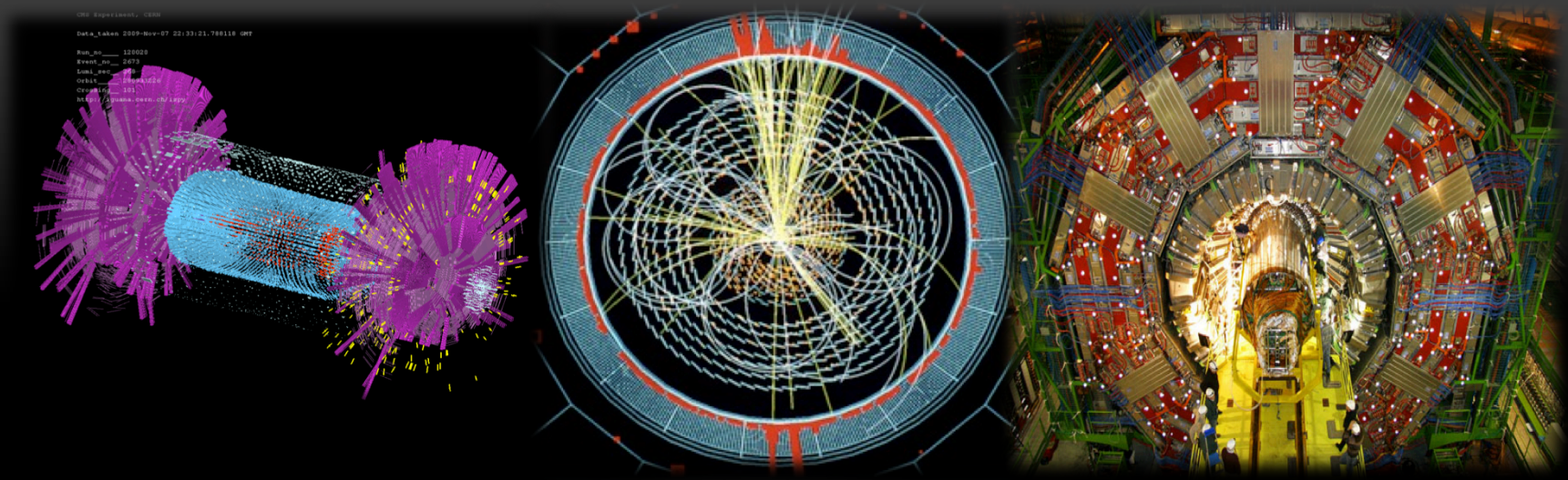


EVENT GENERATION IN CMS

Altan CAKIR

Deutsches Elektronen-Synchroton (DESY)

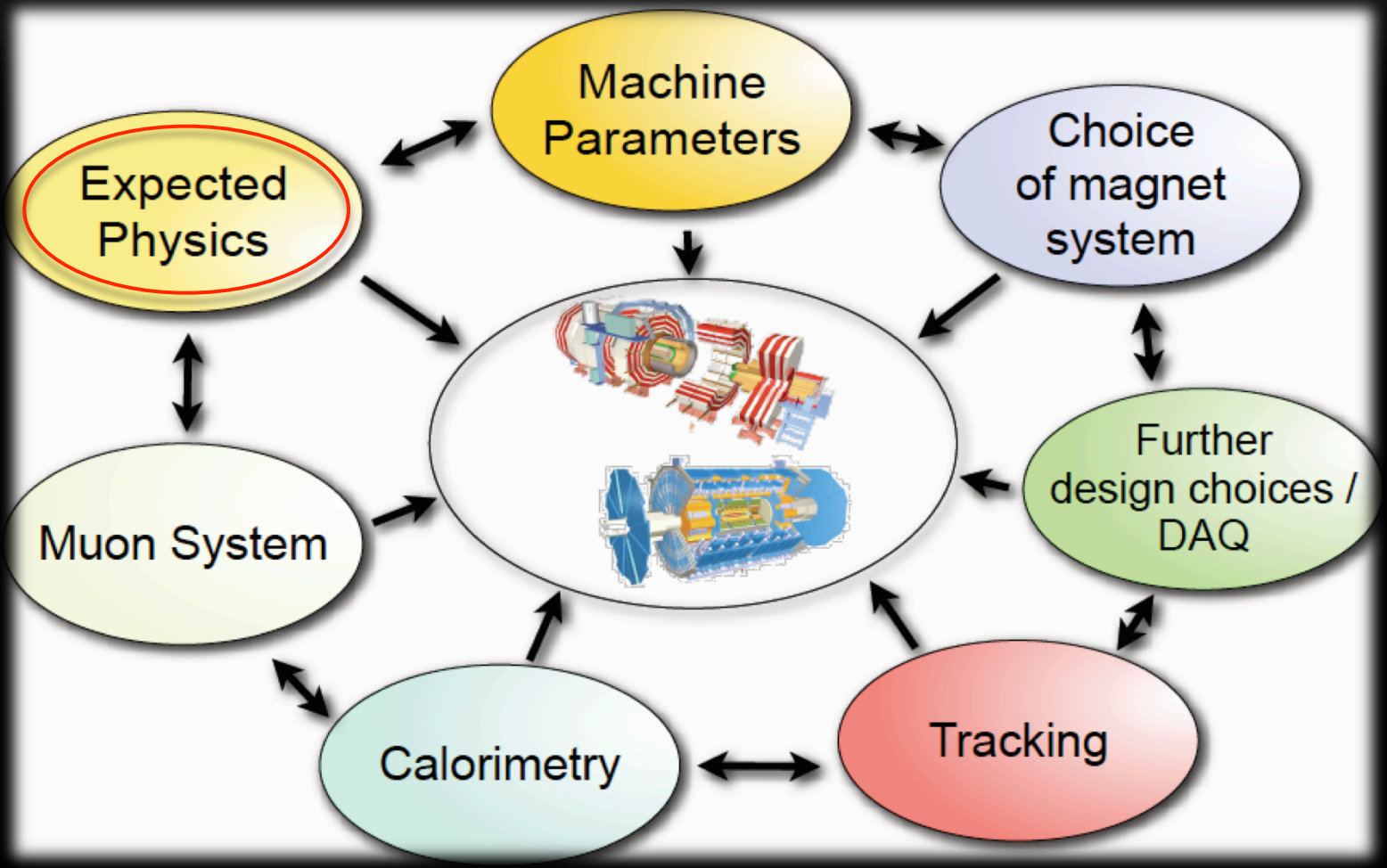
17/18 March 2011

DESYMC2011

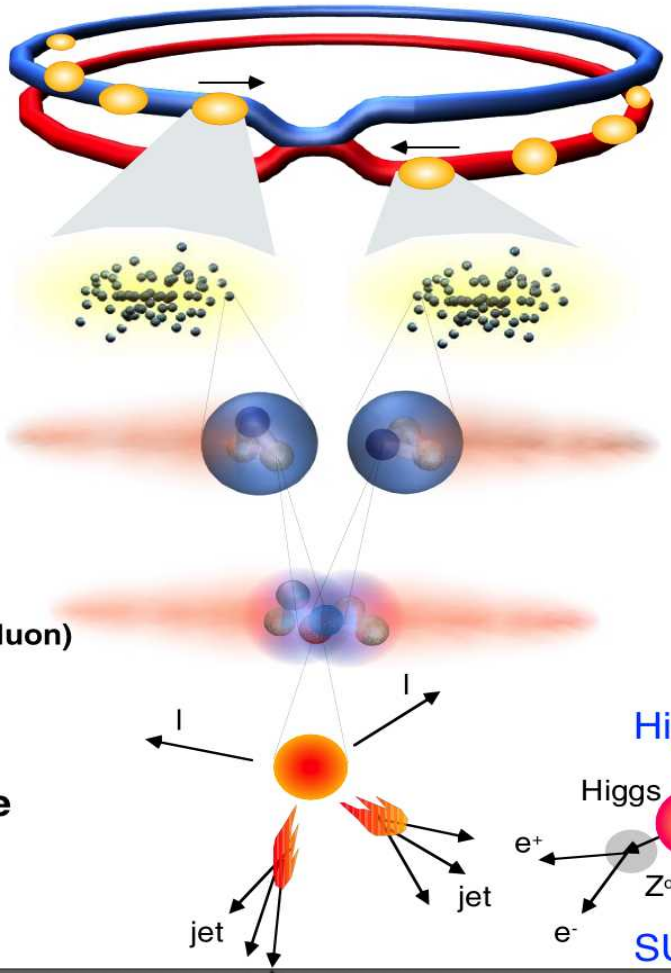
OUTLINE

- Design your experiment at the LHC - **CMS** detector
- Offline Tools – **CMSSW** Introduction
- Physics tools in CMSSW
- MC Simulations
- Detector simulations
- New Physics? How to simulate?

DESIGNING YOUR EXPERIMENT



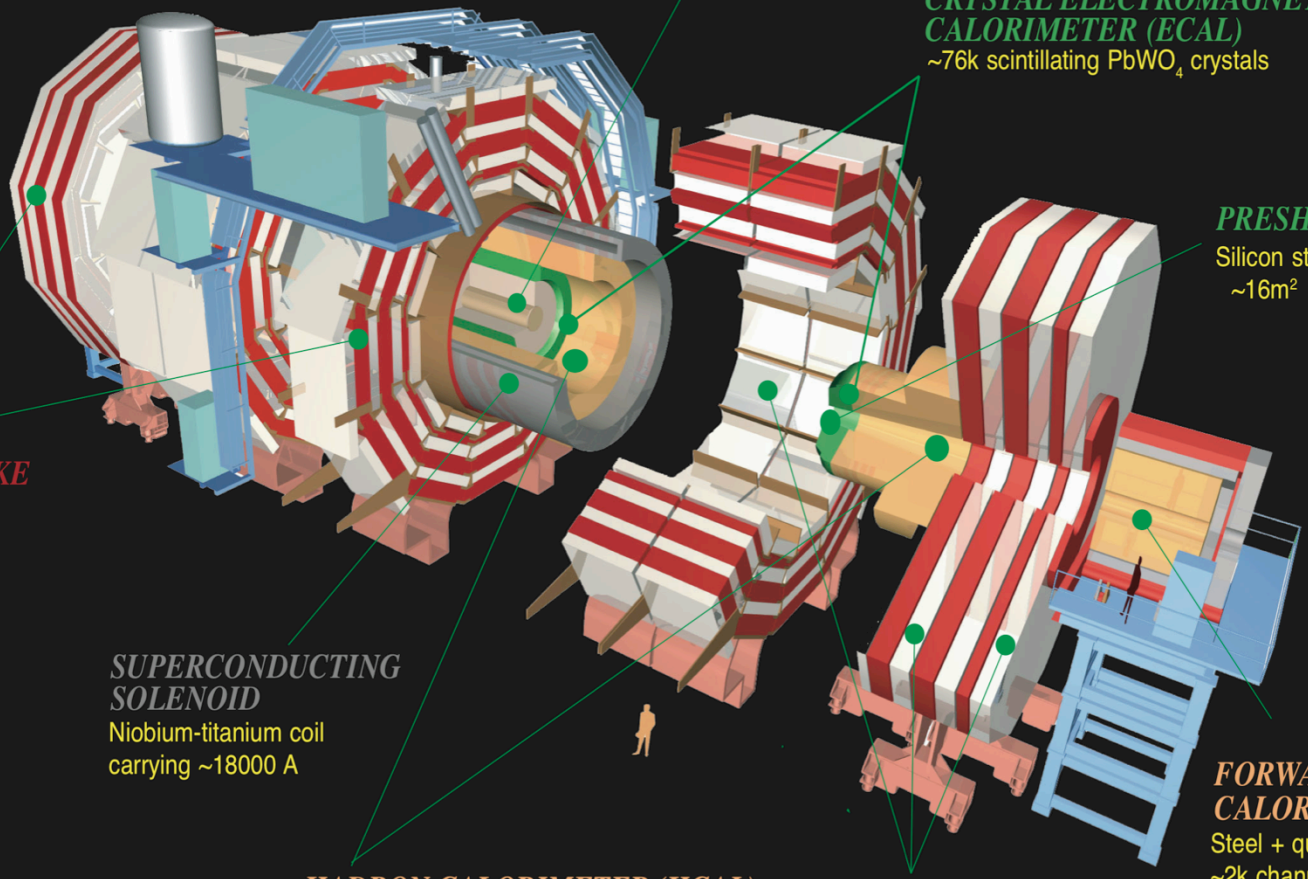
FROM BUNCH TO PARTICLE



Proton-Proton	2835 bunch/beam
Protons/bunch	10^{11}
Beam energy	7 TeV (7×10^{12} eV)
Luminosity	10^{34} cm^{-2} s^{-1}
Crossing rate	40 MHz
Collisions ≈	10^{7} - 10^{9} Hz

Selection of 1 in 10,000,000,000,000

CMS Detector



SILICON TRACKER
Pixels (100 x 150 μm²)
~1m² ~66M channels
Microstrips (80-180μm)
~200m² ~9.6M channels

CRYSTAL ELECTROMAGNETIC CALORIMETER (ECAL)
~76k scintillating $PbWO_4$ crystals

PRESHOWER
Silicon strips
~16m² ~137k channels

STEEL RETURN YOKE
~13000 tonnes

SUPERCONDUCTING SOLENOID
Niobium-titanium coil
carrying ~18000 A

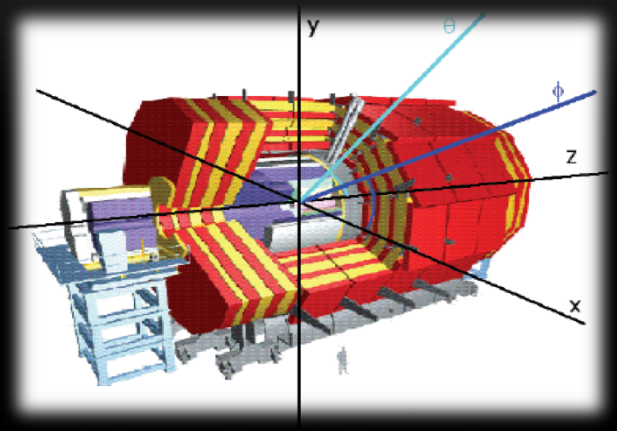
FORWARD CALORIMETER
Steel + quartz fibres
~2k channels

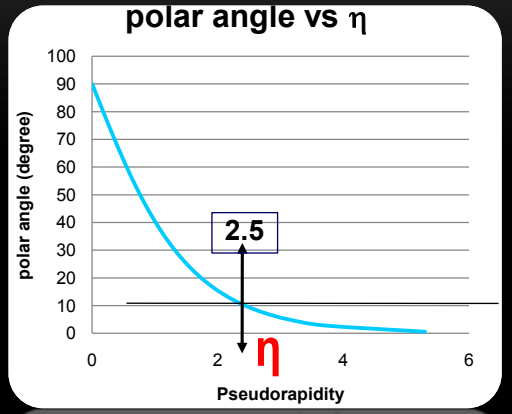
HADRON CALORIMETER (HCAL)
Brass + plastic scintillator
~7k channels

MUON CHAMBERS
Barrel: 250 Drift Tube & 480 Resistive Plate Chambers
Endcaps: 473 Cathode Strip & 432 Resistive Plate Chambers

Total weight	: 14000 tonnes
Overall diameter	: 15.0 m
Overall length	: 28.7 m
Magnetic field	: 3.8 T

CMS COORDINATES AND CONVENTIONS





- Azimuthal angle:

φ = azimuthal angle

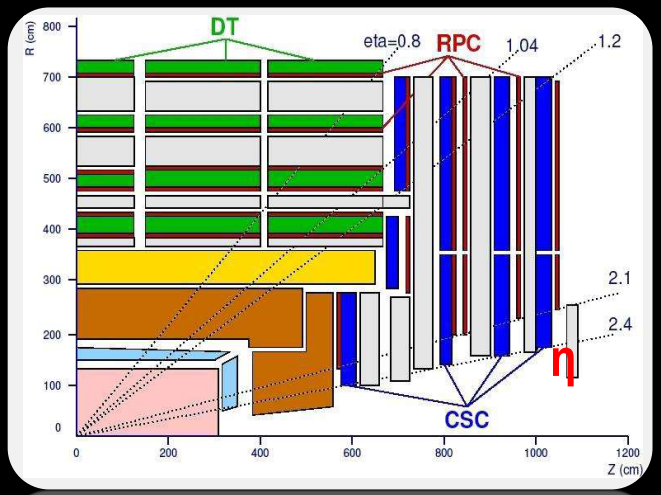
$-\pi < \varphi < \pi$

- Polar angle:

Θ = polar angle

$0 < \Theta < \pi$

Also pseudurapidity $\eta = -\ln[\tan(\Theta/2)]$



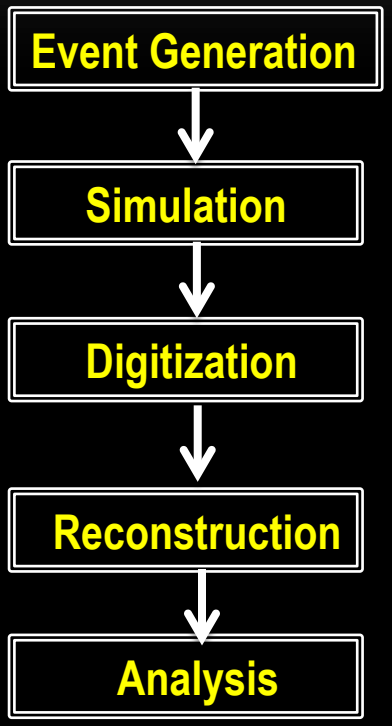
A spectrometer that covers 2π in azimuthal angle and down to 10^0 on each end in polar angle, covers 98% of the full solid angle. It will Accept the light decay products of heavy objects

$p_T = |p| \sin\Theta$

$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\varphi)^2}$

- Energy is measured in GeV, momentum in GeV/c and mass GeV/c^2
- Distance and position in cm
- Time in ns

ANALYSIS CHAIN



First step is to generate the events of your required physics problem, using event generators like Pythia, MadGraph, SOFTSUSY etc.

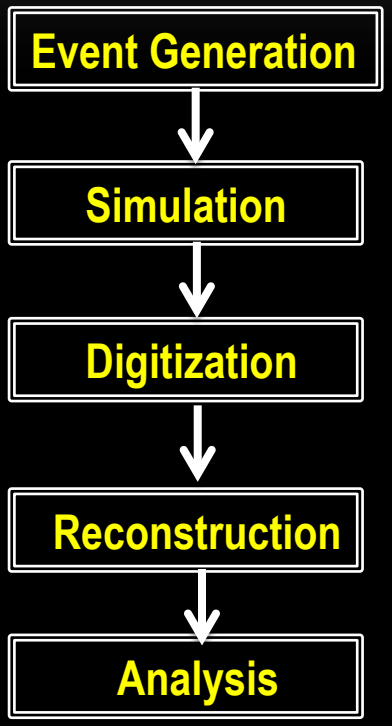
Once you are done with generation, detector effects are included

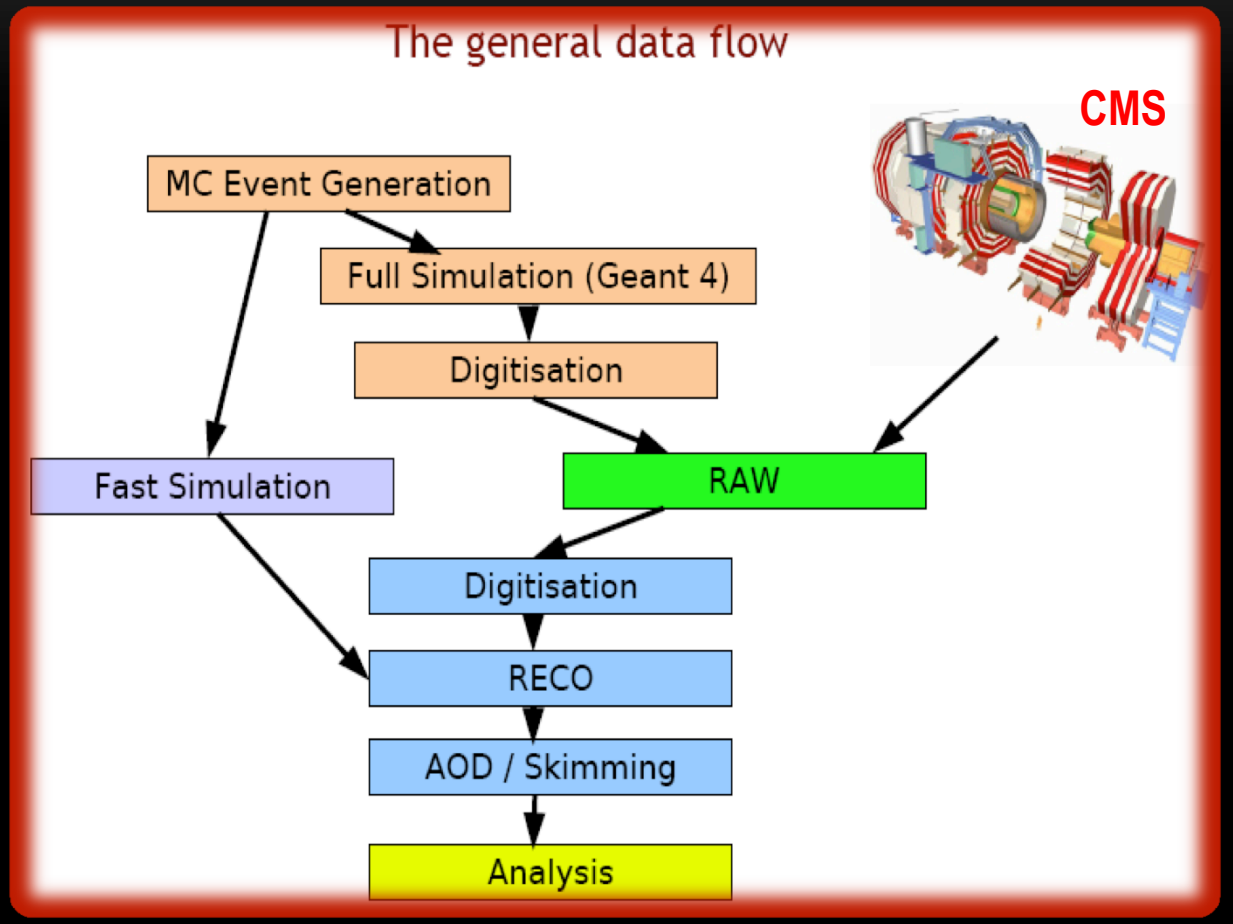
recHits, clusters etc.

Tracks, Electrons, Muons, jets etc.

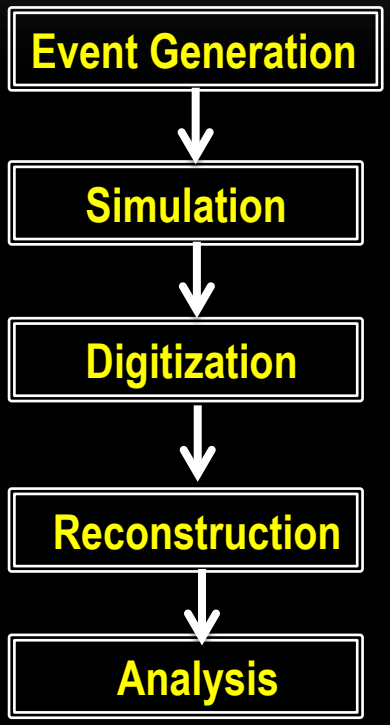
Analysis Code is written by the user

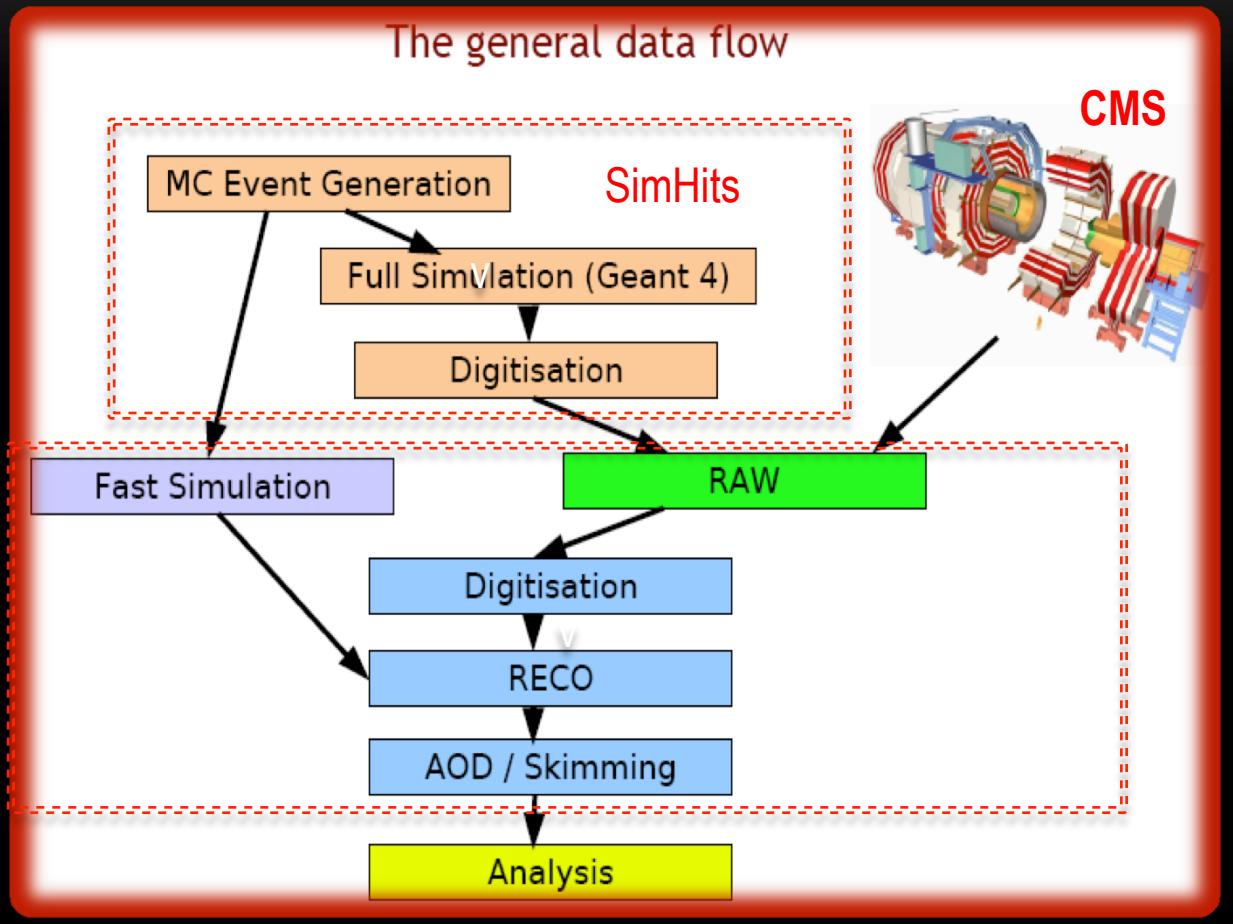
ANALYSIS CHAIN





ANALYSIS CHAIN





DATA STREAM

SimHits

- The simulation process begins with an event generator followed by the **Geant-based** detector simulation.
- The output of the simulation is a set of simulated **energy deposits** (with deposit times) in detector channels.
- These energy deposits are stored in the event as SimHits. The SimHits are labeled by DetIds.

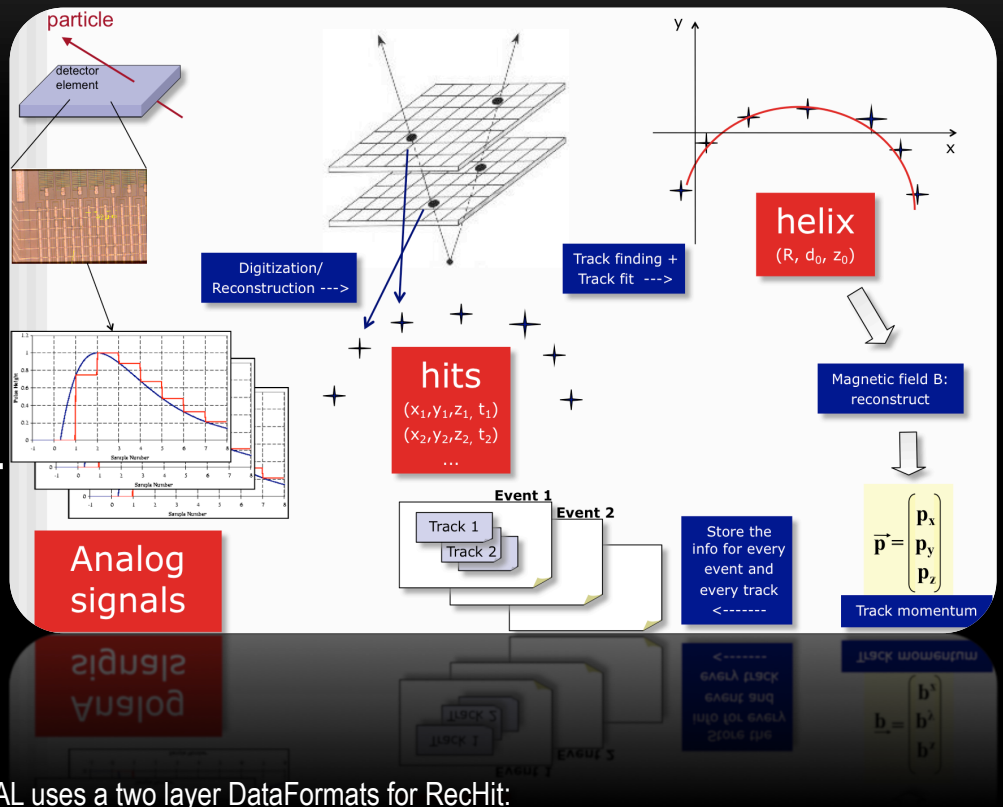
SimHits are converted into digis in the electronics simulation or "digitization process"

DigiHits

- **Digis** are the per-channel data from the detector. They are labeled by DetId. Digis show the pulse shape of the detector and are used to reconstruct the energy and time of the hit in the calorimeter.

RecHits

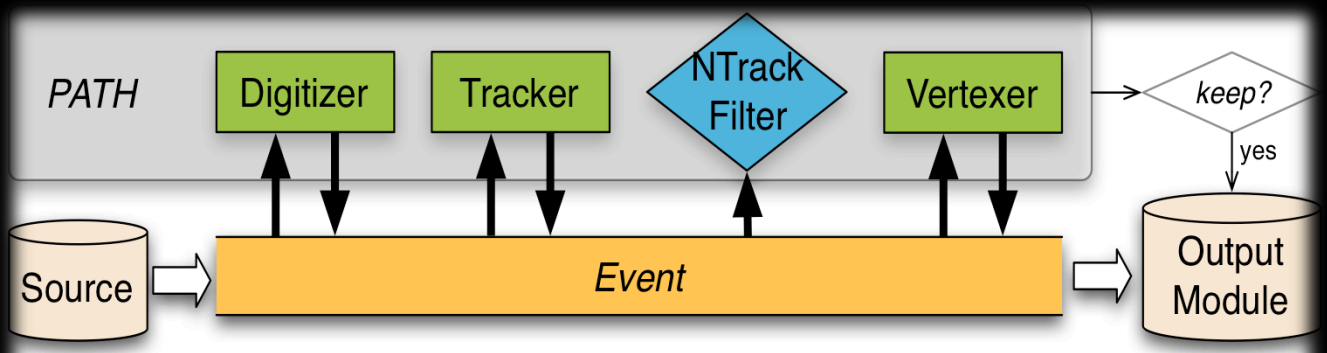
- **RecHits** are the energy and time for each hit (per channel). ECAL uses a two layer DataFormats for RecHit:
  - RecHits are the primary input for many higher-level
  - reconstruction tasks: electron/photon clustering …



EVENTS : THE EVENT DATA MODEL (EDM)

- Physically, an event is the result of a single readout of the detector electronics and the signals that will (in general) have been generated by particles, tracks, energy deposits, present in a number of bunch crossings.

- In software terms, an Event starts as a collection of the RAW data from a **detector** or **MC event**, stored as a single entity in memory, a C++ type-safe container called **edm::Event**. Any C++ class can be placed in an Event, there is no requirement on inheritance from a common base class. As the event data is processed, products (of producer modules) are stored in the Event as reconstructed (RECO) data objects.



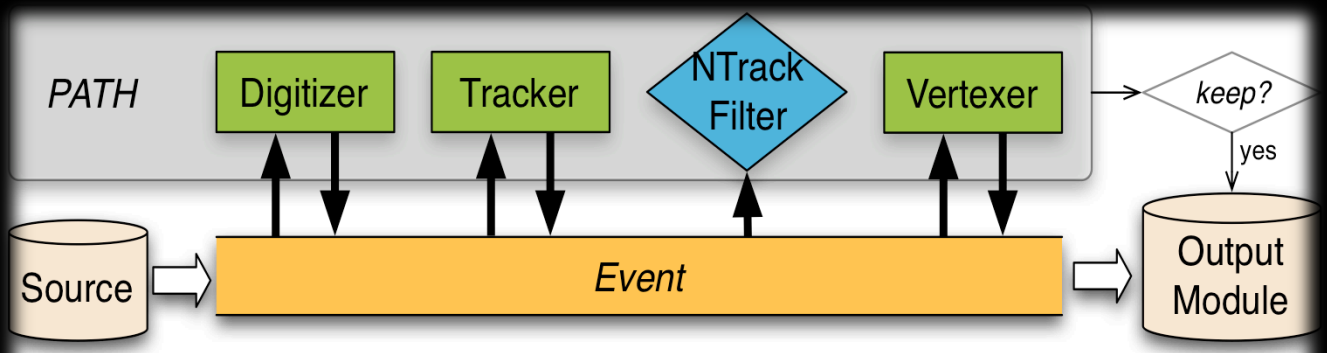
Communication among modules **only** via the event

- Source
- EDProducer
- EDFilter
- EDAnalyzer
- OutputModule
- Sequence
- Path
- EndPath
- Schedule

EVENTS : THE EVENT DATA MODEL (EDM)

- Physically, an event is the result of a single readout of the detector electronics and the signals that will (in general) have been generated by particles, tracks, energy deposits, present in a number of bunch crossings.

- In software terms, an Event starts as a collection of the RAW data from a **detector** or **MC event**, stored as a single entity in memory, a C++ type-safe container called **edm::Event**. Any C++ class can be placed in an Event, there is no requirement on inheritance from a common base class. As the event data is processed, products (of producer modules) are stored in the Event as reconstructed (RECO) data objects.



Communication among modules **only** via the event

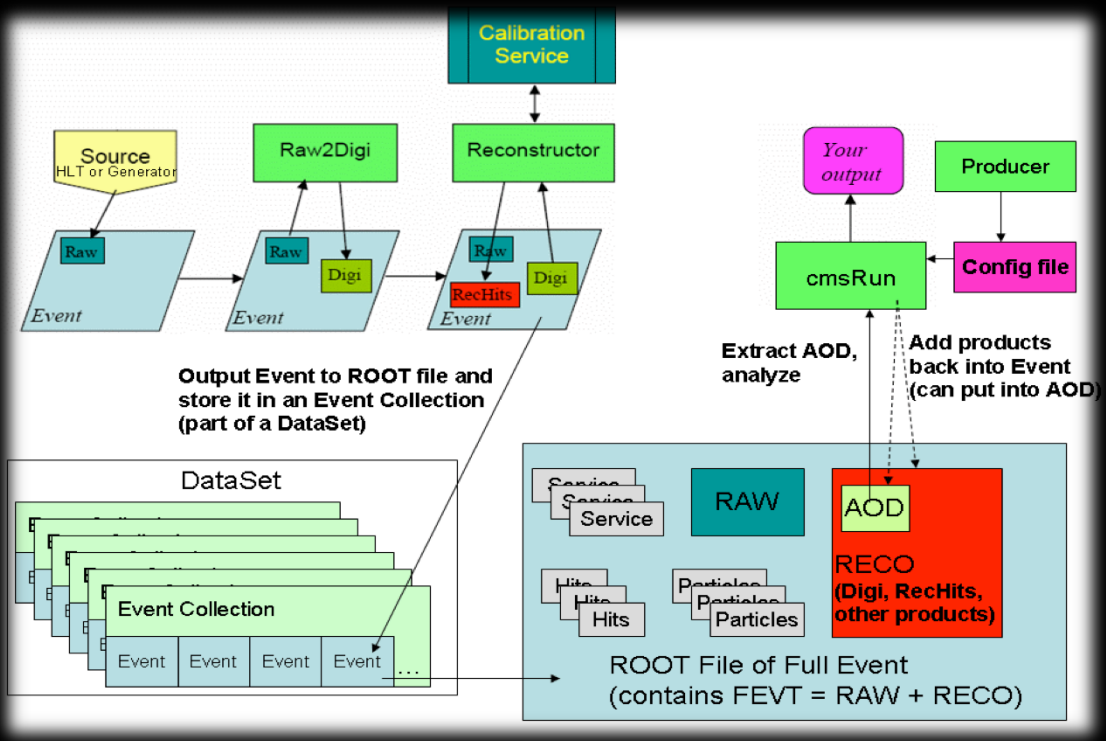
- Source
- EDProducer
- EDFilter
- EDAnalyzer
- OutputModule
- Sequence
- Path
- EndPath
- Schedule

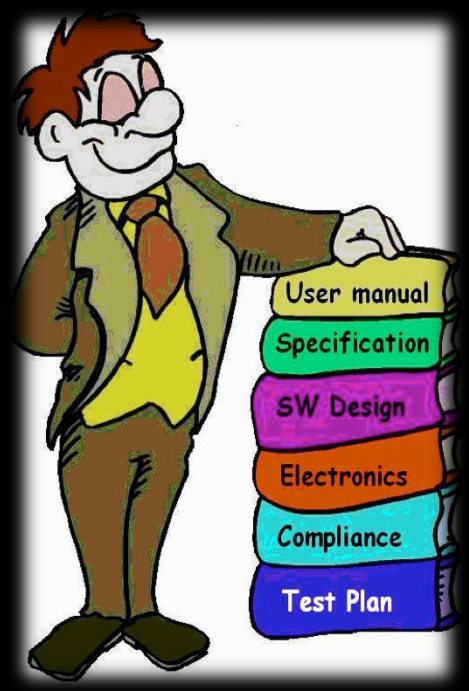
- EDAnalyzer
  - analyse objects from the event
- EDProducer
  - adds objects into the event
- EDFilter
  - can stop an execution path and put objects into the event
- EDLooper
  - For multi-pass looping over events (e.g. for alignment)
- OutputModule
  - Write events to a file. Can use filter decisions

EVENTS : THE EVENT DATA MODEL (EDM)

- Physically, an event is the result of a single readout of the detector electronics and the signals that will (in general) have been generated by particles, tracks, energy deposits, present in a number of bunch crossings.

- In software terms, an Event starts as a collection of the RAW data from a **detector** or **MC event**, stored as a single entity in memory, a C++ type-safe container called **edm::Event**. Any C++ class can be placed in an Event, there is no requirement on inheritance from a common base class. As the event data is processed, products (of producer modules) are stored in the Event as reconstructed (RECO) data objects.

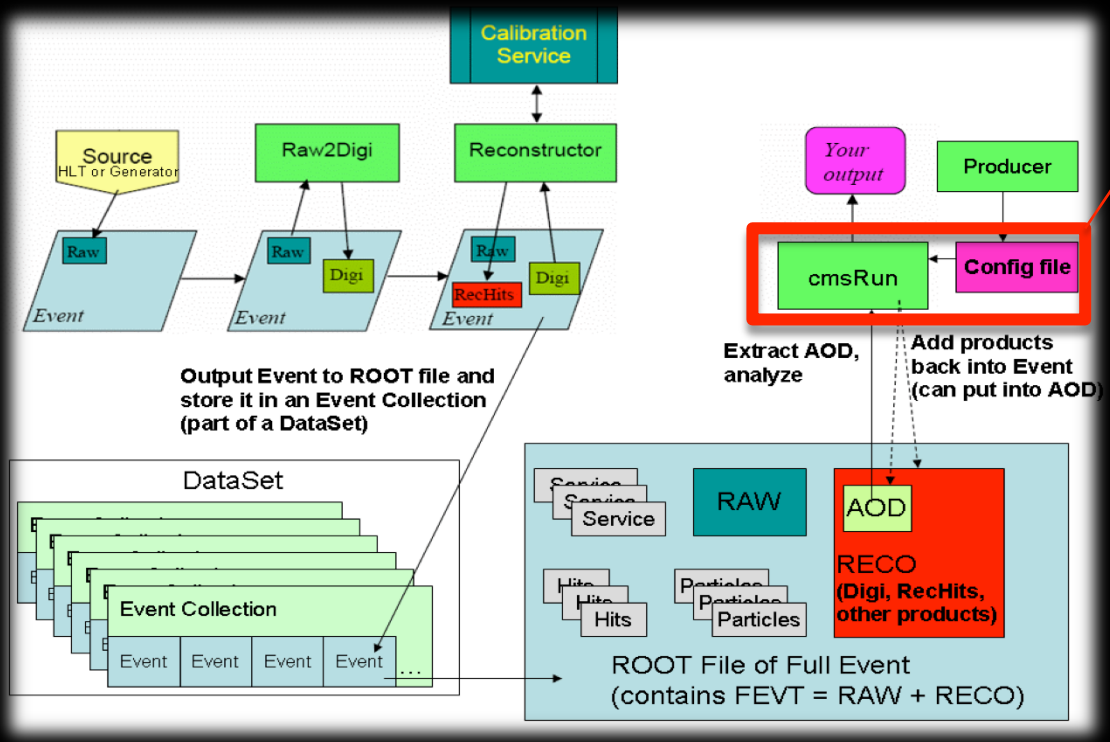




EVENTS : THE EVENT DATA MODEL (EDM)

- Physically, an event is the result of a single readout of the detector electronics and the signals that will (in general) have been generated by particles, tracks, energy deposits, present in a number of bunch crossings.

- In software terms, an Event starts as a collection of the RAW data from a **detector** or **MC event**, stored as a single entity in memory, a C++ type-safe container called **edm::Event**. Any C++ class can be placed in an Event, there is no requirement on inheritance from a common base class. As the event data is processed, products (of producer modules) are stored in the Event as reconstructed (RECO) data objects.



cmsRun <config file>.py

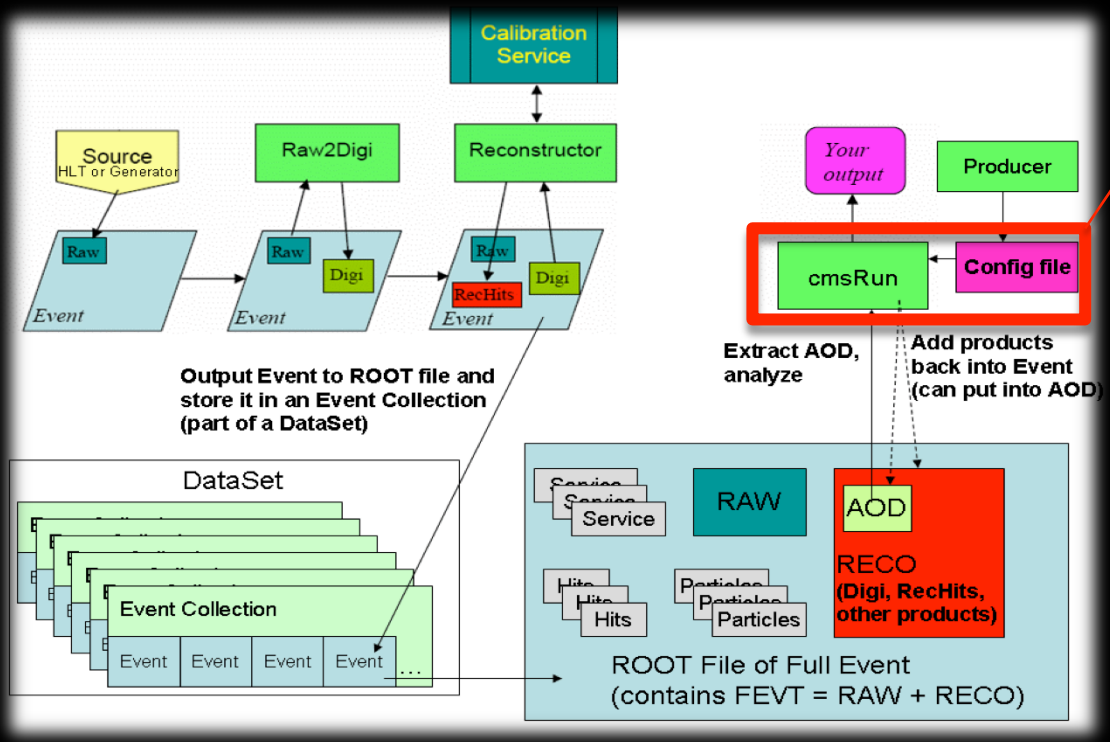
These includes

- Online data taking
- Online High Level Trigger
- **Monte Carlo Event**
- Detector Simulation
- Reconstruction
- Analysis

EVENTS : THE EVENT DATA MODEL (EDM)

- Physically, an event is the result of a single readout of the detector electronics and the signals that will (in general) have been generated by particles, tracks, energy deposits, present in a number of bunch crossings.

- In software terms, an Event starts as a collection of the RAW data from a **detector** or **MC event**, stored as a single entity in memory, a C++ type-safe container called **edm::Event**. Any C++ class can be placed in an Event, there is no requirement on inheritance from a common base class. As the event data is processed, products (of producer modules) are stored in the Event as reconstructed (RECO) data objects.



cmsRun <config file>.py

These includes

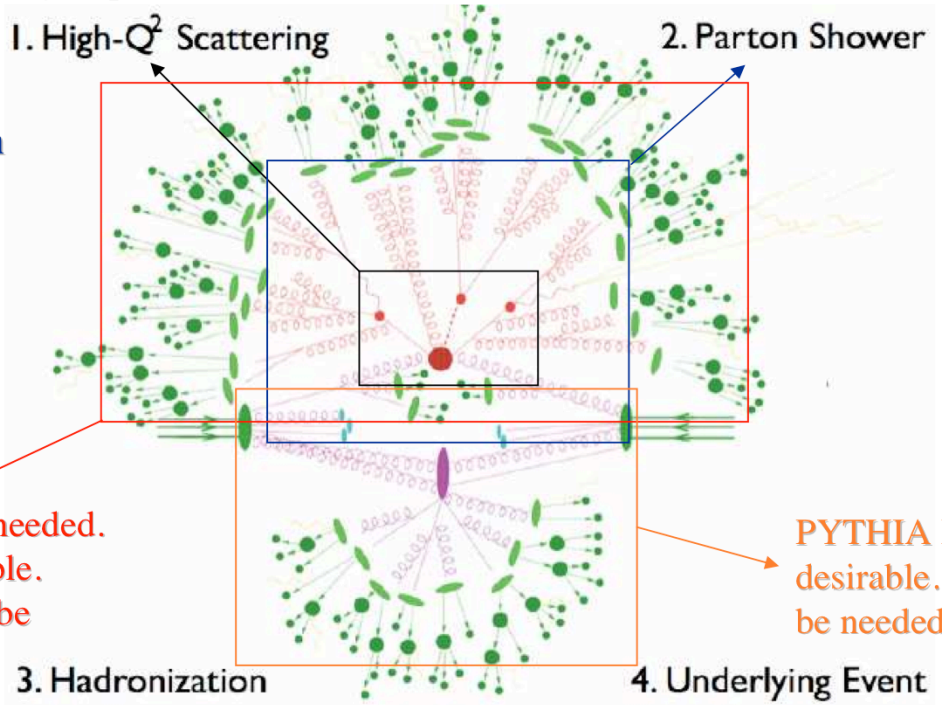
- Online data taking
- Online High Level Trigger
- **Monte Carlo Event**
- Detector Simulation
- Reconstruction
- Analysis

PHYSICS ANALYSIS IN CMS

Extra gluon emission described with ME at the highest possible order (+matching). Spin correlations needed.

Interface to PYTHIA needed. HERWIG very desirable. Tuning with data will be needed.

The MC description of LHC events is tremendously complex



Interface to PYTHIA needed. HERWIG very desirable. Tuning with data will be needed.

PYTHIA MPI. HERWIG/JIMMY desirable. Tuning with data will be needed.

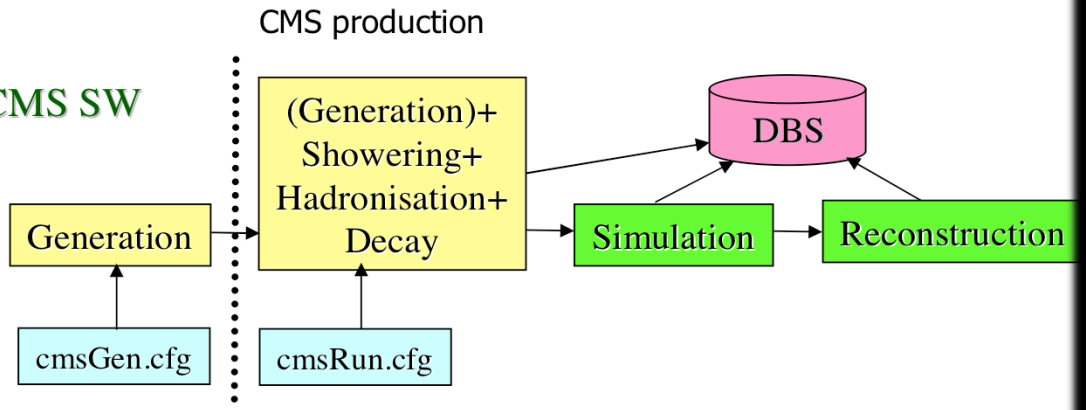
Other desirable features, from the experimentalist's viewpoint:

- output in the Les Houches standard format
- as much complete as possible coverage of SM phase space
- user friendly inclusion of new physics signals

MC GENERATORS IN CMS(SW)

- CMS have preference for the generators integrated in the experiment software, that can directly be used in production. Methods also exist to start from an externally produced parton level (LHE) file

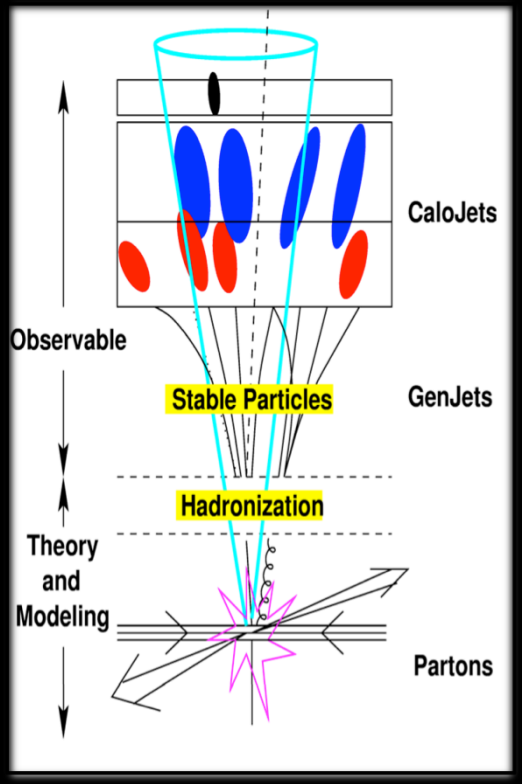
- The generator interfaced to the CMS SW must undergo:
  - ✓ a technical validation to show its ability to run standalone and in production
  - ✓ a physics validation wrt a similar content generator



- The physics program at the LHC is very rich:
  - ✓ pp General purpose: Pythia6, Herwig6, (Pythia8, Herwig++), Sherpa
  - ✓ pp HLO: Alpgen, MadGraph, Helac, Sherpa
  - ✓ pp NLO: MCatNLO
  - ✓ pp Others: CompHEP, TopRex, Phantom
  - ✓ Diffractive physics: Pomwig, Exhume, EDDE
  - ✓ Decayers: EvtGen, Tauola, Photos
  - ✓ Heavy Ions: Hydjet, Pyquen
  - ✓ Detector specific: Cosmic muons, particle guns, beam halo, beam-gas
  - ✓ New physics specific: Charybdis

MC SIMULATION :: PYTHIA6

Pythia is a general purpose event generator, containing theory and models for a number of physics aspects, including hard and soft interactions, parton distributions, initial and final state parton showers, multiple interactions, fragmentation ("Lund" model) and decays.



The two major software components are

- ✓ Pythia6GeneratorFilter - full event generation
- ✓ Pythia6HadronizerFilter - processing of parton-level event by an ME generator (MadGraph, Alpgen, SOFTSUSY)

Both components deliver the same output: HepMCProduct and GenEventInfoProduct (these are common across CMSSW interfaces to all multi-purpose generators).

Let us concentrate on Pythia simulation, first setting up CMSSW

SETTING UP CMSSW

- CMSSW runs on Scientific Linux (“SLC”)
  - You'll probably run it either on NAF here or lxplus at CERN
- CMSSW has new releases often
  - The most recent Monte-Carlo production is for 3_9_X data is being taken with 3_8_X or 3_7_X
  - Which version you use will largely depend on what version was used to generate your MC data and what version other members of your group are using
  - You can have areas for many different versions at once
- CMSSW is a very large piece of software
  - However, when you create a project area you do not get a copy of all code.
  - You only check out those bits you want to change, and everything else is loaded from a server containing a complete copy, with your modifications overriding the original copy
- All the code for CMSSW is kept in a CVS repository (Concurrent Version System)
  - http://cmssw.cvs.cern.ch/
  - More on CVS later

SETTING UP CMSSW

- CMSSW runs on Scientific Linux ("SLC")
  - You'll probably run it either on NAF here or lxplus at CERN
- CMSSW has new releases often
  - The most recent Monte-Carlo production is for 3_9_X data is being taken with 3_8_X or 3_7_X
  - Which version you use will largely depend on what version was used to generate your MC data and what version other members of your group are using
  - You can have areas for many different versions at once
- CMSSW is a very large piece of software
  - However, when you create a project area you do not get a copy of all code.
  - You only check out those bits you want to change, and everything else is loaded from a server containing a complete copy, with your modifications overriding the original copy
- All the code for CMSSW is kept in a CVS repository (Concurrent Version System)
  - http://cmssw.cvs.cern.ch/
  - More on CVS later

SETTING UP CMSSW @ LXPLUS

CMS Computing Concepts: This chapter introduces you to the CMS computing environment and to CMSSW, the software framework used to analyze data in CMS. You will learn how to connect to lxplus machines at CERN and to find out which CMSSW releases are available for MC/Detector simulations.

- `ssh –X username@lxplus.cern.ch`
- `kinit username@CERN.CH`
- `scram list CMSSW`

```
......
CMSSW      CMSSW_3_9_7    -> /afs/cern.ch/cms/slc5_ia32_gcc434/cms/cmssw/CMSSW_3_9_7
CMSSW      CMSSW_3_9_8    -> /afs/cern.ch/cms/slc5_ia32_gcc434/cms/cmssw/CMSSW_3_9_8
CMSSW      CMSSW_3_9_9    -> /afs/cern.ch/cms/slc5_ia32_gcc434/cms/cmssw/CMSSW_3_9_9
......
```

From tutorial!

MC GENERATORS IN CMSSW

➢ `ls /afs/cern.ch/cms/slc5_ia32_gcc434/cms/cmssw/CMSSW_3_9_7/src/`

```
Alignment          FastSimDataFormats    PerfTools          SimCalorimetry
AnalysisAlgos      FastSimulation        CMS.PhysicsTools   SimDataFormats
.................
AnalysisExamples   GeneratorInterface    RecoBTag           SimG4Core
```

Each subsystem contains several packages, for example in **/GeneratorInterface**

AlpgenInterface Configuration ExhumeInterface GenFilter HijingInterface MCatNLOInterface

PyquenInterface SherpaInterface AMPTInterface Core ExternalDecays Herwig6Interface HydjetInterface

PartonShowerVeto Pythia6Interface ThePEGInterface BeamHaloGenerator CosmicMuonGenerator

GenExtension HiGenCommon LHEInterface PomwigInterface Pythia8Interface

GENERATOR INTERFACES

Generator	Documentation	Interface	Responsible	Status
Pythia6	View Twiki	yes	Julia Yarba	ready
Herwig6	under construction	yes	Fabian Stoeckli	ready
Pythia8	under construction	yes	Mikhail Kirsanov	ready
ThePEG (Herwig++, Ariadne 5)	under construction	yes	Christoph Hackstein, Oliver Oberst, Fred Stober	ready
ALPGEN	View Twiki	yes	Thiago Tomei	ready
MadGraph	View Twiki	no	Roberto Chierici, Silvano Tosi	ready
MC@NLO	View Twiki	yes	Fabian Stoeckli	ready
POWHEG	under construction	no	Martina Malberti	ready
SHERPA	View Twiki	yes	Martin Niegel, Markus Merschmeyer, Altan Cakir	ready
Phantom	under construction	no	Sara Bolognesi	ready
Hydjet	under construction	yes	Yetkin Yilmaz	ready
Pyquen	under construction	yes	Yetkin Yilmaz	ready
Cosmic Muon Generator	under construction	yes	Lars Sonnenschein , Kerstin Hoepfner	ready
Beam Halo Muon Generator	no doc	yes	?	ready
ExHuME	View Twiki	yes	Antonio Vilela Pereira	ready
Pomwig	View Twiki	yes	Antonio Vilela Pereira	ready
BcGenerator	View Twiki	no	Silvia Taroni	ready
HARDCOL	under constructions	no	Sheila Amaral	ready

MC GENERATORS IN CMSSW

```
➢ ls /afs/cern.ch/cms/slc5_ia32_gcc434/cms/cmssw/CMSSW_3_9_7/src/
Alignment      FastSimDataFormats   PerfTools          SimCalorimetry
AnalysisAlgos  FastSimulation       CMS.PhysicsTools   SimDataFormats
.....
AnalysisExamples  GeneratorInterface  RecoBTag          SimG4Core
```

Each subsystem contains several packages, for example in **/GeneratorInterface**

```
AlpgenInterface   Configuration ExhumeInterface  GenFilter  HijingInterface  MCatNLOInterface
PyquenInterface SherpaInterface AMPTInterface Core ExternalDecays Herwig6Interface  HydjetInterface
PartonShowerVeto  Pythia6Interface  ThePEGInterface  BeamHaloGenerator  CosmicMuonGenerator
GenExtension  HiGenCommon  LHEInterface   PomwigInterface  Pythia8Interface
```

```
➢ cmsrel CMSSW_X_Y_Z
➢ cd CMSSW_X_Y_Z/src
➢ cmsenv
```

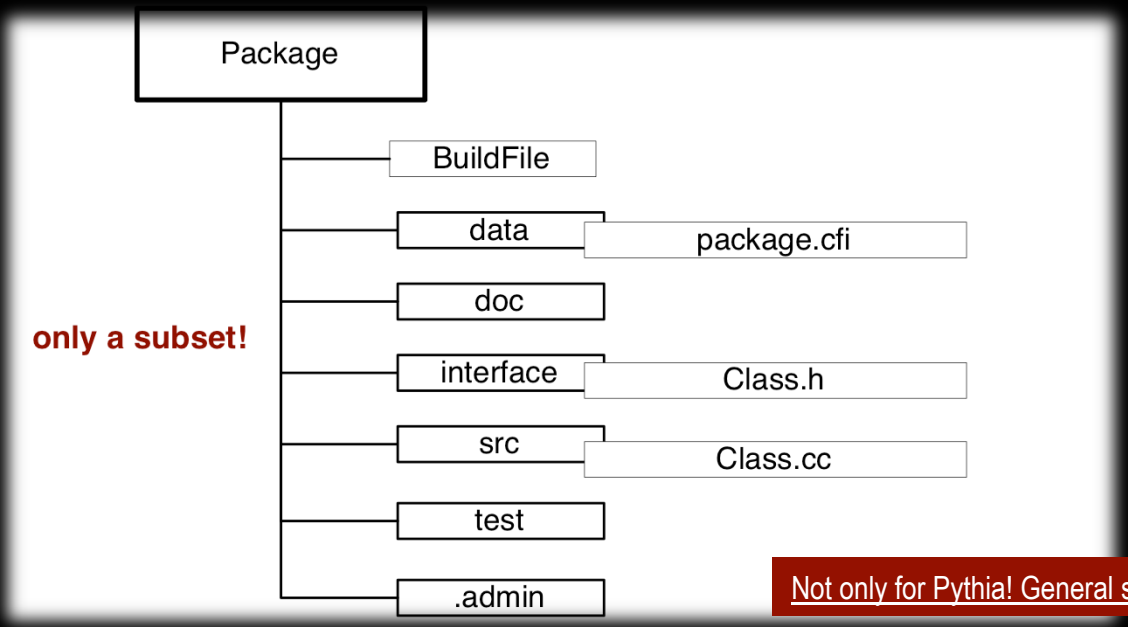
A new project area created in your lxplus machine in the */afs/cern.ch/user/u/username*.

```
➢ addpkg GeneratorInterface/Pythia6Interface
➢ OR cvs co -r CMSSW_X_Y_Z GeneratorInterface/Pythia6Interface
➢ scram b
➢ cd GeneratorInterface/Pythia6Interface/test
```

THE RELEASE AREA

```
Project/Subproject/Package
```

CMSSW_3_9_7/src/GeneratorInterface/Pythia6Interface



CONFIGURATION FILES

Definition of terms: configuration file

- 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
```

Process Object:
"the protagonist" of the configuration

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

SCRAM is a configuration management tool, a distribution system, a build system and resource manager, with local resources and applications managed in a transparent way.

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

HOW TO IMPORT SETTINGS/OBJECT

- 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)
```

THE OUTPUT

- Standard event contents are defined centrally:

```
Configuration.EventContent.EventContent_cff
```

- Output files are written via the PoolOutputModule

```
cms.OutputModule("PoolOutputModule",
    outputCommands = RECOEventContent.outputCommands,
    fileName = cms.untracked.string('TTbar_cfi_GEN_SIM_DIGI.root'),
    SelectEvents = cms.untracked.PSet(
        SelectEvents = cms.vstring('*Electron:HLT')
    )
)
```

PYTHIA CONFIGURATION CARD

From tutorial

The two major software components are

- Pythia6GeneratorFilter - full event generation
- Pythia6HadronizerFilter - processing of parton-level event by an ME generator

Both components deliver the same output: **HepMCProduct** and **GenEventInfoProduct** (these are common across CMSSW interfaces to all multi-purpose generators). For a Pythia-MC test run in the CMSSW, execute the following commands:

Generator Level Simulation:

- `cd CMSSW_X_Y_Z/src/`
- `cmsenv`
- `cp /afs/cern.ch/user/c/cakir/public/MC2011/testPythia.py .`
- `cmsRun –p testPythia.py &> log_Pythia6MC`

PYTHIA CONFIGURATION CARD

From tutorial

The two major software components are

- Pythia6GeneratorFilter - full event generation
- Pythia6HadronizerFilter - processing of parton-level event by an ME generator

Both components deliver the same output: **HepMCProduct** and **GenEventInfoProduct** (these are common across CMSSW interfaces to all multi-purpose generators). For a Pythia-MC test run in the CMSSW, execute the following commands:

Generator Level Simulation:

```
cd CMSSW_X_Y_Z/src/
cmsenv
cp /afs/cern.ch/user/c/cakir/public/MC2011/testPythia.py .
cmsRun –p testPythia.py &> log_Pythia6MC
```

Tutorial

PYTHIA SIMULATION

```
lxplus243] ~/public/MC2011 $ vi testPythia6.py
lxplus243] ~/public/MC2011 $ cmsRun testPythia6.py
     MSTU(12)       changed from              0 to          12345
1****************** PYINIT: initialization of PYTHIA routines *****************
 ==== PYTHIA WILL USE LHAPDF ====
 ************************************
 *       LHAPDF Version 5.6.0       *
 ************************************

 >>>>>> PDF description: <<<<<<
 CTEQ6L1 - LO with LO alpha_s
 Reference:
 J. Pumplin, D.R. Stump, J. Huston, H.L. Lai, P. Nadolsky,
 W.K. Tung
 hep-ph/0201195
 >>>>>>                    <<<<<<

 Parametrization: CTEQ6

 ================================================
 PDFset name /afs/cern.ch/cms/slc5_ia32_gcc434/external/lhapdf/5.6.0-cms2/share/lhapdf/PDFset
 with          1 members
 ====  initialized. ============================
 Strong coupling at Mz for PDF is:  0.12978

 ================================================================================
 I                                                                              I
 I               PYTHIA will be initialized for a p on p collider               I
 I                   at  14000.000 GeV center-of-mass energy                    I
 I                                                                              I
 ================================================================================

 ******** PYMAXI: summary of differential cross-section maximum search ********

           ============================================================
           I                                      I                   I
           I  ISUB  Subprocess name               I  Maximum value  I
           I                                      I                   I
           ============================================================
           I                                      I                   I
           I    1   f + fbar -> gamma*/Z0         I    2.5332D-05   I
           I   96   Semihard QCD 2 -> 2           I    1.3343D+04   I
           I                                      I                   I
           ============================================================

 ****** PYMULT: initialization of multiple interactions for MSTP(82) = 4 ******
        pT0 = 2.52 GeV gives sigma(parton-parton) = 8.29D+02 mb: accepted

 ****** PYMIGN: initialization of multiple interactions for MSTP(82) = 4 ******
        pT0 = 2.52 GeV gives sigma(parton-parton) = 3.27D+02 mb: accepted
```

PYTHIA SIMULATION

```
lxplus243] ~/public/MC2011 $ vi testPythia6.py
lxplus243] ~/public/MC2011 $ cmsRun testPythia6.py
     MSTU(12)       changed from              0 to          12345
1****************** PYINIT: initialization of PYTHIA routines *****************
 ==== PYTHIA WILL USE LHAPDF ====
 ***************************************
 *       LHAPDF Version 5.6.0          *
 ***************************************

 >>>>>> PDF description: <<<<<<
 CTEQ6L1 - LO with LO alpha_s
 Reference:
 J. Pumplin, D.R. Stump, J. Huston, H.L. Lai, P. Nadolsky,
 W.K. Tung
 hep-ph/0201195
 >>>>>>                     <<<<<<

 Parametrization: CTEQ6

 ================================================
 PDFset name /afs/cern.ch/cms/slc5_ia32_gcc434/external/lhapdf/5.6.0-cms2/share/lhapdf/PDFset
 with          1 members
 ====  initialized. ============================
 Strong coupling at Mz for PDF is:  0.12978

 ============================================================================
 I                                                                          I
 I             PYTHIA will be initialized for a p on p collider             I
 I                 at   14000.000 GeV center-of-mass energy                 I
 I                                                                          I
 ============================================================================

 ******** PYMAXI: summary of differential cross-section maximum search ********

           ============================================================
           I                                      I                   I
           I  ISUB  Subprocess name               I  Maximum value    I
           I                                      I                   I
           ============================================================
           I                                      I                   I
           I    1   f + fbar -> gamma*/Z0         I    2.5332D-05     I
           I   96   Semihard QCD 2 -> 2           I    1.3343D+04     I
           I                                      I                   I
           ============================================================

 ****** PYMULT: initialization of multiple interactions for MSTP(82) = 4 ******
        pT0 = 2.52 GeV gives sigma(parton-parton) = 8.29D+02 mb: accepted

 ****** PYMIGN: initialization of multiple interactions for MSTP(82) = 4 ******
        pT0 = 2.52 GeV gives sigma(parton-parton) = 3.27D+02 mb: accepted
```

pdf

cms Energy @LHC

subprocesses

EVENT LISTING

I	particle/jet	KS	KF	orig	p_x	p_y	p_z	E	m
1	!p+!		21	2212	0	0.000	0.000	7000.000	7000.000	0.938
2	!p+!		21	2212	0	0.000	0.000	-7000.000	7000.000	0.938
3	!u!		21	2	1	0.421	-1.555	1249.886	1249.887	-0.000
4	!u!		21	2	2	-2.458	-0.968	-1357.893	1357.896	-0.000
5	!ubar!		21	-2	3	-1.487	1.539	-5.894	6.271	0.000
6	!u!		21	2	4	1.110	-2.039	-802.972	802.976	0.000
7	!Z0!		21	23	0	-0.377	-0.500	-808.866	809.246	24.787
8	!mu-!		21	13	7	-4.098	-0.027	-789.990	790.001	0.106
9	!mu+!		21	-13	7	3.721	-0.473	-18.876	19.245	0.106
10	(Z0)		11	23	7	-0.377	-0.500	-808.866	809.246	24.787
11	mu-		1	13	8	-4.098	-0.027	-789.990	790.001	0.106
12	mu+		1	-13	9	3.721	-0.473	-18.876	19.245	0.106
13	(u)	A	12	2	3	0.329	-1.156	-2.630	2.911	0.330
14	(g)	I	12	21	3	1.324	2.089	-4.450	5.092	0.000
15	(g)	I	12	21	3	-0.498	0.568	-4.238	4.305	0.000
16	(ud_0)	V	11	2101	1	-0.421	1.555	5743.472	5743.472	0.579
17	(u)	A	12	2	3	0.341	-4.629	1210.653	1210	
18	(g)	I	12	21	3	0.518	-0.097	38.535	3	
19	(g)	I	12	21	0	-3.557	-1.545	6.004		
20	(g)	I	12	21	0	3.557	1.545	-57.205	5	
21	(g)	I	12	21	4	0.051	-0.398	-98.108	9	
22	(g)	I	12	21	4	-3.725	1.598	-438.902	43	
23	(ud_1)	V	11	2103	2	2.458	0.968	-5584.263	558	

```
Begin processing the 2nd record. Run 1, Event 2, LumiSection 1 at 15-Mar-2011 12:56:53.111 CET
Begin processing the 3rd record. Run 1, Event 3, LumiSection 1 at 15-Mar-2011 12:56:53.123 CET
Begin processing the 4th record. Run 1, Event 4, LumiSection 1 at 15-Mar-2011 12:56:53.141 CET
Begin processing the 5th record. Run 1, Event 5, LumiSection 1 at 15-Mar-2011 12:56:53.148 CET
Begin processing the 6th record. Run 1, Event 6, LumiSection 1 at 15-Mar-2011 12:56:53.165 CET
Begin processing the 7th record. Run 1, Event 7, LumiSection 1 at 15-Mar-2011 12:56:53.177 CET
Begin processing the 8th record. Run 1, Event 8, LumiSection 1 at 15-Mar-2011 12:56:53.192 CET
Begin processing the 9th record. Run 1, Event 9, LumiSection 1 at 15-Mar-2011 12:56:53.208 CET
Begin processing the 10th record. Run 1, Event 10, LumiSection 1 at 15-Mar-2011 12:56:53.217 CET
 ********* PYSTAT:  Statistics on Number of Events and Cross-sections *********

 ================================================================================
 I                                  I                                  I           I
 I           Subprocess             I      Number of points            I    Sigma  I
 I                                  I                                  I           I
 I----------------------------------I----------------------------------I    (mb)   I
 I                                  I                                  I           I
 I N:o Type                         I    Generated          Tried      I           I
 I                                  I                                  I           I
 ================================================================================
 I                                  I                                  I           I
 I   0 All included subprocesses    I          10            33        I 8.227D-06 I
 I   1 f + fbar -> gamma*/Z0        I          10            33        I 8.227D-06 I
 I                                  I                                  I           I
 ================================================================================
```

EVENT LISTING

I	particle/jet	KS	KF	orig	p_x	p_y	p_z	E	m
1	!p+!	21	2212	0	0.000	0.000	7000.000	7000.000	0.938
2	!p+!	21	2212	0	0.000	0.000	-7000.000	7000.000	0.938
3	!u!	21	2	1	0.421	-1.555	1249.886	1249.887	-0.000
4	!u!	21	2	2	-2.458	-0.968	-1357.893	1357.896	-0.000
5	!ubar!	21	-2	3	-1.487	1.539	-5.894	6.271	0.000
6	!u!	21	2	4	1.110	-2.039	-802.972	802.976	0.000
7	!Z0!	21	23	0	-0.377	-0.500	-808.866	809.246	24.787
8	!mu-!	21	13	7	-4.098	-0.027	-789.990	790.001	0.106
9	!mu+!	21	-13	7	3.721	-0.473	-18.876	19.245	0.106
10	(Z0)	11	23	7	-0.377	-0.500	-808.866	809.246	24.787
11	mu-	1	13	8	-4.098	-0.027	-789.990	790.001	0.106
12	mu+	1	-13	9	3.721	-0.473	-18.876	19.245	0.106
13	(u) A	12	2	3	0.329	-1.156	-2.630	2.911	0.330
14	(g) I	12	21	3	1.324	2.089	-4.450	5.092	0.000
15	(g) I	12	21	3	-0.498	0.568	-4.238	4.305	0.000
16	(ud_0) V	11	2101	1	-0.421	1.555	5743.472	5743.472	0.579
17	(u) A	12	2	3	0.341	-4.629	1210.653	1210	
18	(g) I	12	21	3	0.518	-0.097	38.535	3	
19	(g) I	12	21	0	-3.557	-1.545	6.004		
20	(g) I	12	21	0	3.557	1.545	-57.205	5	
21	(g) I	12	21	4	0.051	-0.398	-98.108	9	
22	(g) I	12	21	4	-3.725	1.598	-438.902	43	
23	(ud_1) V	11	2103	2	2.458	0.968	-5584.263	558	

Proton-proton collision @ 14 TeV

$Z \to \mu^+\mu^-$

originated

```
...gin processing the 2nd record. Run 1, Event 2, LumiSection 1 at 15-Mar-2011 12:56:53.111 CET
Begin processing the 3rd record. Run 1, Event 3, LumiSection 1 at 15-Mar-2011 12:56:53.123 CET
Begin processing the 4th record. Run 1, Event 4, LumiSection 1 at 15-Mar-2011 12:56:53.141 CET
Begin processing the 5th record. Run 1, Event 5, LumiSection 1 at 15-Mar-2011 12:56:53.148 CET
Begin processing the 6th record. Run 1, Event 6, LumiSection 1 at 15-Mar-2011 12:56:53.165 CET
Begin processing the 7th record. Run 1, Event 7, LumiSection 1 at 15-Mar-2011 12:56:53.177 CET
Begin processing the 8th record. Run 1, Event 8, LumiSection 1 at 15-Mar-2011 12:56:53.192 CET
Begin processing the 9th record. Run 1, Event 9, LumiSection 1 at 15-Mar-2011 12:56:53.208 CET
Begin processing the 10th record. Run 1, Event 10, LumiSection 1 at 15-Mar-2011 12:56:53.217 CET
1********* PYSTAT:  Statistics on Number of Events and Cross-sections *********
```

Subprocess N:o	Type	Number of points Generated	Tried	Sigma (mb)
0	All included subprocesses	10	33	8.227D-06
1	f + fbar -> gamma*/Z0	10	33	8.227D-06

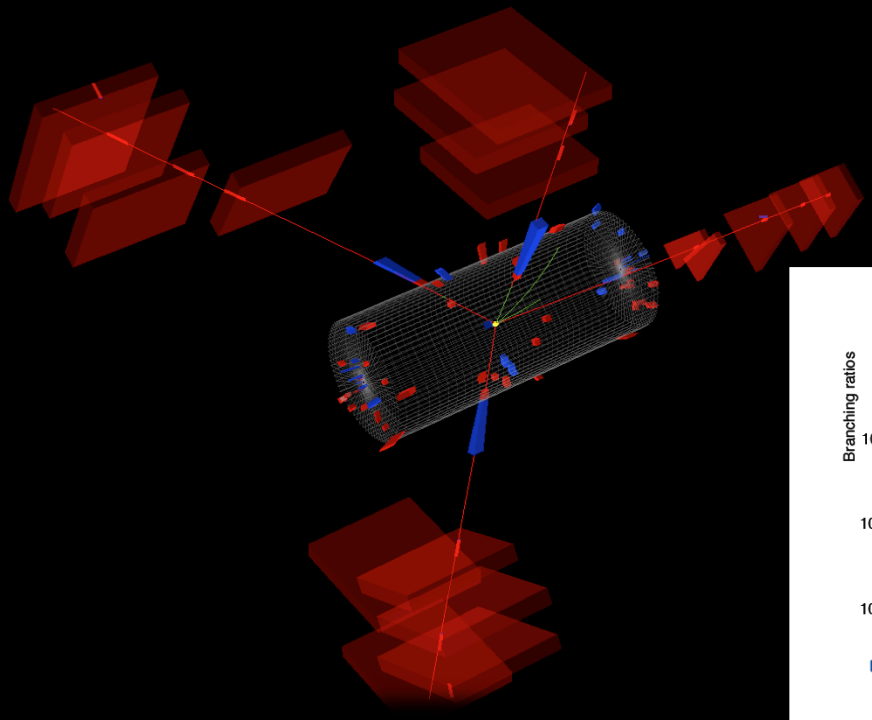
Xsection (mb)

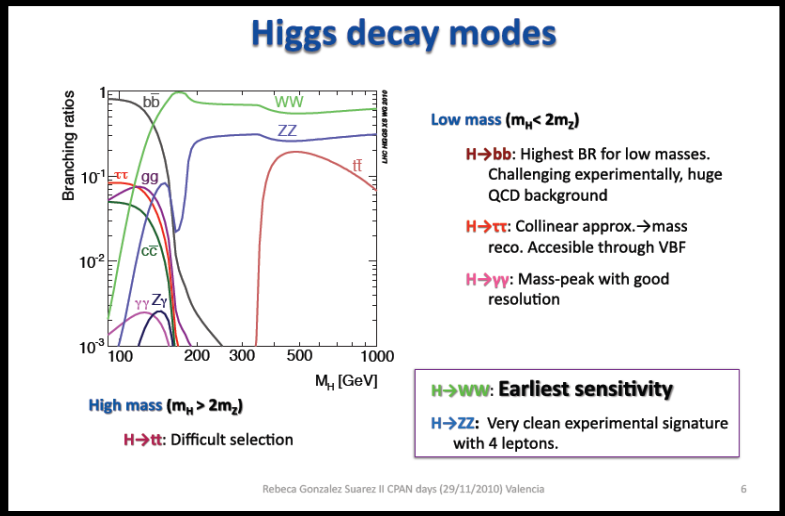
TUTORIAL1

- Simulate Z → 2l (l=e,mu) 1000K events with Pythia and analyze mOSSF at generator level
- Simulate ZZ→ 4l (l=mu) 1000K events with Pythia and analyze mOSSF for Z and ZZ (Higgs).

TUTORIAL1

- Simulate Z → 2l (l=e,mu) 1000K events with Pythia and analyze mOSSF at generator level
- Simulate ZZ→ 4l (l=mu) 1000K events with Pythia and analyze mOSSF for Z and ZZ (Higgs).





SUMMARY

- Here are the topics covered in this lecture:
  - ✓ Analysis chain in CMS Detector
  - ✓ Data stream and event data model in CMSSW
  - ✓ Setting up CMSSW environment
  - ✓ Generator interface in CMSSW
  - ✓ MC/Detector simulations in the CMSSW
    - ✧ Pythia simulations: Generator/Hadronization Filters
    - ❑ cmsDriver usage for the MC production
    - ❑ HLT, RECO and FASTSIM
    - ❑ External files (LHE/SLHA etc.) usage in Pythia Hadronization
  - ❑ Systematics studies for MC (not covered – time? - discussion?)

DETECTOR SIMULATION

- The software that is related to the IR modeling and detector simulation, including related components, is quite extensive and resides in several subsystems:

  IOMC, SimGeneral, SimG4Core, SimTracker, SimCalo, SimMuon

- The simulation part in the event processing chain is performed by several CMSSW modules, that need to be composed in the specific order in the that need to be composed in the specific order in the configuration application as each module in the chain will require certain output from an earlier step, performed by another module. Also, these modules need to be assigned certain labels, because it that by the label that each module will access necessary information delivered into edm::Event.

An Event starts as a collection of the RAW data from detector or MC event.
In software terms: an Event is a single entity in memory, a C++ type-safe container called

edm::Event

Data within the Event are uniquely identified by four quantities:

- *C++ class type of the data*
- *module label*
- *product instance label (usually empty string)*
- *process name*

recoTracks_globalMuons__RECO

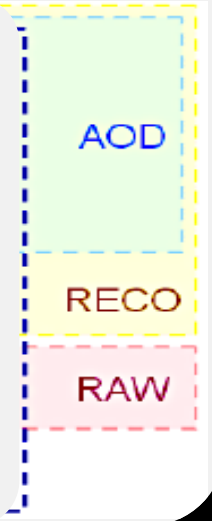
class name module label process name

EDM::CONTENT

What is stored in the event files?

- A collection of products generated by CMSSW producers
- These products are grouped into useful categories"
  - FEVT: Full Event
  - RECO: RECOnstruction
  - RECOSIM: RECOnstruction + selected simulation information
  - AOD: Analysis Object Data (a compact subject of RECO fromat)
  - AODSIM: AOD + generator information

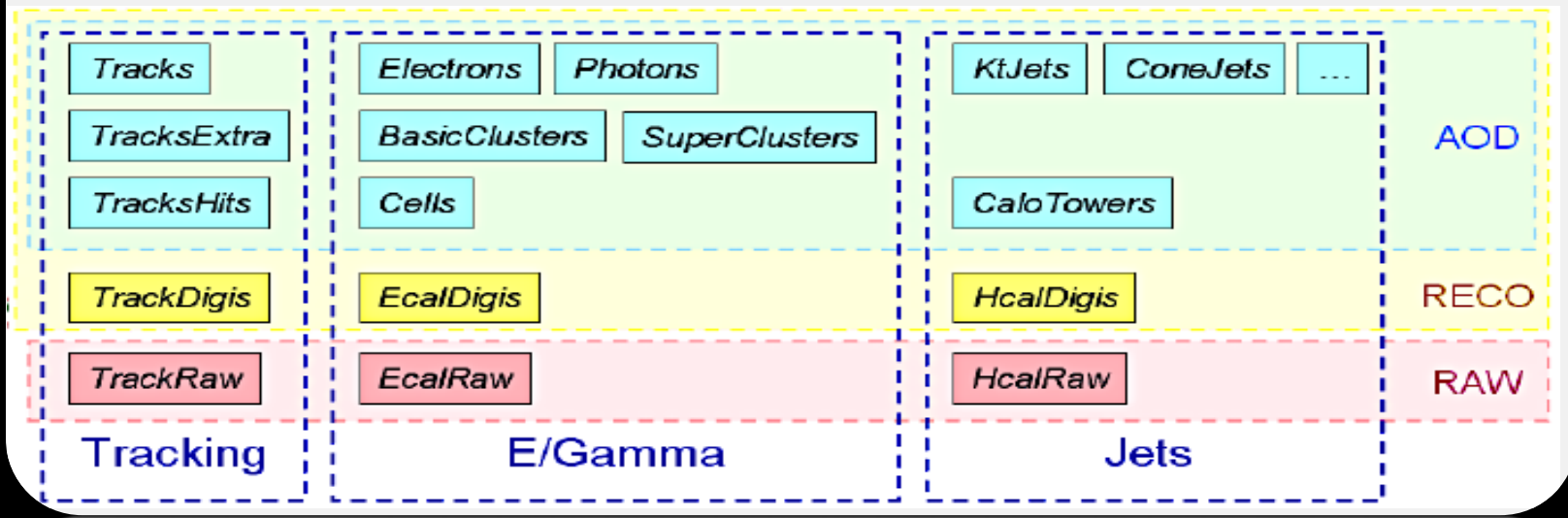
- Based of regular root tree
  - Content viewable by *TBrowser*
- Stores complex C++ objects as hierarchical branches
  - By loading so called dictionaries, ROOT becomes aware of C++ class details, called DataFormates.
- Contains two main TTrees
  - Run
  - Events
- Contains metadata, the "provenance".



EDM::CONTENT

What is stored in the event files?

- A collection of products generated by CMSSW producers
- These products are grouped into useful categories"
  - FEVT: Full Event
  - RECO: RECOnstruction
  - RECOSIM: RECOnstruction + selected simulation information
  - AOD: Analysis Object Data (a compact subject of RECO fromat)
  - AODSIM: AOD + generator information



WORKFLOWS

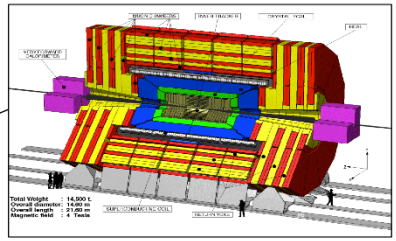
MC Workflow

SimHits
Pileup Events

SimHits
Signal Event

Simulation of
Electronics

Tracker
Calorimeter
Muon Det.
TriggerSim



Raw Data

STEP 1 = `GEN:ProductionFilterSequence,SIM,DIGI,L1,DIGI2RAW,HLT`

Digi's
RecHits,
Segments

Low-level Reconstruction

High-level Reconstruction and Analysis

Tracks, Jets,
Muons, Vertices,
MET, Electrons,
Photons, b-tags,
etc...

STEP 2 = `RAW2DIGI,L1Reco,RECO,VALIDATION`

The set of processes that both MC and real data flow through.

STANDARD WORKFLOWS - CMSDRIVER

- Because of the daily testing of workflows, they should all work "out of the box". All you need to know is how to reproduce them.

To create the **full configuration file** from the generation fragment we will use the same local scram area (CMSSW_3_9_7).

```
cd ~/CMSSW_3_9_7/src
cmsenv
cvs co Configuration/Generator
scram b
ls Configuration/Generator/python
```

from tutorial

```
cmsDriver.py Configuration/Generator/python/H200ZZ4L_cfi.py -s
GEN,SIM,DIGI,L1,DIGI2RAW,HLT:GRun --conditions auto:mc --datatier GEN-SIM-RAW --
eventcontent RAWSIM -n 10 --fileout RAWSIM.root  --no_exec
```

Steps? - Step1-

STEP1 – CONFIGURATION CARD

```
# Auto generated configuration file
# using:
# Revision: 1.232.2.6.2.1
# Source: /cvs_server/repositories/CMSSW/CMSSW/Configuration/PyReleaseValidation/python/ConfigBuilder.py,v
# with command line options: Configuration/Generator/python/H200ZZ4L_cfi.py -s GEN,SIM,DIGI,L1,DIGI2RAW,HLT:GRun --conditions auto:mc --datatier GEN-SIM-RAW --eventcontent RAWSIM -n -1 --no_exec
import FWCore.ParameterSet.Config as cms

process = cms.Process('HLT')

# import of standard configurations
process.load('Configuration.StandardSequences.Services_cff')
process.load('SimGeneral.HepPDTESSource.pythiapdt_cfi')
process.load('FWCore.MessageService.MessageLogger_cfi')
process.load('Configuration.StandardSequences.MixingNoPileUp_cff')
process.load('Configuration.StandardSequences.GeometryDB_cff')
process.load('Configuration.StandardSequences.MagneticField_38T_cff')
process.load('Configuration.StandardSequences.Generator_cff')
process.load('Configuration.StandardSequences.VtxSmearedRealistic7TeVCollision_cff')
process.load('Configuration.StandardSequences.SimIdeal_cff')
process.load('Configuration.StandardSequences.Digi_cff')
process.load('Configuration.StandardSequences.SimL1Emulator_cff')
process.load('Configuration.StandardSequences.DigiToRaw_cff')
process.load('HLTrigger.Configuration.HLT_GRun_cff')
process.load('Configuration.StandardSequences.EndOfProcess_cff')
process.load('Configuration.StandardSequences.FrontierConditions_GlobalTag_cff')
process.load('Configuration.EventContent.EventContent_cff')
```

Complicated??

```
# Path and EndPath definitions
process.generation_step = cms.Path(process.pgen)

process.simulation_step = cms.Path(process.psim)

process.digitisation_step = cms.Path(process.pdigi)
```

```
process.L1simulation_step = cms.Path(process.SimL1Emulator)

process.digi2raw_step = cms.Path(process.DigiToRaw)

process.endjob_step = cms.EndPath(process.endOfProcess)

process.RAWSIMoutput_step = cms.EndPath(process.RAWSIMoutput)


# Schedule definition
process.schedule = cms.Schedule(process.generation_step,process.simulation_step,process.digitisation_step,process.L1simulation_step,process.digi2raw_step)
process.schedule.extend(process.HLTSchedule)
process.schedule.extend([process.endjob_step,process.RAWSIMoutput_step])

# special treatment in case of production filter sequence
for path in process.paths:
    getattr(process,path)._seq = process.generator*getattr(process,path)._seq
```

STEP1 – CONFIGURATION CARD

```
# Auto generated configuration file
# using:
# Revision: 1.232.2.6.2.1
# Source: /cvs_server/repositories/CMSSW/CMSSW/Configuration/PyReleaseValidation/python/ConfigBuilder.py,v
# with command line options: Configuration/Generator/python/H200ZZ4L_cfi.py -s GEN,SIM,DIGI,L1,DIGI2RAW,HLT:GRun --conditions auto:mc --datatier GEN-SIM-RAW --eventcontent RAWSIM -n -1 --no_exec
import FWCore.ParameterSet.Config as cms

process = cms.Process('HLT')

# import of standard configurations
process.load('Configuration.StandardSequences.Services_cff')
process.load('SimGeneral.HepPDTESSource.pythiapdt_cfi')
process.load('FWCore.MessageService.MessageLogger_cfi')
process.load('Configuration.StandardSequences.MixingNoPileUp_cff')
process.load('Configuration.StandardSequences.GeometryDB_cff')
process.load('Configuration.StandardSequences.MagneticField_38T_cff')
process.load('Configuration.StandardSequences.Generator_cff')
process.load('Configuration.StandardSequences.VtxSmearedRealistic7TeVCollision_cff')
process.load('Configuration.StandardSequences.SimIdeal_cff')
process.load('Configuration.StandardSequences.Digi_cff')
process.load('Configuration.StandardSequences.SimL1Emulator_cff')
process.load('Configuration.StandardSequences.DigiToRaw_cff')
process.load('HLTrigger.Configuration.HLT_GRun_cff')
process.load('Configuration.StandardSequences.EndOfProcess_cff')
process.load('Configuration.StandardSequences.FrontierConditions_GlobalTag_cff')
process.load('Configuration.EventContent.EventContent_cff')
```

cmsDriver

Include files –magnetic files, detector simulations etc.

```
# Path and EndPath definitions
process.generation_step = cms.Path(process.pgen)

process.simulation_step = cms.Path(process.psim)

process.digitisation_step = cms.Path(process.pdigi)

process.L1simulation_step = cms.Path(process.SimL1Emulator)

process.digi2raw_step = cms.Path(process.DigiToRaw)

process.endjob_step = cms.EndPath(process.endOfProcess)

process.RAWSIMoutput_step = cms.EndPath(process.RAWSIMoutput)

# Schedule definition
process.schedule = cms.Schedule(process.generation_step,process.simulation_step,process.digitisation_step,process.L1simulation_step,process.digi2raw_step)
process.schedule.extend(process.HLTSchedule)
process.schedule.extend([process.endjob_step,process.RAWSIMoutput_step])

# special treatment in case of production filter sequence
for path in process.paths:
    getattr(process,path)._seq = process.generator*getattr(process,path)._seq
```

Defines which modules and sequences to run

STEP2 – RECONSTRUCTION STEP

After HLT events we should reconstruct the samples: For the **RECO** step

```
cmsDriver.py H200ZZ4L_cfi_py_GEN_SIM_DIGI_L1_DIGI2RAW_RECO.py
RAW2DIGI,L1Reco,RECO --conditions auto::mc --datatier GEN-SIM-RECO
--eventcontent RECOSIM -n 10 --filein file:RAWSIM.root --fileout RECOSIM.root
--cust_function customisePPMC --no_exec
```

tutorial

For more details https://twiki.cern.ch/twiki/bin/view/CMSPublic/WorkBookSimDigi

FOR FASTSIMULATION

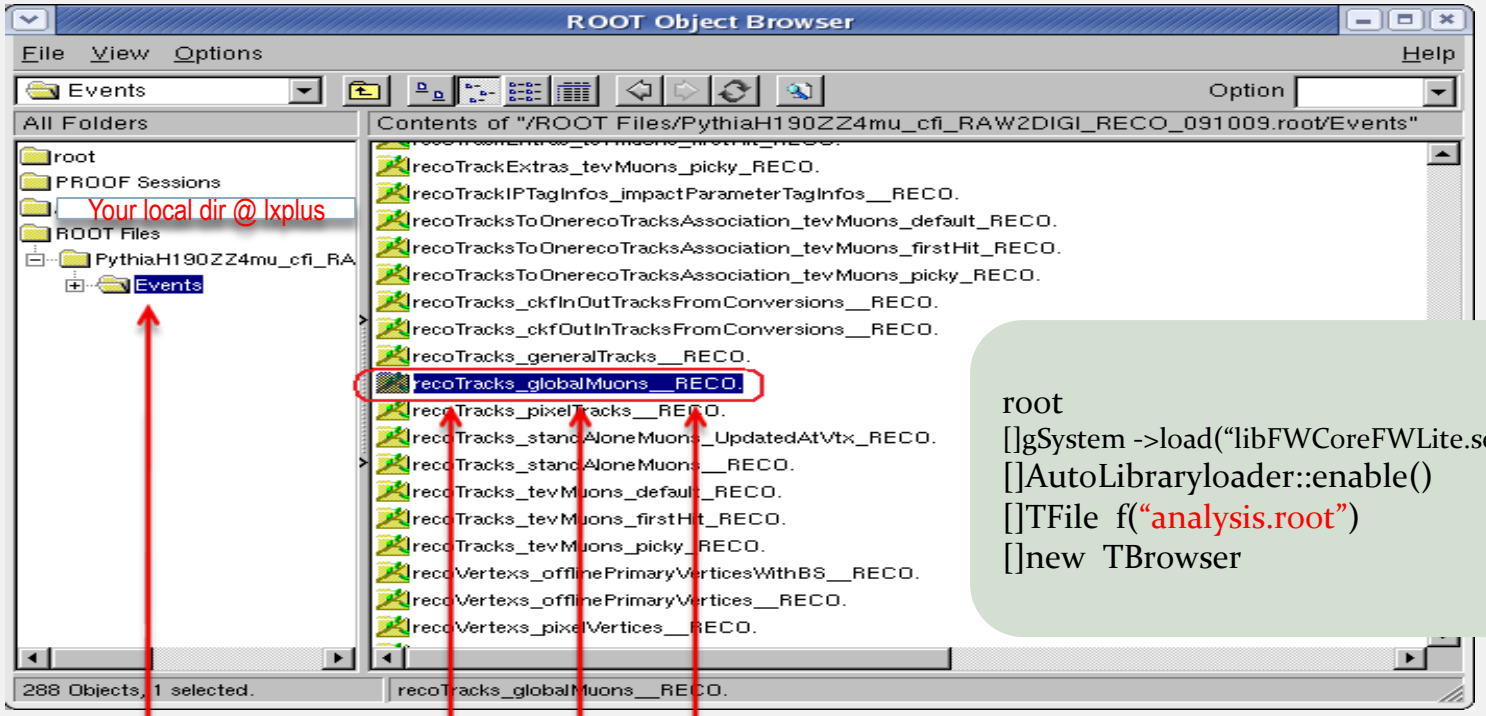
```
cmsDriver.py Configuration/Generator/python/H200ZZ4L_cfi.py GEN,FASTSIM --
pileup=NoPileUp --conditions FrontierConditions_GlobalTag,IDEAL_V11::All --
beamspot=Early7TeVCollision --datatier 'GEN-SIM-DIGI-RECO' --eventcontent AODSIM -n
10 --no_exec
```

tutorial

TUTORIAL2

- Generate step1 (HLT) and step2 (RECO) simulation cards for H→ZZ-> 4l (l=mu) in CMS Detector simulation. Generate the fast simulation configuration card for the same physical process.
- For understanding physics process, please look at the pythia paramaters and settings from Pythia manual.
- Walk-through a Configuration File(s): Simulation Related Steps and Components, try to understand the different simulation steps (for all conf.cards)
- Open edm files (HLT-RECO-FASTSIM) with bare root (next slide) and check the events.
- For analysis of these files, please follow the appendix(tutorial) and the link 4 in the references.

TUTORIAL2



root
[]gSystem ->load("libFWCoreFWLite.so")
[]AutoLibraryloader::enable()
[]TFile f("analysis.root")
[]new TBrowser

The "*Process*" name

The product "*label*"

The C++ class name(Here: vector <reco::Track>)

The main EDM TTree

SUMMARY

- Here are the topics covered in this lecture:
  - ✓ Analysis chain in CMS Detector
  - ✓ Data stream and event data model in CMSSW
  - ✓ Setting up CMSSW environment
  - ✓ Generator interface in CMSSW
  - ✓ MC/Detector simulations in the CMSSW
    - ✓ Pythia simulations: Generator/Hadronization Filters
    - ✓ cmsDriver usage for the MC production
    - ✓ HLT, RECO and FASTSIM
    - ❑ External files (LHE/SLHA etc.) usage in Pythia Hadronization
  - ❑ Systematics studies for MC (not covered – time? - discussion?)

PYTHIA SIMULATION WITH EXTERNAL INPUT

The two major software components are

- Pythia6GeneratorFilter - full event generation
- Pythia6HadronizerFilter - processing of parton-level event by an ME generator

Both components deliver the same output: **HepMCProduct** and **GenEventInfoProduct** (these are common across CMSSW interfaces to all multi-purpose generators). For a Pythia-MC test run in the CMSSW, execute the following commands:

Generator Level Simulation:

- `cd CMSSW_X_Y_Z/src/`
- `cmsenv`

Remember?
Now we will have a look at Pythia hadronization for LHE or SLHA files?

PYTHIA SIMULATION WITH EXTERNAL INPUT

The two major software components are

- Pythia6GeneratorFilter - full event generation
- Pythia6HadronizerFilter - processing of parton-level event by an ME generator

Both components deliver the same output: **HepMCProduct** and **GenEventInfoProduct** (these are common across CMSSW interfaces to all multi-purpose generators). For a Pythia-MC test run in the CMSSW, execute the following commands:

Generator Level Simulation:

- `cd CMSSW_X_Y_Z/src/`
- `cmsenv`

Remember?
Now we will have a look at Pythia hadronization for LHE or SLHA files?

Pythia Hadronization for LHE file: There is a dedicated module Pythia6HadronizerFilter, which is different from Pythia6GeneratorFilter (previous examples) used for full event generation.

- `cd GeneratorInterface/Pythia6Interface/test`
- `less Py6HadFilter_cfg.py` OR `less Py6HadFilter_mgmatching_cfg.py`
- `cmsRun Py6HadFilter_mgmatching_cfg.py &> log_LHE_MadGraph_MGMMatching`

These files are set for Pythia Hadronization and they read LHE flies from MadGraph simulation.

Tutorial. Ready example in Pythia interface

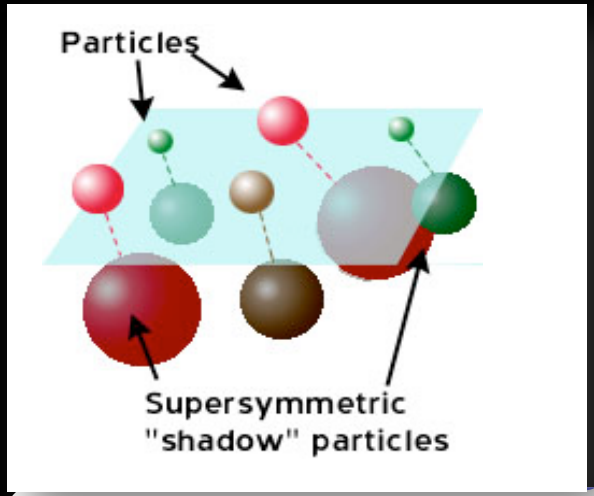
TTBAR SIMULATION EXAMPLE – MADGRAPH

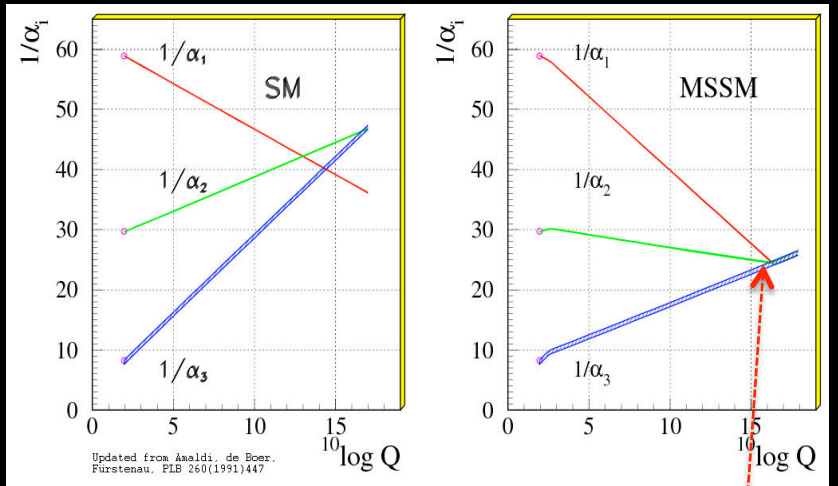
```
mgInfoCMS: maxjets 3
mgInfoCMS: minjets 0
 MGINIT: ickkw is             1
 MGINIT: ktscheme is             0
 MGINIT: QCut is     30.000000000000000
 MGINIT: Showerkt is     0.0000000000000000
 NLJETS=            0
 Minimum number of jets in file:             6
 Maximum number of jets in file:             0
 NPRUP=            3
  Number of Events Read::             0
  Total cross section (pb)::     1475.5137500000001
  Process    Cross Section (pb)::
    1             0.16932E+03
    2             0.87079E+03
    3             0.43540E+03
 Minimum number of jets allowed:             0
 Maximum number of jets allowed:             3
 IEXCFILE =             0
 jetprocs =  F
 IEXCPROC(            0 ) =            -2
 IEXCPROC(            1 ) =            -2
 IEXCPROC(            2 ) =            -2
 Found parameter  ptj    20.000000000000000
 Found parameter  etaj    5.0000000000000000
 Found parameter  ptb    20.000000000000000
 Found parameter  etab    5.0000000000000000
 Found parameter  drjj    1.0000000000000002E-003
 Found parameter  xqcut    20.000000000000000
 KT JET PARAMETERS FOR MATCHING:
 QCUT=   30.000000000000000
 ETA(JET)<   5.0000000000000000
 Note that in ME generation, qcut =    20.000000000000000
 the showerkt param is    0.0000000000000000
Begin processing the 1st record. Run 1, Event 1, LumiSection 1 at 16-Mar-2011 15:27:57.338 CET

 ****************************************************
 KTCLUS: written by Mike Seymour, July 1992.
 Last modified November 2000.
 Please send comments or suggestions to Mike.Seymour@rl.ac.uk

       Collision type = 4 = pp
    Angular variable = 3 = f(DeltaR)
 Monotonic definition = 1 = no                Recombination scheme = 3 = Pt**2
```

SUPERSYMMETRY



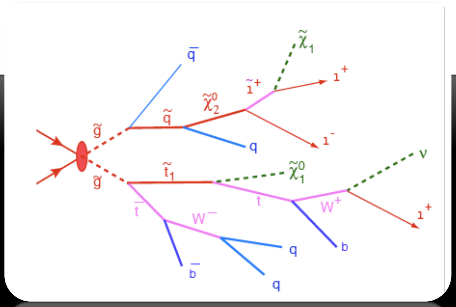


It provides a solution to the hierarchy problem

It allows unification of the gauge couplings at high scales and therefore a GUT?

It can provide a dark matter candidate

Signature? Missing energy



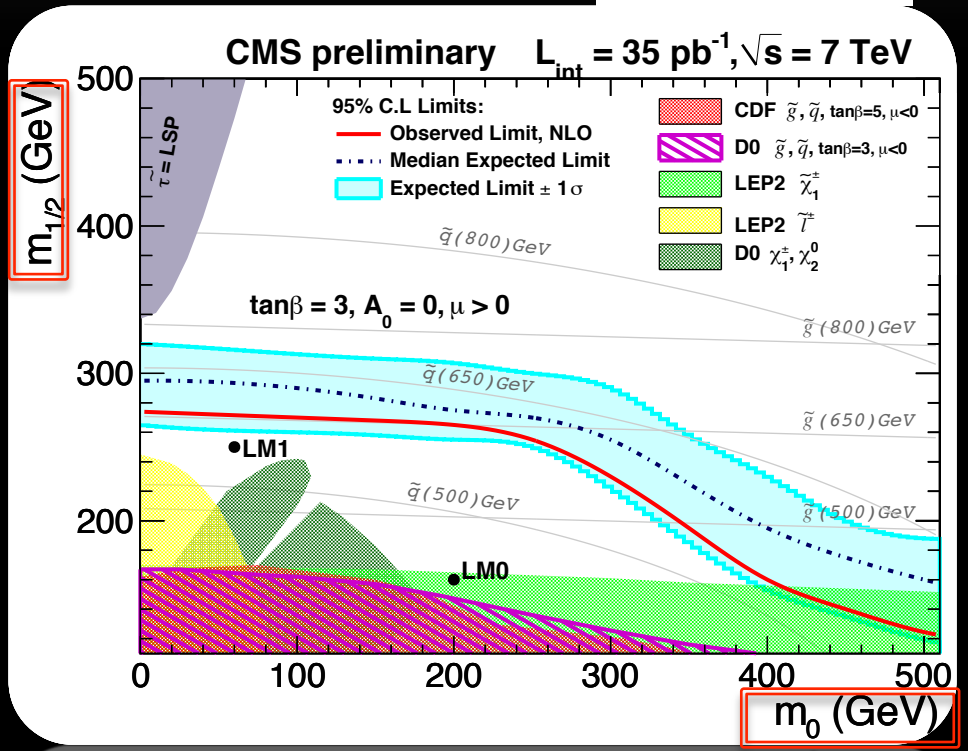
SUSY - M0 AND M1/2 PLANE

mSugra (minimal supergravity) depends on the 5 main parameters (reduces the parameters)

- m0 and m1/2 (common scalar and gauigino mass at he GUT scale)
- A^0 , tanβ and μ (common gaugino terms, ratio of vev values between 2 higgs doublets, mixing parm. them)

How can we produce this plot?

arXiv:1101.1628

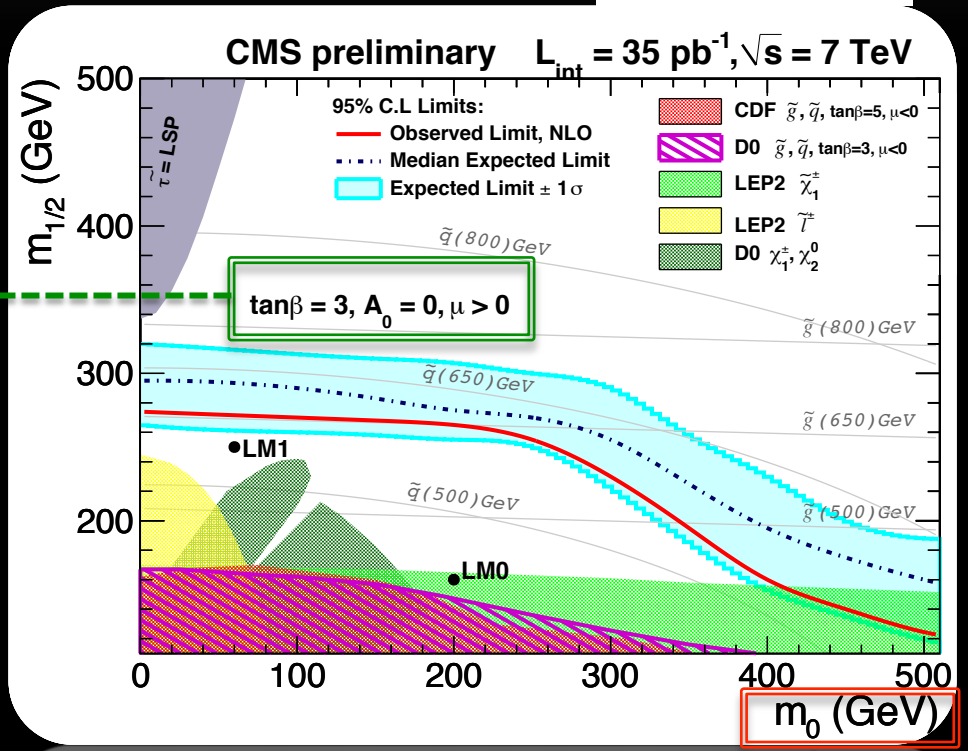


SUSY M0 AND M1/2 PLANE

mSugra (minimal supergravity) depends on the 5 main parameters (reduces the parameters)

- m0 and m1/2 (common scalar and gauigino mass at he GUT scale)
- A^0 , tanβ and μ (common gaugino terms, ratio of vev values between 2 higgs doublets, mixing parm. them)

arXiv:1101.1628



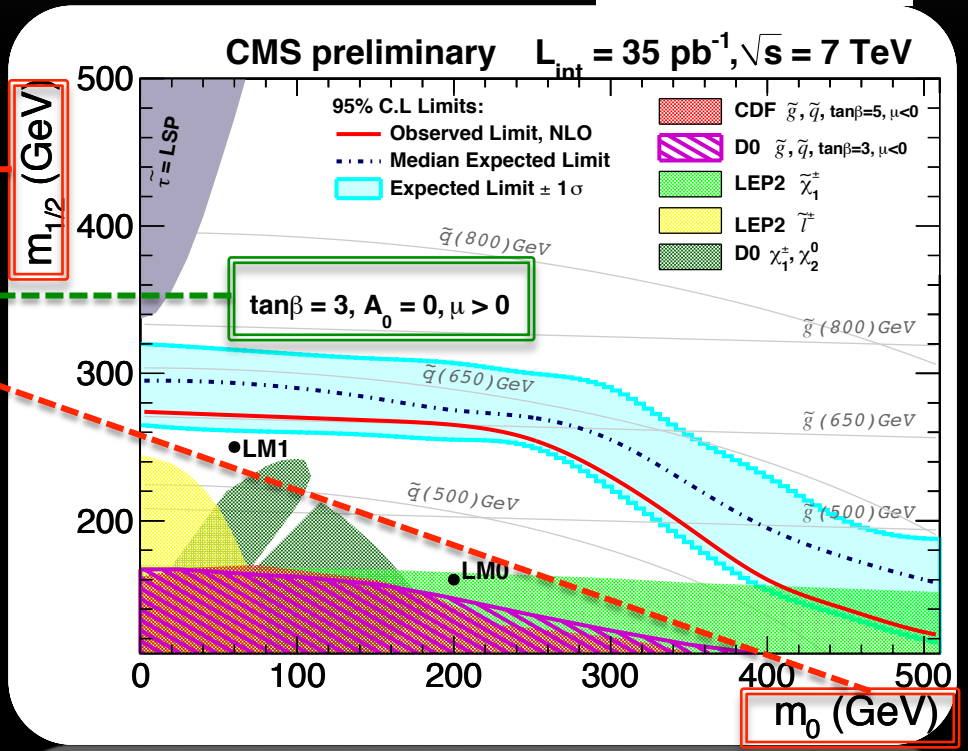
SUSY M0 AND M1/2 PLANE

mSugra (minimal supergravity) depends on the 5 main parameters (reduces the parameters)

- m0 and m1/2 (common scalar and gauigino mass at he GUT scale)
- A^0 , tanβ and μ (common gaugino terms, ratio of vev values between 2 higgs doublets, mixing parm. them)

Change the m0 and m1/2 values for 2d plane! HOW?

arXiv:1101.1628



HOW TO GENERATE SUSY SAMPLES USING AN EXTERNAL SLHA SPECTRUM FILE

- For an accurate calculation of SUSY particle masses, you can use a dedicated program (ISASUGRA, SUSPECT, SOFTSUSY, ...) to calculate the spectrum and let Pythia6 use this to generate the events.
- SLHA file reading option is available by both Pythia6-based generator modules, Pythia6GeneratorFilter and Pythia6HadronizerFilter. This means that, in addition to Pythia6 SUSY event generation, one can also develop such events from externally generator partons. Let us consider SOFTSUSY,

```
mkdir SLHA_Production
cd SLHA_Production
wget http://www.hepforge.org/archive/softsusy/softsusy-3.1.6.tar.gz
tar -zxvf softsusy-3.1.6.tar.gz
./configure
make
make install
ls spsSLHAfiles/
```

Tutorial, *.out files there?

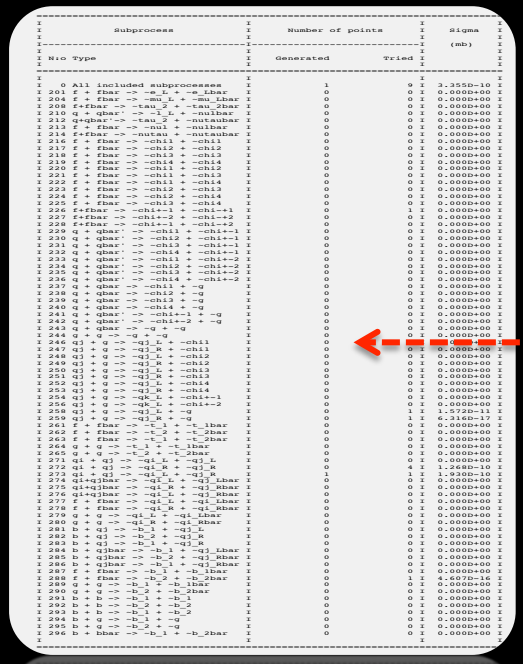
HOW TO GENERATE SUSY SAMPLES USING AN EXTERNAL SLHA SPECTRUM FILE

- For different m0-m1/2 values (so called LM points), the following syntax can be used:

```
➢  ./softpoint.x leshouches < spsSLHAfiles/<your-file>.in > spsSLHAfiles/<yourout>.out
```

Tutorial

- The ouput of this sythax can be included in Pythia-MC simulation:



All SUSY processes xsections !

TUTORIAL3

- Simulate 12 different m0-m1/2 points (starting from m0=50GeV and m1/2=100) with 50GeV steps in the SOFTSUSY. Read these files via Pythia simulation and reconstruct them with FastSimulation in the CMSSW. Finally analyze them related to their cross-section values with the corresponding m0-m1/2 values on the SUSY plane.

- For analysis of these files please ask Francesco Costanza.

SUMMARY

- Here are the topics covered in this lecture:
  - ✓ Analysis chain in CMS Detector
  - ✓ Data stream and event data model in CMSSW
  - ✓ Setting up CMSSW environment
  - ✓ Generator interface in CMSSW
  - ✓ MC/Detector simulations in the CMSSW
    - ✓ Pythia simulations: Generator/Hadronization Filters
    - ✓ cmsDriver usage for the MC production
    - ✓ HLT, RECO and FASTSIM
    - ✓ External files (LHE/SLHA etc.) usage in Pythia Hadronization
    - ✓ Understanding of SUSY m0-m1/2 plane and simulation/simple analysis
  - ✓ Systematics studies for MC (not covered – time? - discussion?)