pat::Candidate = reco:Candidate + more

Add extra informations to pat Candidates wrt reco Candidates such as:

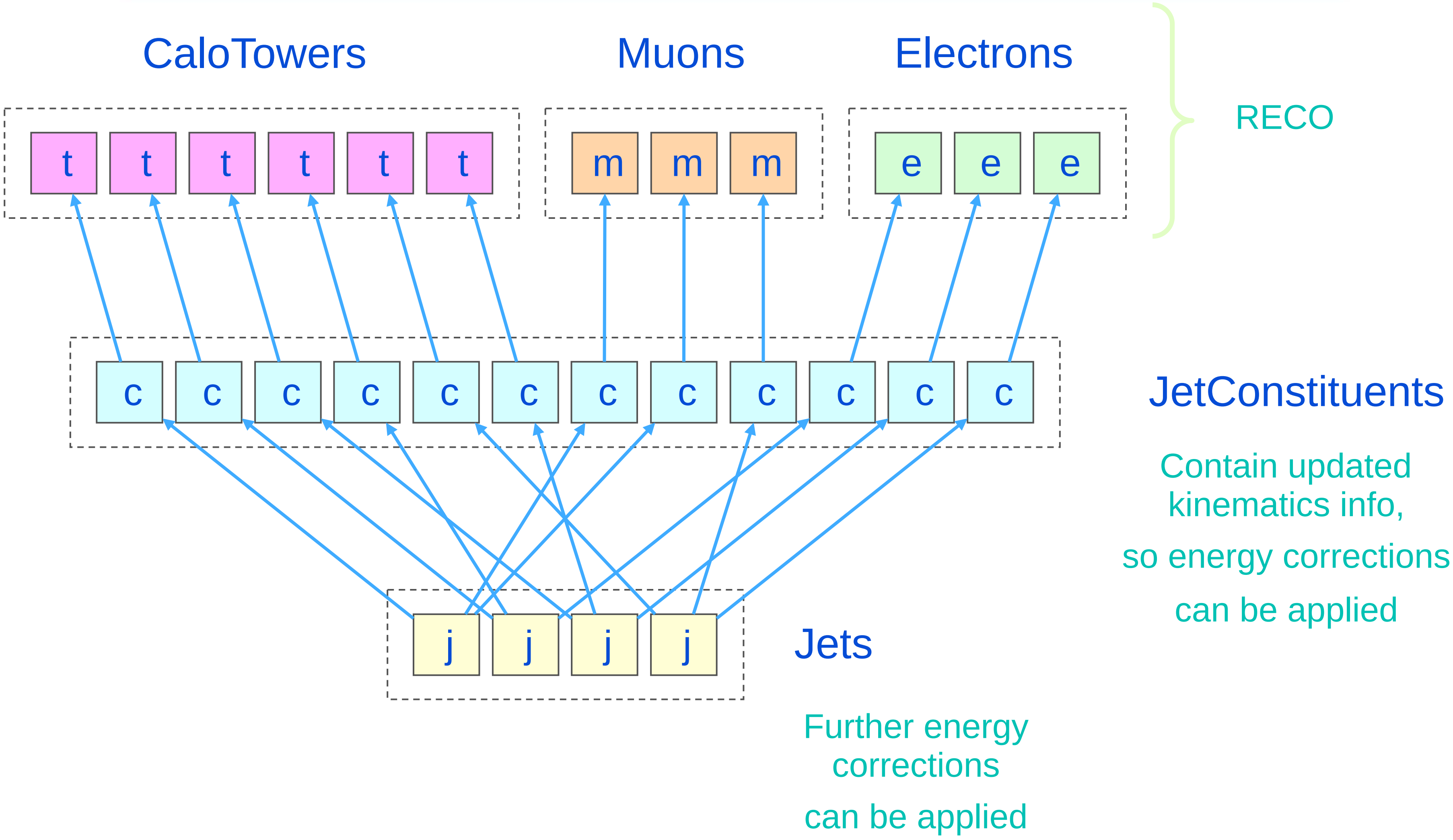
- Isolation
- MC matching
- Trigger matching

PAT - DATA Formats 2



This is the hierarchy of pat::Candidates

Analysis a la CMS: Particle Candidates for Jets



PAT - Common modules

PAT provides a series of modules common to different analysis task, such as:

- **Cleaning** - to remove disambiguities on the identifications of particles in the event
- **MC matching** - to associate PAT objects with generator objetcs
- **Trigger matching** - to associate PAT objects with trigger objects

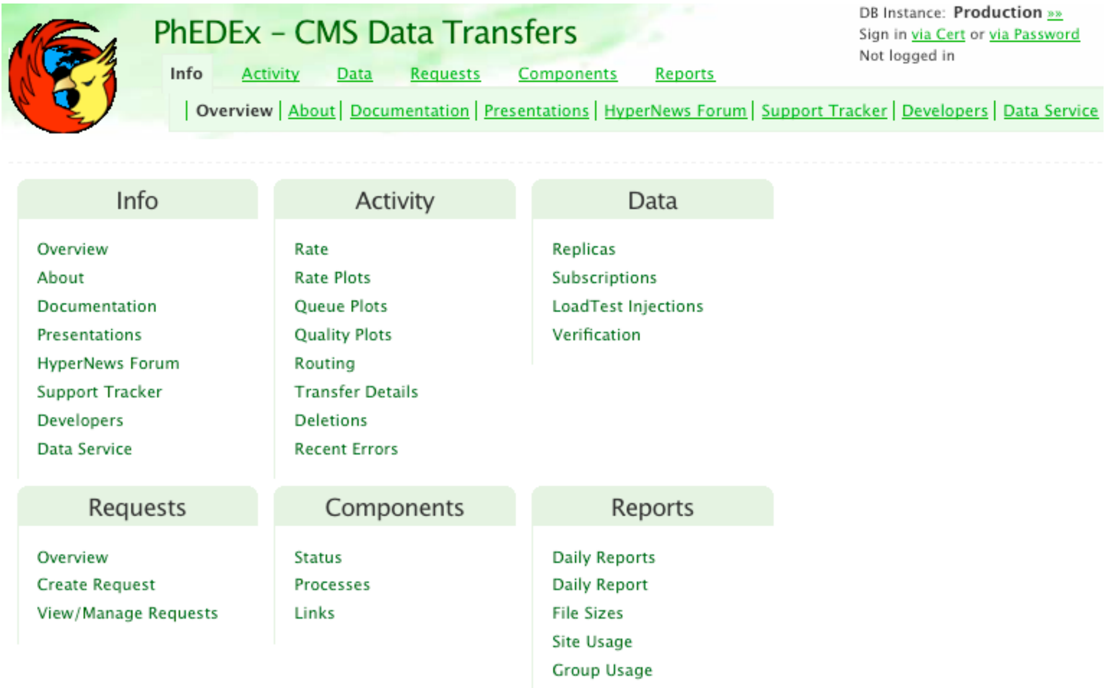
Moreover PAT provides a set of tools to performe easily the configuration of the Workflow → **PAT Tools**

These tools have been conceived to be common to different kind of analysis and, at the same time, to be customized according to the specific analysis requirement. They are well documented in SWGuidePATTools:

https://twiki.cern.ch/twiki/bin/view/CMS/SWGuidePATTools

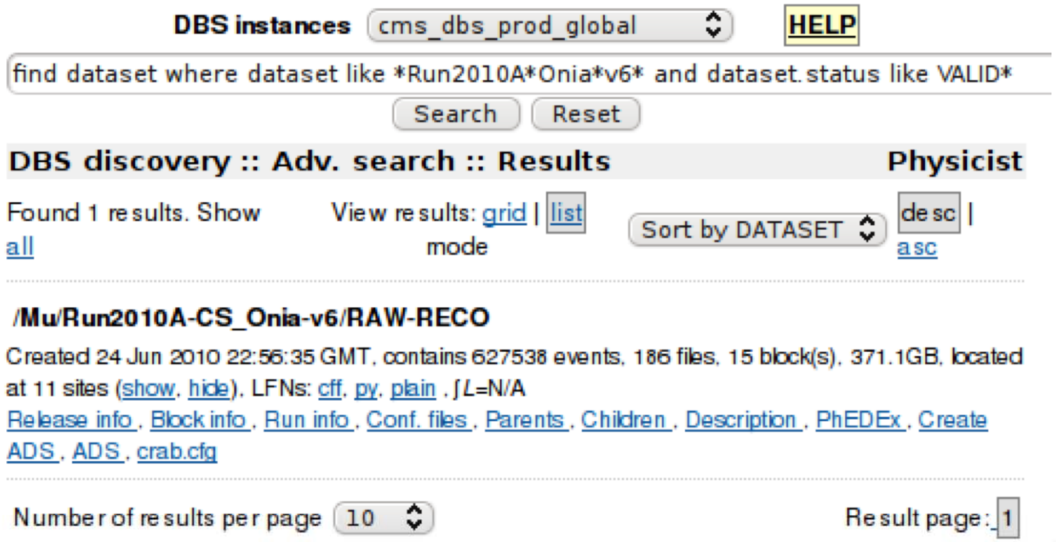
Transfer tool: PhEDEx

- PhEDEx is CMS' tool to request and manage data transfers
  - http://cmsweb.cern.ch/phedex
  - Every user can request the transfer of a data sample to a T2 site for analysis
  - Every T2 site (also the T1 sites and the T0) have data managers which approve or disapprove transfer requests according to global policies and available storage space



Dataset Bookkeeping system (DBS)

- DBS handles to bookkeeping of datasets
  - https://cmsweb.cern.ch/dbs_discovery
- A dataset name is composed of:
  - **/<primary dataset name>/<processed dataset name>/<data tier>**
    - **Primary dataset name**: specifies the physics content of the sample
    - **Processed dataset name:** specifies the processing conditions and data taking or MC production period, for Data: "<AcquisitionEra>-<FilterName>-<ProcessingVersion>"
    - **Data tier**: specifies the format of content of the files (RAW, RECO, AOD, …)
- Primary tool to look up and discovery datasets and their location on the T2 level for your analysis



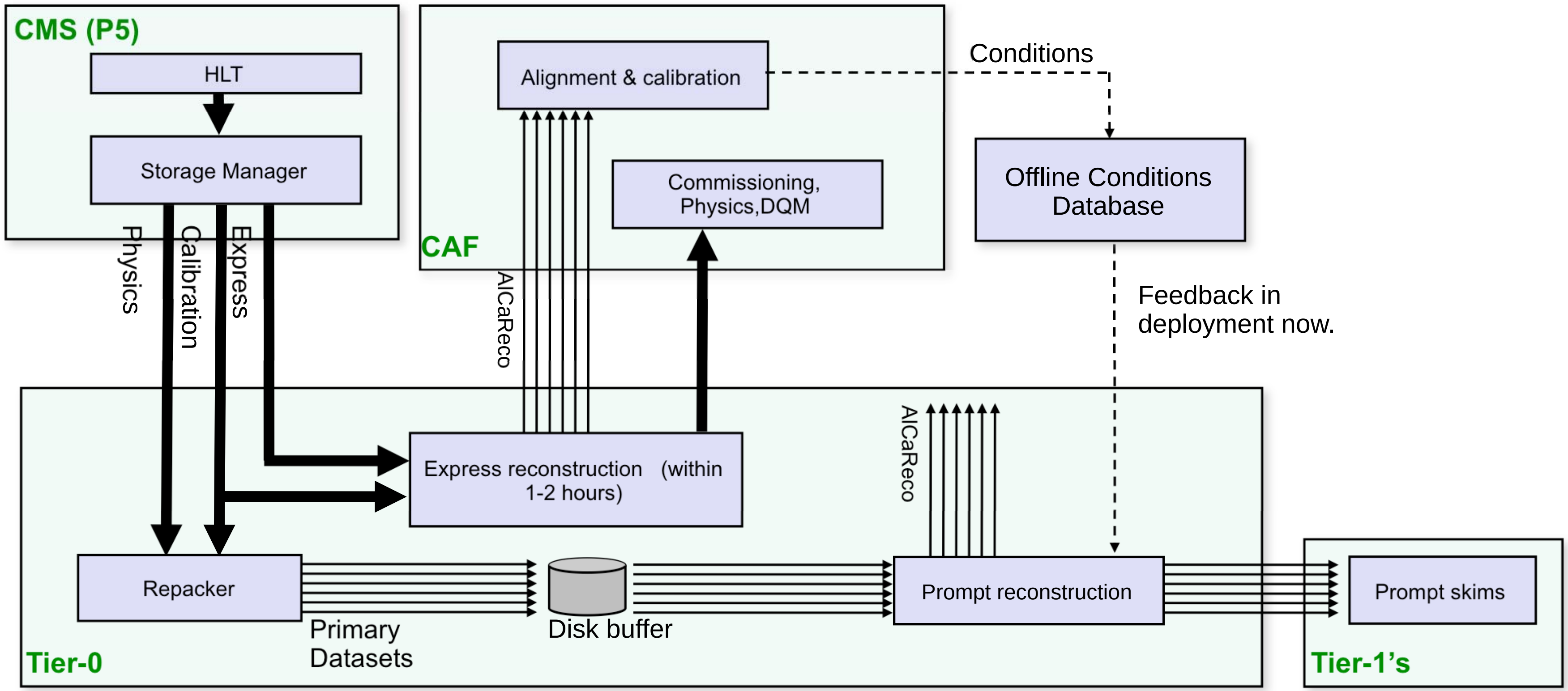
GRID submission tool: CRAB

- CMS Remote Analysis Builder
  - https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrab
- Enables every user to send her/his analysis code to the T2 sites to process stored data and MC samples
- Represents a wrapper to the GRID tools used to execute jobs on the GRID

CMS Remote Analysis Builder - CRAB

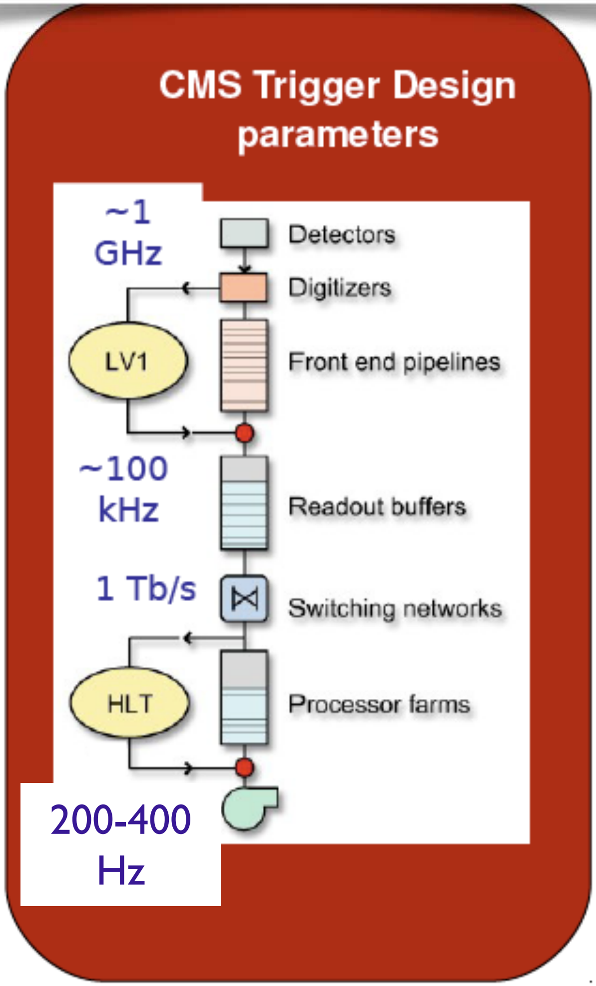
Contents:

Data Flow &
Offline and Computing Operations

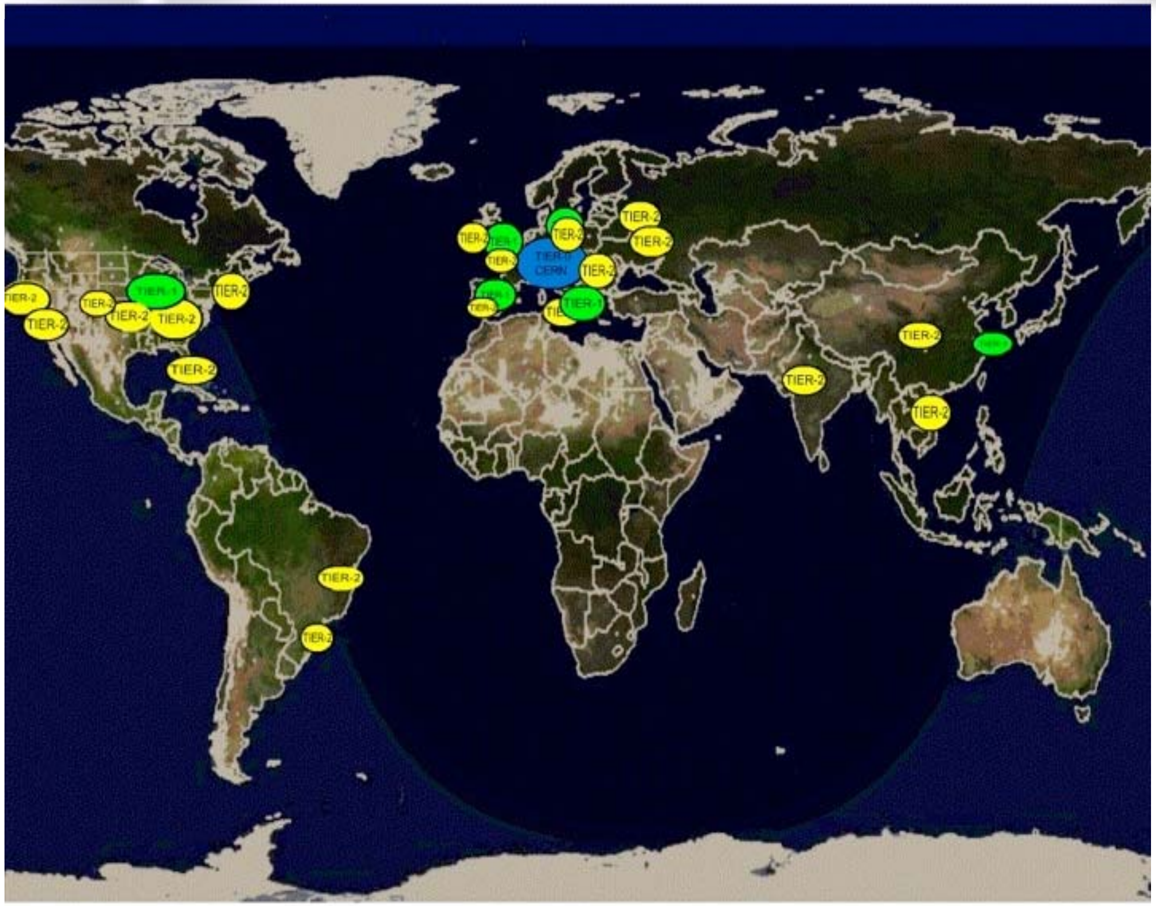


Data Flow: from Detector to Tier4 (your PC..)

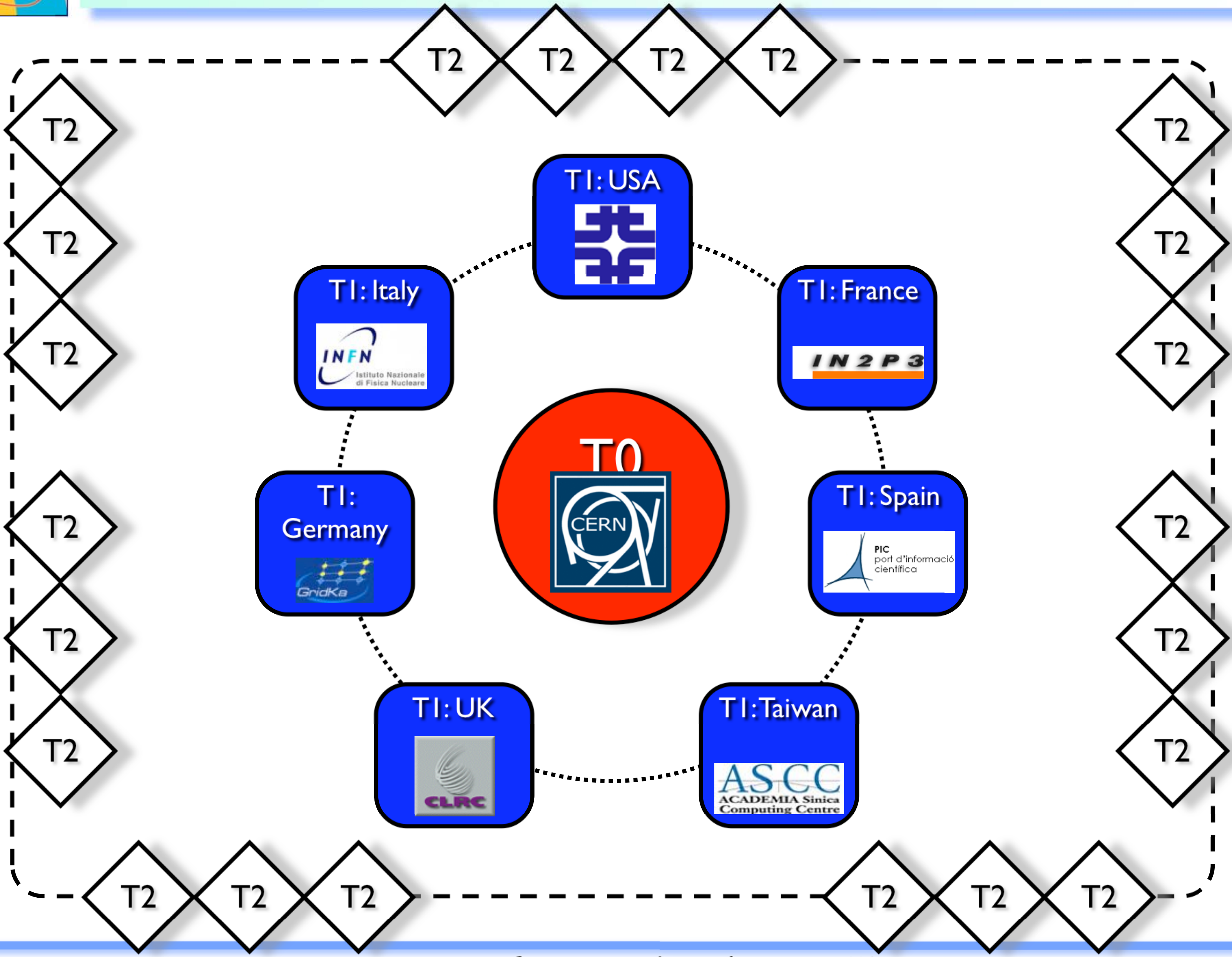
DETECTOR AND L1 & HLT



TIERS WORLDWIDE DISTRIBUTED



CMS Computing model: tiers



How data are distributed & role for the different Tiers

- Computing Model (key ingredients)
  - 2 safe copies of RAW data on tape at CERN and Tier-1 sites
  - 2-3 large re-reconstruction passes per year in first years at Tier-1 sites
  - Monte Carlo production matches collision data
  - Production and user jobs go where the data is
  - Full network connectivity. All sites are connected with each other
- A lot a work went into preparation and testing
- Provides flexibility to tackle unforeseen scenarios (very high turn-around before conferences)

- Tier0 activities
  - Prompt data processing, Prompt calibration and alignmente, Storage of Raw data backup
- Tier1 activities
  - Custodial storage of Raw data, Prompt skimming, Reprocessing of data and MC, MC production.
- Tier 2 activities
  - MC production, User analysis
- Tier3 activities
  - User analysis

Offline data-taking Operations

Offline workflows deliver validated & calibrated reconstructed data for physics analysis

offline reconstruction

- prompt feedback on detector status and data quality
- sample for physics analysis

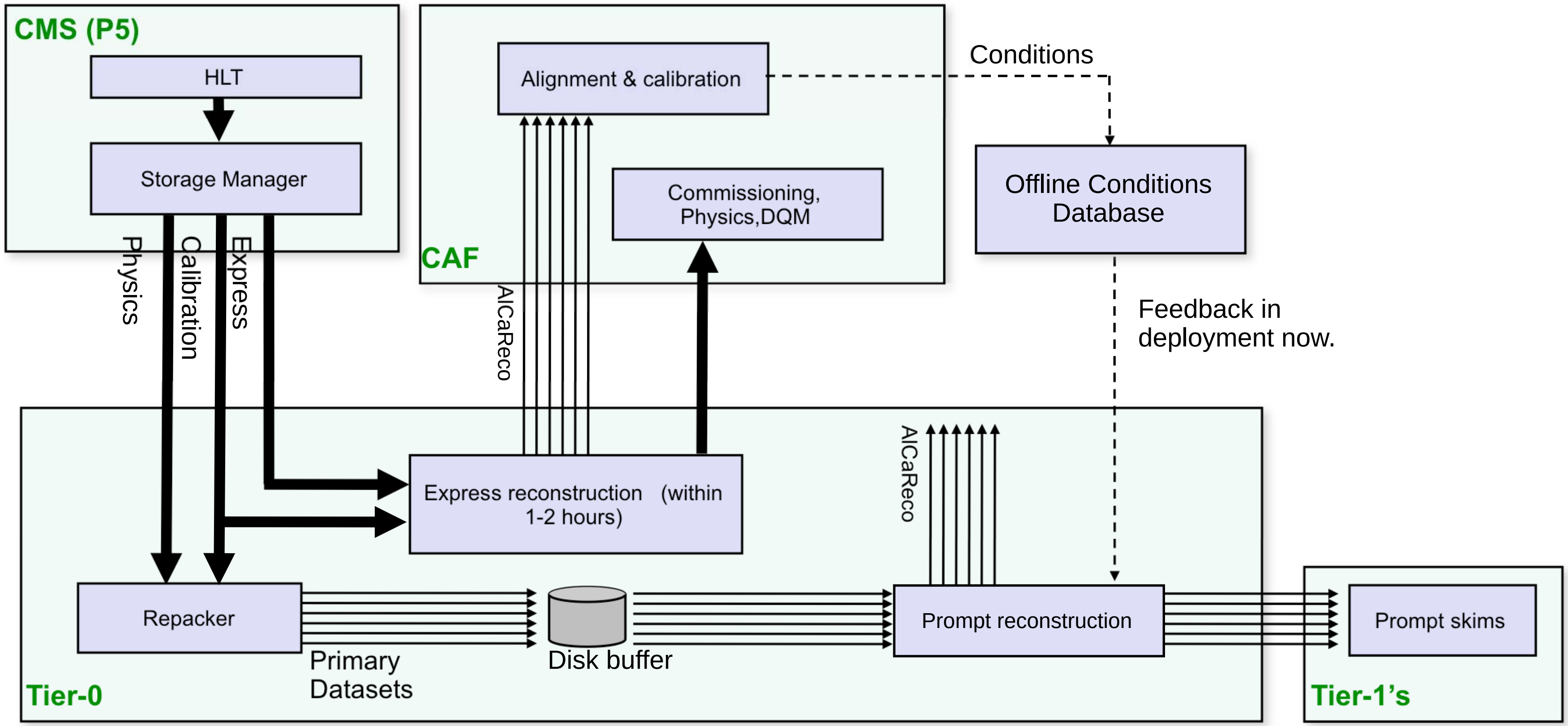
up-to-date alignment & calibration (AlCa)

- calibration workflows with short latency
- provide samples for calibration purposes: AlCa streams
- consistent set of conditions for data and MC

data validation and certification for analysis

- data quality monitoring (DQM)

Processing at Tier0/CAF (CERN)

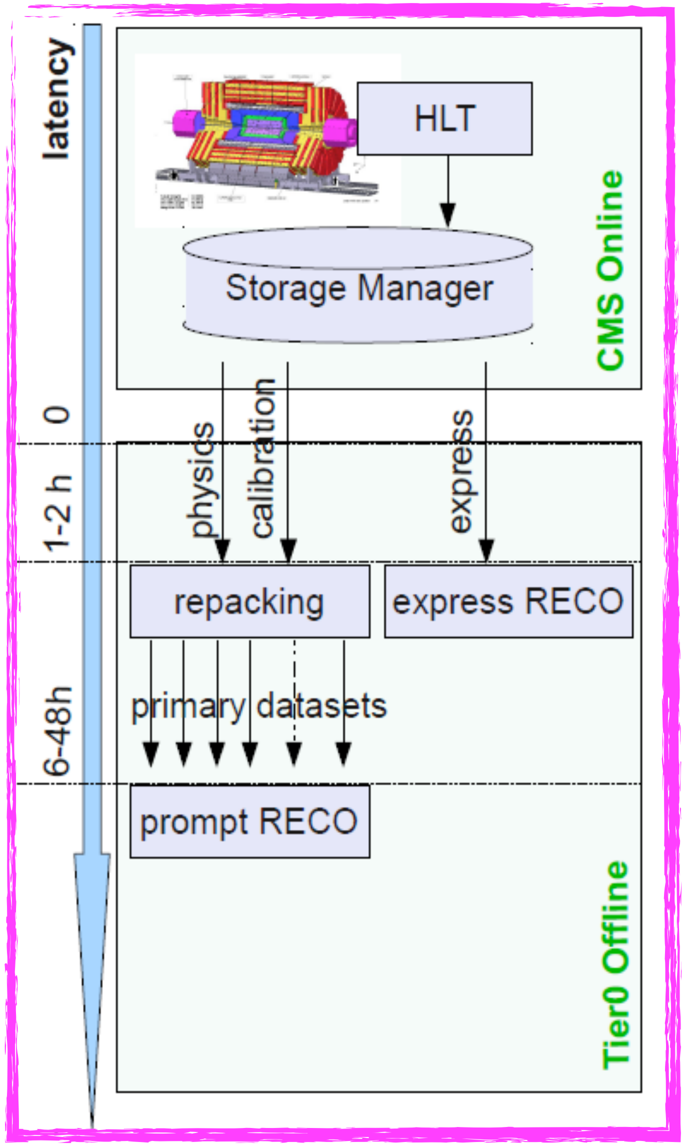


Data Streams and Tier0

Data streams & Tier0 workflows

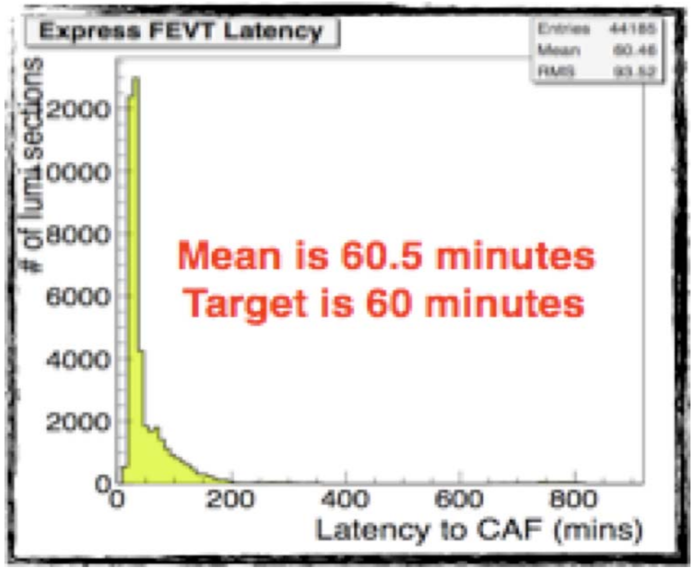
Depending on the latency

- Express: prompt feedback & calibrations
  - short latency: 1-2 hours
  - ~40Hz bandwidth shared by:
    - calibration ($\frac{1}{2}$)
    - detector monitoring ($\frac{1}{4}$)
    - physics monitoring ($\frac{1}{4}$)
- Alignment & Calibration (AlCa) streams
- Prompt reconstruction: sample for physics analysis
  - split in Primary Datasets using High Level
  - will be delayed of 48h (latest calibrations)
  - writing ~300Hz



Prompt reconstruction and express stream

- Rolling workflows are fully automated
- Express processing provides quick feedback for commissioning, data quality monitoring and physics
- Alignment and calibration loop to improve quality of prompt reconstruction
- Operational experience at Tier-0 is excellent. **Success rate of 99.9%**
- Categorize data according to trigger selection in primary dataset



Calibration Workflows

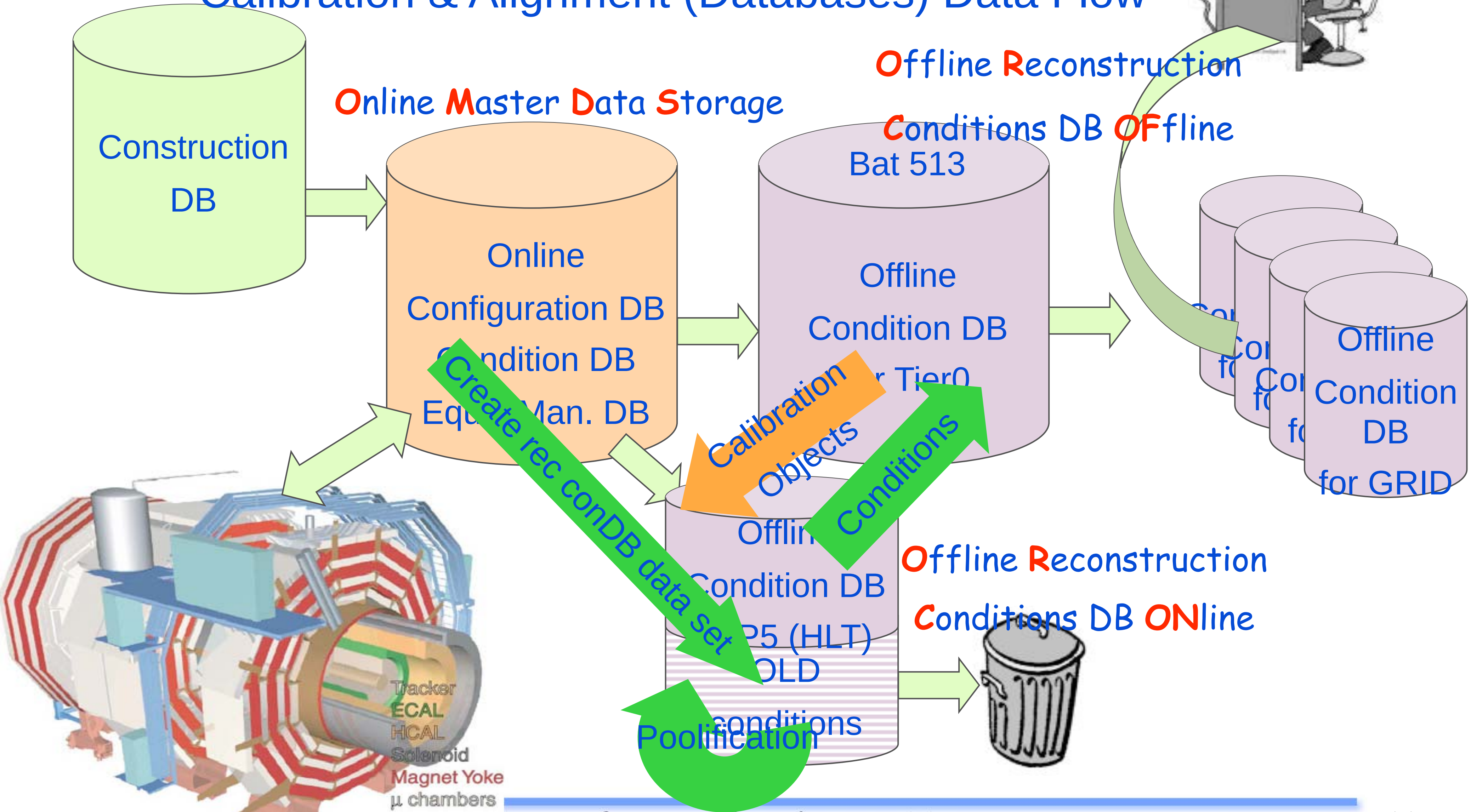
Provide most up-to-date conditions @ all stages of the data processing

Different workflows depending on the time scale of updates:

- quasi-online calibrations for HLT and express:
  - e.g. beam-spot → quick determination online
- prompt calibrations: monitor/update conditions expected to vary run-by-run (or even more frequently):
  - updated conditions must be ready before prompt-reconstruction
- offline re-reco workflows:
  - more stable conditions
  - workflows which need higher statistics:
    run on AlCa streams produced during prompt-reco or offline re-reco

Calibration/Alignment: Databases Data Flow

Calibration & Alignment (Databases) Data Flow



Data Certification

INFN Istituto Nazionale di Fisica Nucleare Sezione di Bari

- The complexity of the offline workflows requires robust validation

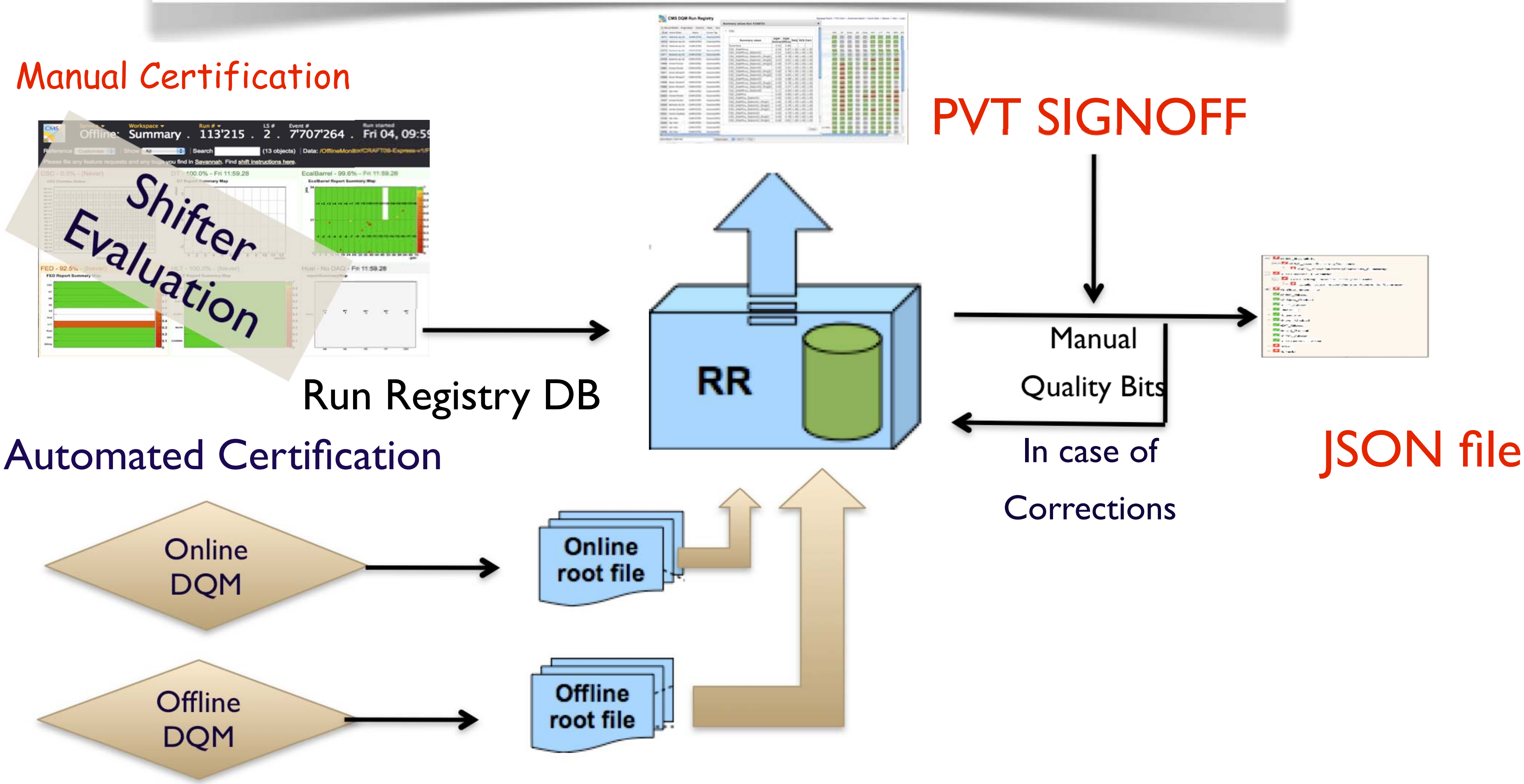
- Several stages of Data Quality Monitoring (DQM):
  - online DQM → monitor detector performance during data-taking: dedicate event stream
  - offline DQM → monitor performance of physics objects
    - runs on full statistics available for analysis:
      - express reco → fast feedback
      - prompt-reco → continuous monitor
      - offline re-reco → validation of software and condition updates

- Physics Validation Team (PVT) → coordinates the validation activity. Feedback from: groups responsible for physics objects; detector performance groups; analysis group

Workflow for Data Certification

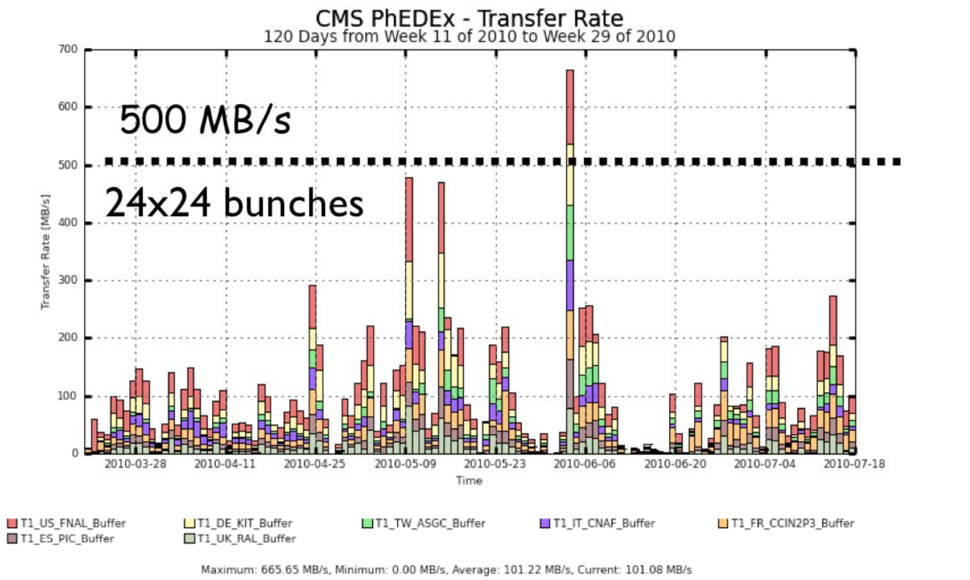
Production of GOOD Runs and Luminosity Section Lists

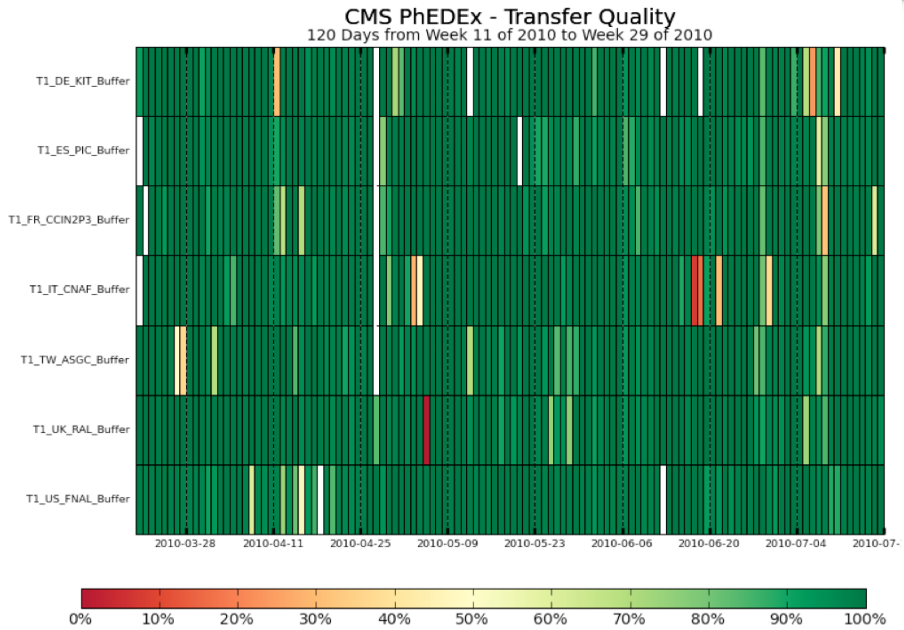
for use in Physics Analyses

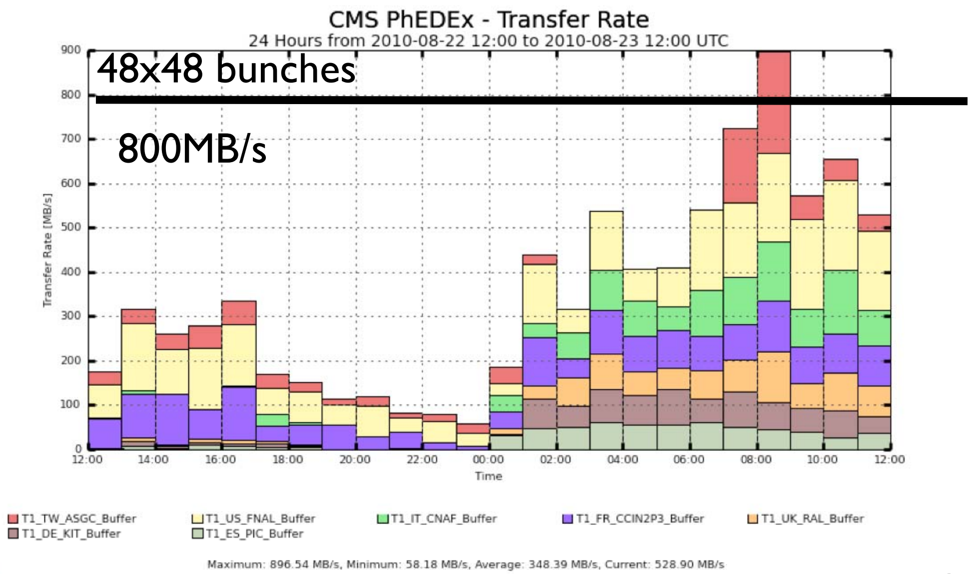


Data Transfer from CERN to Tier-1's

- Resources provisioned for steady data stream from Tier-0 to Tier-1's
- Nice peaks from the fills (good balance on Tier-1s
- Very good transfer quality

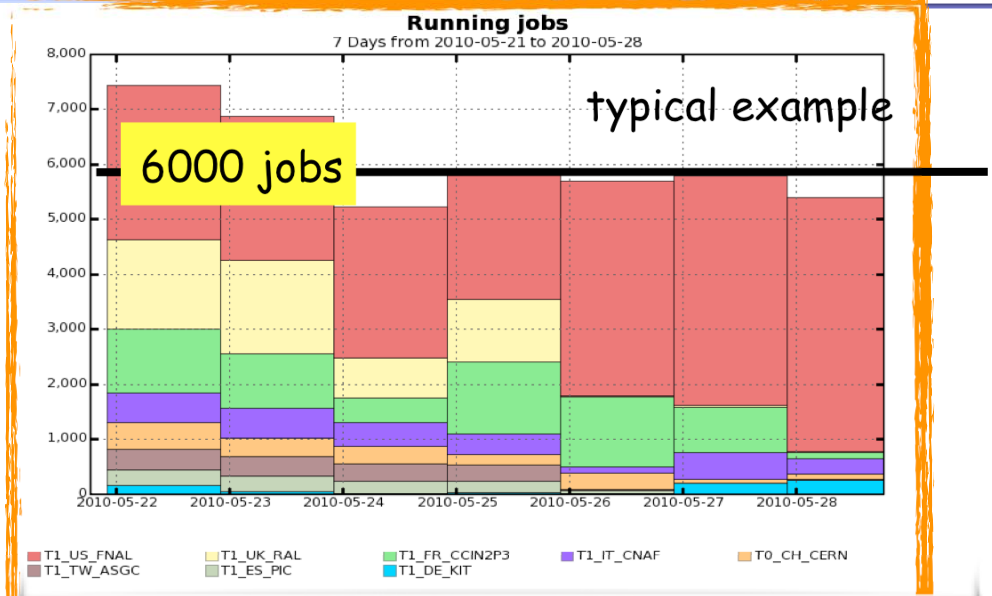


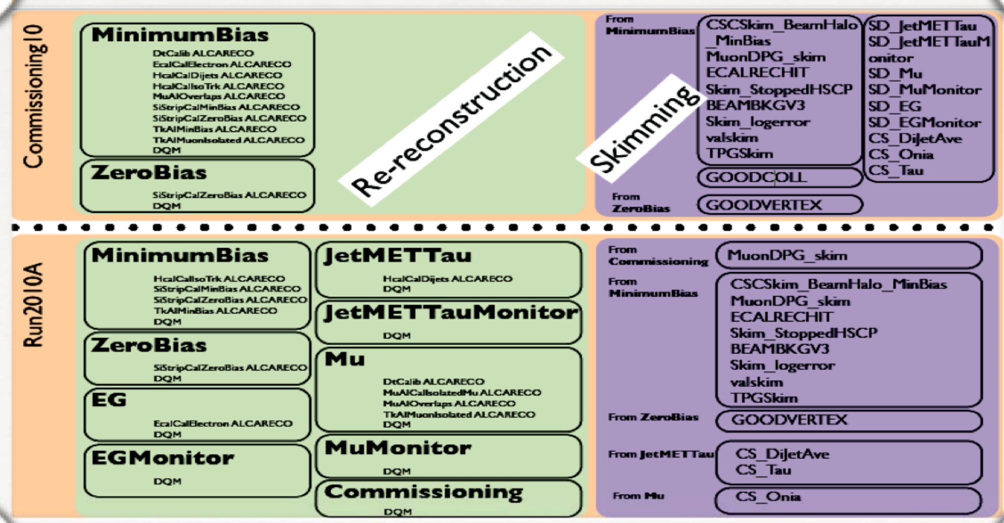




Central Processing @ Tier-1

- All Tier-1 sites used in production
- Upon arrival at Tier-1's, data is being processed and stored on tape
- Prompt skimming
  - Produce small datasets based on trigger selection or reconstructed objects
  - Fully automatized system
- Reprocessing of data and MC
  - Improved software, calibration and alignment
  - ~ 10 data reprocessing passes for 7 TeV (up to now).
  - 3 MC reprocessing passes for 7 TeV



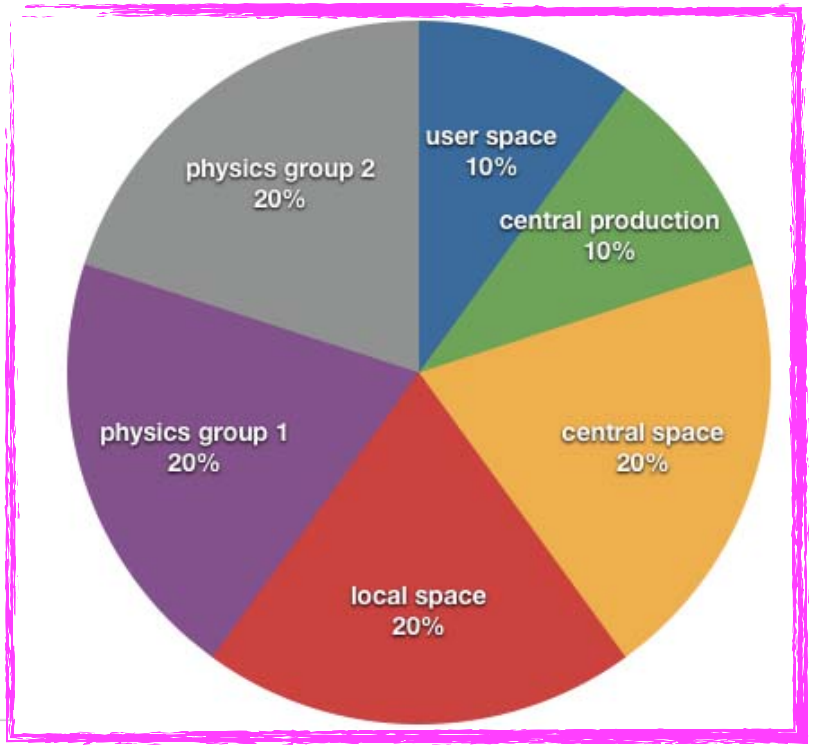


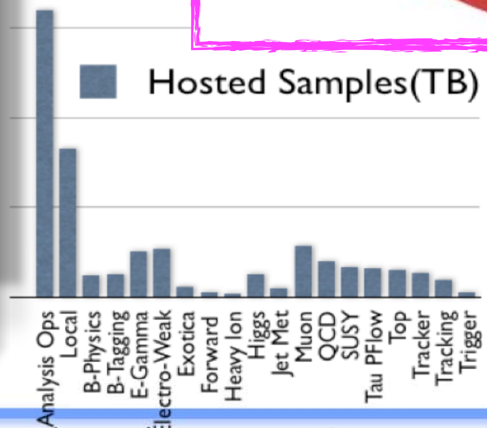
Primary Datasets

Data Distribution for Analysis

- Data distribution to Tier-1 organized centrally to balance resource utilization.
  - Jobs go where the data is
- Data storage serves as temporary buffer
  - Refresh with hot datasets
- Data distribution on Tier-2 organized
  - Centrally (Analysis Operations)
  - By physics groups
  - By local users

Tier-2 storage breakdown (typical example)



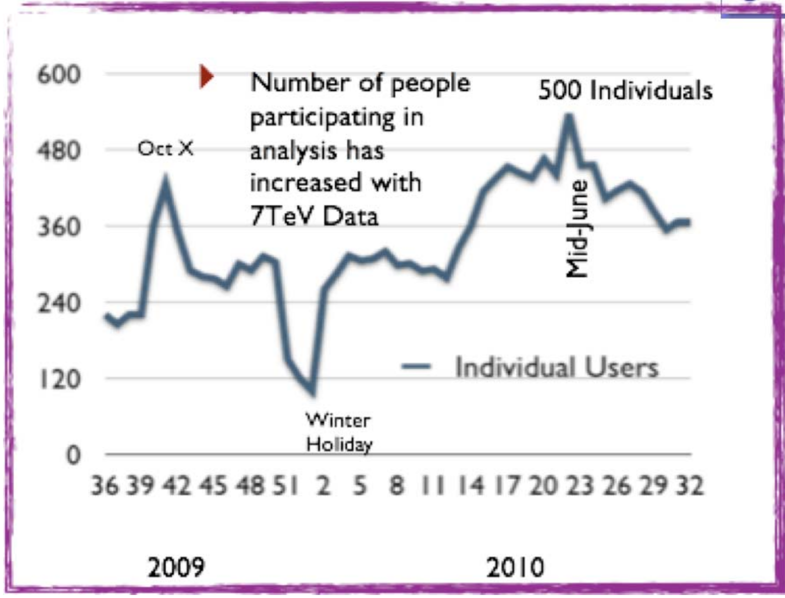


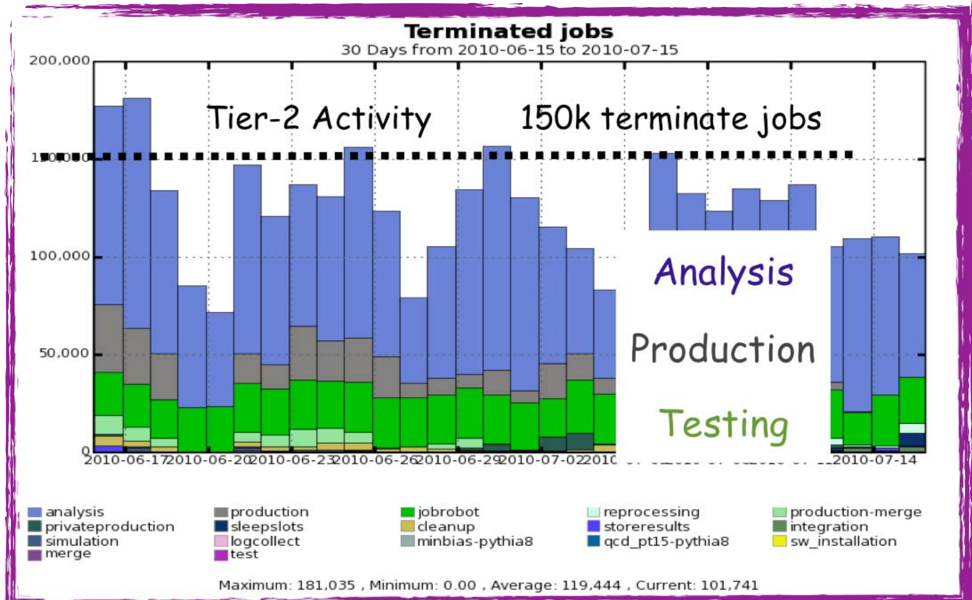
Analysis Activities @ Tier-2/3's

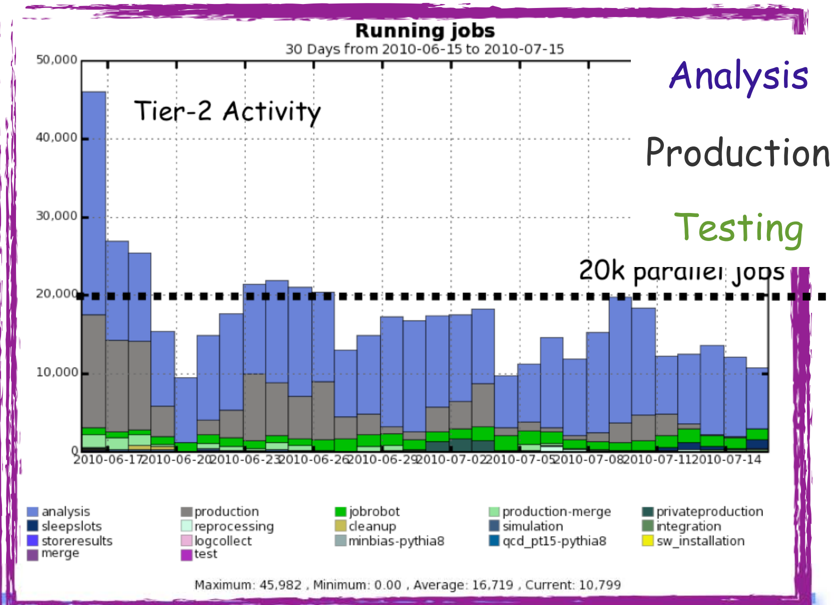
500 individual CMS users active using grid resources

Maximum reached in preparation for ICHEP

Tier-2 resource usage currently dominated by analysis activities

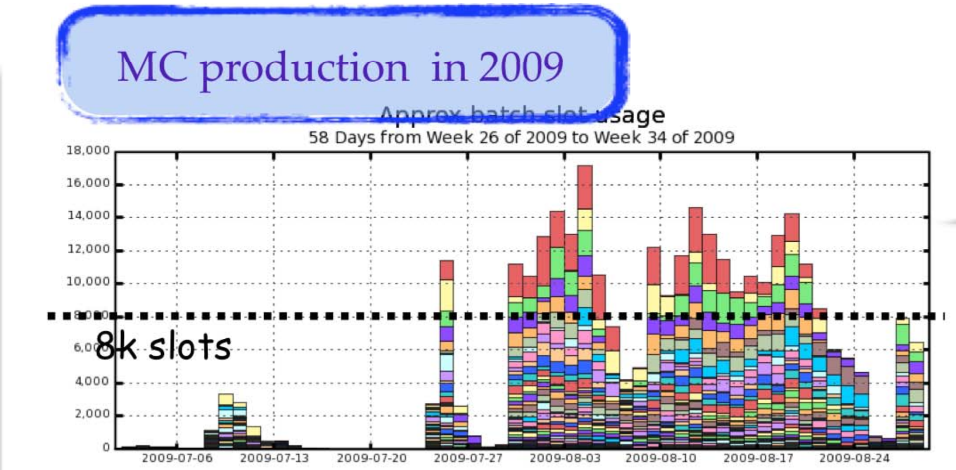


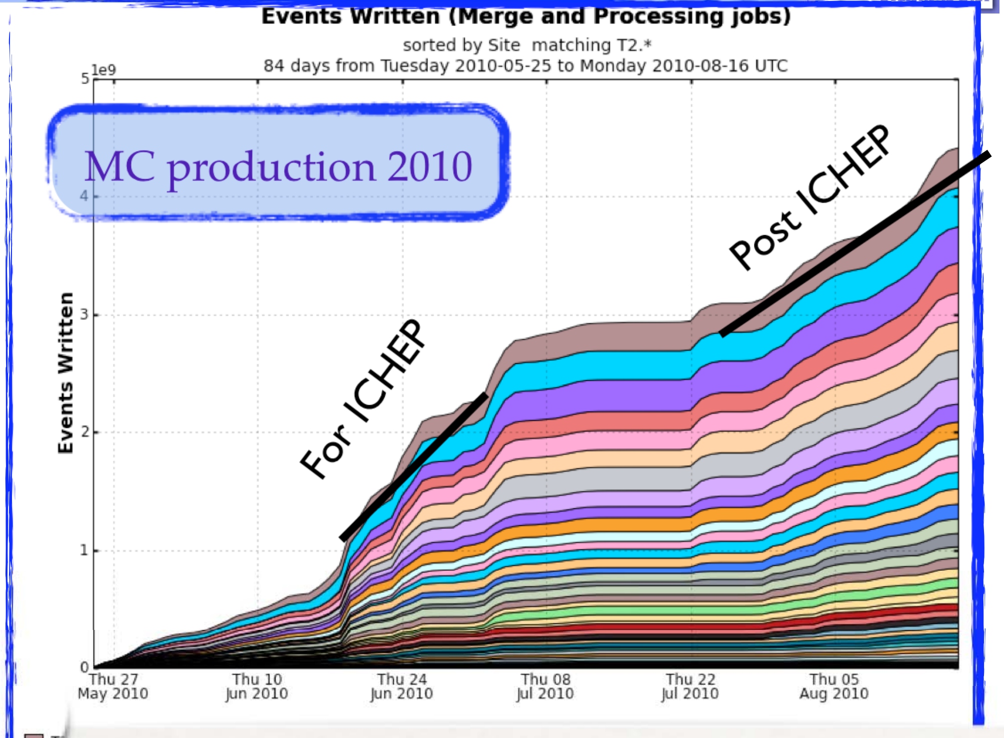




Monte Carlo Production

Successfully exercised for years
64 Tier-1/2/3 sites participating
MC production preparation for 7TeV data started in Summer 2009
Multiple production validation cycle





Before ICHEP: Mostly "Data-like" MC production in 2010 (MinimumBias & low-Pt QCD)

Next generation of CMSSW simulation starting. Large Scale sample with Pile-up to start

End Lecture 2

Back-up slides

Additional Material

More slides

Writing your framework module

Preparing the environment

```
creating your local area
$ cmsrel CMSSW_3_6_2
$ cd CMSSW_3_6_2/src

setting runtime variables
$ cmsenv
```

Writing an EDAnalyzer

```
$ cd CMSSW_3_6_2/src
$ mkdir Tutorial
$ cd Tutorial

Create the skeleton of an EDAnalyzer

$ mkedanlzr DemoAnalyzer
$ cd DemoAnalyzer
& scram b
```

The skeleton of the EDAnalyzer has been created and put in DemoAnalyzer/src/DemoAnalyzer.cc

Find more details on WorkBookWriteFrameworkModule:

https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookWriteFrameworkModule

The Source Code

In the header:

```
private:
     virtual void beginJob() ;
     virtual void analyze(const edm::Event&, const edm::EventSetup&);
     virtual void endJob() ;
```

The methods *beginJob()*, *analyze()* and *endJob()* are called for each event!

```
void
DemoAnalyzer::analyze(const edm::Event& iEvent, const edm::EventSetup& iSetup) {

using namespace edm;
using reco::TrackCollection;

Handle<reco::TrackCollection> tracks;

iEvent.getByLabel("generalTracks", tracks);

for(TrackCollection::const_iterator itTrack = tracks->begin();
itTrack != tracks->end(); ++itTrack)
    {
    int charge = itTrack->charge();
    }
}
```

Get the TrackCollection and loop on all the tracks in the event

```
DEFINE_FWK_MODULE(DemoAnalyzer);
```

Define the Module!

TFileService

We can create ROOT histograms and store them using the TFileService, a framework service.

```
// access the TFileService
edm::Service<TFileService> fs;
// create your histogram
TH1F * h_pt = fs->make<TH1F>( "pt" , "p_{t}", 100, 0., 100. );
// fill it
h_pt->Fill( pt );
// create subdirectories if you like
TFileDirectory subDir = fs->mkdir( "mySubDirectory" );
```

```
# make the TFileService known to the config
process.TFileService = cms.Service("TFileService",
                                   fileName = cms.string("histo.root")
                                   )
```

https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideTFileService

HistoAnalyzer

We can get histograms without writing C++ code ☺

```
plotJets = cms.EDAnalyzer("CandViewHistoAnalyzer",
    src = cms.InputTag(”ak5CaloJets"),
    histograms = cms.VPSet(
        cms.PSet( itemsToPlot = cms.untracked.int32(5),
        # plots the first 5 jets
        min = cms.untracked.double(0.0),
        max = cms.untracked.double(200),
        nbins = cms.untracked.int32(50),
        name = cms.untracked.string("jet %d E_{T} [GeV/c]"),
        description = cms.untracked.string("jet_%d_et"),
        plotquantity = cms.untracked.string("et")
        )
    )
)
```

https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideHistogramUtilities

CMSSW Configuration Files

Configuration Files

Controls the final job to be run

Written in Python

Contains a **cms.Process** object named **process**

Usually placed in a package's python/ or test/

Can be checked for completeness doing

python myExample_cfg.py (Python interpreter)

Can be run using cmsRun

cmsRun myExample_cfg.py

you may want to inspect your config file in python interactive mode:

```
$ python –i config_file_cfg.py
# to inspect the process path called "path"
>>>process.path
```

Configuration Files

Definition of terms: Python module

A python file that is meant to be included by other files

Placed in **`Subsystem/Package/python/`** or a subdirectory of it

Naming conventions

- Definition of a single object: **`_cfi.py`**
- A configuration fragment: **`_cff.py`**
- A full process definition: **`_cfg.py`**

To make your module visible to other python modules:

- Be sure your SCRAM environment is set up
- Go to your package and do scram b or scram b python
- Needed only once

Correctness of python config files is checked on a basic level every time scram is used.

How to import objects

- To fetch all modules from some other module into local namespace

```
from Subsystem.Package.Foo_cff import *
```

(looks into Subsystem/Package/python/Foo_cff.py)

- To load everything from a python module into your process object you can say:

```
process.load('Subsystem.Package.Foo_cff')
```

- Don't forget that all imports create references, not copies:

changing an object at one place

changes the object at other places

Sequences and Paths

Sequence:

- Defines an execution order and acts as building block for more complex configurations and contains modules or other sequences.

```
trDigi = cms.Sequence(siPixelDigis + siStripDigis)
```

Path:

- Defines which modules and sequences to run.

```
p1 = cms.Path(pdigi * reconstruction)
```

EndPath:

- A list of analyzers or output modules to be run after all paths have been run.

```
outpath = cms.EndPath(myOutput)
```

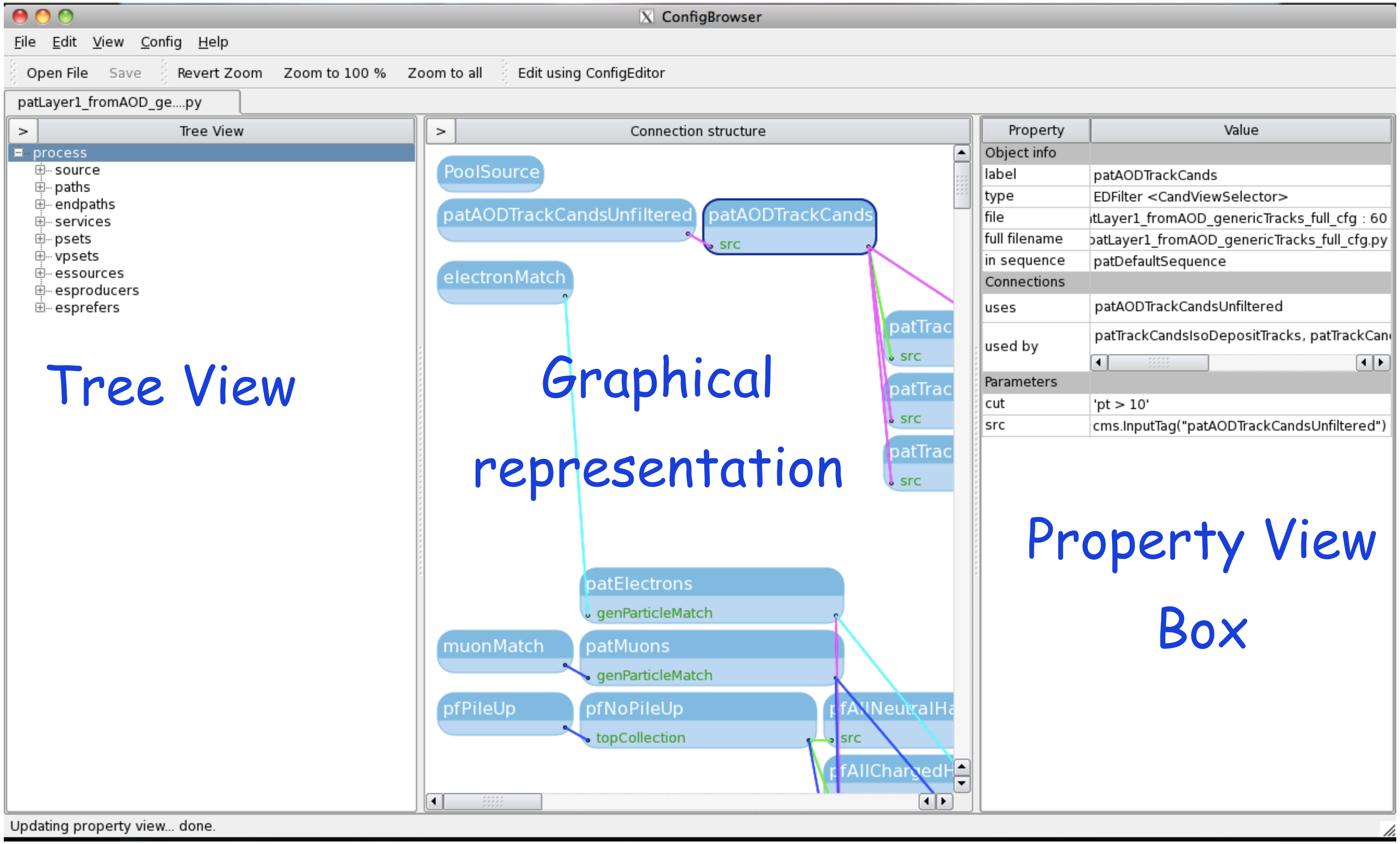
Filters in Paths

- Each path corresponds to a trigger bit
- When an EDFilter is in a path, returning False will cause the path to terminate
- Two operators ~ and - can modify this.
  - ~ means not. The filter will only continue if the filter returns False.
  - - means to ignore the result of the filter and proceed regardless

```
jet500_1000 = cms.Path( ~jet1000filter + jet500filter + jetAnalysis )
```

Inspect your configuration: the ConfigBrowser

You can inspect your config file using a graphical tool as well: the **ConfigBrowser**



https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideConfigBrowser

PAT (Physics Analysis ToolKit)

PAT

PAT is a toolkit as part of the CMSSW framework aimed at performing analysis

It provides:

- data format
- common modules

- It serves as well tested and supported common ground for group and user analyses.
- It facilitates reproducibility and comprehensibility of analyses,
- It is an interface between the sometimes complicated EDM and the simple mind of the common user.
- You can view it as a common language between CMS analysts:
- If another CMS analyst describes you a PAT analysis you can easily know what he/she is talking about

PAT provides a very quick start for beginners

PAT - DATA Formats

Representation of reconstructed physics particles

pat::Candidate (pat::Jet, pat::Photon, pat::Muon,etc..)

There is a base class common to all kind of "Particles": the reco::Candidate

It provides access:

- kinematics (pt, mass, eta, phi, etc.)
- underlying componenents (link to track, superclaster, etc.)
- navigation among the daughters (to access the daughter particles and their attributes)

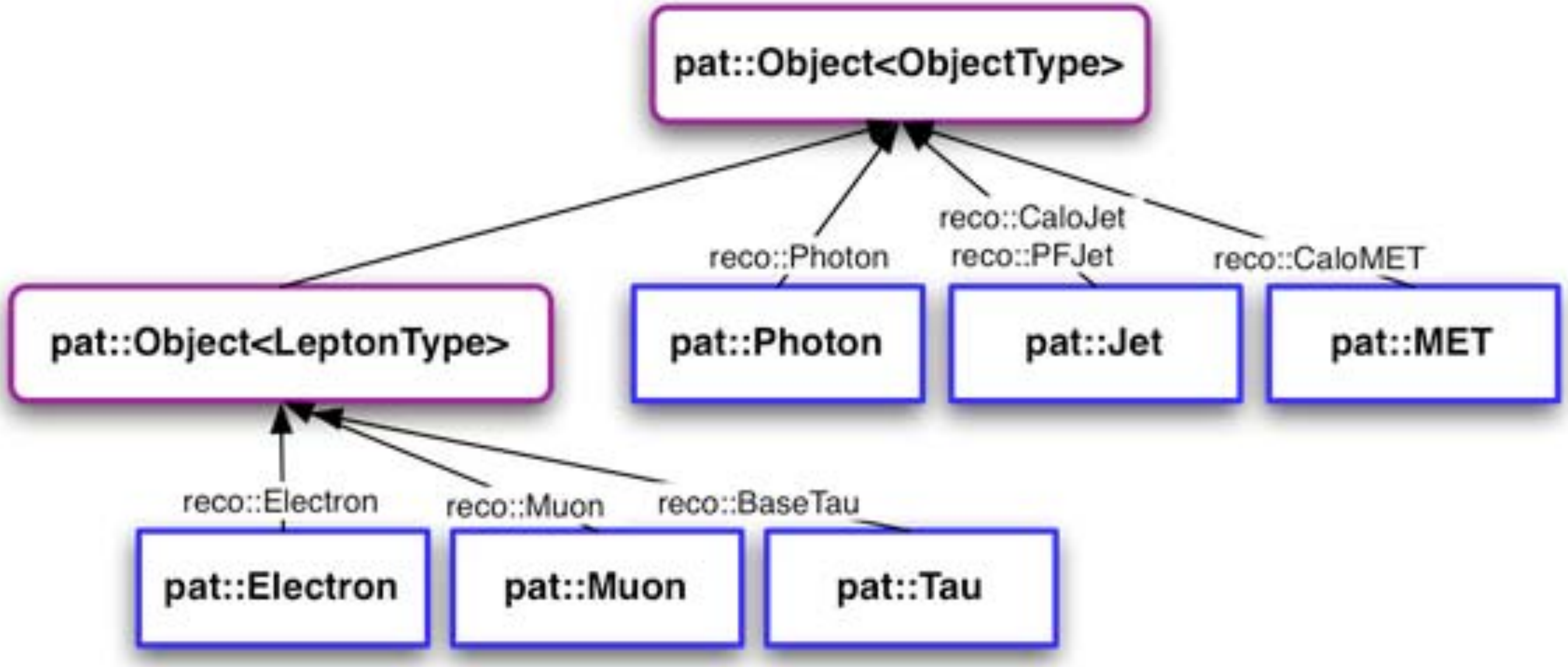
The pat::Object inherits from the reco:Candidate

pat::Candidate = reco:Candidate + more

Add extra informations to pat Candidates wrt reco Candidates such as:

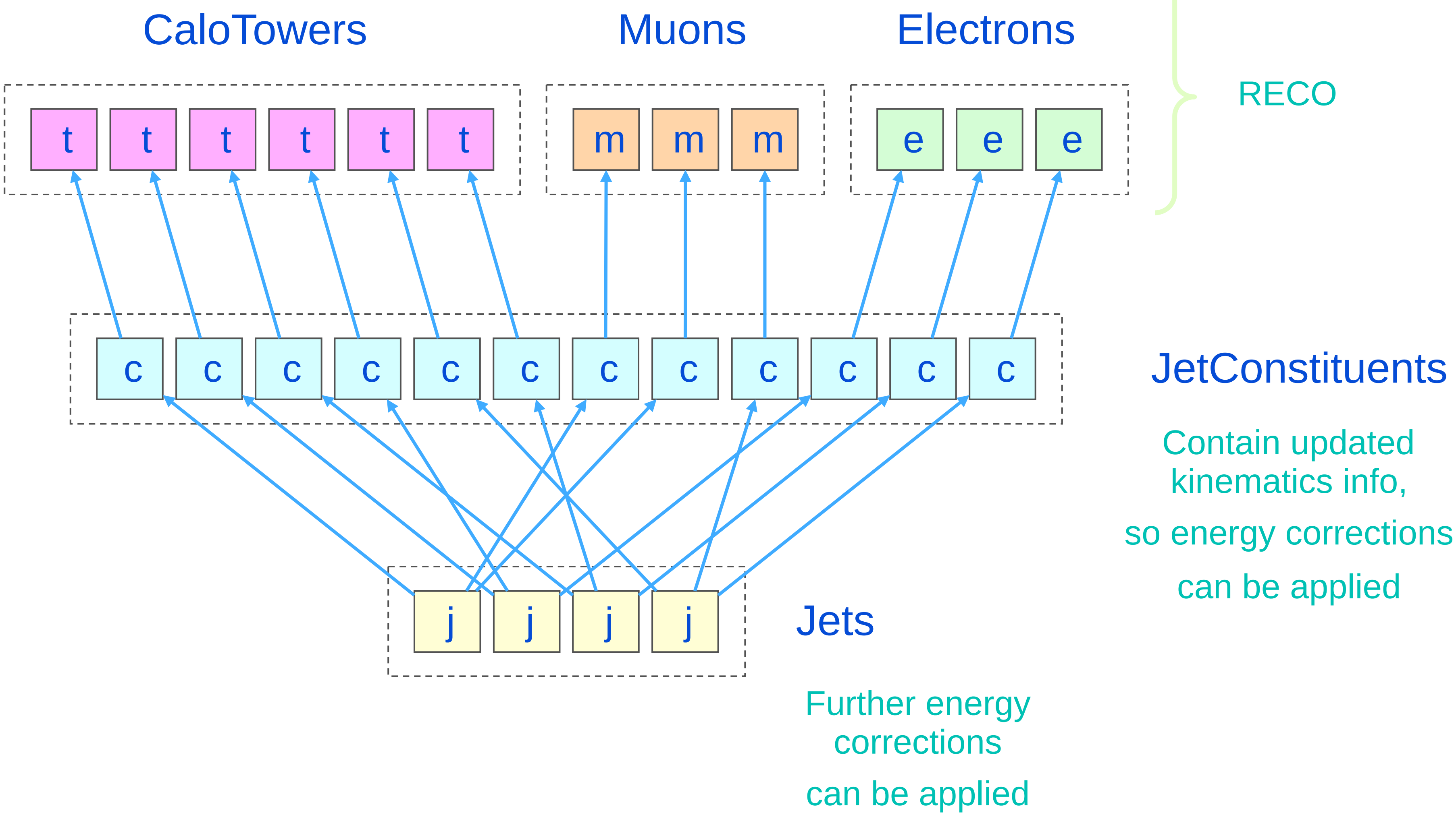
- Isolation
- MC matching
- Trigger matching

PAT - DATA Formats 2



This is the hierarchy of pat::Candidates

Analysis a la CMS: Particle Candidates for Jets



PAT - Common modules

PAT provides a series of modules common to different analysis task, such as:

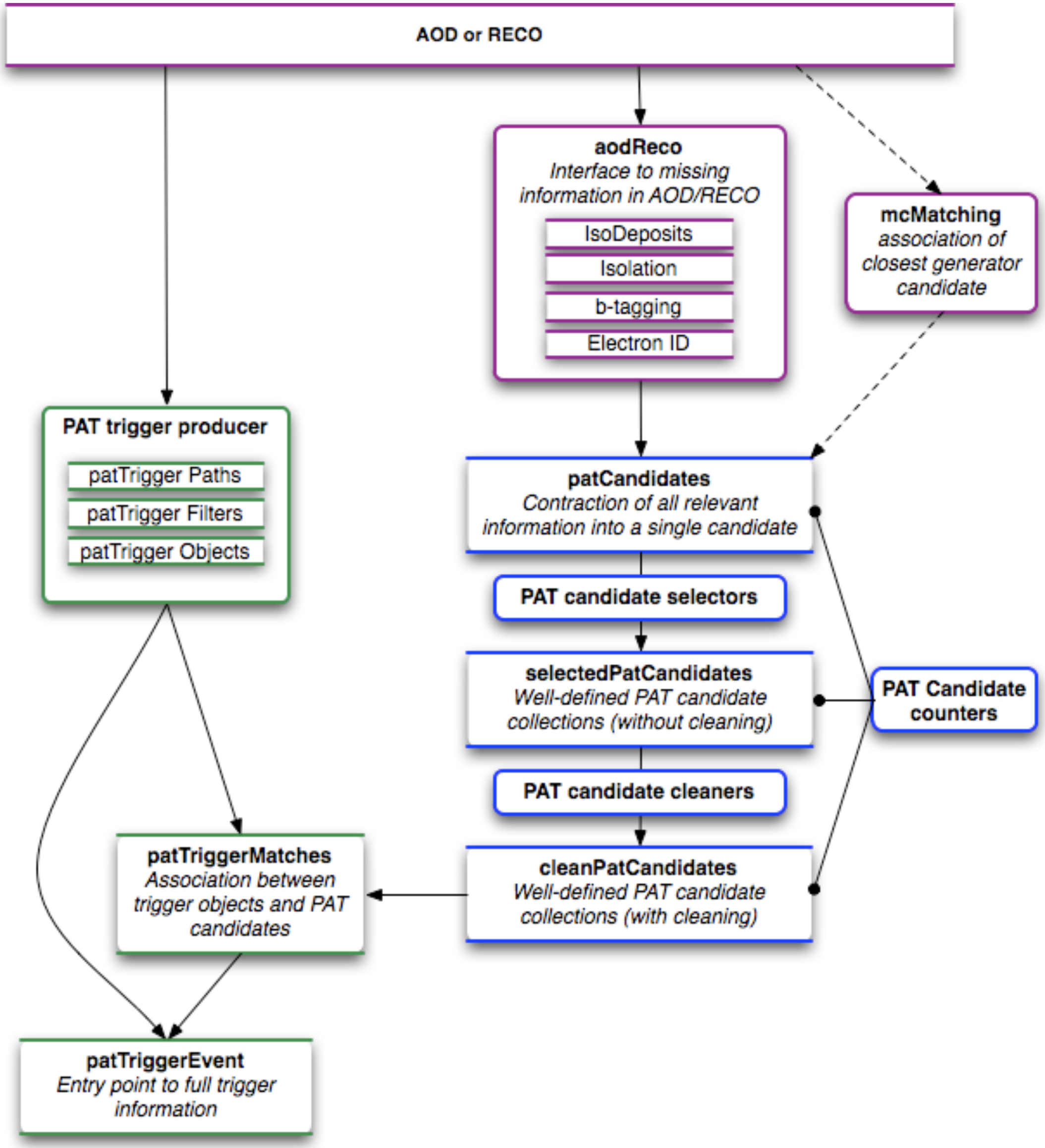
- **Cleaning** - to remove disambiguities on the identifications of particles in the event
- **MC matching** - to associate PAT objects with generator objetcs
- **Trigger matching** - to associate PAT objects with trigger objects

Moreover PAT provides a set of tools to performe easily the configuration of the Workflow → **PAT Tools**

These tools have been conceived to be common to different kind of analysis and, at the same time, to be customized according to the specific analysis requirement. They are well documented in SWGuidePATTools:

https://twiki.cern.ch/twiki/bin/view/CMS/SWGuidePATTools

PAT workflow



PAT workflow is organised in a principal sequence and a parallel one for associating trigger informations.

The main steps of the patTuple production are:

- PAT preproduction
- Candidate Production
- Candidate Selection
- Candidate Disambiguation
- PAT Trigger Event

Browsing configuration

The PAT workflow is defined in python configuration files.

We can learn how it is structured inspecting it by looking at which are the modules involved and in which way they are related to each other, at which values parameters are set and so on.

Some tools have been implemented to investigate workflow configuration and the pat::Tuple produced:

- edmConfigEditor (https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookPATConfigEditor)
- edmDumpEventContent

Editing configuration

The production of an user-defined PAT analysis consists of three steps:

1. Start from the standard configuration file.
2. Apply the PAT tools to change the configuration of the standard configuring file according to the specific needs of your analysis.
3. Replace parameter values according to the needs of your analysis.

To browse and edit configuration file you can use a graphical tool:

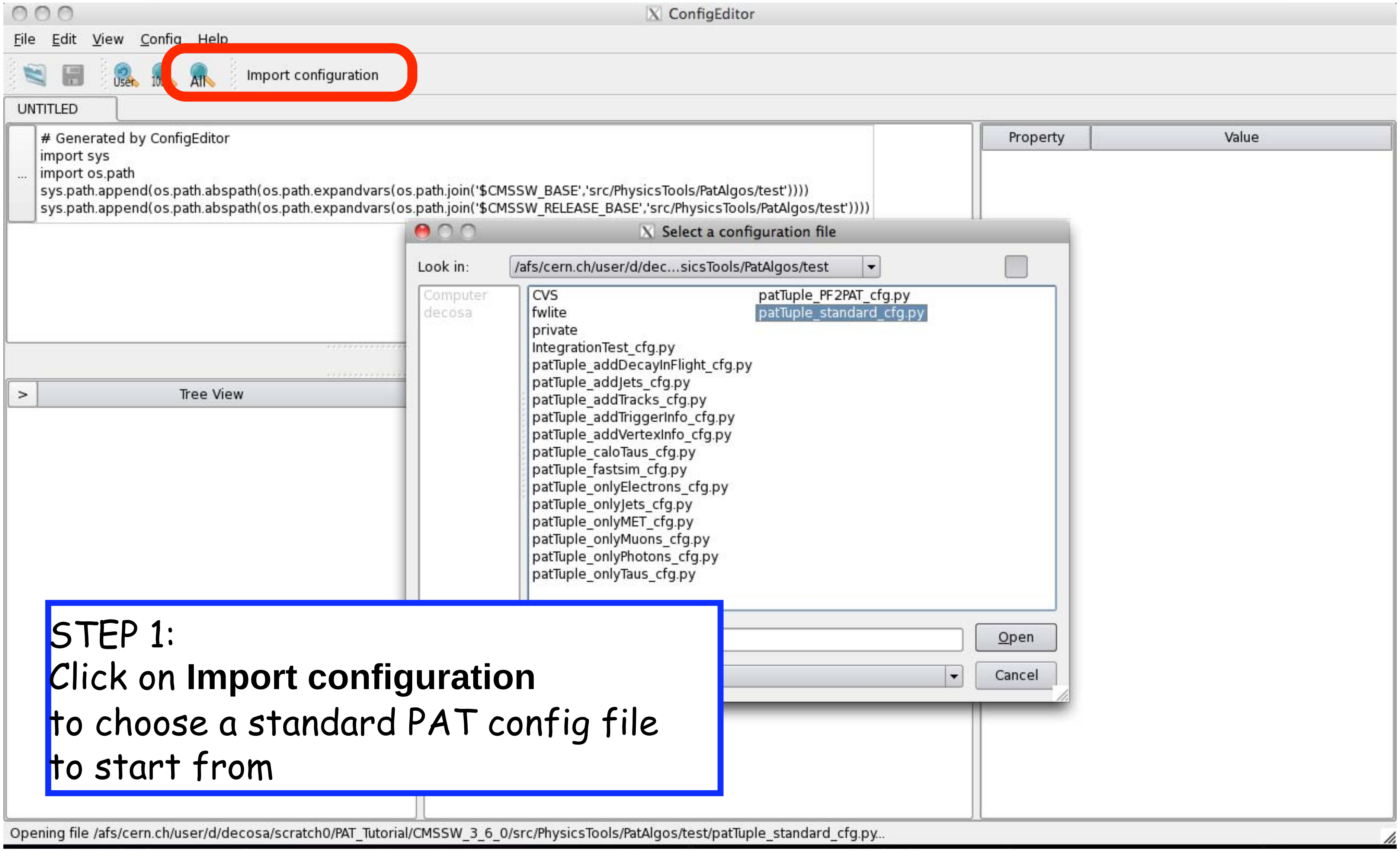
the edmConfigEditor

NOTE: You may want to learn more about ConfigEditor and PAT Tools, look the SWGuideConfigExercise page:

SWGuidePATConfigExercise

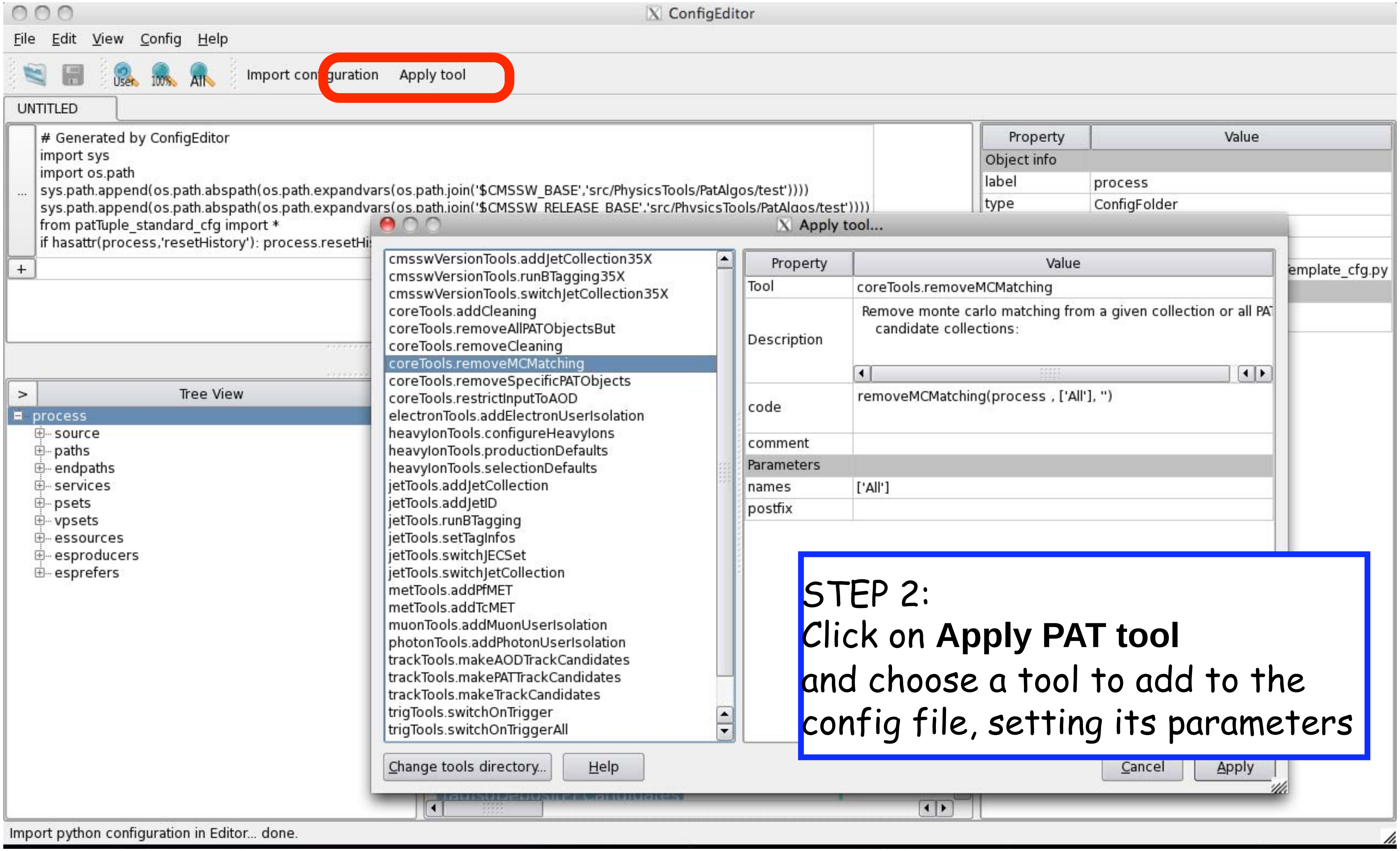
https://twiki.cern.ch/twiki/bin/view/CMS/SWGuidePATConfigExercise

The ConfigEditor - Import



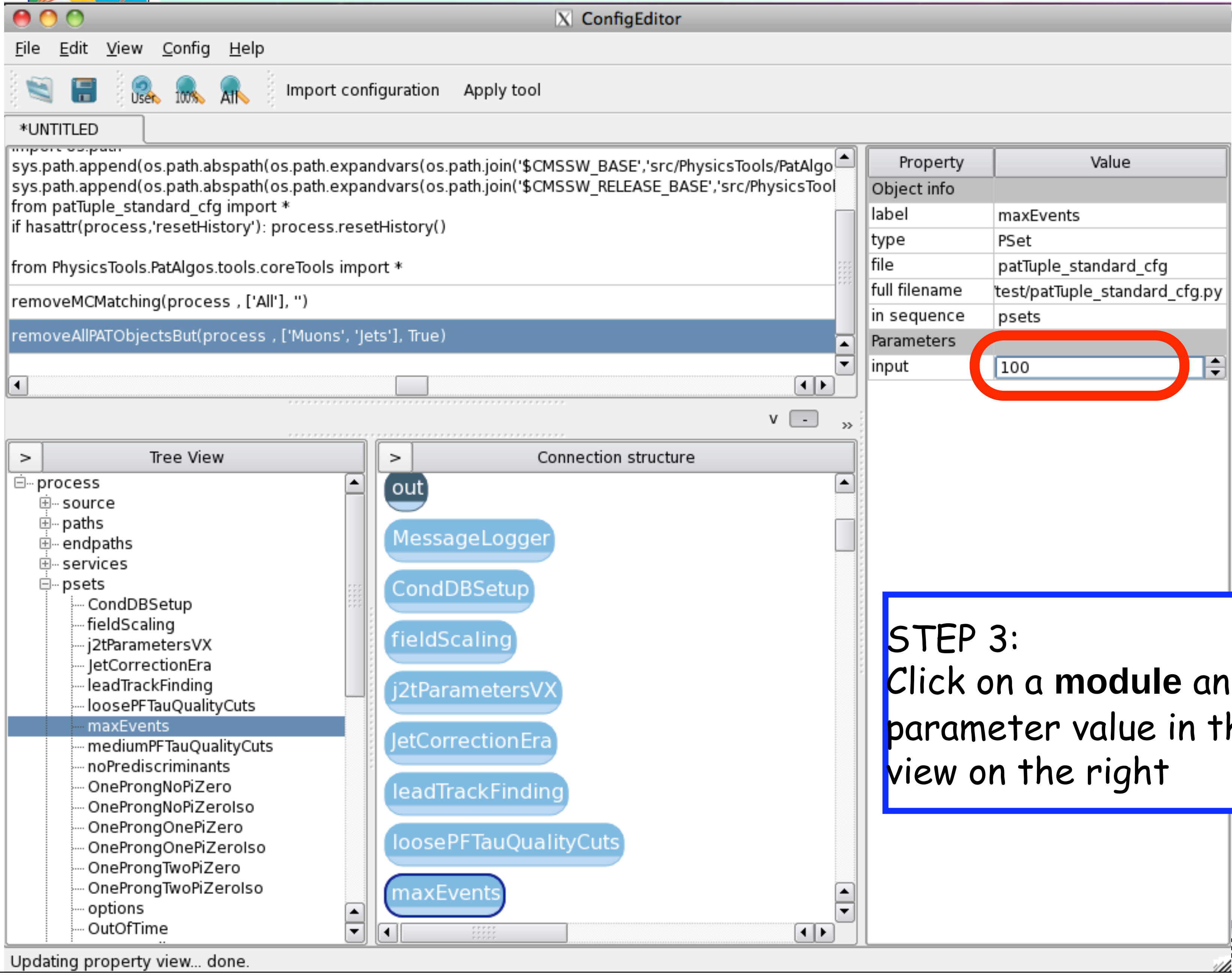
STEP 1:
Click on **Import configuration**
to choose a standard PAT config file
to start from

ConfigEditor - Apply tool



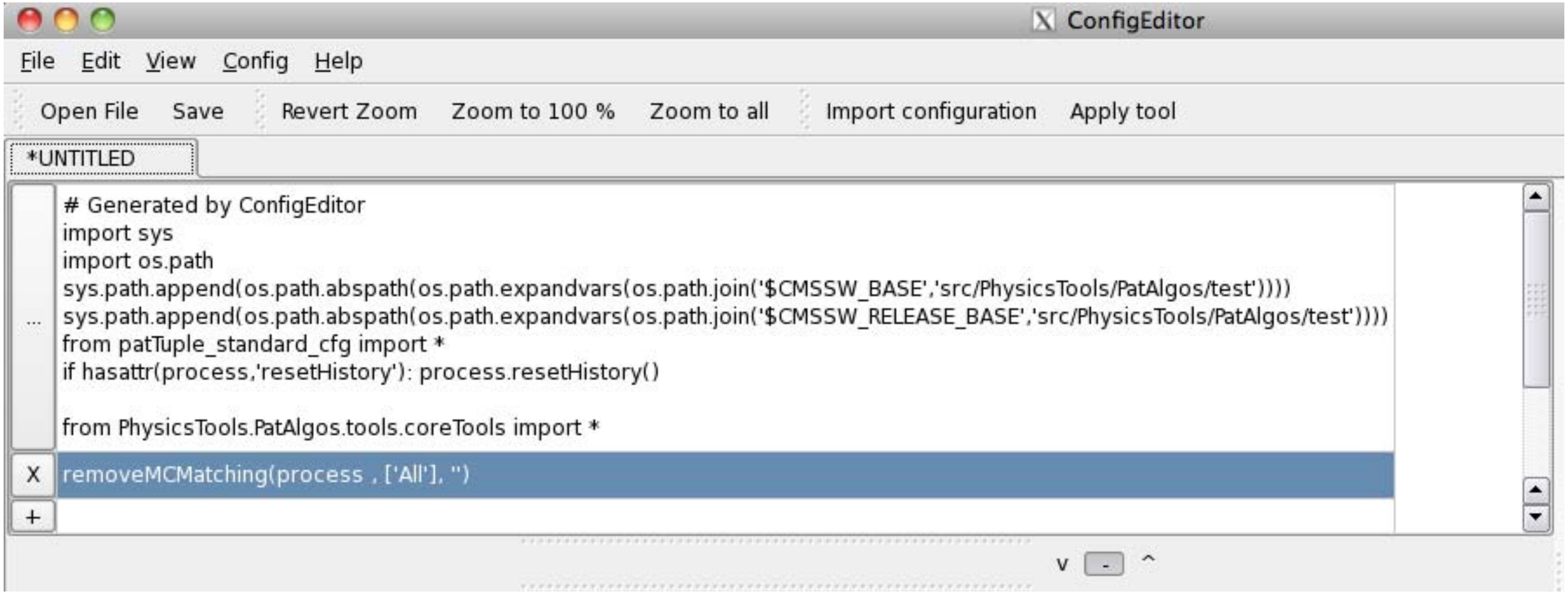
STEP 2:
Click on **Apply PAT tool**
and choose a tool to add to the
config file, setting its parameters

ConfigEditor - Replace parameters



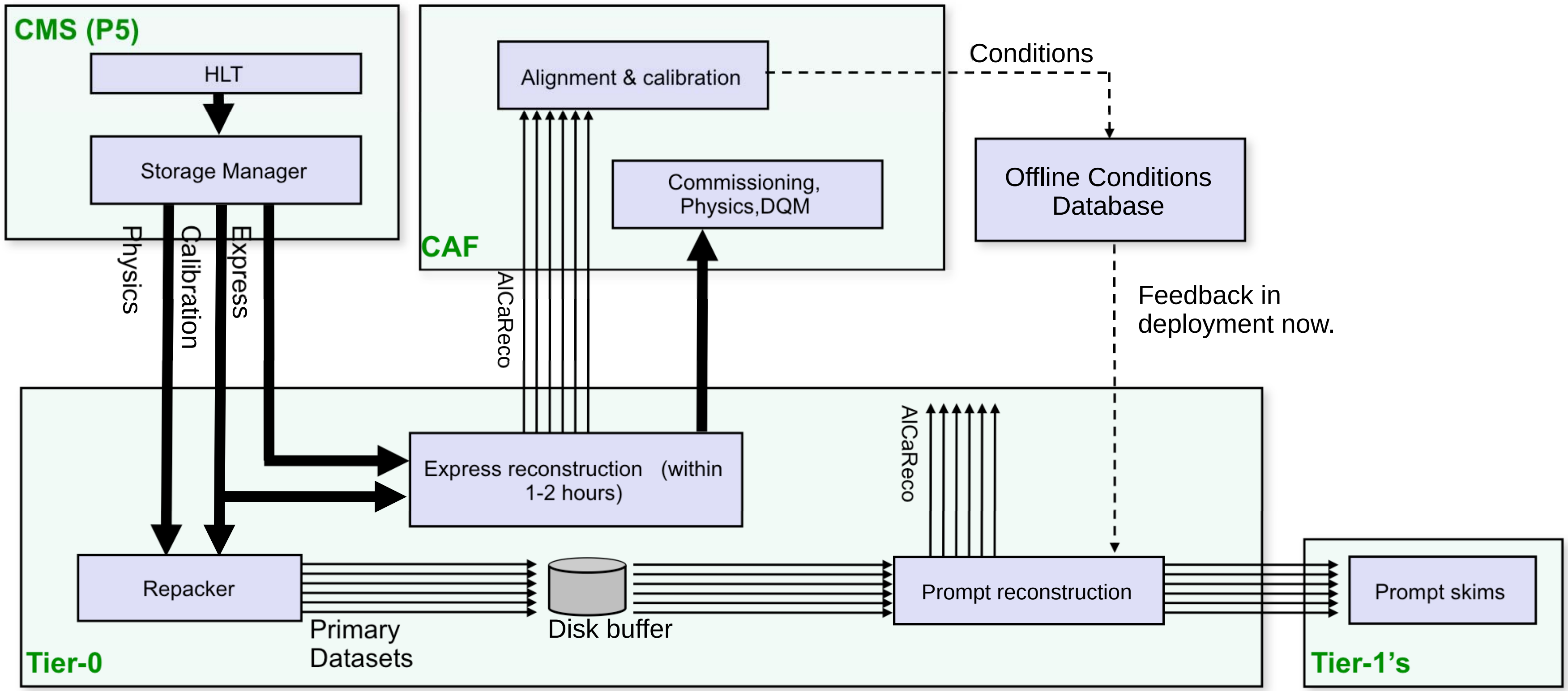
STEP 3:
Click on a **module** and replace a parameter value in the Property view on the right

ConfigEditor - The resulting code



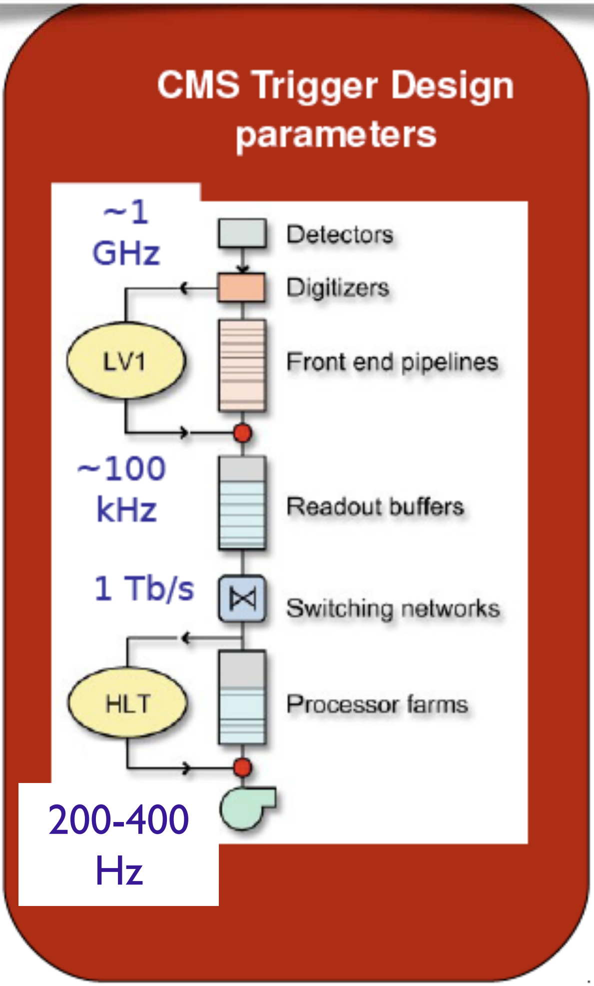
In the left corner inspect the produced code

Data Flow & Offline Operations

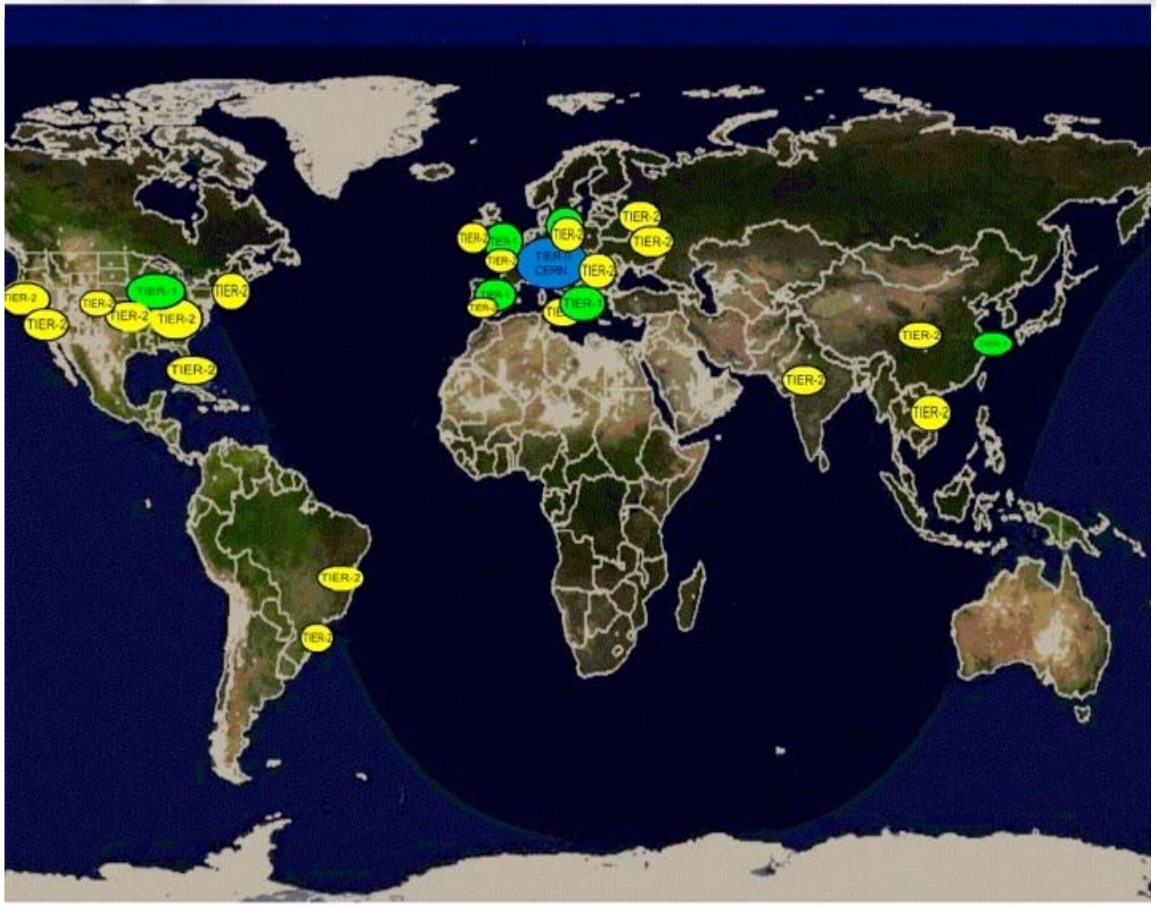


Data Flow: from Detector to Tier4 (your PC..)

DETECTOR AND L1 & HLT



TIERS WORLDWIDE DISTRIBUTED



How data are distributed & role for the different Tiers

- Computing Model (key ingredients)
  - 2 safe copies of RAW data on tape at CERN and Tier-1 sites
  - 2-3 large re-reconstruction passes per year in first years at Tier-1 sites
  - Monte Carlo production matches collision data
  - Production and user jobs go where the data is
  - Full network connectivity. All sites are connected with each other
- A lot a work went into preparation and testing
- Provides flexibility to tackle unforeseen scenarios (very high turn-around before conferences)

- Tier0 activities
  - Prompt data processing, Prompt calibration and alignmente, Storage of Raw data backup
- Tier1 activities
  - Custodial storage of Raw data, Prompt skimming, Reprocessing of data and MC, MC production.
- Tier 2 activities
  - MC production, User analysis
- Tier3 activities
  - User analysis

Offline data-taking Operations

Offline workflows deliver validated & calibrated reconstructed data for physics analysis

offline reconstruction

- prompt feedback on detector status and data quality
- sample for physics analysis

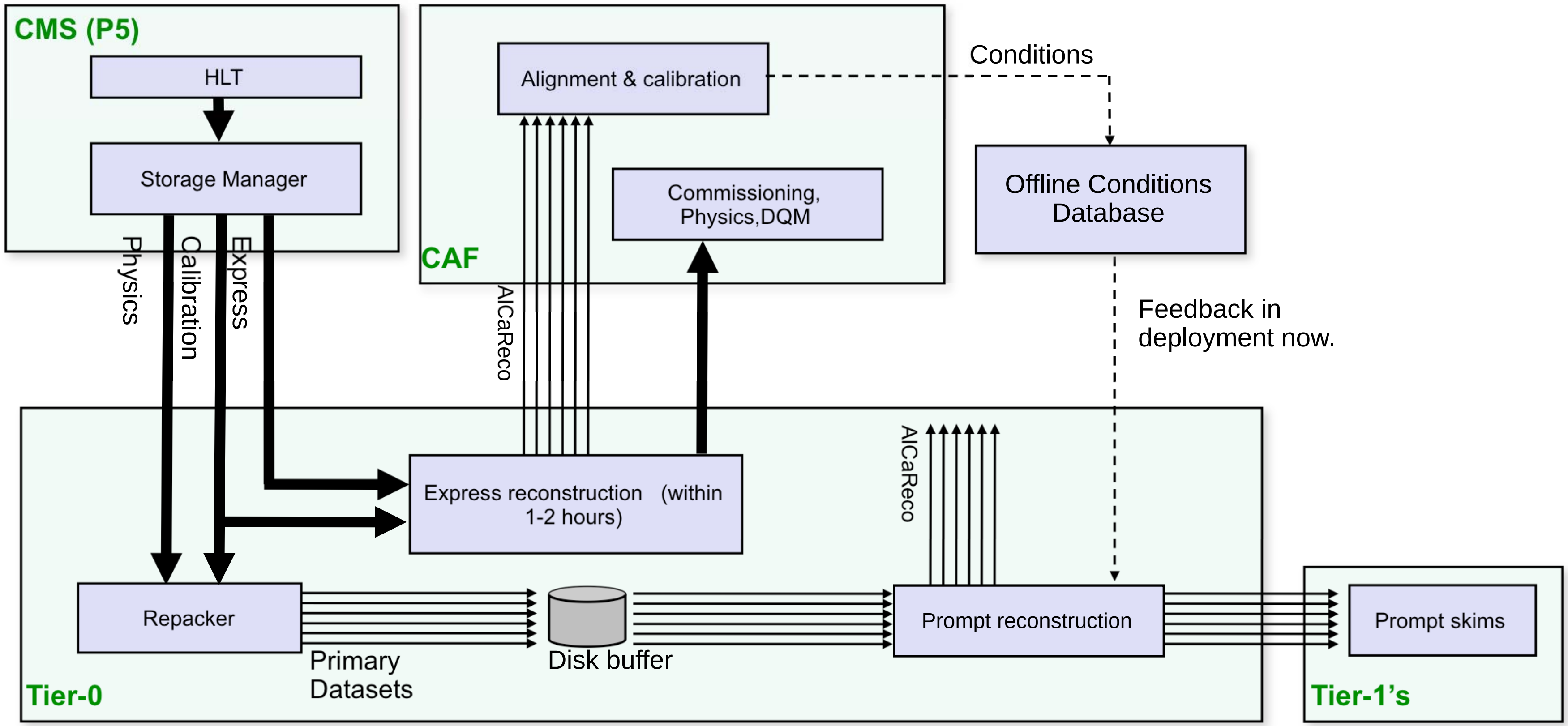
up-to-date alignment & calibration (AlCa)

- calibration workflows with short latency
- provide samples for calibration purposes: AlCa streams
- consistent set of conditions for data and MC

data validation and certification for analysis

- data quality monitoring (DQM)

Processing at Tier0/CAF (CERN)

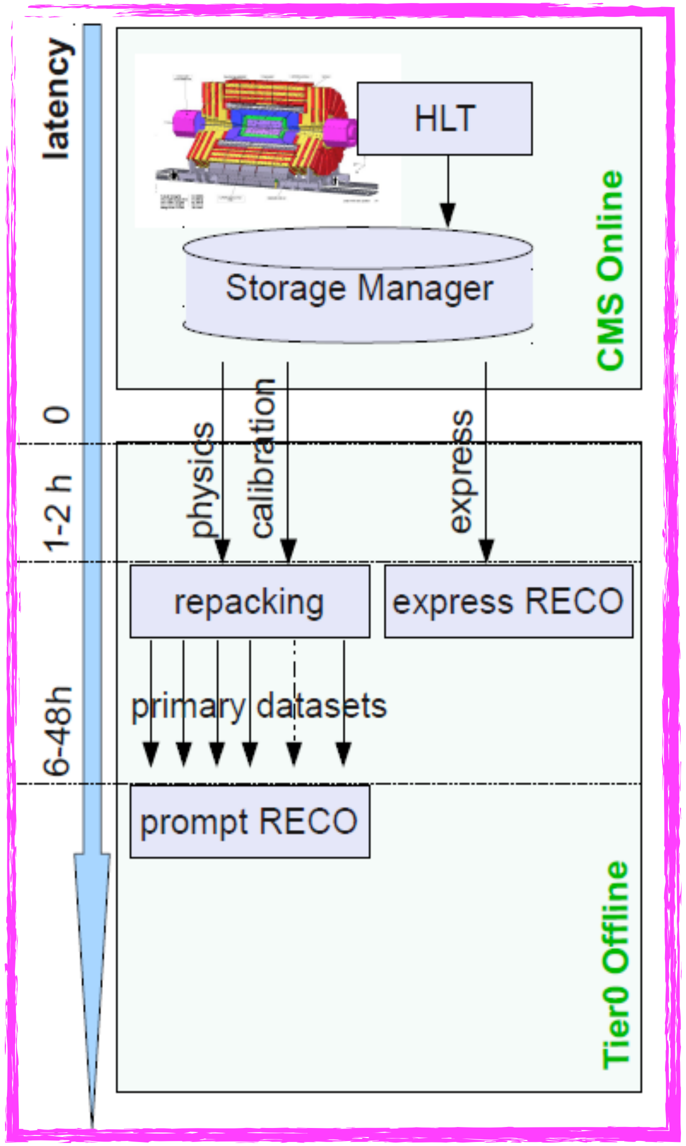


Data Streams and Tier0

Data streams & Tier0 workflows

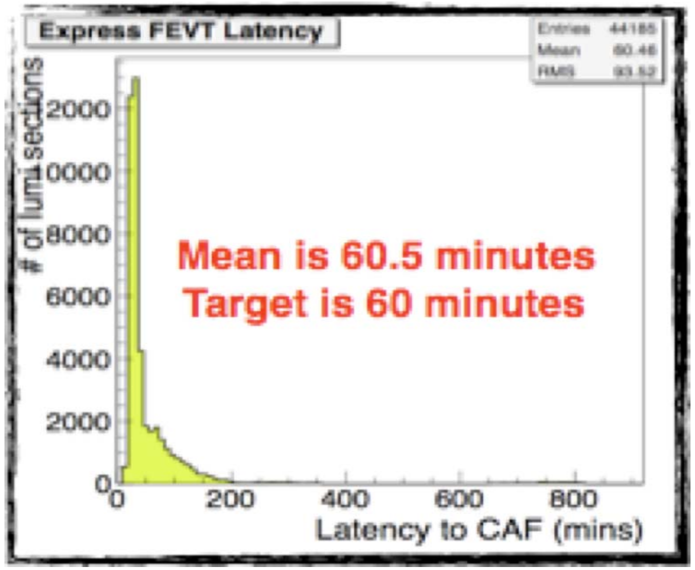
Depending on the latency

- Express: prompt feedback & calibrations
  - short latency: 1-2 hours
  - ~40Hz bandwidth shared by:
    - calibration (½)
    - detector monitoring (¼)
    - physics monitoring (¼)
- Alignment & Calibration (AlCa) streams
- Prompt reconstruction: sample for physics analysis
  - split in Primary Datasets using High Level
  - will be delayed of 48h (latest calibrations)
  - writing ~300Hz



Prompt reconstruction and express stream

- Rolling workflows are fully automated
- Express processing provides quick feedback for commissioning, data quality monitoring and physics
- Alignment and calibration loop to improve quality of prompt reconstruction
- Operational experience at Tier-0 is excellent. **Success rate of 99.9%**
- Categorize data according to trigger selection in primary dataset



Calibration Workflows

Provide most up-to-date conditions @ all stages of the data processing

Different workflows depending on the time scale of updates:

- quasi-online calibrations for HLT and express:
  - e.g. beam-spot → quick determination online
- prompt calibrations: monitor/update conditions expected to vary run-by-run (or even more frequently):
  - updated conditions must be ready before prompt-reconstruction
- offline re-reco workflows:
  - more stable conditions
  - workflows which need higher statistics:
    run on AlCa streams produced during prompt-reco or offline re-reco

Example: beam Spot determination

Track beam-spot 3D position and width as a function of time:

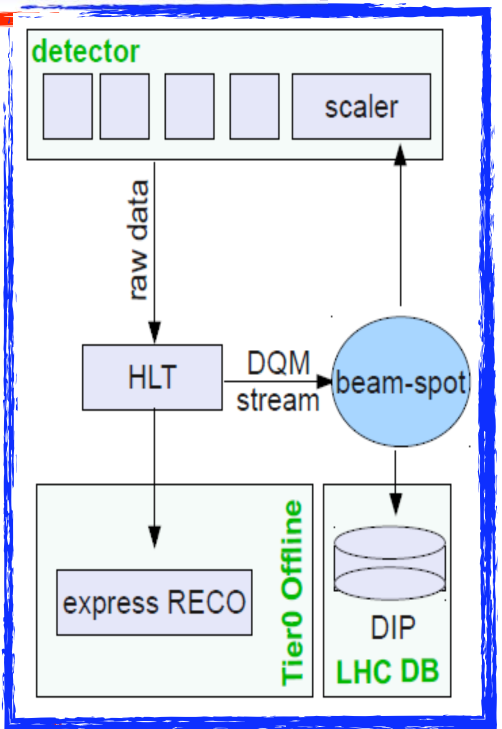
- track based: correlation of impact parameter and azimuthal angle (d_0 -Φ)
- vertex based: 3D fit to distribution of primary-vertexes

Quasi-online workflow for express (and HLT) reconstruction

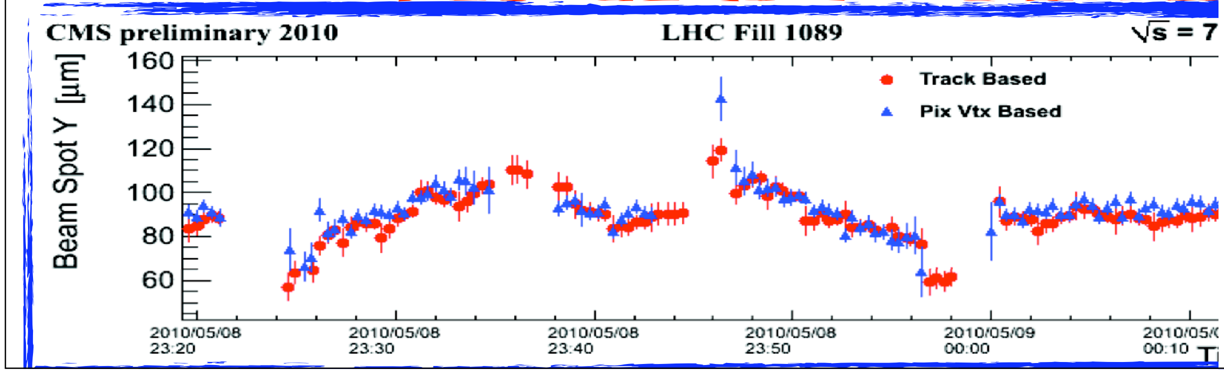
- using DQM-dedicated stream (sampling @ ~ 100Hz max)
- using track based and pixel-only vertexing → very fast
  - 1 value every 5 Lumi-Section (~2 min)

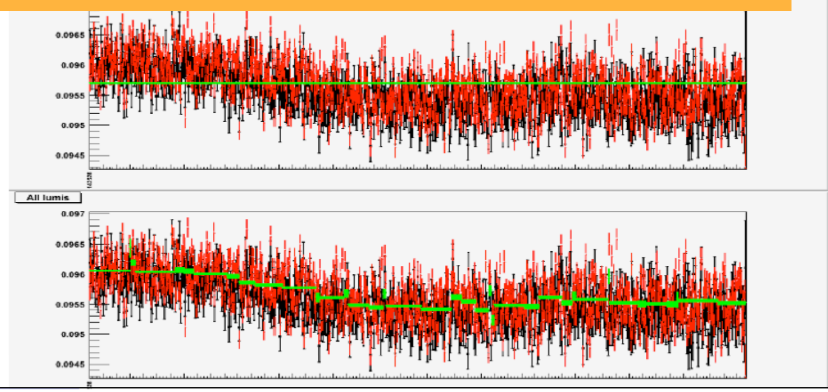
Runs also in prompt-calibration loop (in deployment phase)

- full statistics and tracking capabilities
  - 1 value every LS (=23s)



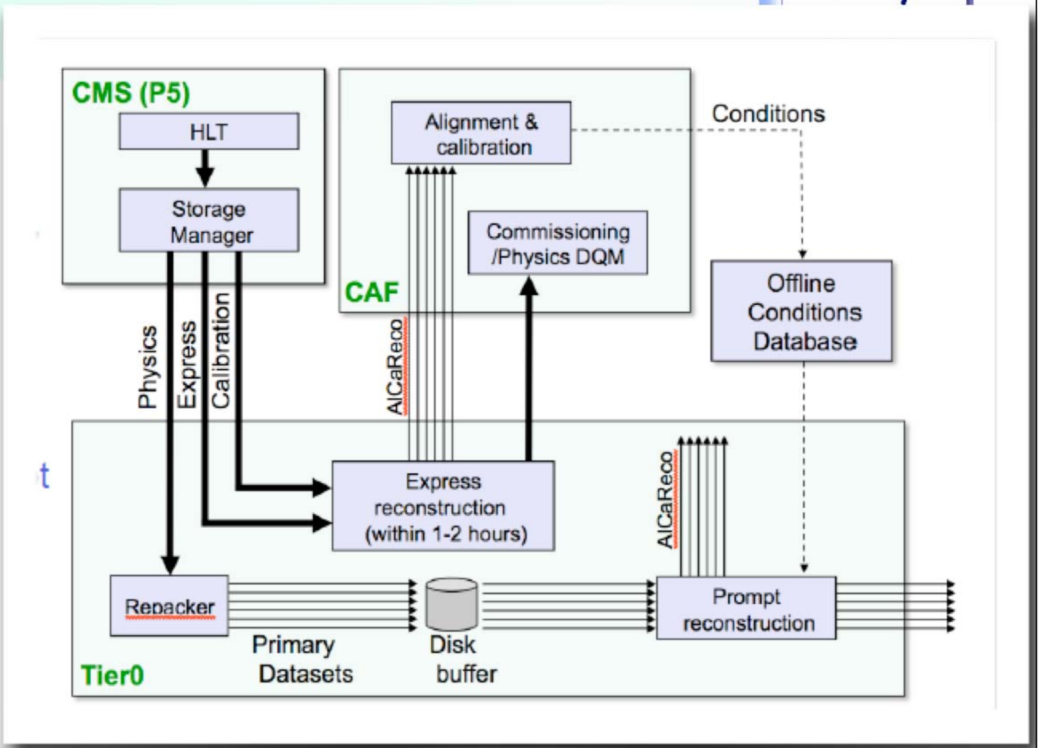
Additional granularity is especially useful in long runs to correct for slow drift in beam position





Prompt Calibration Loop

- Prompt calibration workflows:
  - conditions which need continuous updates:
    - beam-spot position
      - measured every 23s (1 LS)
    - tracker problematic channels
      - respond to HV trips/noise
  - conditions which need monitoring
    - calorimeter problematic channels
      - mask hot channels
    - tracker alignment
      - monitor movements of large structures



- Update strategy based on delay between express and prompt reco
  - AlCa streams out of express used for calibration
    - compute conditions in time for prompt-reco (start 48h later)

Just started the deployment

- **<u>Reduce need for offline re-reco just after data-taking</u>**
- Dedicated resources @ CERN: CMS Analysis Facility (CAF)

Calibration & Alignment

All workflows fed using dedicated skims or datasets:

- event selection tuned on the needs of the workflow
- event content reduced to optimize bandwidth/disk space usage

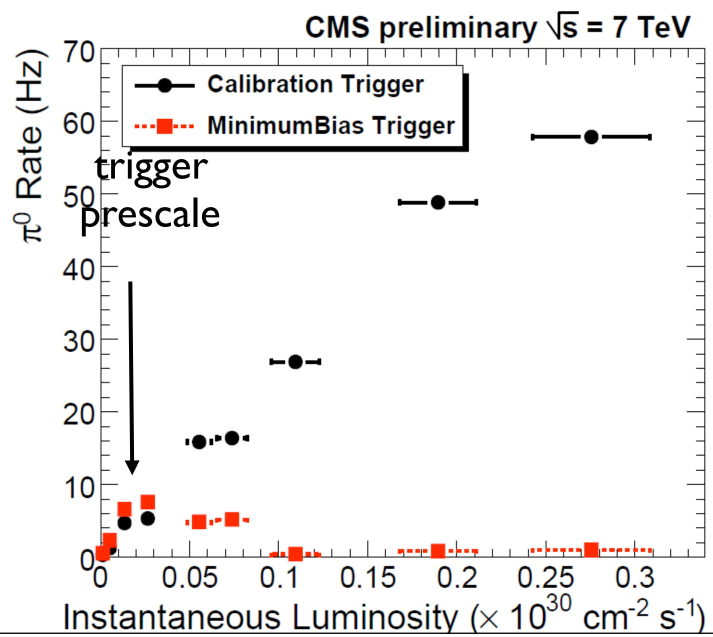
2 kind of calibration streams:

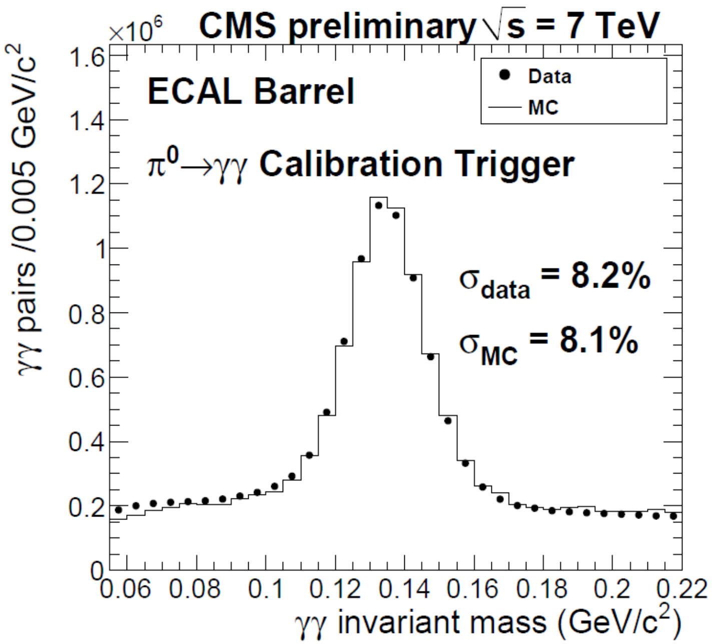
- produced directly @ HLT level
  - workflows statistically limited or requiring dedicated selection:
    - e.g. ECAL π^0 stream and Φ-symmetry....
  - profit from High Level Trigger flexibility → software based
- produced offline during express and prompt reconstruction (and offline re-processing)
  - just skimming events dedicated to calibrations

Example: ECAL Calibration

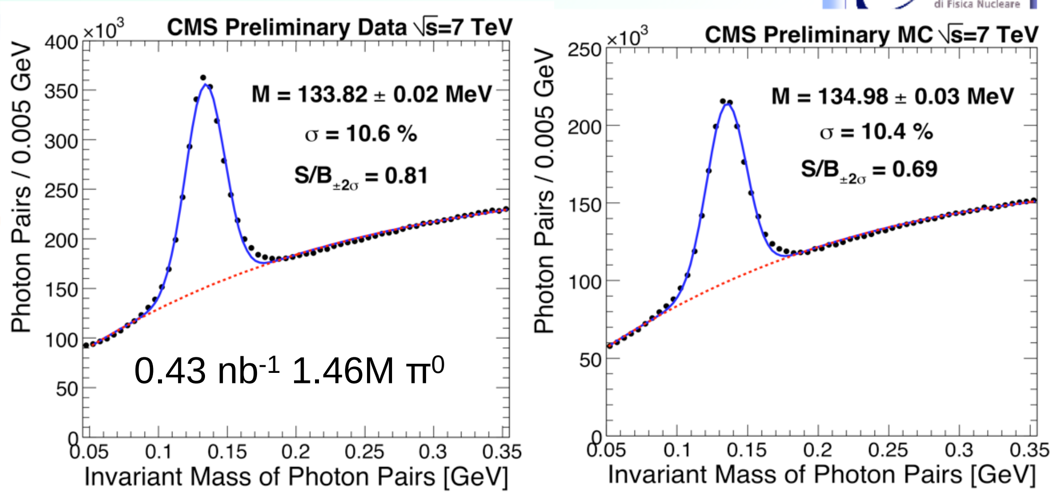
- Calibration stream produced @ HLT level: π^0 and η calibration events
- Stream optimized for:
  - low CPU usage @ HLT:
    - seeded by Level1 single-e/γ or single-Jet triggers
    - regional unpacking ($\Delta\eta \times \Delta\Phi$ = 0.25 x 0.4 around the seed)
    - event selection based on info @ crystal-level only
  - low bandwidth
    - store data only for interesting crystals (ROI)

Produced directly on Online Stream

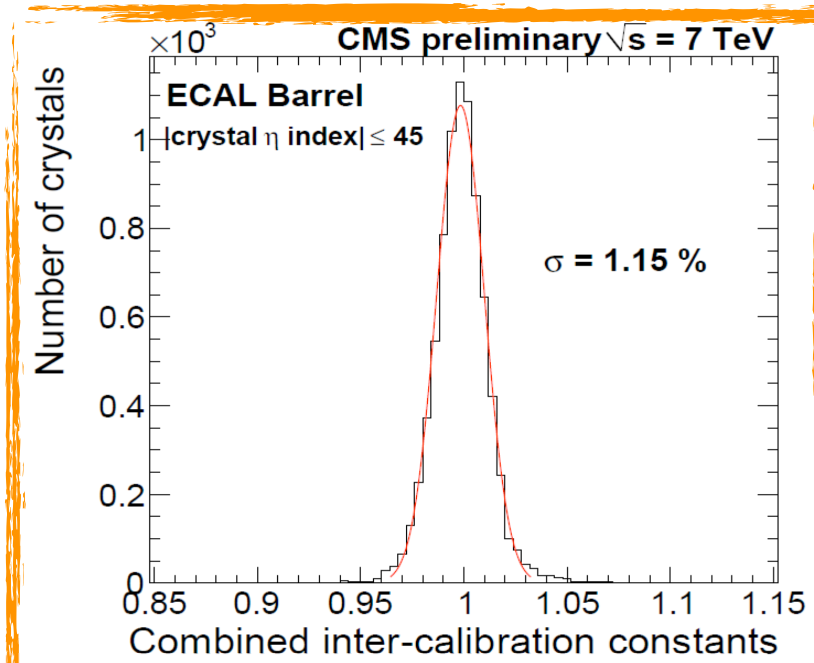




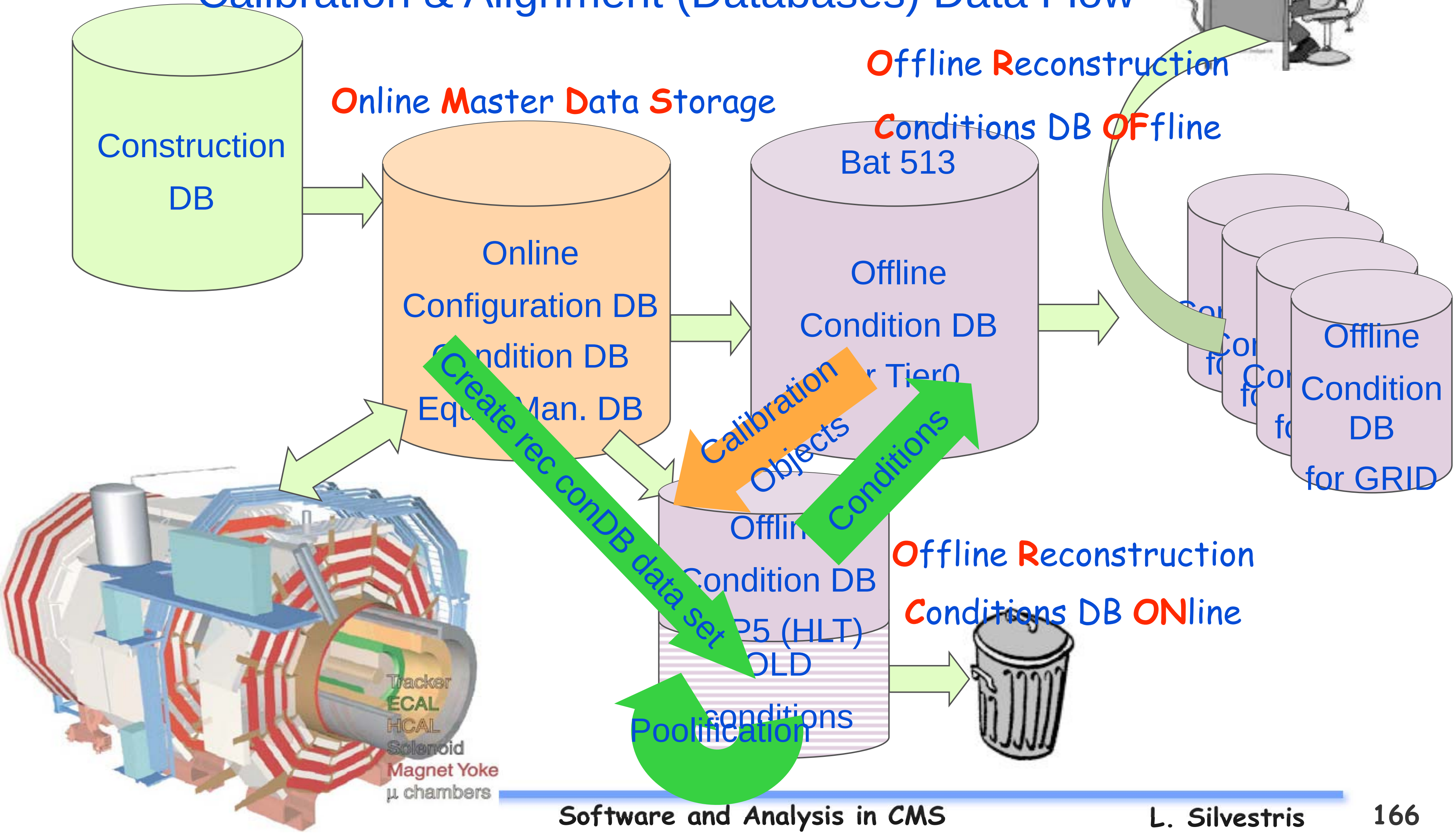
Example: ECAL Calibration



- Inter-calibration based on several (complementary) techniques:
  - Φ-symmetry → Φ invariance of energy fixed pseudo-rapidity
- dedicated stream (@ HLT) of Minimum-Bias events
- already ~ asymptotic in terms of performance
  - π^0 and η calibration → photon pairs $\pi^0(\eta) \rightarrow \gamma\gamma$
  - isolated electrons and di-electron resonances (larger dataset $O(fb^{-1})$)
  - monitoring of crystal transparency and light yield (only @ higher lumi)
- Combination allows to reach 1.15% precision in the barrel (design goal for $H \rightarrow \gamma\gamma$ is 0.5%)



Calibration & Alignment (Databases) Data Flow



Data Certification

- The complexity of the offline workflows requires robust validation

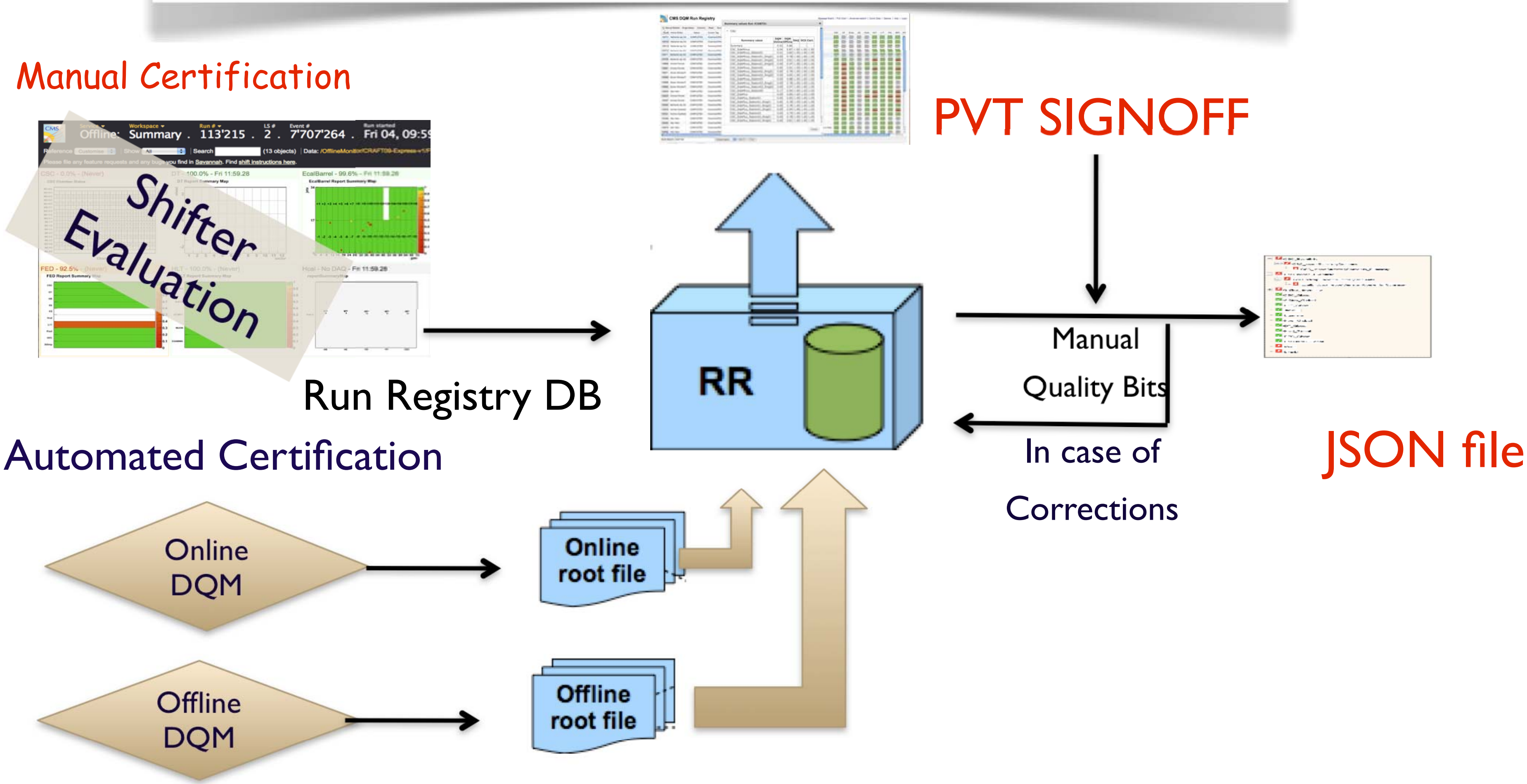
- Several stages of Data Quality Monitoring (DQM):
  - online DQM → monitor detector performance during data-taking: dedicate event stream
  - offline DQM → monitor performance of physics objects
    - runs on full statistics available for analysis:
      - express reco → fast feedback
      - prompt-reco → continuous monitor
      - offline re-reco → validation of software and condition updates

- Physics Validation Team (PVT) → coordinates the validation activity. Feedback from: groups responsible for physics objects; detector performance groups; analysis group

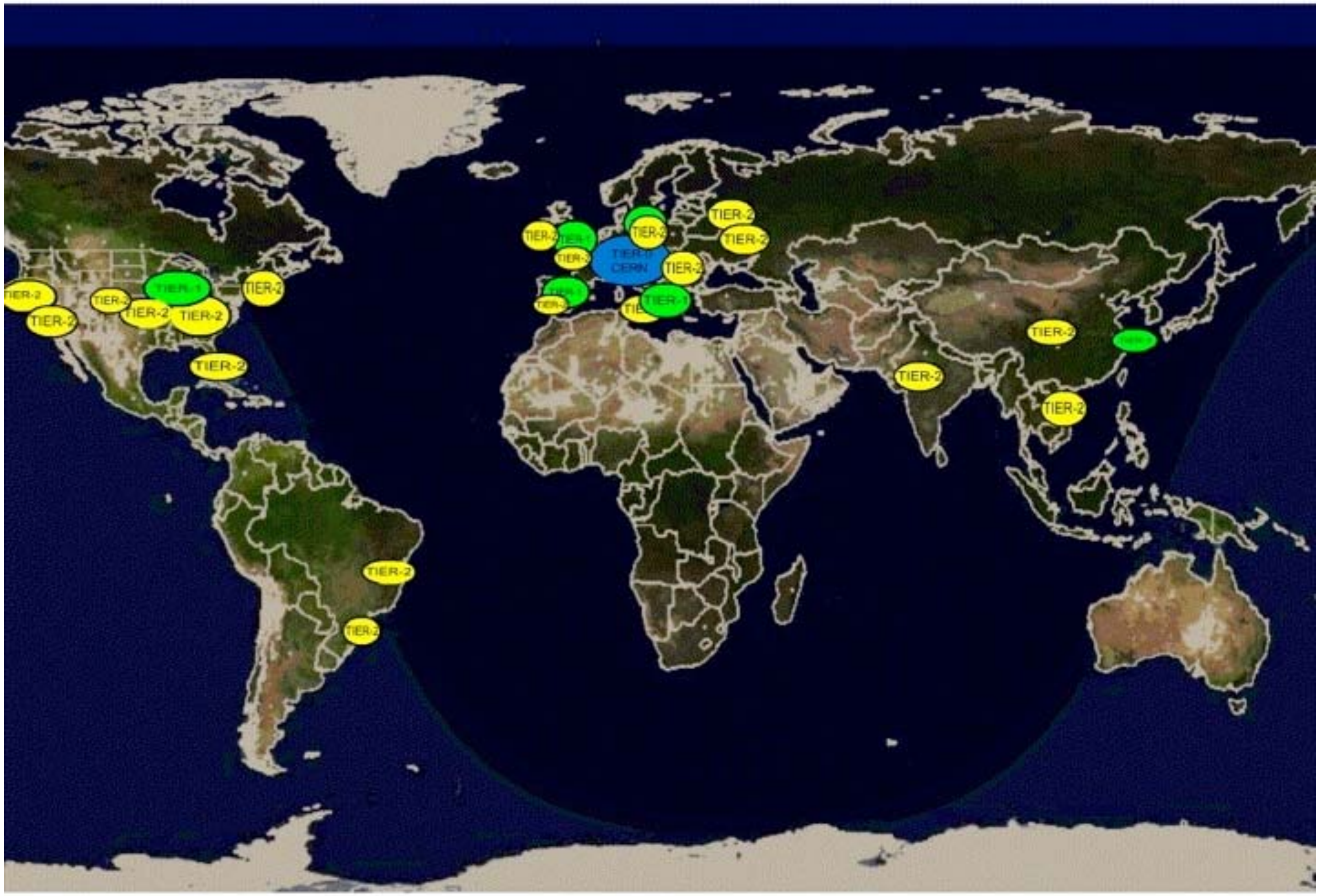
Workflow for Data Certification

Production of GOOD Runs and Luminosity Section Lists

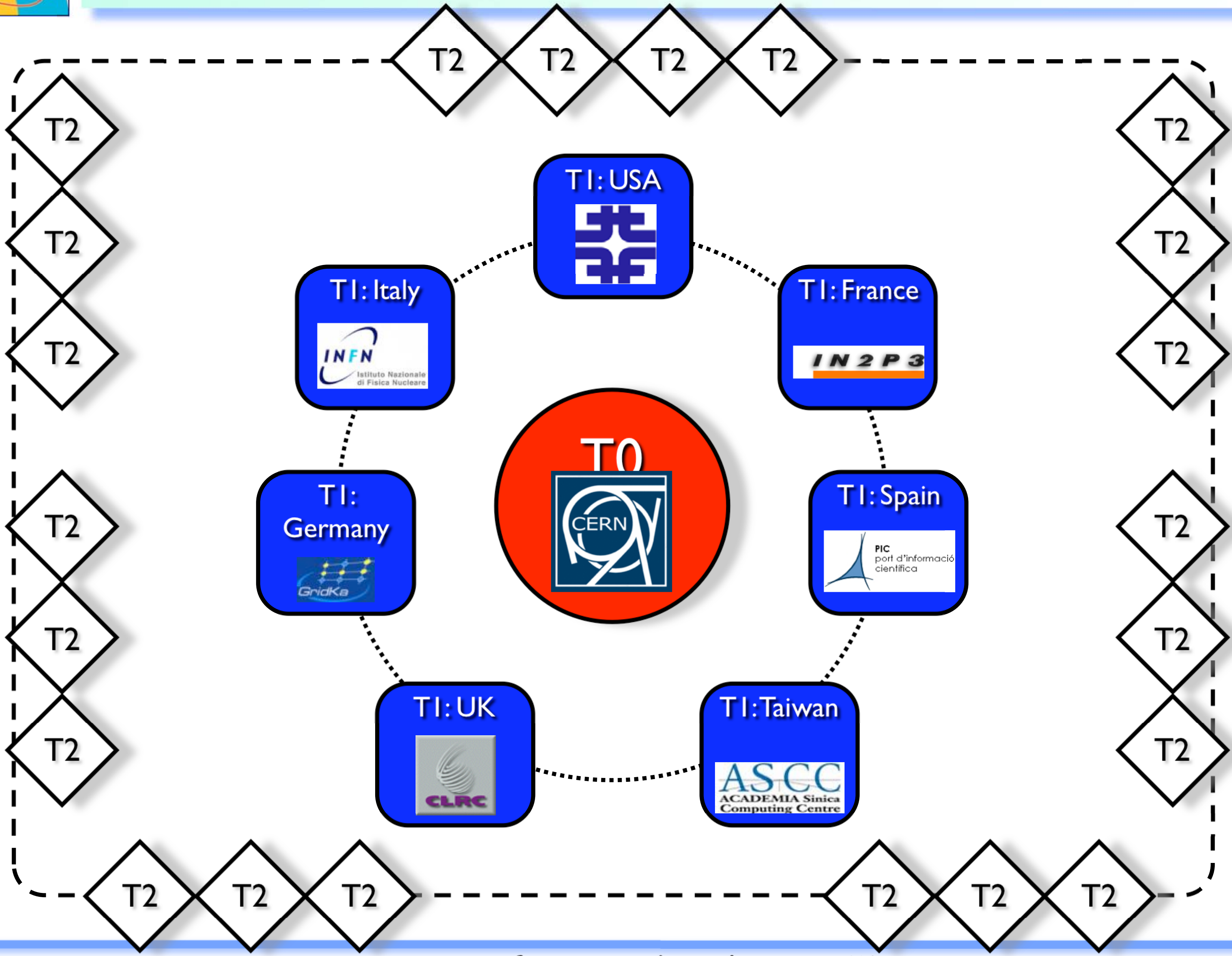
for use in Physics Analyses



Computing Operations



CMS Computing model: tiers



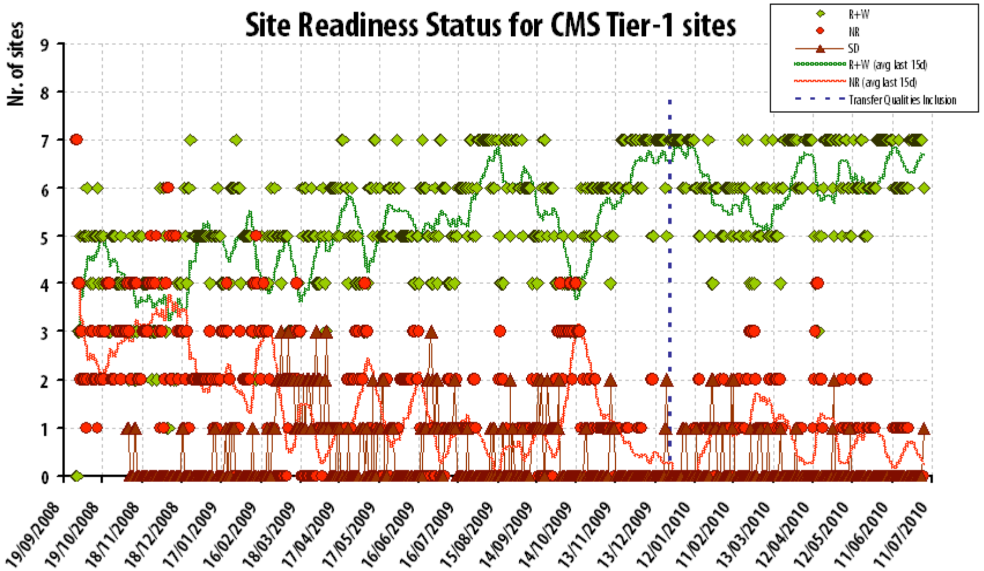
CMS Computing model

- **Tier 0 (T0)** at CERN **(20% of all CMS computing resources)**
  - Record and prompt reconstruct collision data
  - Calculate condition and alignment constants
  - Store data on tape (only archival copy, no access)
  - Only central processing, *no user access*
- **Tier 1 (T1)**: regional centers in 7 countries **(40% of all CMS computing resources)**
  - Store data fraction on tape (served copy)
    - Every T1 site gets a fraction of the data according to its respective size
  - Archive fraction of produced MC on tape
  - Skim data to reduce data size and make data more easily handleable
  - Rereconstruct data with newer software and conditions/alignment constants
  - Only central processing, *limited user access*
- **Tier 2 (T2)**: local computing centers at Universities and Laboratories **(40% of all CMS computing resources)**
  - Simulate MC events
  - **User access to data for analysis**

CMS distributed computing MODEL

- "Data driven" computing model
  - Data and MC samples are distributed centrally
  - Jobs (processing, analysis) "go" to the data
- Requires very fast network connections between the different centers:
  - T0→T1: handled via the LHC-OPN (**O**ptical **P**rivate **N**etwork) consisting of dedicated 10 Gbit/s network links
    - Distributes the recorded data for storage on tape at T1 sites
  - T1→T1: also handled via the OPN
    - Redistribute parts of the data produced during rereconstruction
  - T1→T2: handled via national high speed network links
    - Transfer datasets for analysis to T2 sites
  - T2→T1: handled via national high speed network links
    - Transfer produced MC to T1 for storage on tape

Computing Resources and site readiness



Resources currently available

Tier-0: 55 kHS06, 3 PB disk, 9 PB tape

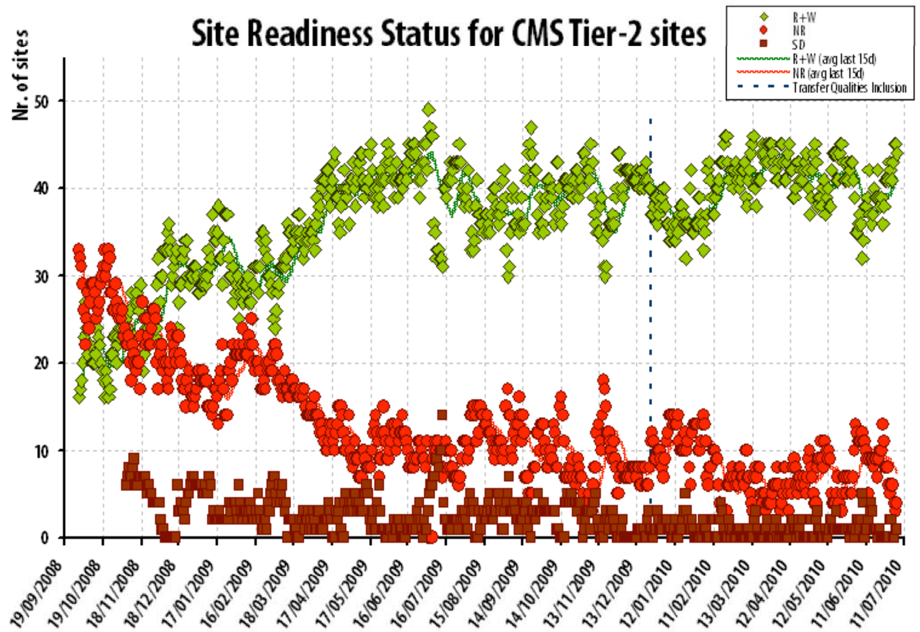
Tier-1: 100 kHS06, 11 PB disk, 20 PB tape

Tier-2: 192 kHS06, 12PB disk

Excellent site readiness

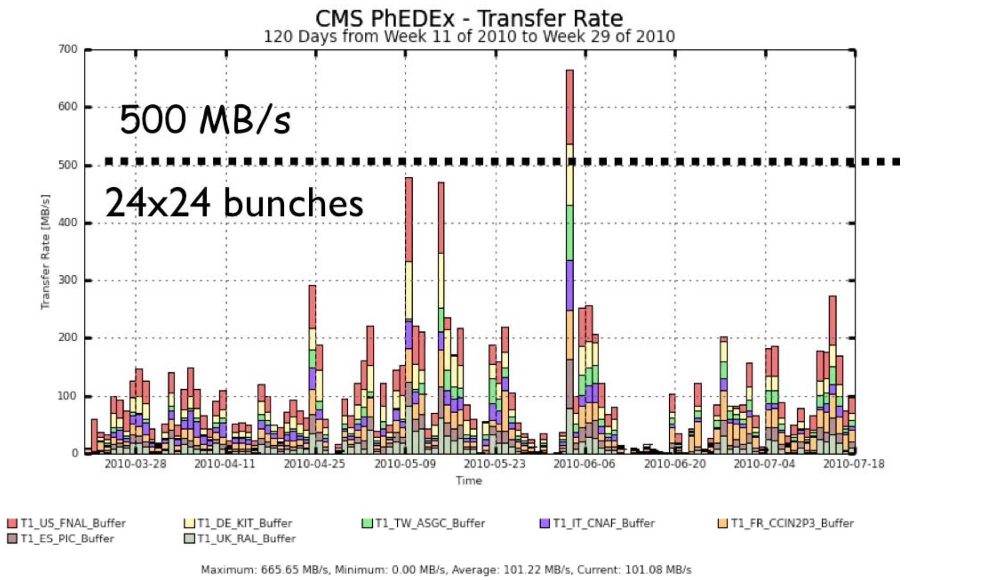
Key ingredient for successfull operations

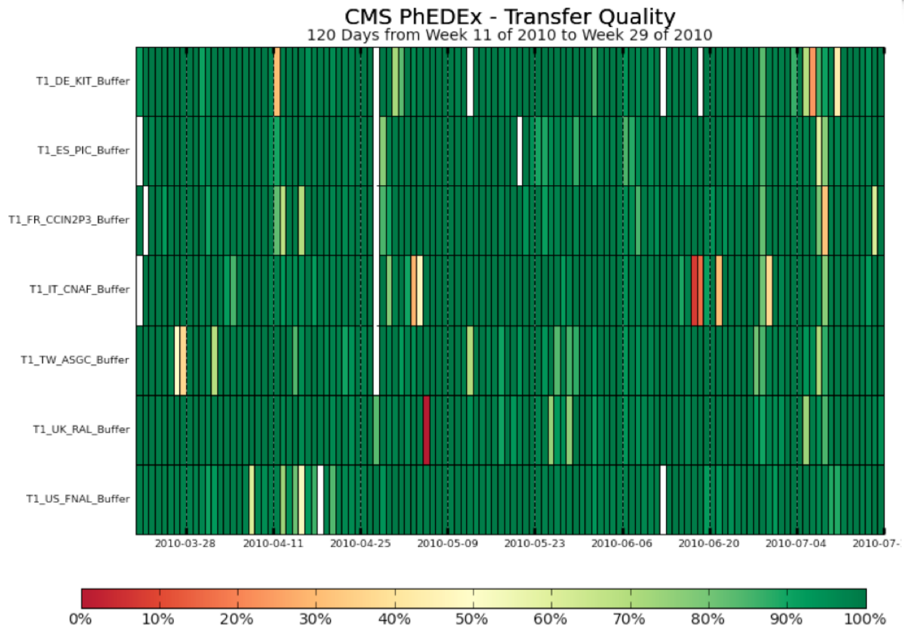
Close relationship with sites through contact person and data manager

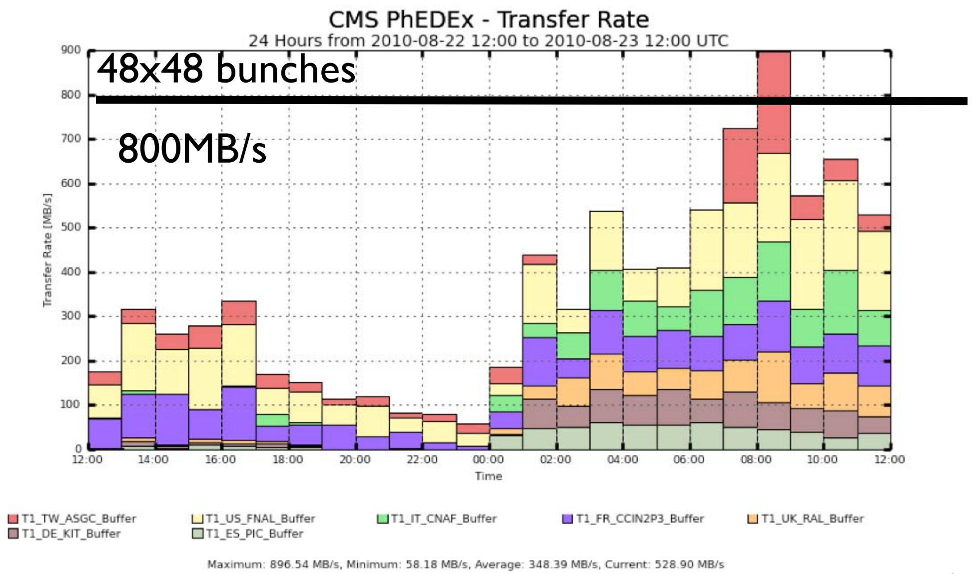


Data Transfer from CERN to Tier-1's

- Resources provisioned for steady data stream from Tier-0 to Tier-1's
- Nice peaks from the fills (good balance on Tier-1s
- Very good transfer quality

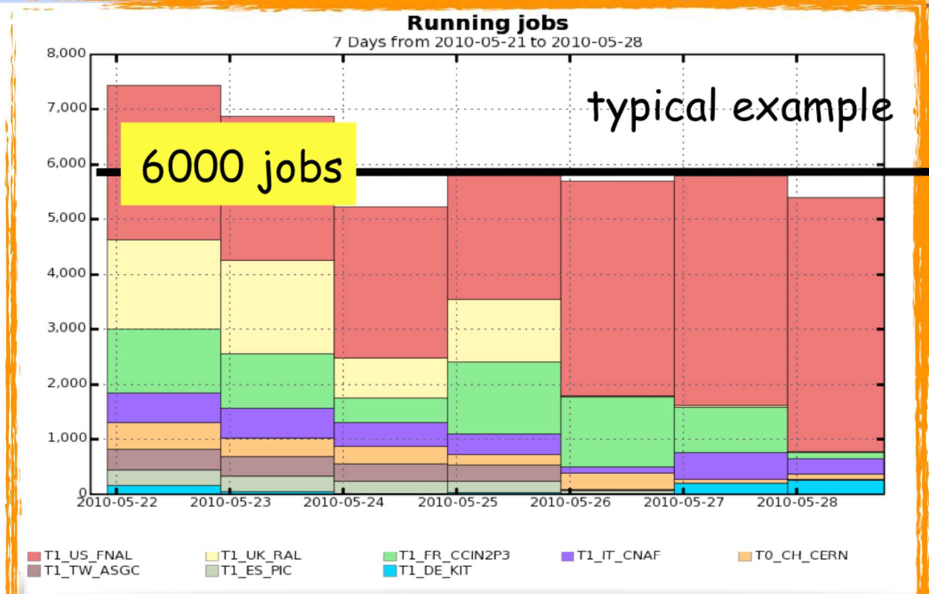


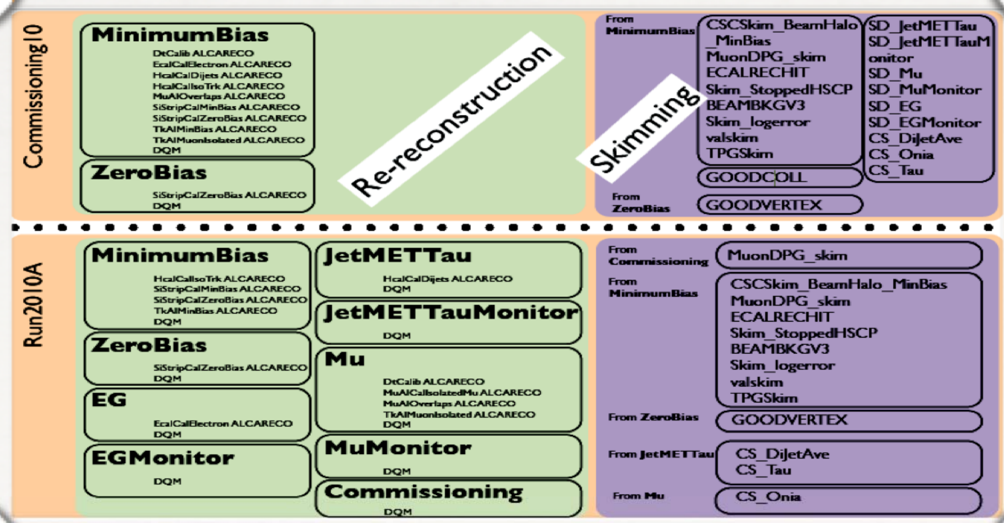




Central Processing @ Tier-1

- All Tier-1 sites used in production
- Upon arrival at Tier-1's, data is being processed and stored on tape
- Prompt skimming
  - Produce small datasets based on trigger selection or reconstructed objects
  - Fully automated system
- Reprocessing of data and MC
  - Improved software, calibration and alignment
  - ~ 10 data reprocessing passes for 7 TeV (up to now).
  - 3 MC reprocessing passes for 7 TeV



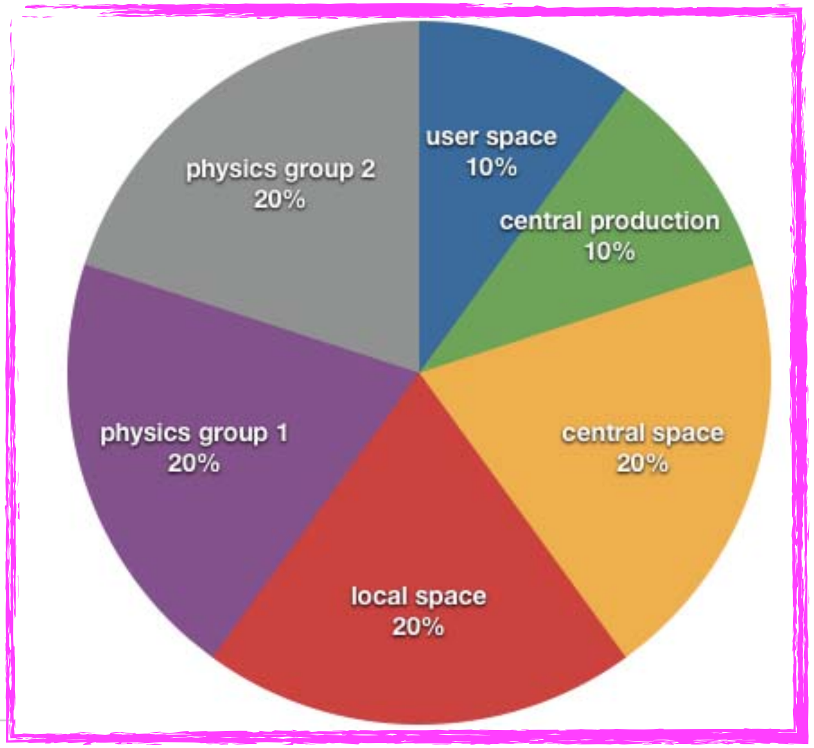


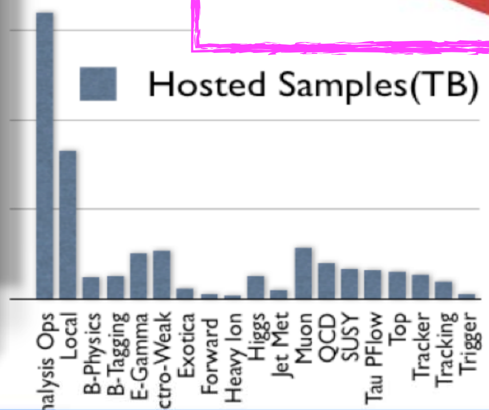
Primary Datasets

Data Distribution for Analysis

- Data distribution to Tier-1 organized centrally to balance resource utilization.
  - Jobs go where the data is
- Data storage serves as temporary buffer
  - Refresh with hot datasets
- Data distribution on Tier-2 organized
  - Centrally (Analysis Operations)
  - By physics groups
  - By local users

Tier-2 storage breakdown (typical example)



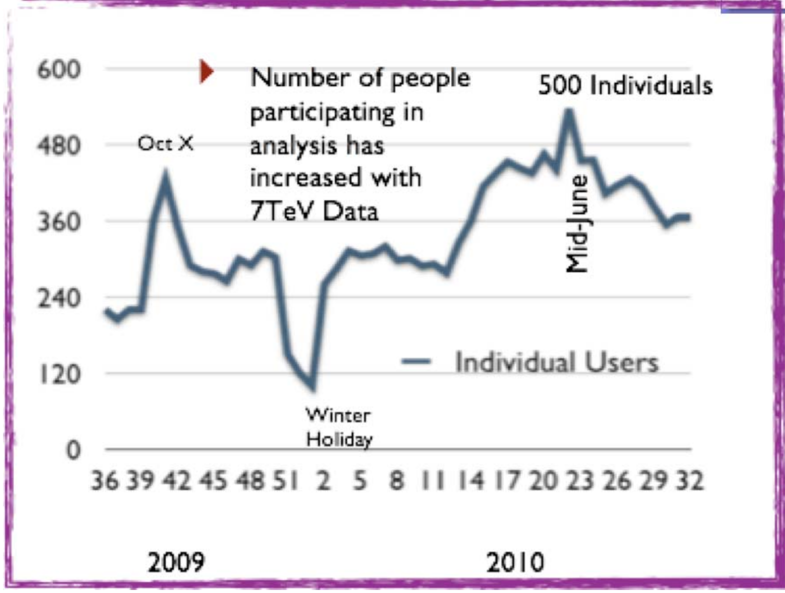


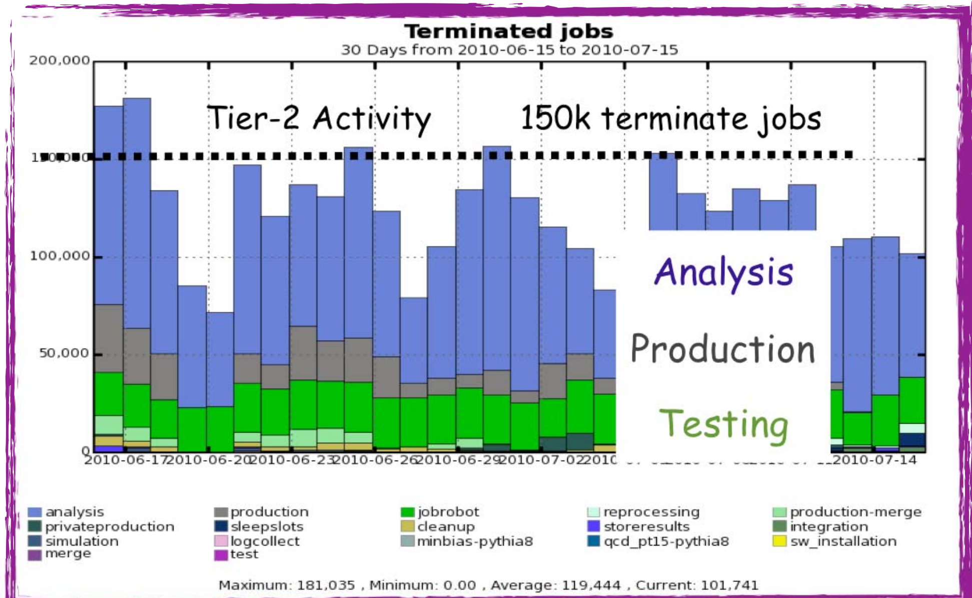
Analysis Activities @ Tier-2/3's

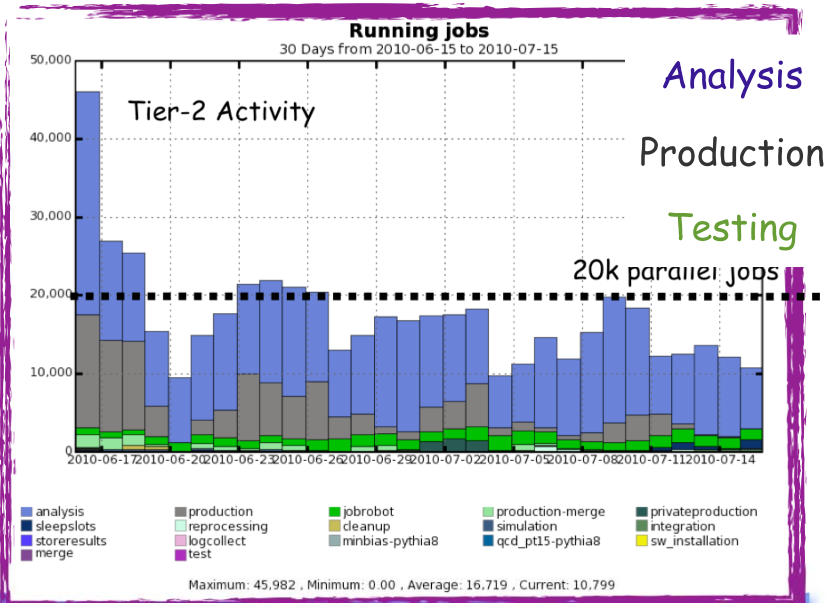
500 individual CMS users active using grid resources

Maximum reached in preparation for ICHEP

Tier-2 resource usage currently dominated by analysis activities

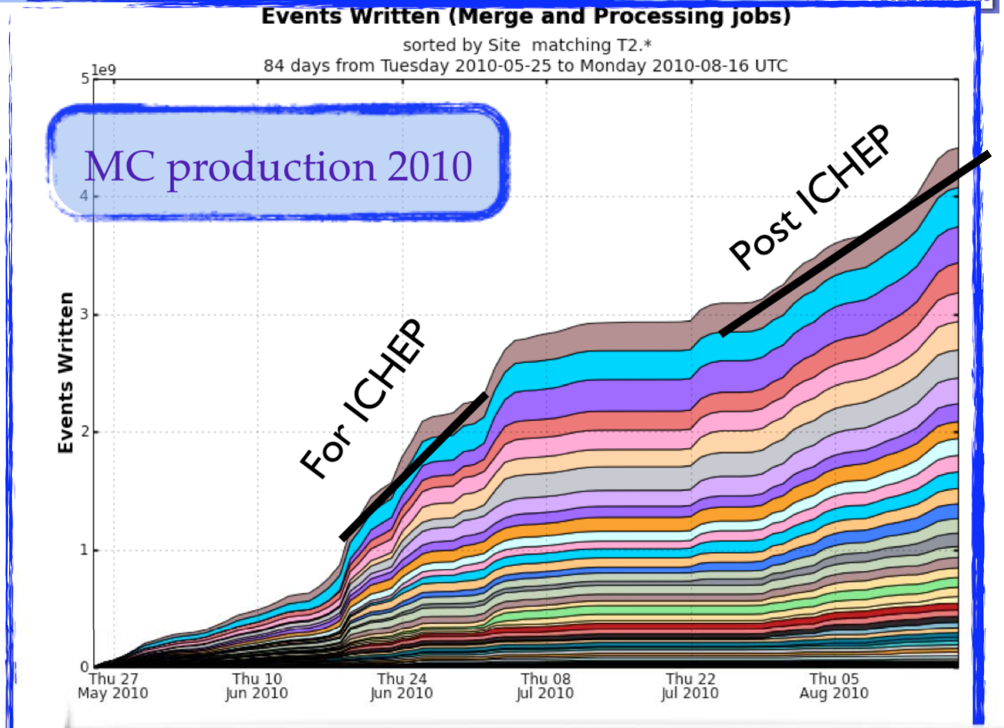


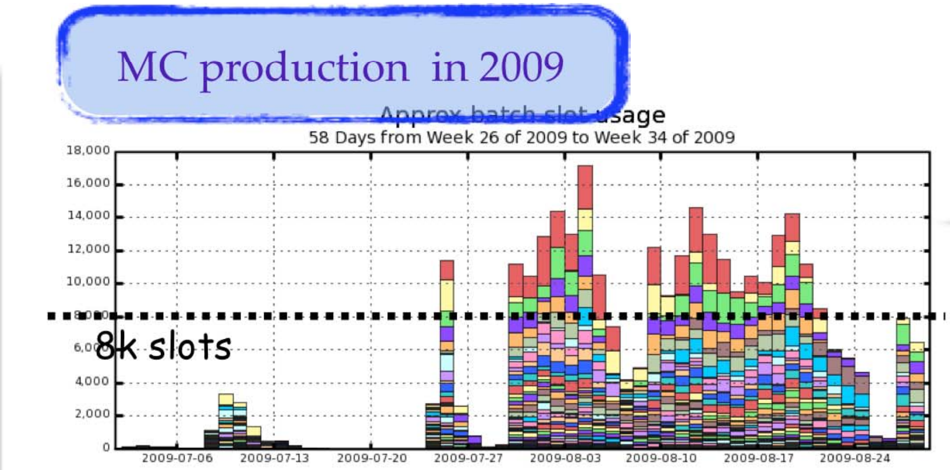




Monte Carlo Production

Successfully exercised for years
64 Tier-1/2/3 sites participating
MC production preparation for 7TeV data started in Summer 2009
Multiple production validation cycle





Before ICHEP: Mostly "Data-like" MC production in 2010 (MinimumBias & low-Pt QCD)

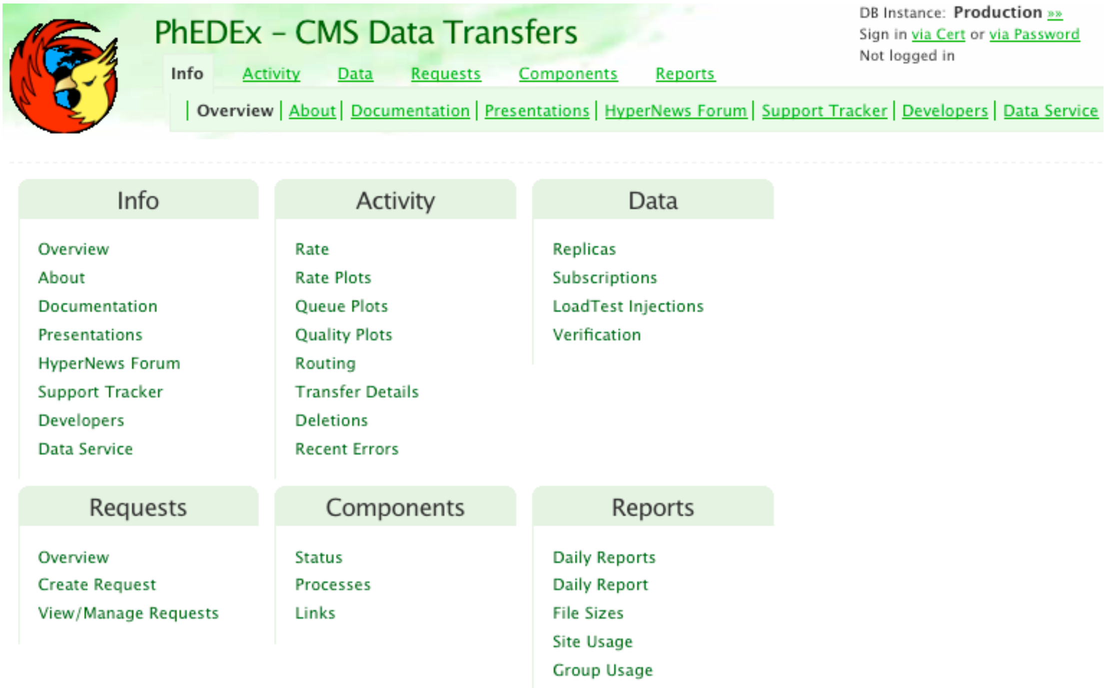
Next generation of CMSSW simulation starting. Large Scale sample with Pile-up to start

How user interact with

CMS Computing Infrastructure

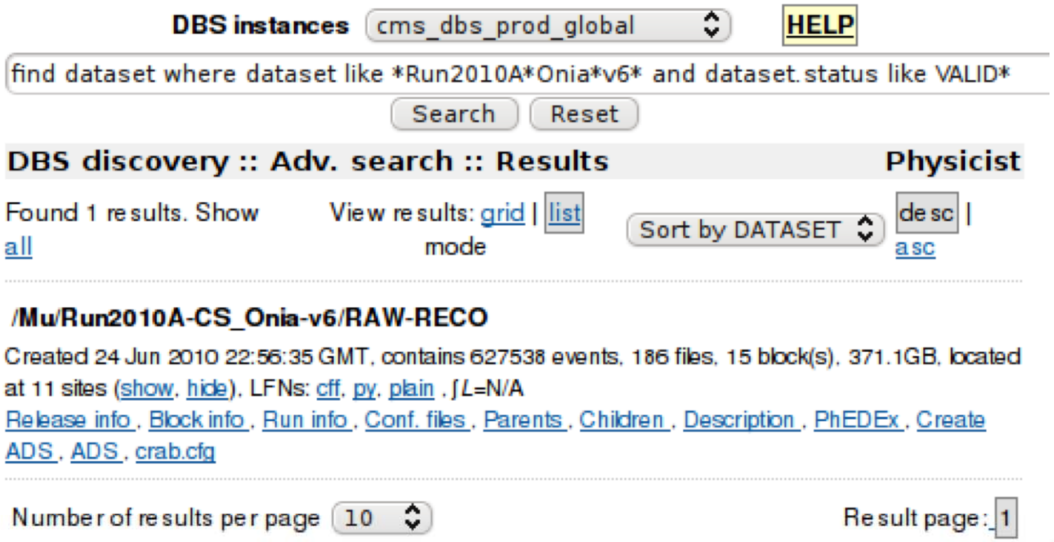
Transfer tool: PhEDEx

- PhEDEx is CMS' tool to request and manage data transfers
  - http://cmsweb.cern.ch/phedex
  - Every user can request the transfer of a data sample to a T2 site for analysis
  - Every T2 site (also the T1 sites and the T0) have data managers which approve or disapprove transfer requests according to global policies and available storage space



Dataset Bookkeeping system (DBS)

- DBS handles to bookkeeping of datasets
  - https://cmsweb.cern.ch/dbs_discovery
- A dataset name is composed of:
  - **/<primary dataset name>/<processed dataset name>/<data tier>**
    - **Primary dataset name**: specifies the physics content of the sample
    - **Processed dataset name:** specifies the processing conditions and data taking or MC production period, for Data: "<AcquisitionEra>-<FilterName>-<ProcessingVersion>"
    - **Data tier**: specifies the format of content of the files (RAW, RECO, AOD, …)
- Primary tool to look up and discovery datasets and their location on the T2 level for your analysis



GRID submission tool: CRAB

- CMS Remote Analysis Builder
  - https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrab
- Enables every user to send her/his analysis code to the T2 sites to process stored data and MC samples
- Represents a wrapper to the GRID tools used to execute jobs on the GRID

CMS Remote Analysis Builder - CRAB

Contents:

Helper Utilities

Finding Code

How can we know where an object is defined?

- If you already know where to look, you might use cvs browser:

  http://cmssw.cvs.cern.ch/cgi-bin/cmssw.cgi/CMSSW

- In all other 99% of the cases you might use lxr browser:

  http://cmslxr.fnal.gov/lxr/

Handling Source Code

- To add a package from the cms repository type:

```
addpkg PhysicsTools/Utilities [tag]
```

- If no tag is given the default one from the release is taken
- List which packages are in your release area:

```
showtags –r
```

```
Test Release based on: CMSSW_3_8_x
Base Release in: /afs/cern.ch/cms/sw/slc5_ia32_gcc434/cms/cmssw/
CMSSW_3_8_x
Your Test release in: /afs/cern.ch/user/d/decosa/scratch0/CMSSW_3_8_x
--- Tag ---     --- RelTag --- -------- Package --------
V05-09-14-02   V05-09-14-00   DataFormats/PatCandidates
V07-13-15-05   V07-13-15-03   PhysicsTools/PatAlgos
---------------------------------------
total packages: 2 (2 displayed)
```

How to check what is going on?

In CMSSW a Service module, called 'Tracer', helps us giving trace of each step of the processing → this is a clean way to understand what's happening

```
process.trace = cms.Service('Trace')
process.p = cms.Path(... + trace + ...)
```

The output:

```
++++source
 Begin processing the 1st record. Run 1, Event 1, LumiSection 1 at 09-Sep-2008  10:30:22 CEST
 ++++finished: source
 ++++ processing event:run: 1 event: 1 time:5000000
 ++++++ processing path:generation_step
 ++++++++ module:randomEngineStateProducer
 ++++++++ finished:randomEngineStateProducer
 ++++++++ module:VtxSmeared
 ++++++++ finished:VtxSmeared
```

Support and documentation

- The WorkBook:

  https://twiki.cern.ch/twiki/bin/view/CMS/WorkBook

- The SWGuide:

  https://twiki.cern.ch/twiki/bin/view/CMS/SWGuide

- LXR:

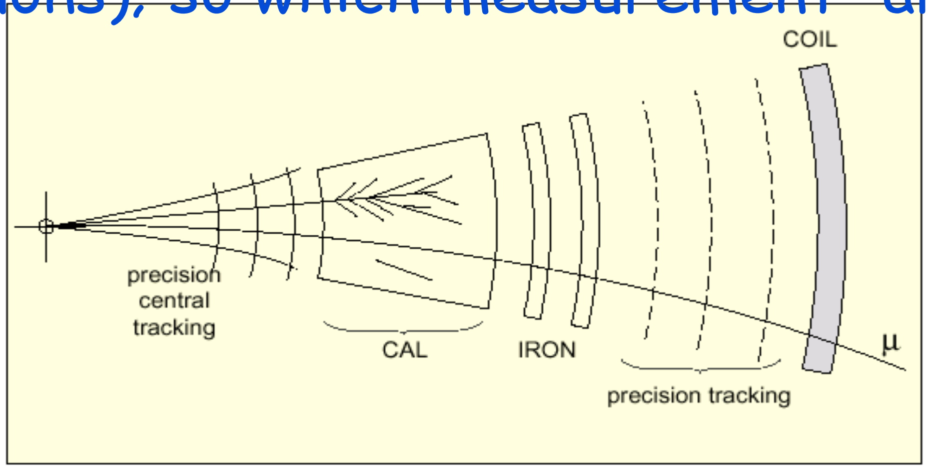
  http://cmslxr.fnal.gov/lxr/

- PAT Tutorial

https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookPATTutorial

- Many, many hypernews lists (at least one for each areas Simu, Reco, Alca, Physics Tools...)

THE issue: measure momenta of charged particles (e.g. muons); so which measurement "architecture"?

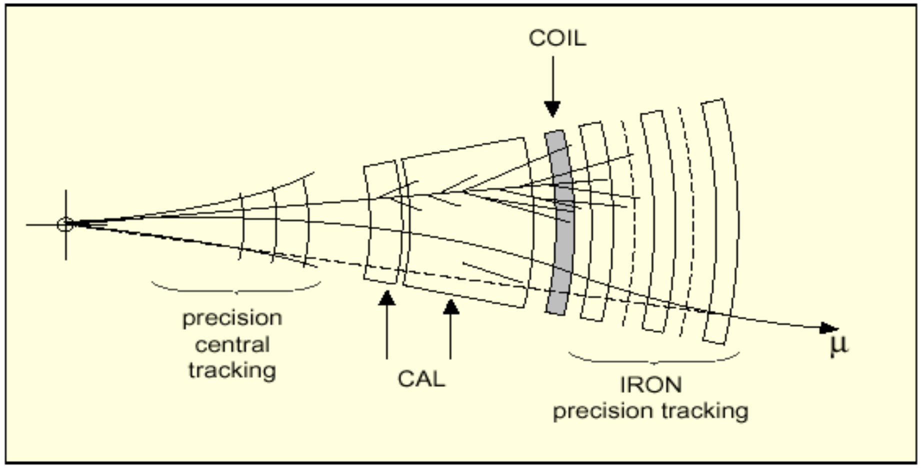


ATLAS

Standalone p measurement; safe for high multiplicities;

Air-core torroid

Property: σ flat with η



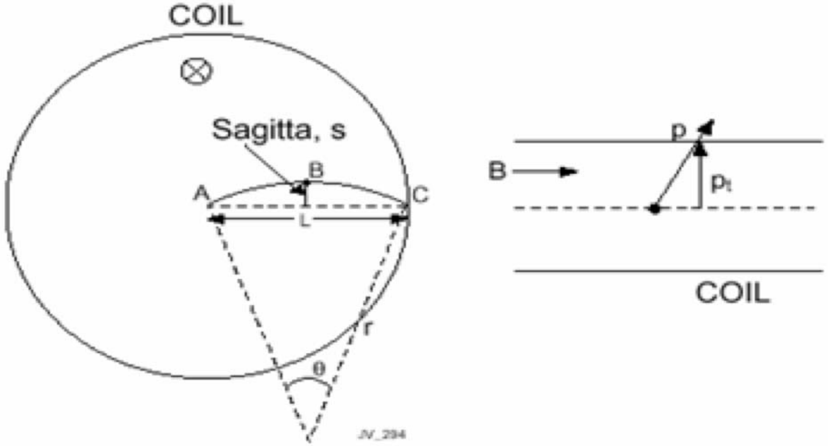
CMS

Measurement of p in tracker and B return flux; Iron-core solenoid Property: muon tracks point back to vertex

Tracking Momentum Measurements

Quick reminder: $p_t = 0.3Br$

Need high BL^2 or small σ_s:



In practice, measure s, not r

$$\sin(\theta/2) = \frac{L}{2r} \Rightarrow \theta \approx \frac{L}{r} = \frac{0.3BL}{p_t}$$

$$s = r - r\cos(\theta/2) \approx r\left[1 - \frac{1}{2}\frac{\theta^2}{4}\right] = \frac{r\theta^2}{8} \approx \frac{0.3BL^2}{8p_T}$$

$$\frac{\sigma(p_t)}{p_t} = \frac{\sigma_s}{s}$$

For a detector with N sensitive layers equally spaced

$$\frac{\sigma(p_t)}{p_t} \approx \sqrt{\frac{720}{N+4}}\,\sigma_x \frac{p_t}{0.3BL^2}$$

High p_t

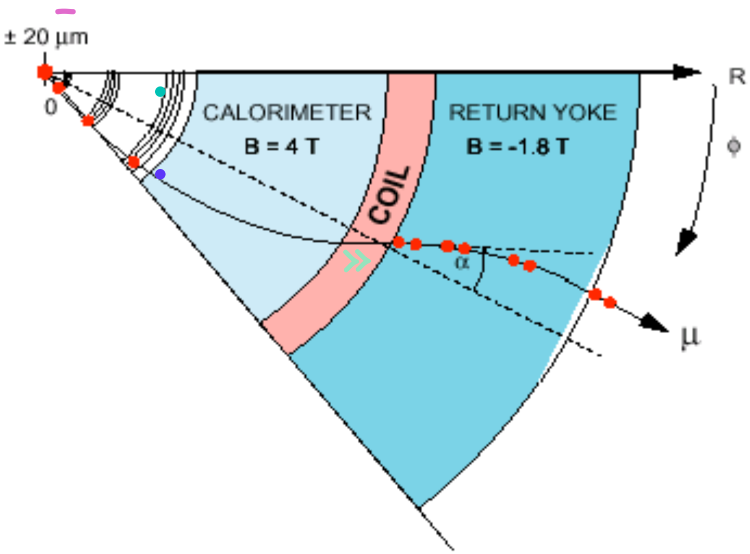
The dp/p increases linearly with p from about 30 Gev. At lower p it is constant.

$$\left.\frac{\sigma(p_t)}{p_t}\right|_{ms} = \frac{0.05}{B\sqrt{LX_0}}$$

Low p_t

Choice of magnet

Solenoid:

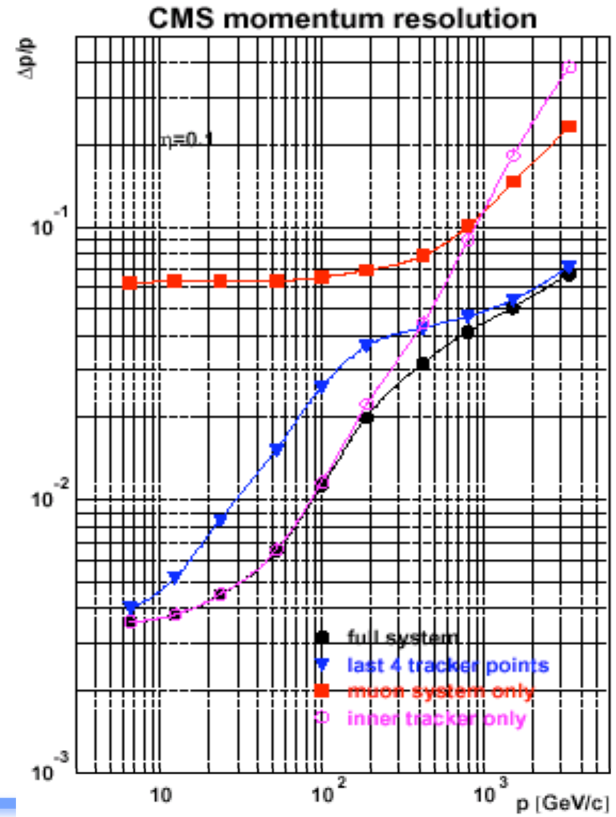


Bending in transverse plane

Use 20μm beam spot

BUT: 4T brings problems

(e.g. cannot use PM tubes)



Iron-core → multiple scattering

- Tracking in magnetized iron:

$$\frac{\Delta p}{p} = \frac{40\%}{B\sqrt{L}}$$

- BUT measurement much better when combined with the tracker

Back-up slides

More slides