

# Computing in High Energy Physics

## An Introductory Overview

Frank Gaede  
DESY IT Physics Computing  
Summer Student Lecture  
DESY, August 18, 2009

# Introduction

- The aim of this lecture is to provide an overview and some understanding of the basic concepts of Computing in High Energy Physics
- the (randomly :) chosen topics are of course a subset of possible topics
- we will just scratch the surface of this wide field
- for an overview of topics that are currently discussed and under development see the programme of the CHEP 2006 Conference in Mumbai (next two slides) :

Welcome to CHEP06-Mumbai - Mozilla Firefox

File Edit View Go Bookmarks Tools Help


http://www.tifr.res.in/~chep06/

simulation/geant4 LCIO Linux Conferences DESY IT Group LEO English/Ger... Google MyHome Ctime HOWTO: Remaste...

chep06

## Computing in High Energy and Nuclear Physics

13-17 February 2006, T.I.F.R. Mumbai, India



**TIFR**  
A Deemed University

**Thank you for your attendance. We hope to see you all again for**  
[CHEP'07](#) **at Victoria BC Canada.**

**To view some of the photographs taken during CHEP'06, please click**  
[here](#)

**Bulletin**

- » Bulletin 5
- » Bulletin 4
- » Bulletin 3
- » Bulletin 2
- » Bulletin 1

**Home**

**Poster**

**Programme**

- » Time Table
- » Template
- » Guideline
- » Publications

**Contributions**

**Preconference**

- » Workshop
- » School

**Postconference**

**Participants**

- » Conference
- » SC Workshop
- » School

**General Information**

**Registration**

**Accommodation**

**Social Programme**

**Organisation**

- » International
- » Local
- » Programme

**Past Events**

**Tours**

**About India**

**Contact**

## Tata Institute of Fundamental Research


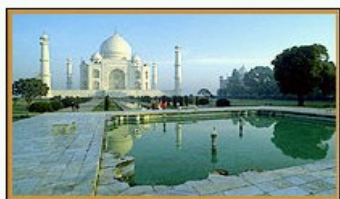
Welcomes you to the international conference

on

## Computing in High Energy and Nuclear Physics

13-17 February 2006, T.I.F.R. Mumbai, India

*CHEP conferences provide an international forum to exchange information on computing experience and needs for the High Energy Physics and Nuclear Physics communities, and to review recent, ongoing and future activities. CHEP conferences are held every 18 months.*

Site designed and maintained by EHEP Group, TIFR.

Transferring data from www.tifr.res.in...

# CHEP 06 Programme I

- **Online Computing**
  - CPU farms for high-level triggering; Farm configuration and run control; Describing and managing configuration data and conditions databases; Online software frameworks and tools
- **Event processing applications**
  - Event simulation and reconstruction; Physics analysis; Event visualisation and data presentation; Toolkits for simulation and analysis; Event data models; Detector geometry models; Specialised algorithms for event processing
- **Software Components and Libraries**
  - Persistency; Interactivity; Foundation and utility libraries; Mathematical libraries; Component models; Object dictionaries; Scripting; Graphics; Use of 3rd party software components (open source and commercial)
- **Software Tools and Information Systems**
  - Programming techniques and tools; Software testing; Configuration management; Software build, release and distribution tools; Quality assurance; Documentation

# CHEP 06 Programme II

- **Computing Facilities and Networking**
  - Global network status and outlook; Advanced technologies and their use in applications; HENP networks and their relation to future grid systems; The digital divide and issues of access, readiness and cost; Collaborative systems, progress in technologies and applications
- **Grid middleware and e-Infrastructure operation**
  - Integral systems (cpu/storage) and their operation and management; Functionality and operation of regional centres; Global usage and management of resources; Grid infrastructure and its exploitation in distributed computing models.
- **Distributed Event production and processing**
  - Development of the distributed computing models of experiments; Real experience in prototypes and production systems; Emphasis on the early days of LHC running.
- **Distributed Data Analysis**
  - Large distributed data-base over wide area network; Low-latency

CHEP 2007

http://www.chep2007.com/

LCIO ilcsoft simulation/geant4 Google DESY IT Group MyHome LEO English/... Dictionary Apple (160)

iGoogle CHEP 2007

**CHEP'07**  
VICTORIA, BC

**International Conference on Computing in High Energy and Nuclear Physics**  
2-7 Sept 2007 Victoria BC Canada

CHEP'07 Home Contact

CHEP 2007

CHEP 2009

http://www.particle.cz/conferences/chep2009/

rBuilder - ...Development itps11 HEPonX LCIO ilcsoft simulation/geant4

CHEP 2009

**CHEP 2009**

Home  
Invitation  
General information  
Organisation  
Programme  
Posters  
Conference venue  
Registration  
Abstract submission  
Registered participants  
Accommodation  
Social events & Tours

**Computing in High Energy and Nuclear Physics**  
Prague | Czech Republic | 21 - 27 March 2009

> Home

**CHEP'09**  
**17<sup>th</sup> International Conference on Computing in High Energy and Nuclear Physics**  
21 - 27 March 2009 Prague, Czech Republic

**CHEP 2010 in Taipei, Taiwan**  
17-22 October, 2010

Hosted by:  
Academia Sinica Grid Computing (ASGC)

Bulletin 2 (PDF) (0...  
Bulletin 3 (PDF) (0...

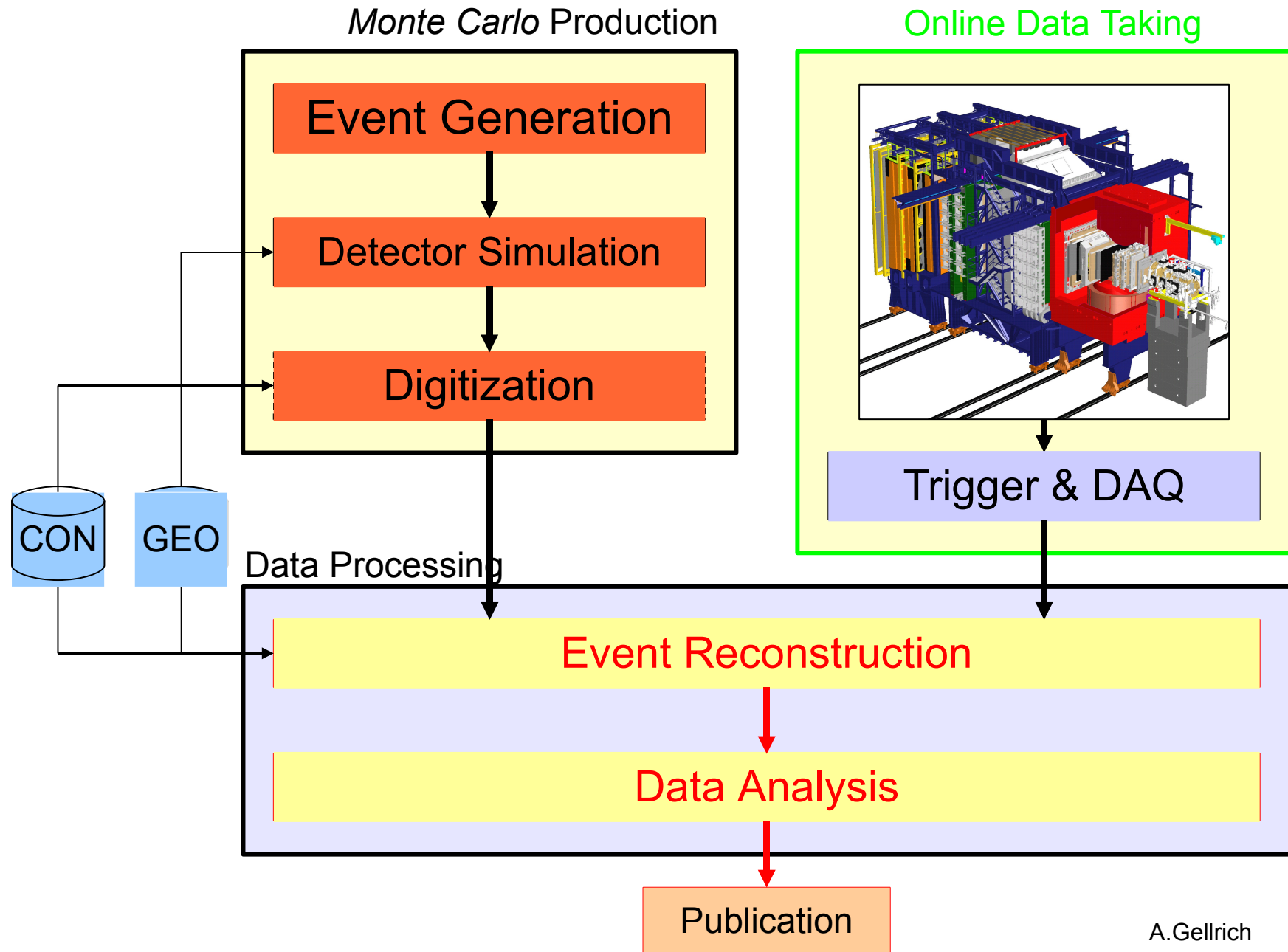
# Selected Topics

- Online Computing - DAQ (data acquisition)
  - Readout software
  - Monitoring
  - Trigger
- Offline Computing
  - Monte Carlo Simulation
  - Reconstruction
  - Analysis - Software Frameworks
- Computing infrastructure (hardware)
  - large PC farms
- GRID Computing

break

---

# HEP Computing overview

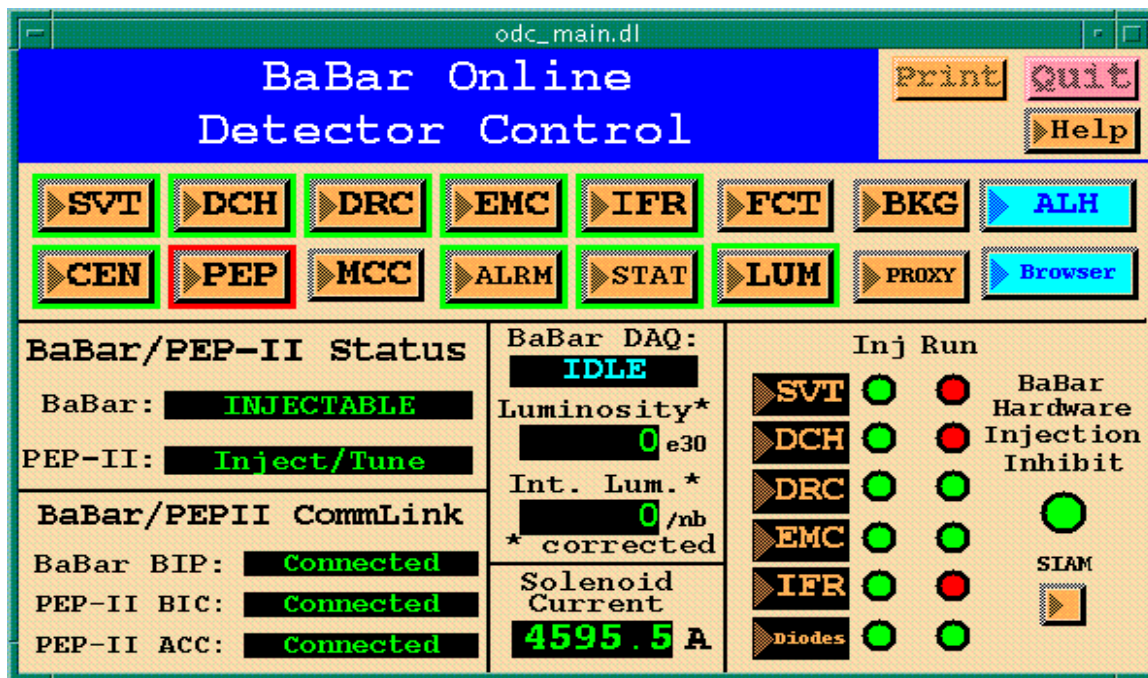




# Online - DAQ

- The Online/DAQ computing makes sure that the interesting physics data is read out from the detector and written to tape/disk (mass storage)
- it is typically divided in three main tasks:
  - **Online Monitoring (slow control)**
    - temperature readings, high voltage, gas supplies...
    - manage the running of the detector
  - **Trigger (software/hardware)**
    - give signal that data needs to be read out 'coz sth. interesting happened in the detector
  - **readout (data flow)**
    - actual readout is tightly coupled to hardware (front end electronics)

# Online detector/run control

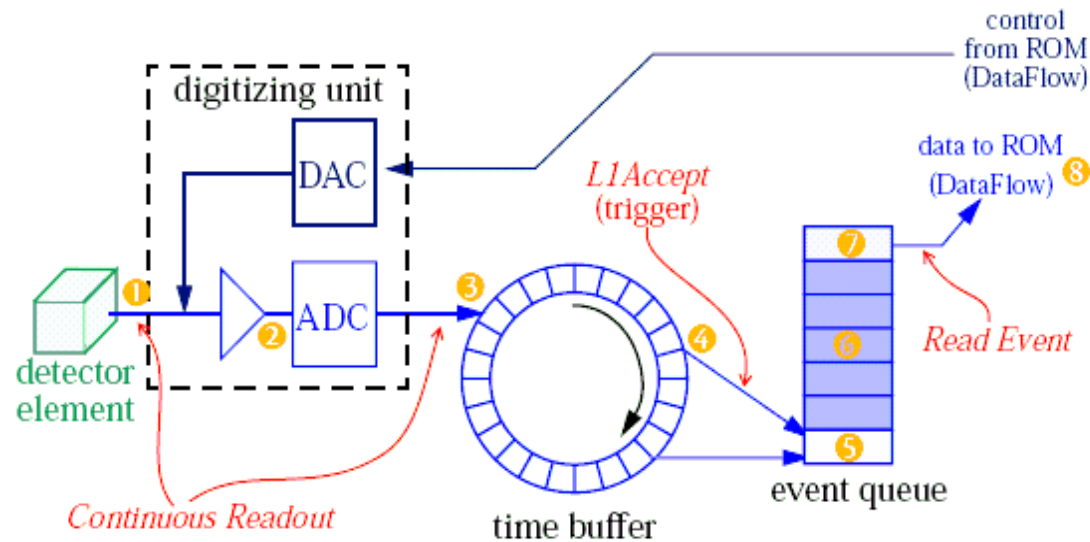


Modern particle physics detectors are run using online software tools, *example: BaBar ODC*

- Online Monitoring – Slow Control systems typically provide a GUI that allows the physicist to run and monitor the detector, by;
  - configuring the detector / online software / trigger
  - start & stop data taking runs
  - monitor temperature readings, high voltage, gas supplies...

# Readout software

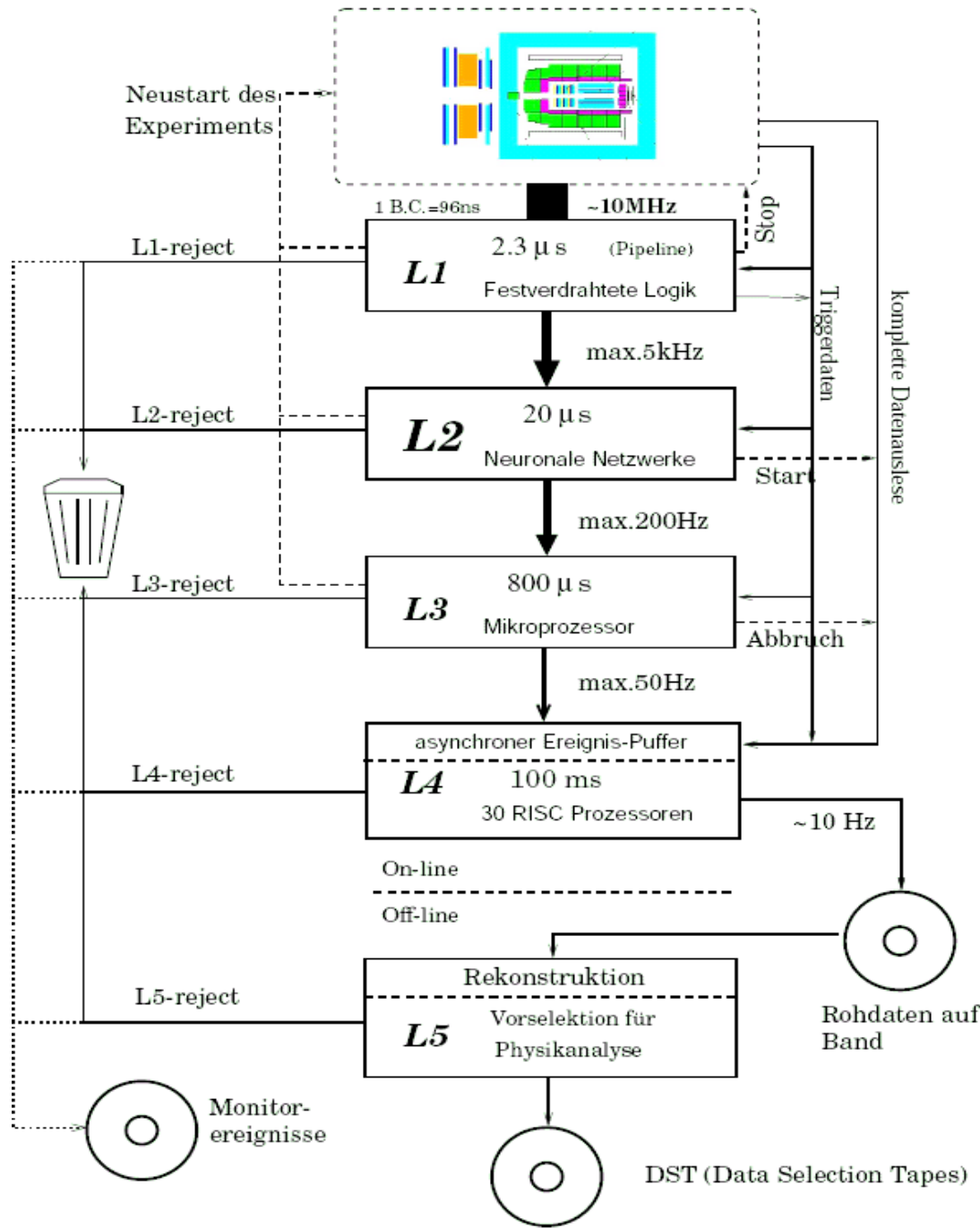
## Specific features (the FEE model)...



*example: front end readout software (Data Flow) of the BaBar experiment*

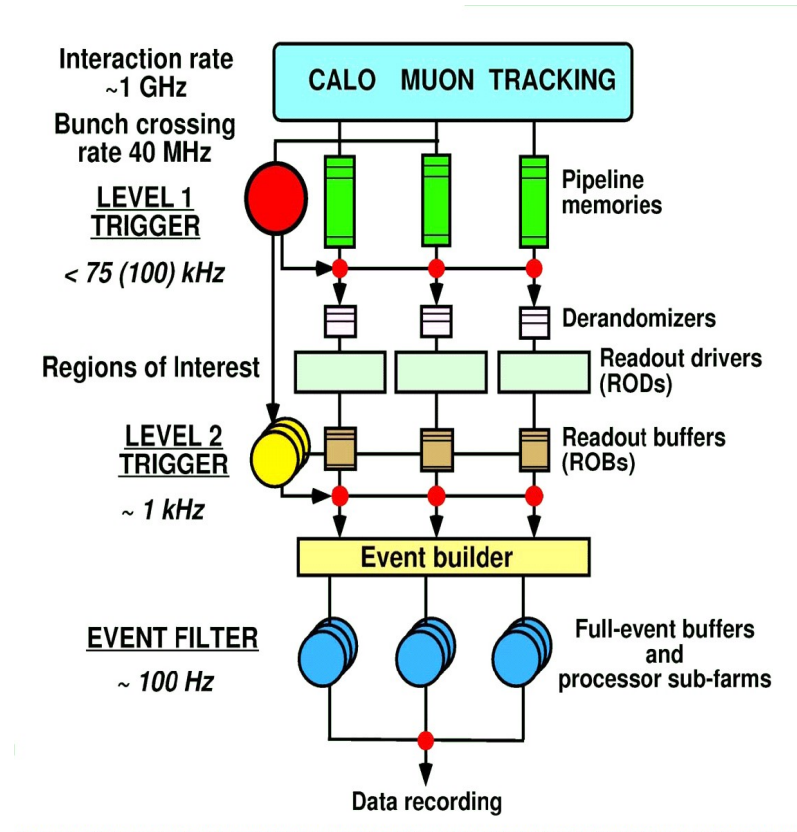
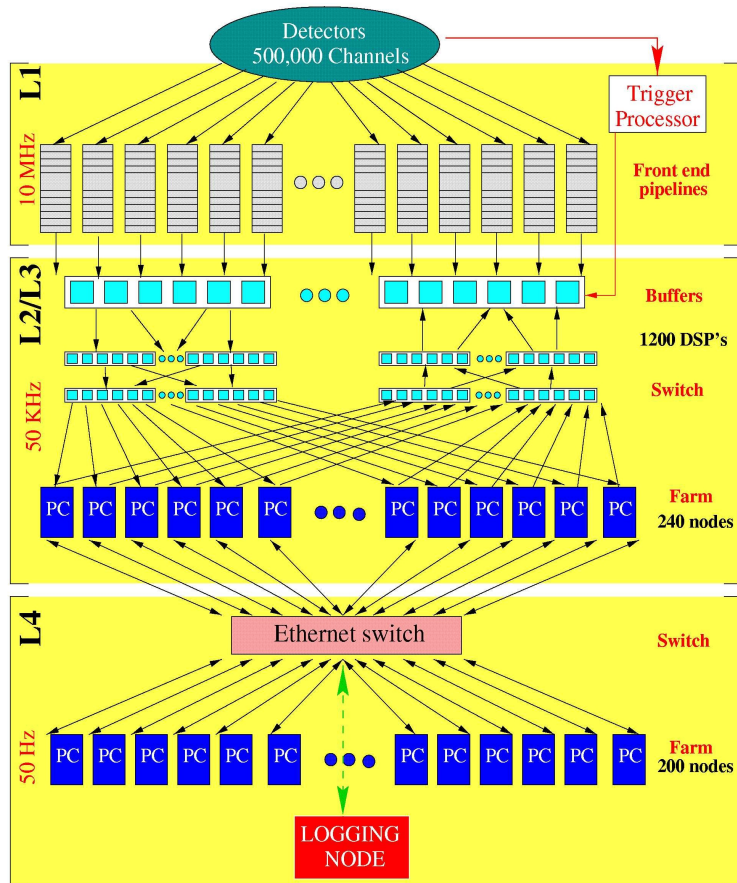
- the readout software is very tightly coupled to the hardware. ie. **front end electronics and readout boards** it typically involves tasks as:
  - buffering of data read out from the detector
  - feature extraction (zero suppression, integrating electronics signals, fitting of peak positions,...)

# Multilevel Trigger System I



- **trigger**
- typically collider
- experiments have far more
- activity in sensitive parts
- than can be read out,
- stored or analyzed
- due to:
  - background from beam-gas interactions
  - high cross sections of (soft) relatively uninteresting physics, e.g. *photoproduction*
- multilevel trigger system reduce the rate through
  - successive application of more advanced algorithms
- buffering pipelines help to reduce the dead time

# Multilevel Trigger System II



other examples:

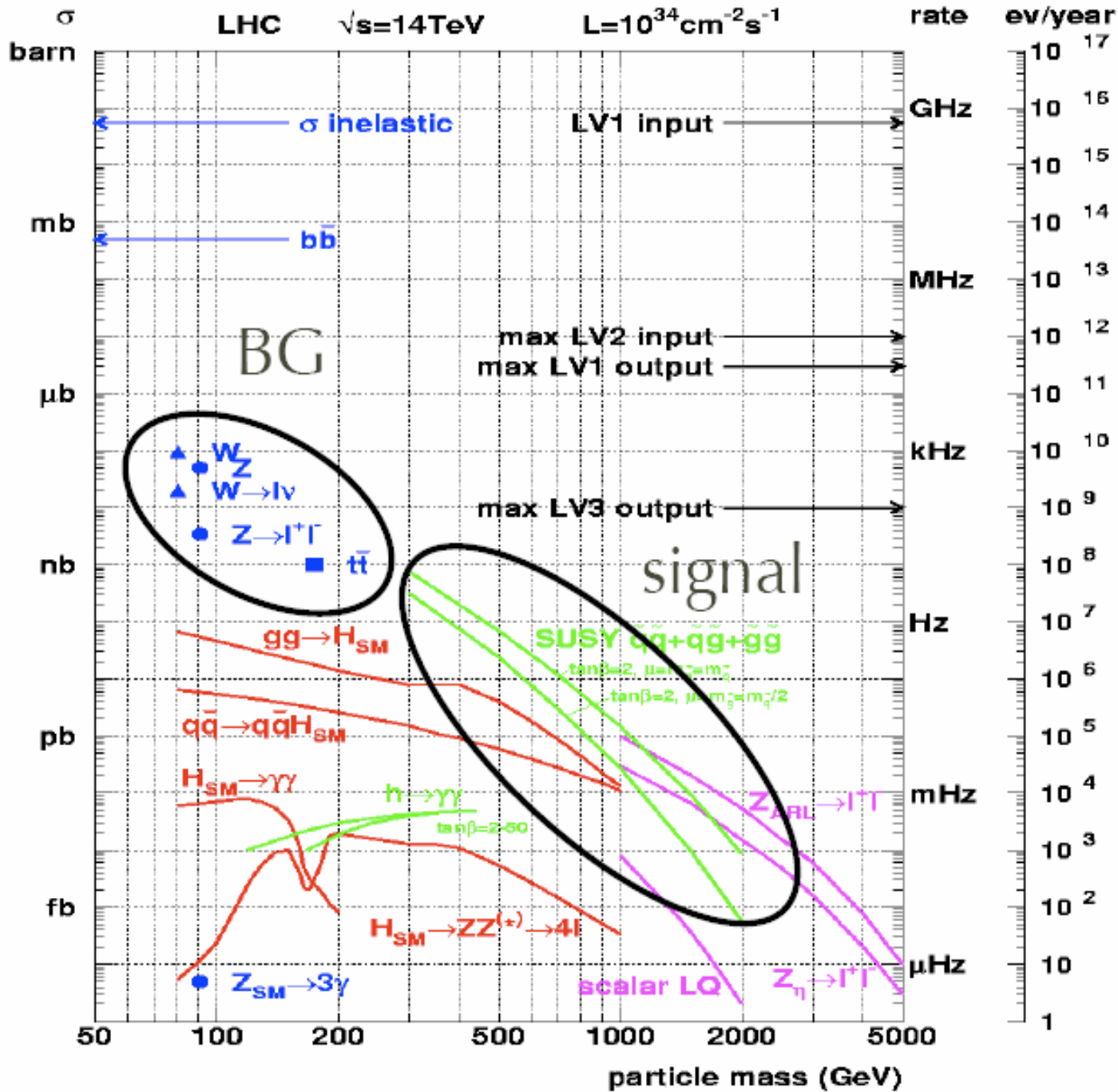
HERA-B

and

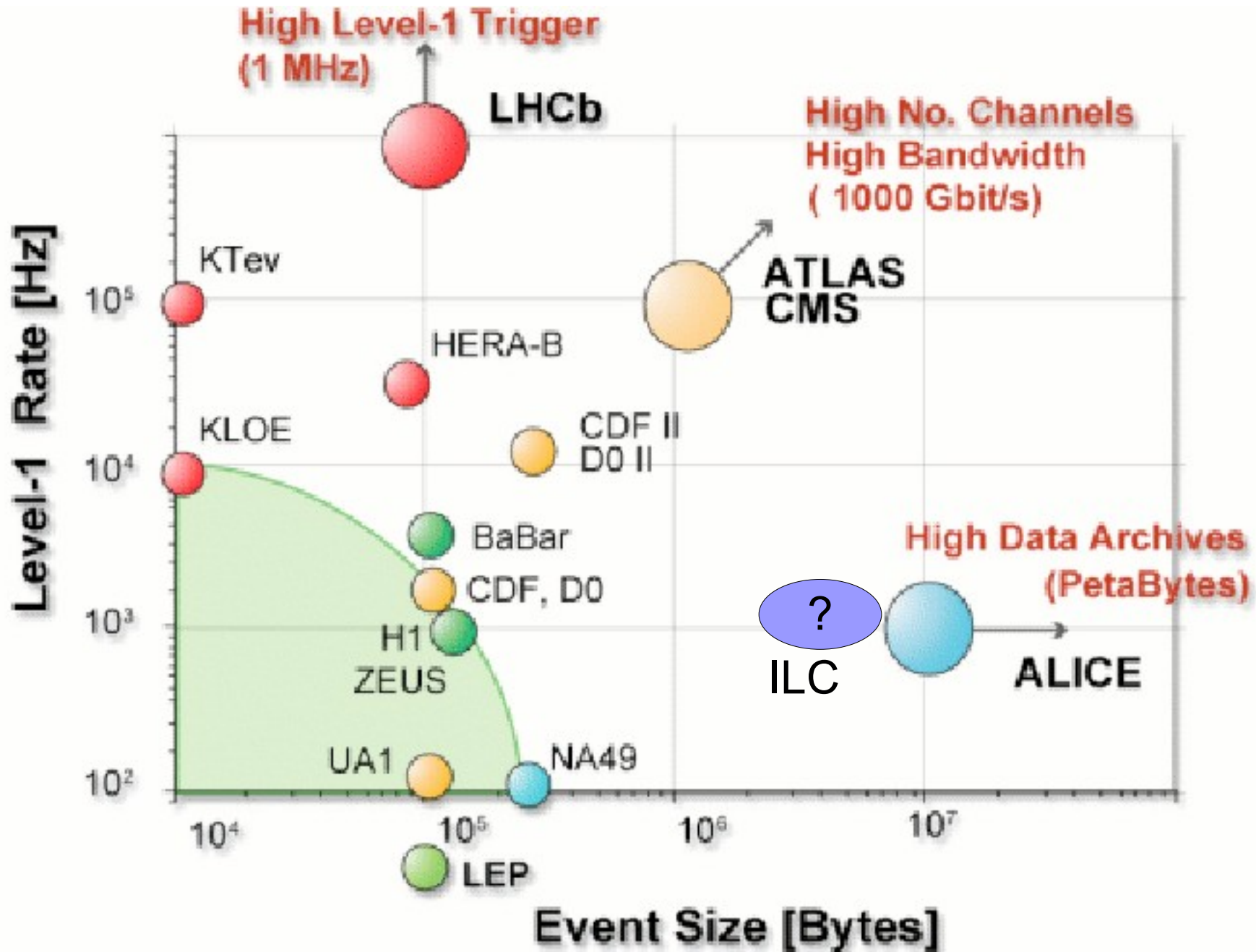
ATLAS trigger systems

exception: planned **ILC** (Linear Collider) due to comparatively low rates and high extrapolated bandwidth (~2015?) no trigger system foreseen but **continuous read out planned** -> no dead time !

# Atlas trigger and physics rates



# Trigger rates and event sizes



# Monte Carlo Simulation



# Why Monte Carlo Simulations ?

- **R&D phase: (planning phase, e.g. ILC )**
  - determine the best geometry of the detector
  - study the (needed) performance of subdetectors
  - compare different designs (competition)
  - evaluate feasibility (bg-rates/radiation)
- **Data taking (running experiments)**
  - study **efficiency** of the detector for all the different physics channels (cross sections)
  - determine the **fundamental parameters** of underlying **physics**

# Monte Carlo Simulation Programs

“Physics”

“Measurement”

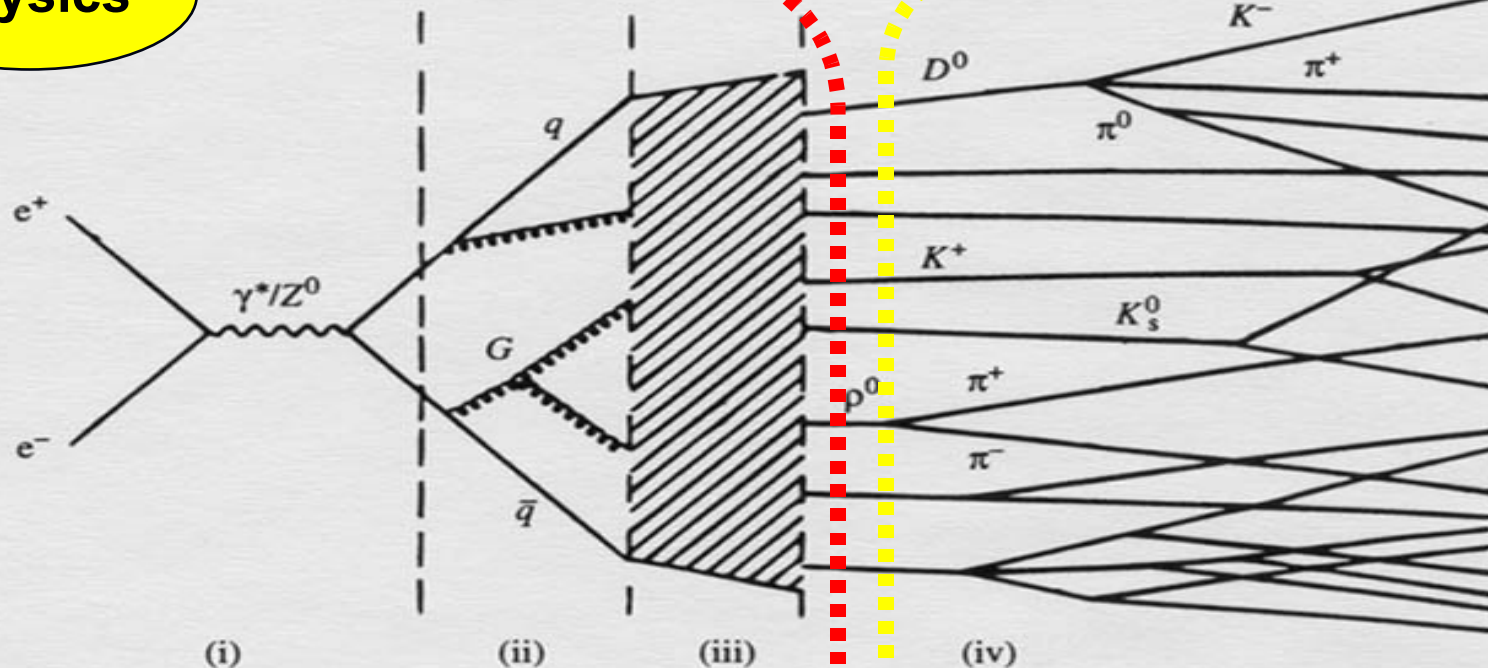
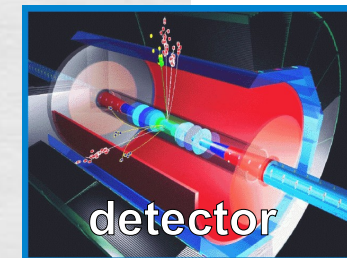


Fig. 25.15. Schematic illustration of  $e^+e^- \rightarrow \text{hadrons}$ .




## Generator:

generates 4-vectors of all particles in the event  
e.g. PYTHIA, Herwig

## Simulation:

detector response to longlived particles -> HITS  
e.g. Geant4

# Simulating the detector response



**European Organization for Nuclear Research**

**Geant4**

A toolkit to simulate the interaction of particles with matter

**Concept**

Geant4 simulates the passage of particles through matter. It provides a complete set of tools for all domains of radiation transport:

- Geometry and Tracking
- Physics processes and models
- Biasing and Scoring
- Graphics and User Interfaces
- Propagation in fields.

Geant4 physics processes describe electromagnetic and nuclear interactions of particles with matter, at energies from eV to TeV. A choice of physics models exists for many processes providing options for applications with different accuracy and time requirements.

The toolkit is developed, maintained and supported by Geant4, a world-wide collaboration of about 100 scientists from many institutions, contributing in their area of expertise. Developers interact constantly with users, and combine efforts to validate physics results for application in high energy physics experiments, space and medical studies.

**Applications**

**High energy and nuclear physics detectors**  
ATLAS, CMS, HARP and LHCb at CERN and BaBar at SLAC

**Accelerator and shielding**

- Linacs for medical use

**Medicine**

- Radiotherapy
- photon, proton and light ion beams
- brachytherapy
- boron and gadolinium neutron capture therapy

**Simulation of scanners**

- PET & SPECT with GATE (Geant4 Application for Tomographic Emission)

**Space**

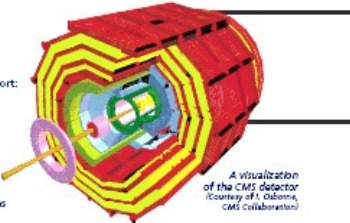
- Satellites
- effect of space environment on components (especially electronic)
- shielding of instruments
- charging effects

**Space environment**

- cosmic ray out-offs

**Astronauts**

- dose estimates



A visualization of the CMS detector (Courtesy of F. Ostrowski, CMS collaboration)

**Advantages**

- Simulates the geometries of complex setups efficiently
- Provides configurations of physics processes for application areas
- Enables user to tailor simulation components and address accuracy needs
- Performant and adaptable
- Easy to embed into specific applications

**The BeppoColombo Mercury orbiter** (Courtesy of ESA)

**Simulation of small PET scanner using GATE** (Courtesy of the OpenGATE collaboration)

**A view of the ATLAS detector** (Courtesy of S. Tanaka, ATLAS collaboration)

**XMM-Newton X-ray telescope: the effects of the radiation environment on its instruments was modeled with Geant4 prior to launch in 1999** (Courtesy of ESA)

The European Organization for Nuclear Research (CERN), one of the world's foremost particle physics laboratories, has introduced an active Technology Transfer policy to establish its competence in European industrial and scientific environments, and to demonstrate clear benefits of the results obtained from the considerable resources made available to particle physics research.

Technology Transfer is an integral part of CERN's principal mission of fundamental research.

**CERN Technology Transfer**

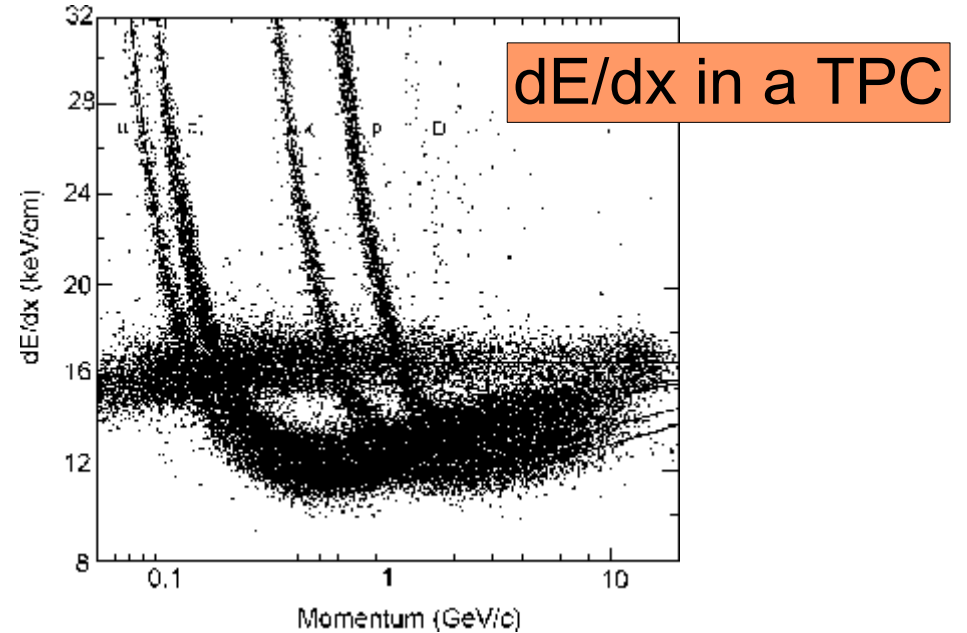
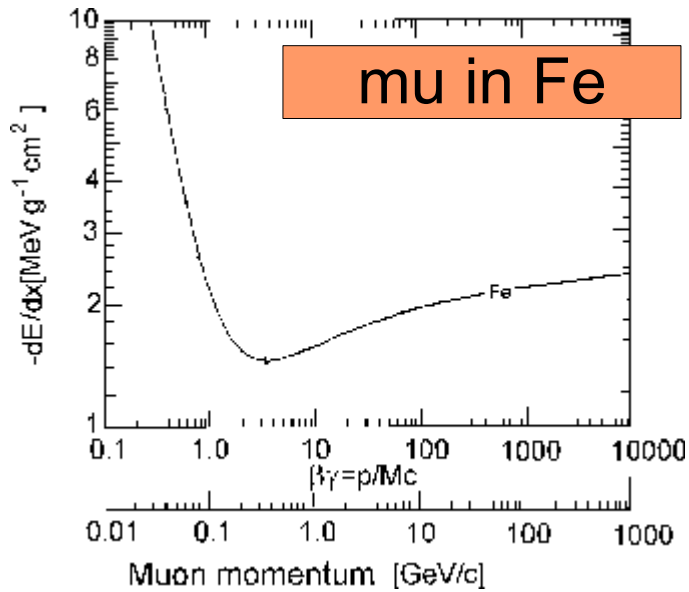
<http://www.cern.ch/ttdb/Technologies/geant4>

- example: **geant4** - a C++ toolkit that simulates passage of particles through matter using "known physics":
- particle decay (lifetime/branching ratios)
- photoelectric effect
- Compton scattering
- pair creation (EM-cascade)
- energy loss due to ionization (exaltation), multiple scattering
- Cherenkov radiation
- Positron - Electron Annihilation
- Bremsstrahlung
- ~hadronic interactions
- cross section tables
- parameterizations
- ... many more ...

# passage of particles through matter

- simulating the detector response only meaningful if the the underlying physics is known well enough
- in general true for all electromagnetic interactions
  - ionization in tracking detectors
  - electromagnetic showers in calorimeters
    - EM-cascade due to repeating Bremsstrahlung/pair-creation
    - QED has a non divergent perturbation series
- in general not so true for hadronic interactions
  - QCD has divergent perturbation series in
  - low energy (soft) hadron interactions

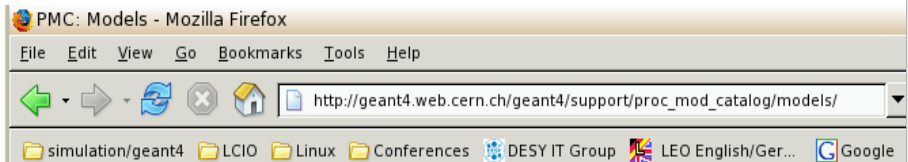
# Ionization energy loss



$$-\frac{dE}{dx} = \kappa z^2 \cdot \frac{Z}{A} \cdot \frac{1}{\beta^2} \left[ \frac{1}{2} \ln \frac{2m_e c^2 \gamma^2 \beta^2}{I^2} E_{\text{kin}}^{\text{max}} - \beta^2 - \frac{\delta}{2} \right]$$

Bethe-Bloch Formula

# hadronic shower parameterization



## Models

- **electromagnetic:**
  - standard:
  - low energy
  - Penelope:
- **hadronic:**
  - **Elastic:** [generic elastic](#), [medium and high energy elastic](#), [coherent elastic pp, nn, coh](#)
  - **Precompound**
  - **Leading particle bias**
  - **Cascade:** [Bertini cascade](#), [Binary cascade](#), [Binary light ion](#)
  - **Low energy parameterized:** [pi+ inelastic](#), [pi- inelastic](#), [K+ inelastic](#), [K- inelastic](#), [K0L inelastic](#), [K0S inelastic](#), [proton inelastic](#), [neutron inelastic](#), [lambda inelastic](#), [sigma+ inelastic](#), [sigma- inelastic](#), [xi- inelastic](#), [xi0 inelastic](#), [omega- inelastic](#), [anti-proton inelastic](#), [anti-neutron inelastic](#), [anti-lambda inelastic](#), [anti-sigma+ inelastic](#), [anti-sigma- inelastic](#), [anti-xi- inelastic](#), [anti-xi0 inelastic](#), [anti-omega- inelastic](#), [deuteron inelastic](#), [triton inelastic](#), [alpha inelastic](#)
  - **High energy parameterized:** [pi+ inelastic](#), [pi- inelastic](#), [K+ inelastic](#), [K- inelastic](#), [K0L inelastic](#), [K0S inelastic](#), [proton inelastic](#), [neutron inelastic](#), [lambda inelastic](#), [sigma+ inelastic](#), [sigma- inelastic](#), [xi- inelastic](#), [xi0 inelastic](#), [omega- inelastic](#), [anti-proton inelastic](#), [anti-neutron inelastic](#), [anti-lambda inelastic](#), [anti-sigma+ inelastic](#), [anti-sigma- inelastic](#), [anti-xi- inelastic](#), [anti-xi0 inelastic](#), [anti-omega- inelastic](#)
  - **High energy:** [Fritiof-CHIPS \(FTFC\)](#), [Fritiof-precompound \(FTFP\)](#), [Quark-gluon string CHIPS \(QGSC\)](#), [Quark-gluon string precompound \(QGSP\)](#)
  - **Nucleus-nucleus:** [electromagnetic dissociation](#), [abrasion/ablation](#), [Binary light ion](#)
  - **Gamma- and Lepto-Nuclear:** [electro-nuclear](#), [gamma-nuclear](#), [muon-nuclear](#)
  - **Neutrons:** [generic capture](#), [generic fission](#), [high precision elastic](#), [high precision inelastic](#), [high precision capture](#), [high precision fission](#), [high precision/parameterized elastic](#), [high precision/parameterized inelastic](#), [high precision/parameterized capture](#), [high precision/parameterized fission](#)

http://geant4.web.cern.ch/geant4/support/proc\_mod\_catalog/models/hadronic/G4ElasticHadrNucleusHE.html

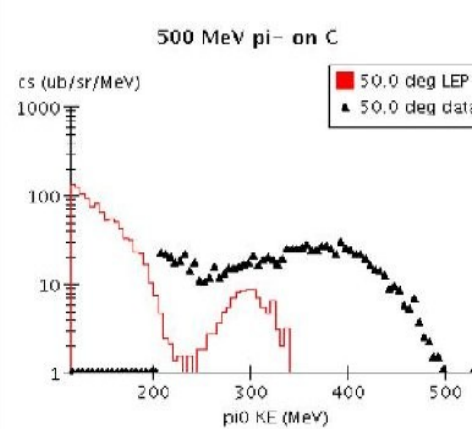


Figure 1: Current GEANT4 LEP physics list setting against data (Ouyang, Peterson 1992)

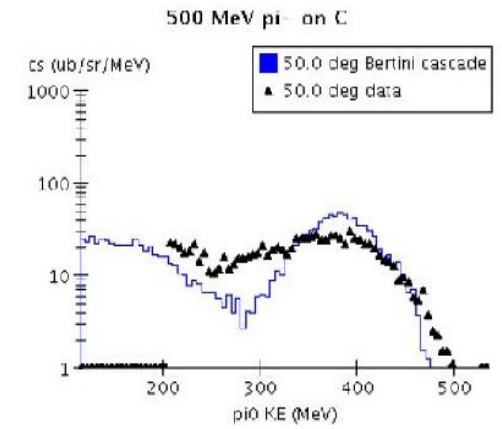
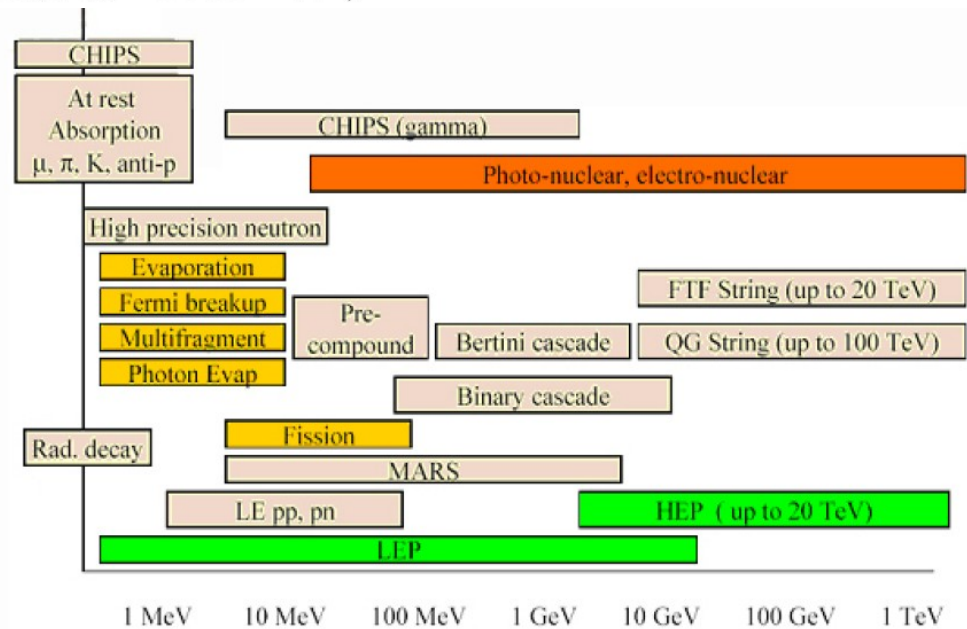


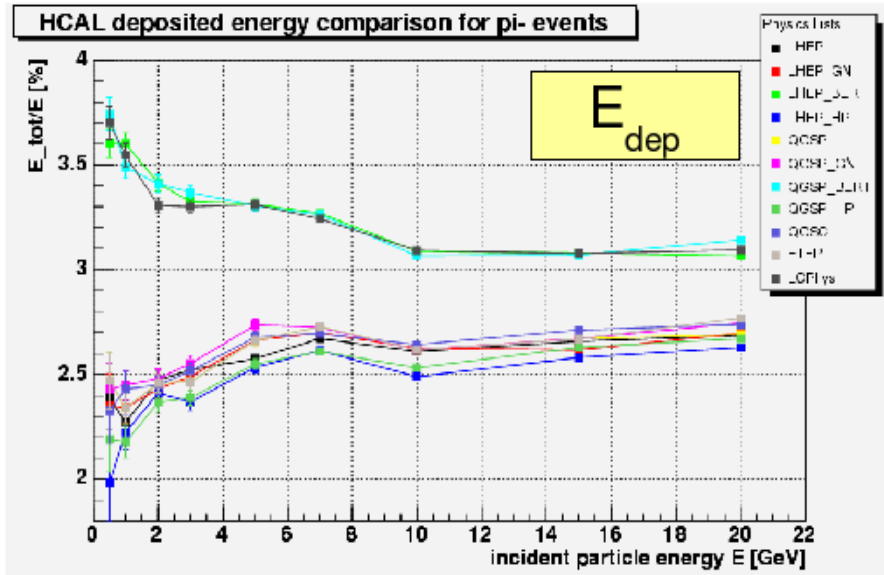
Figure 2: Bertini cascade model



# hadronic showers in ILC-HCal prototype

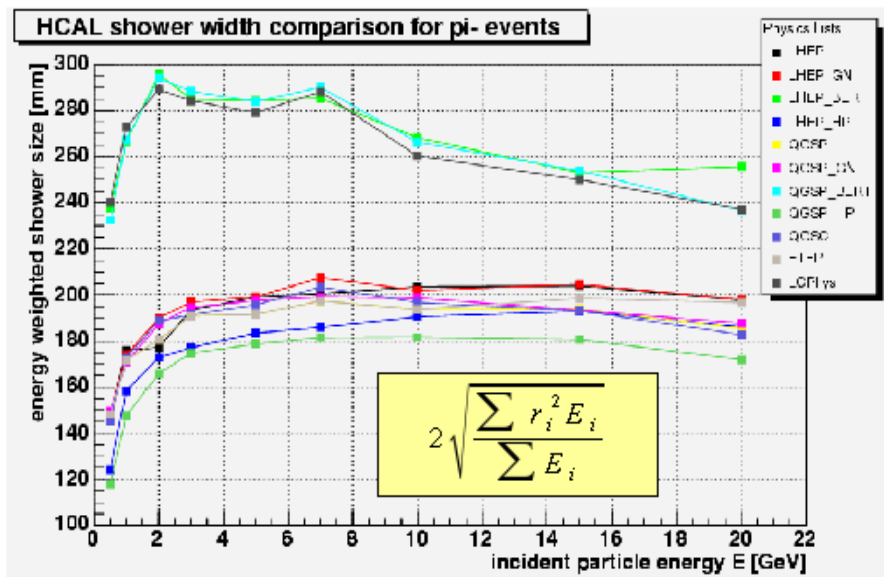
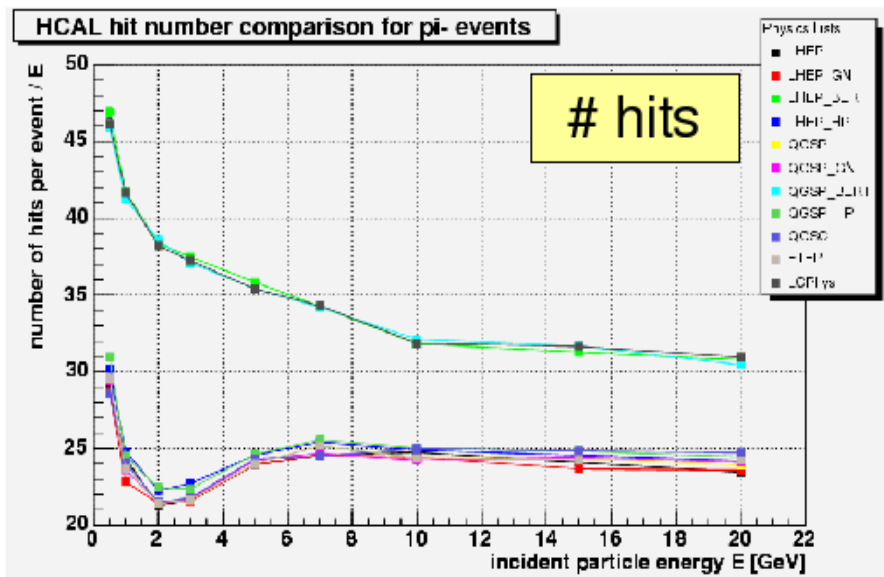
## GEANT4.6.1

- P. Melchior, summer student programme 2004
- need verification with testbeam -> still ongoing !



=> only two classes of physics lists in given energy domain:

- LEP like parameterization
- Bertini cascade



# physics processes in geant4

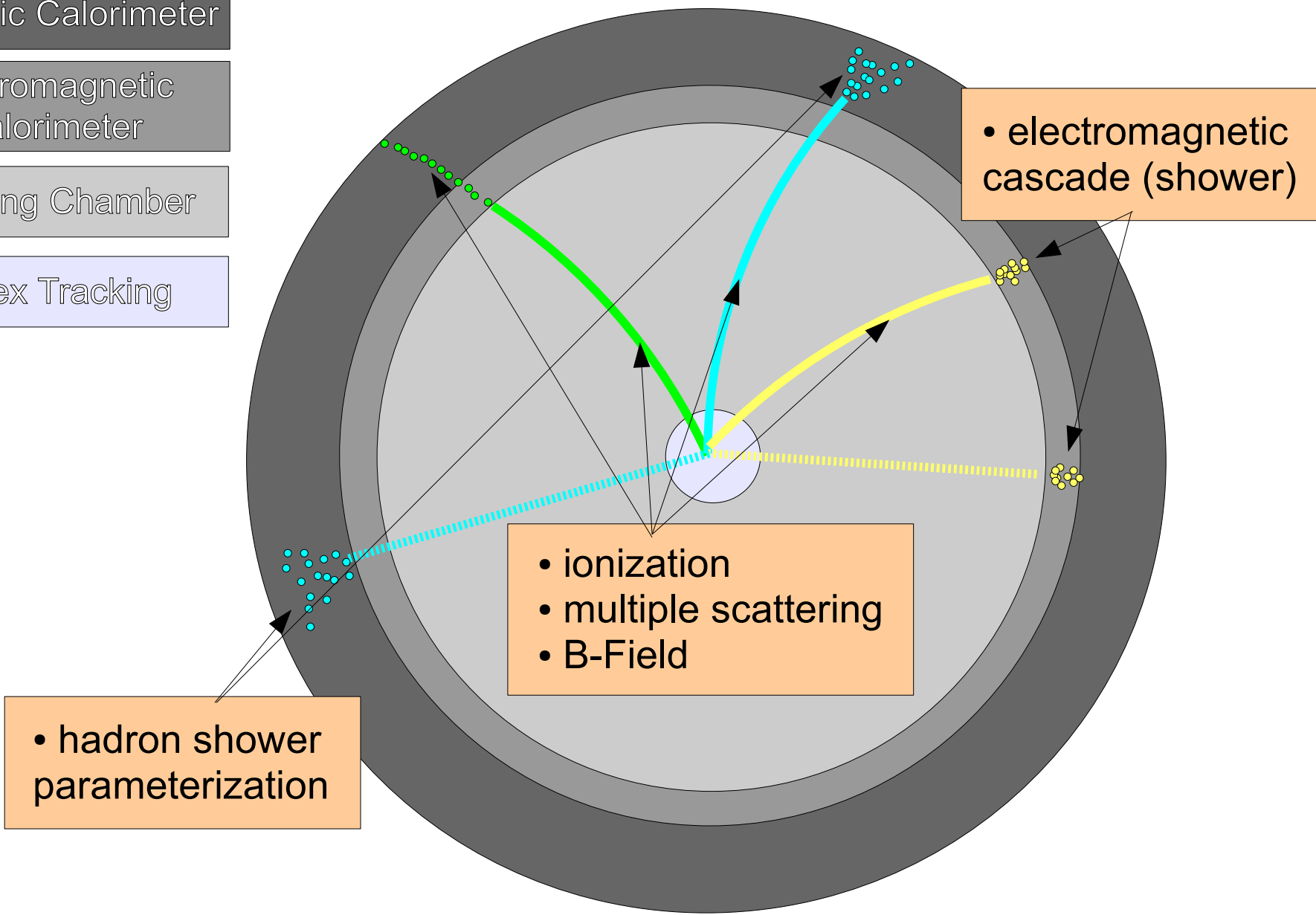
- each particle has its own list of applicable processes
- at each step, all processes listed are invoked to get random physical interaction lengths (Monte Carlo Method!)
- the process with the shortest interaction length limits the step
- each process can have any of the following actions:
  - **AtRest** (e.g. muon decay at rest)
  - **AlongStep** - continuous process (e.g. ionization)
  - **PostStep** - discrete process (e.g. decay in flight)
- every action that is applied to a particle is defined as a process:
  - transportation (E,B fields)
  - decay
  - interactions with Material (ionization, delta-electrons,.....)
  - step length cut off



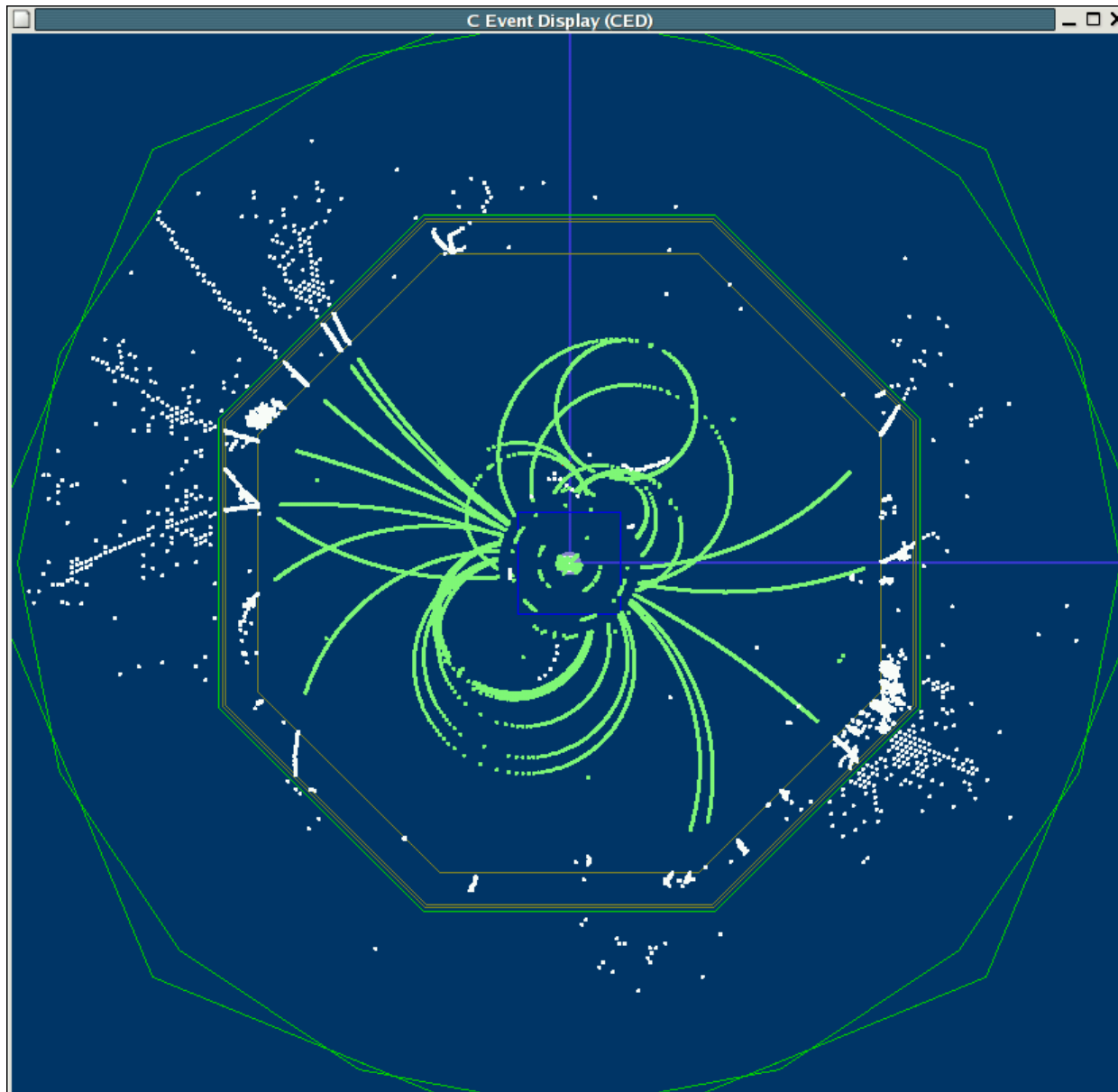
# dominating processes in simulation

and in a real detector...

- Hadronic Calorimeter
- Electromagnetic Calorimeter
- Tracking Chamber
- Vertex Tracking



# simulation output: hits

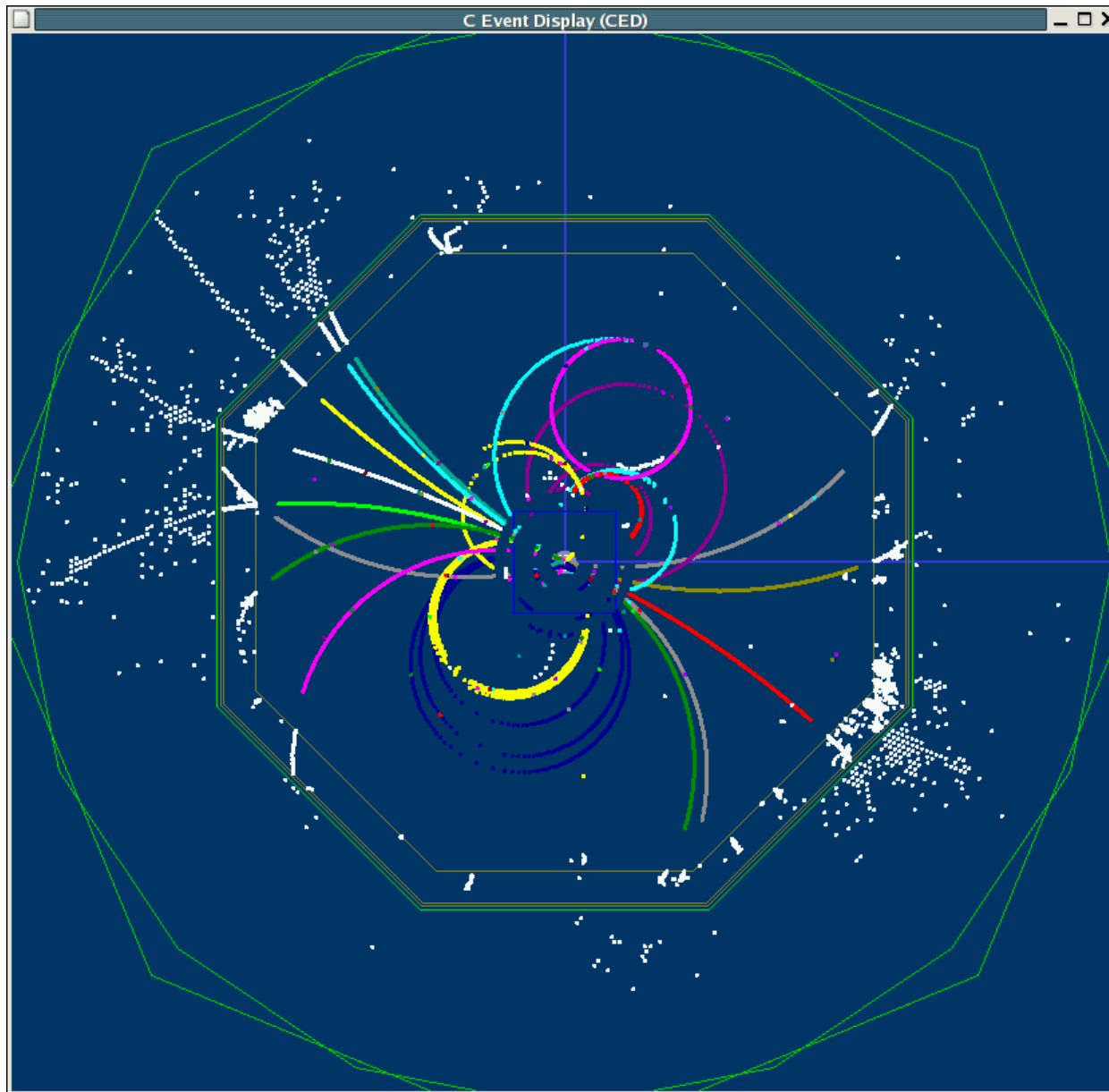


- **simulation** output:
  - *calorimeter hits*
    - cell position
    - amplitude (energy)
  - *tracker hits*
    - amplitude
    - $dE/dx$
- **digitization**
  - smear hits
  - apply noise
    - electronics
    - physics

# Reconstruction

- now we have simulated the detector response (hits) to the generated event
- ideally this is **indistinguishable from real data**
  - (not true in practice as of course MC-Truth is conserved for cross checks !)
- next step: **Reconstruction** – combining hits to reconstructed particles in order to perform the actual physics analysis

# reconstruction - tracking



- **tracking**  
( pattern recognition):
  - *track finding*
    - combine hits that most likely belong to one particle
  - *track fitting*
    - apply fit to all hits taking B-field into account
      - 'Helix approximation'
- Kalman Filter typically perform both steps in one, taking fit to previous points as estimate to next point

# track parameters - fitting

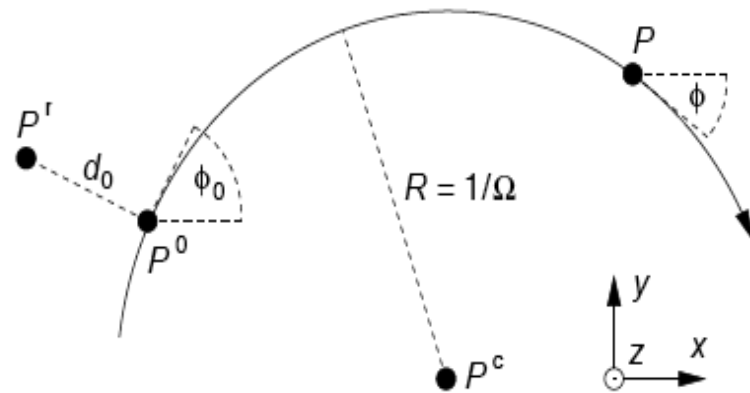


Figure 1: The projection of a helix segment in the  $xy$  plane is a part of an arc with centre  $P^c$  and radius  $R$ . The direction of the particle is shown with the arrow at the arc. All track parameters are given relative to the reference point  $P^r$ .

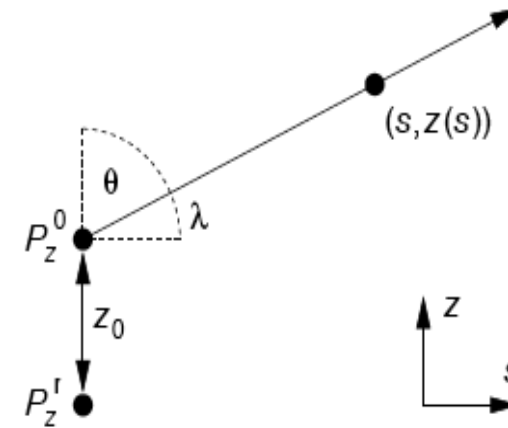
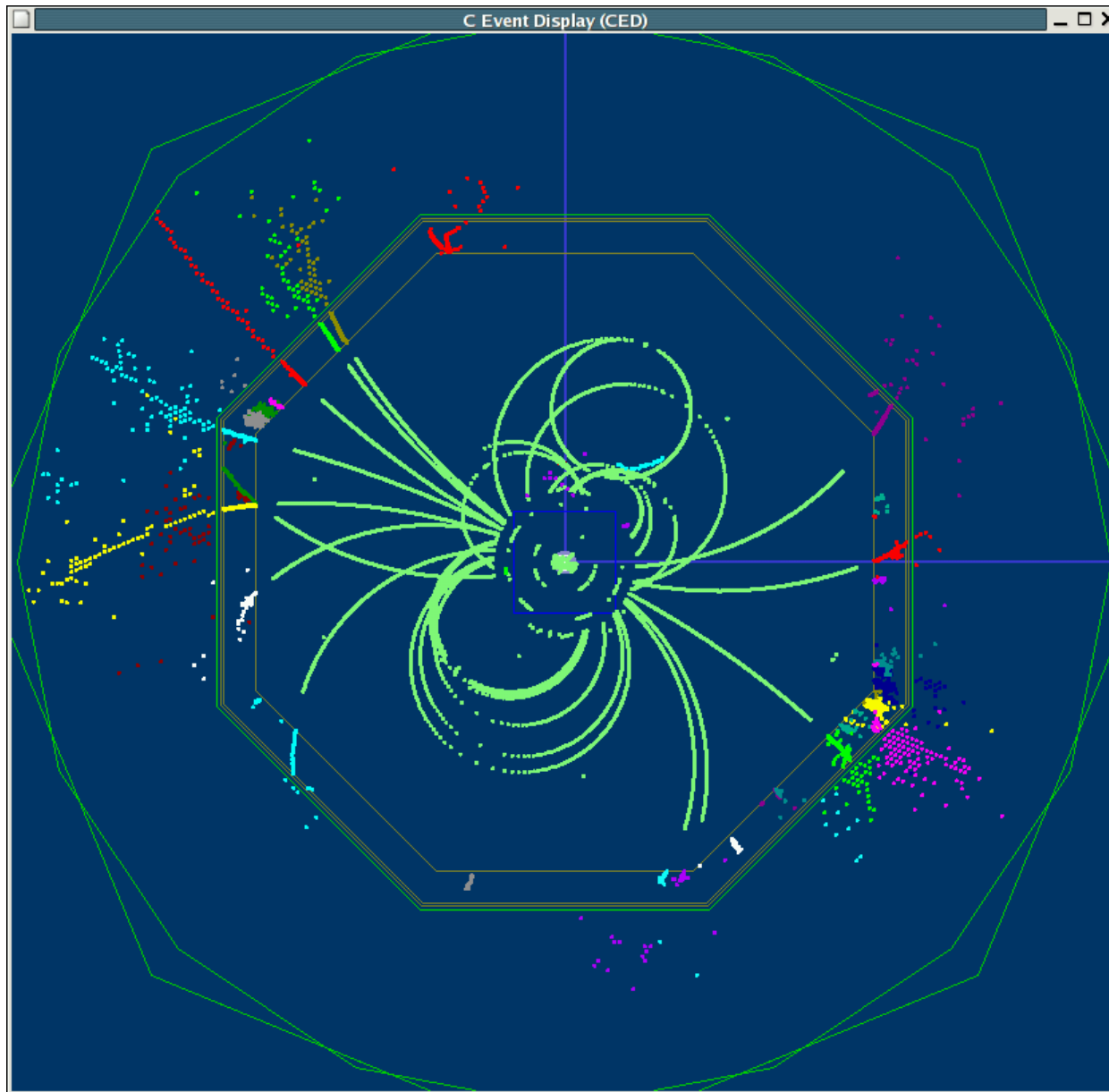


Figure 2: The projection of a helix in the  $sz$  plane is a straight line (see Eq. [10](#)). The variable  $s$  at a point  $P$  is the arc length in the  $xy$  plane from  $P^0$  to  $P$ . This also implies that  $s = 0$ , if  $z = z_0$ .

- a charged particle in a homogenous field describes a helix
- (except for energy loss and multiple scattering)
- a helix is described by five parameters, e.g
- **$d_0$ ,  $\phi_0$ ,  $\omega$ ,  $\tan \lambda$ ,  $z_0$**
- after identifying the hits from one particle fitting the above parameters determines the 3-momentum (and charge):

$$p_t = a \left| \frac{B}{\Omega} \right|, p_x = p_t \sin \phi_0, p_y = p_t \cos \phi_0, p_z = p_t \tan \lambda$$

# reconstruction - clustering



- **clustering**
  - combine hits that most likely belong to a particle shower
  - compute energy of shower
- typically based on some metric that links nearby hits
  - “Nearest-Neighbor”
- could additionally use
  - tracks as seeds
  - hit energy amplitude

# example NNClustering

```
template <class In, class Out, class Pred >
void cluster( In first, In last, Out result, Pred* pred ) {

    typedef typename In::value_type GenericHitPtr ;
    typedef typename Pred::hit_type HitType ;

    typedef std::vector< GenericCluster<HitType >* > ClusterList ;

    ClusterList tmp ;
    tmp.reserve( 256 ) ;

    while( first != last ) {

        for( In other = first+1 ; other != last ; other ++ ) {

            if( pred->mergeHits( (*first) , (*other) ) ) {

                if( (*first)->second == 0 && (*other)->second == 0 ) { // no cluster exists

                    GenericCluster<HitType >* c1 = new GenericCluster<HitType >( (*first) ) ;

                    c1->addHit( (*other) ) ;

                    tmp.push_back( c1 ) ;

                }
                else if( (*first)->second != 0 && (*other)->second != 0 ) { // two clusters

                    (*first)->second->mergeClusters( (*other)->second ) ;

                }
                else { // one cluster exists

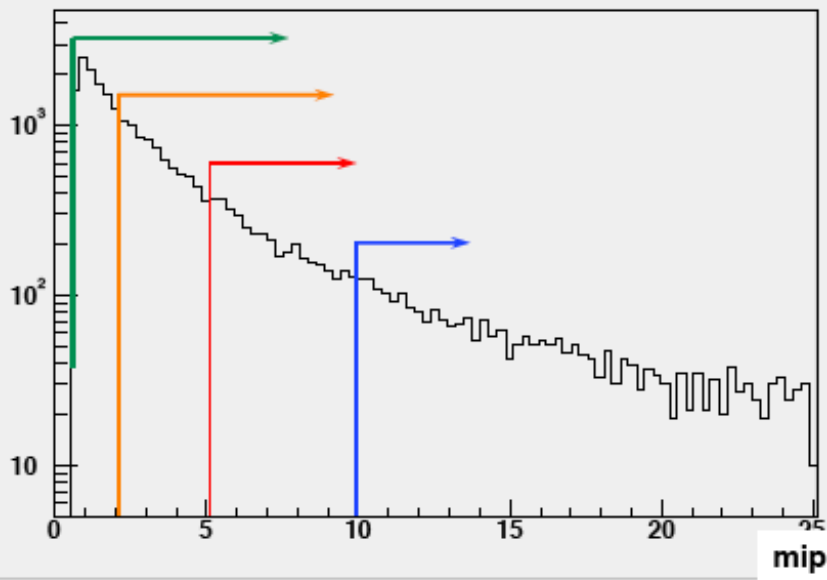
                    if( (*first)->second != 0 ) {
                        (*first)->second->addHit( (*other) ) ;
                    }
                    else {
                        (*other)->second->addHit( (*first) ) ;
                    }
                }
            }
        } // dCut
    }
    ++first ;
}
// remove empty clusters
```

- simplest algorithm: nearest neighbor clustering :
  - loop over all hit pairs
  - merge hits into one cluster if  $d( h_1, h_2 ) < cut$
  - $d()$  could be 3D-distance – typically more complicated

- in real life the NNClustering does not provide the necessary accuracy, e.g. in dense jets where showers overlap
- -> more advanced algorithms needed and under development/study, e.g.
  - *tracking like clustering*
  - *genetic algorithms*
  - *unsupervised learning, ....*

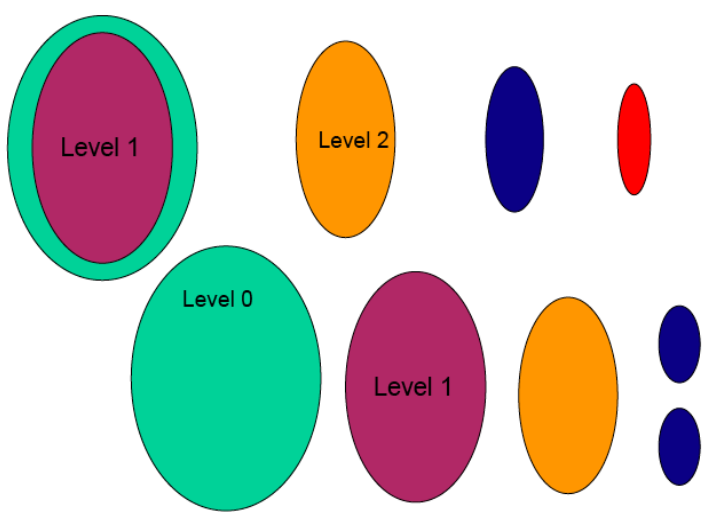
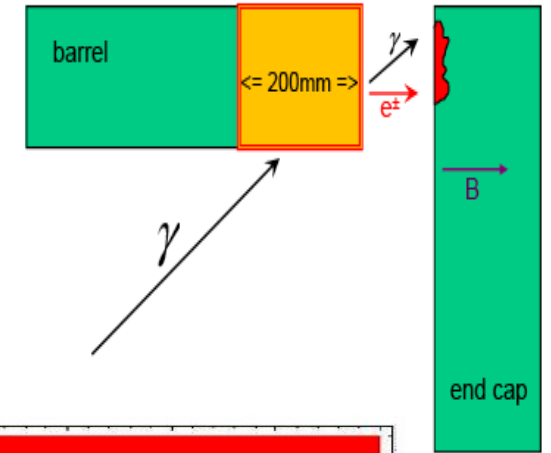
# advanced example: photon shower ID

P.Krstonosic

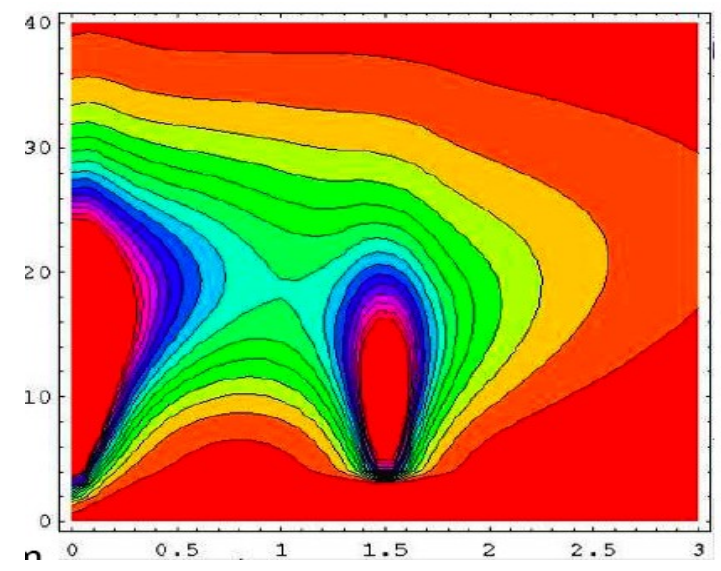


Choose  
N threshold levels  
(N=10 at the moment)  
and get N sets of hits

For each set do a  
NN clustering  
Only in particular  
set!!



Single photon

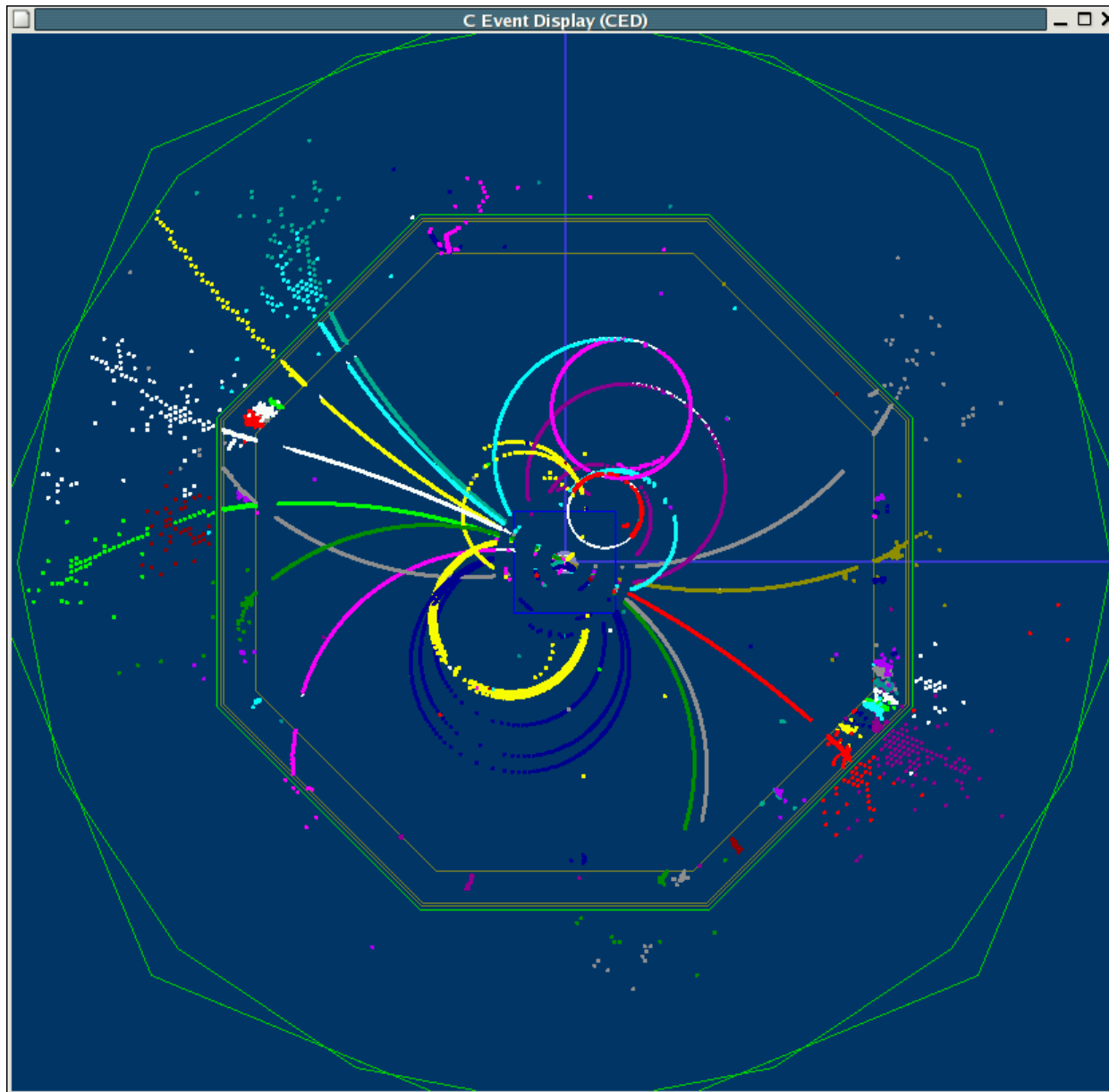


?!

- sophisticated photon ID
- based on N clusterings with different thresholds



# reconstruction - PFA

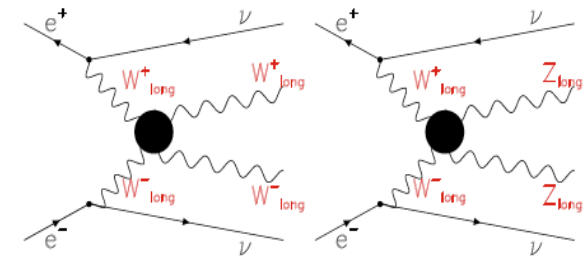
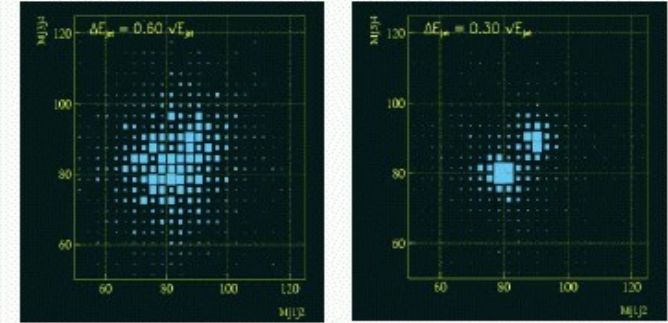


- **track cluster merging**  
*(particle flow)*
- extrapolate the tracks into the calorimeter and merge with clusters that are consistent with the momentum/direction and energy of the track
- the unmerged clusters are then the neutral particles
- ideally one would like reconstruct every single particle (PFA)

# example: reconstruction @ the ILC

- general ILC detector features:
  - precision tracking
  - precision vertexing
  - high granularity in calorimeters
    - ( Ecal ~1cm, Hcal ~1-5cm)
- important: **very high jet-mass resolution**  
 ~30%/sqrt(E/GeV)

WW-ZZ separation



- dominant contribution (E<50 GeV):
  - Hcal resolution
  - confusion term

➔

## Particle Flow

- reconstruct all single particles
- use **tracker** for **charged particles**
- use **Ecal** for **photons**
- use **Hcal** for **neutral hadrons**

$$\sigma_{E_{jet}}^2 = \epsilon_{trk}^2 \sum_i E_{trk,i}^4 + \epsilon_{Ecal}^2 E_{Ecal}^2 + \epsilon_{HCal}^2 E_{HCal}^2 + \sigma_{confusion}^2$$

$$\epsilon_{trk} = \delta(1/p) \approx 5 \cdot 10^{-5}, \quad \epsilon_{Ecal} = \frac{\delta E}{\sqrt{E}} \approx 0.1, \quad \epsilon_{HCal} \approx 0.5$$

# Particle flow calorimetry @ ILC

## Hardware:

★ Need to be able to resolve energy deposits from different particles

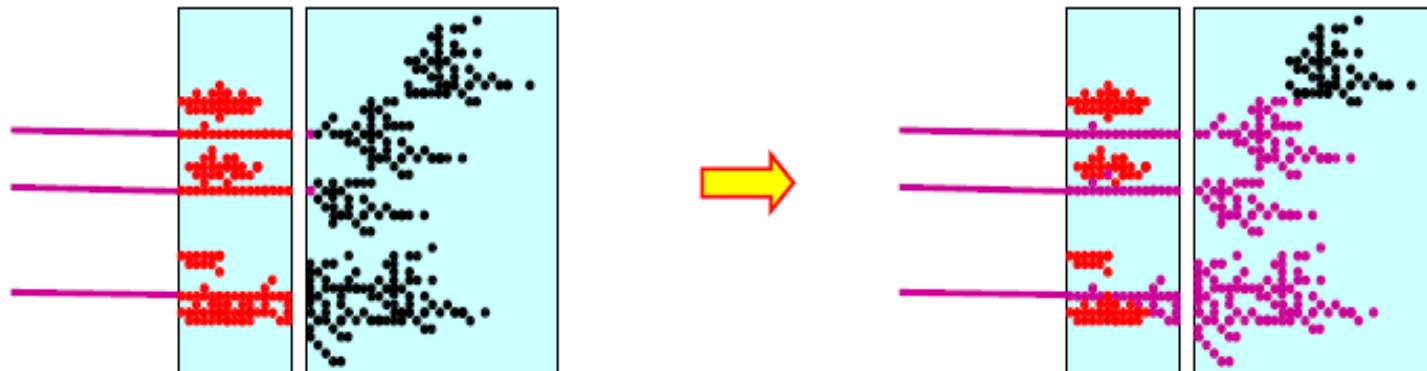
➔ **Highly granular detectors (as studied in CALICE)**



## Software:

★ Need to be able to identify energy deposits from each individual particle !

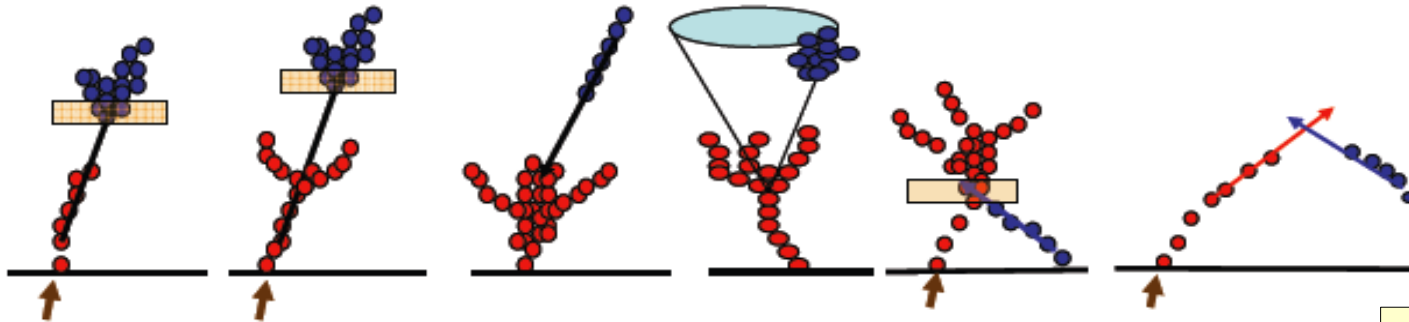
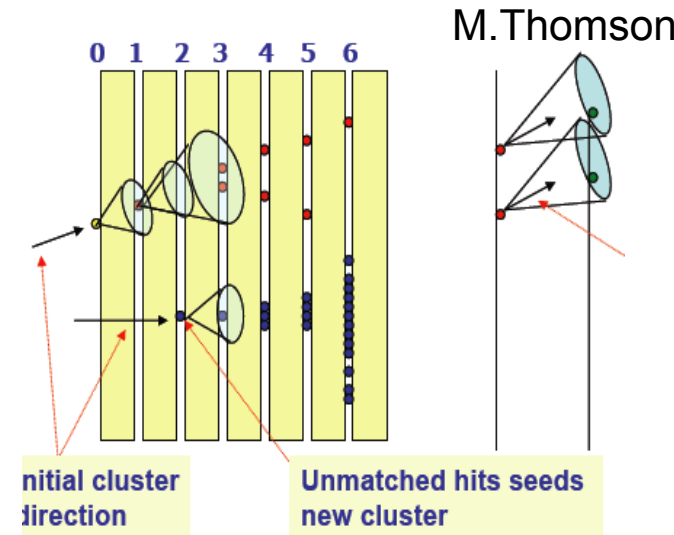
➔ **Sophisticated reconstruction software**



★ **Particle Flow Calorimetry = HARDWARE + SOFTWARE**

# example PandoraPFA clustering

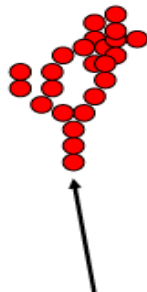
- i. Preparation (MIP hit ID, isolation, tracking)
  - ii. Loose clustering in ECAL and HCAL
  - iii. Topological linking of clearly associated clusters
  - iv. Coarser grouping of clusters
  - v. Iterative reclustering
  - vi. Photon Recovery (NEW)
  - vii. Fragment Removal (NEW)
  - viii. Formation of final Particle Flow Objects (reconstructed particles) – not very sophisticated
- Order inter-changable



If track momentum and cluster energy inconsistent : **RECLUSTER**

e.g.

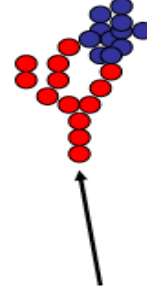
30 GeV



10 GeV Track

18 GeV

12 GeV



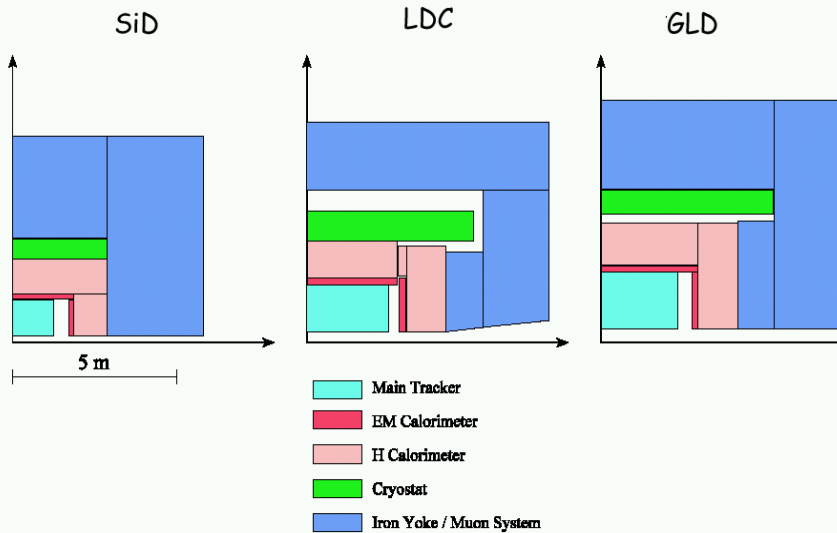
Pandora is the most sophisticated and best performing PFA to date

# example: ILC - Detector Concept Study

recently\* three international detector concepts in R&D

Concepts currently studies differ mainly in **SIZE** and **aspect ratio**

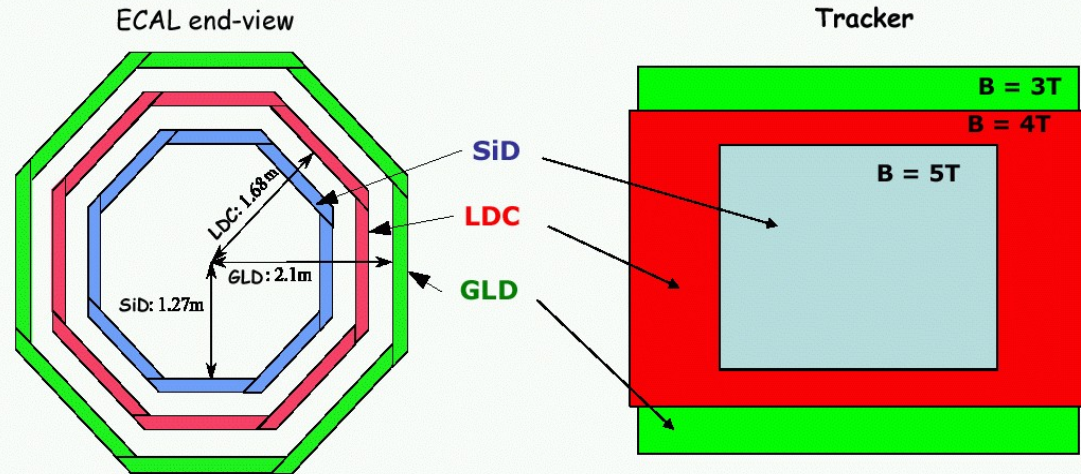
Relevant: inner radius of ECAL: defines the overall scale



"small"

"large"

"huge"



SiD: Silicon based concept

GLD: even larger detector concept

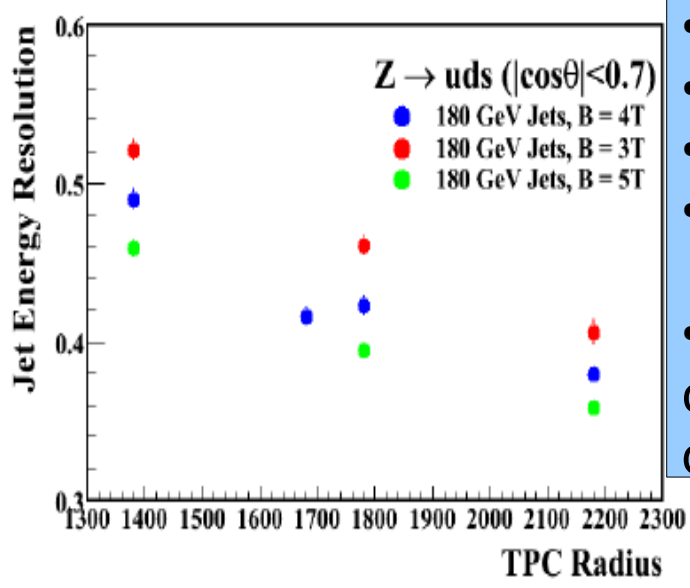
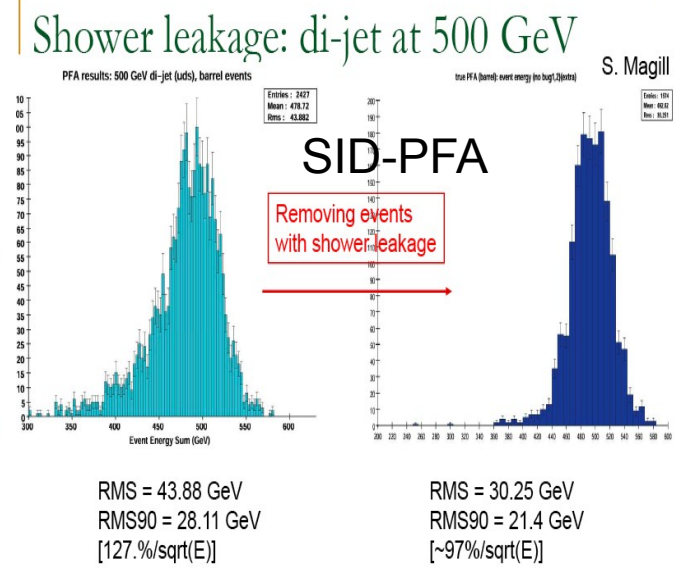
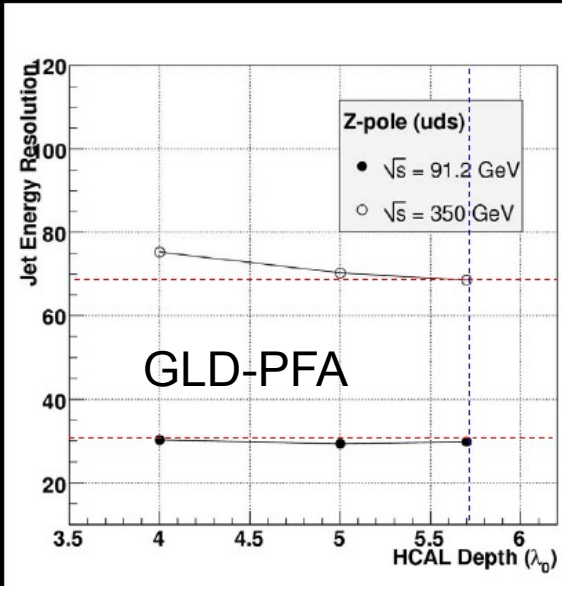
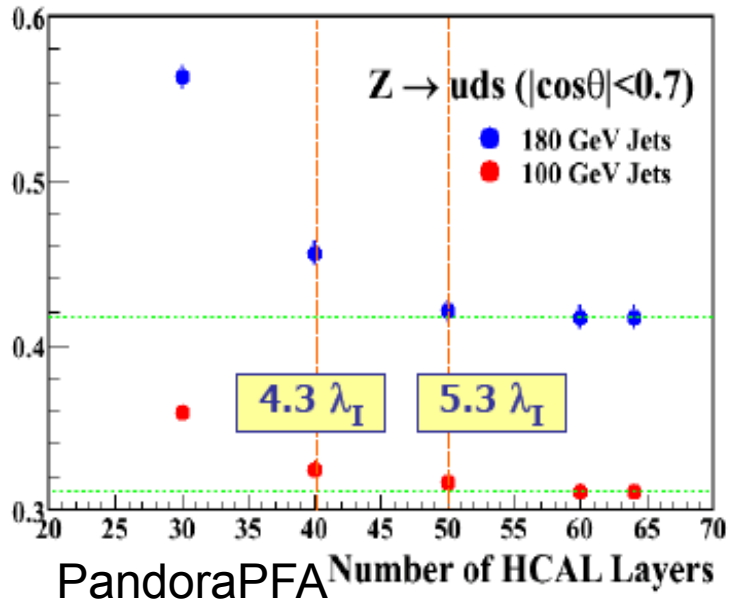
LDC: large detector concept

need of sophisticated **Monte Carlo Simulation** programs as well as full **reconstruction** tools to improve and compare the different detector concepts

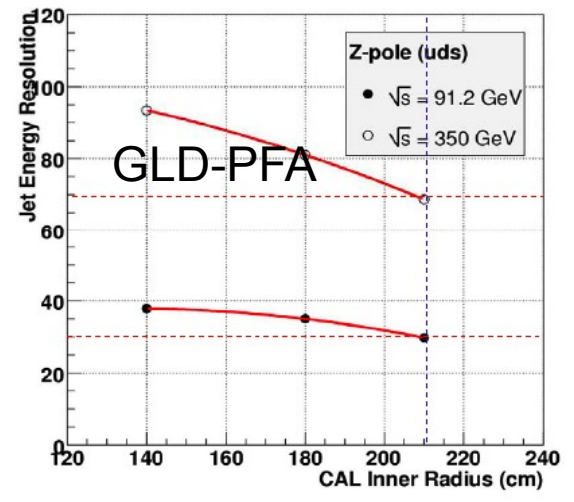
\* now ILD, SID, 4th

# Monte Carlo for detector optimization

Frank Gaede, DESY, Summerstudent Lectures August 19, 2009



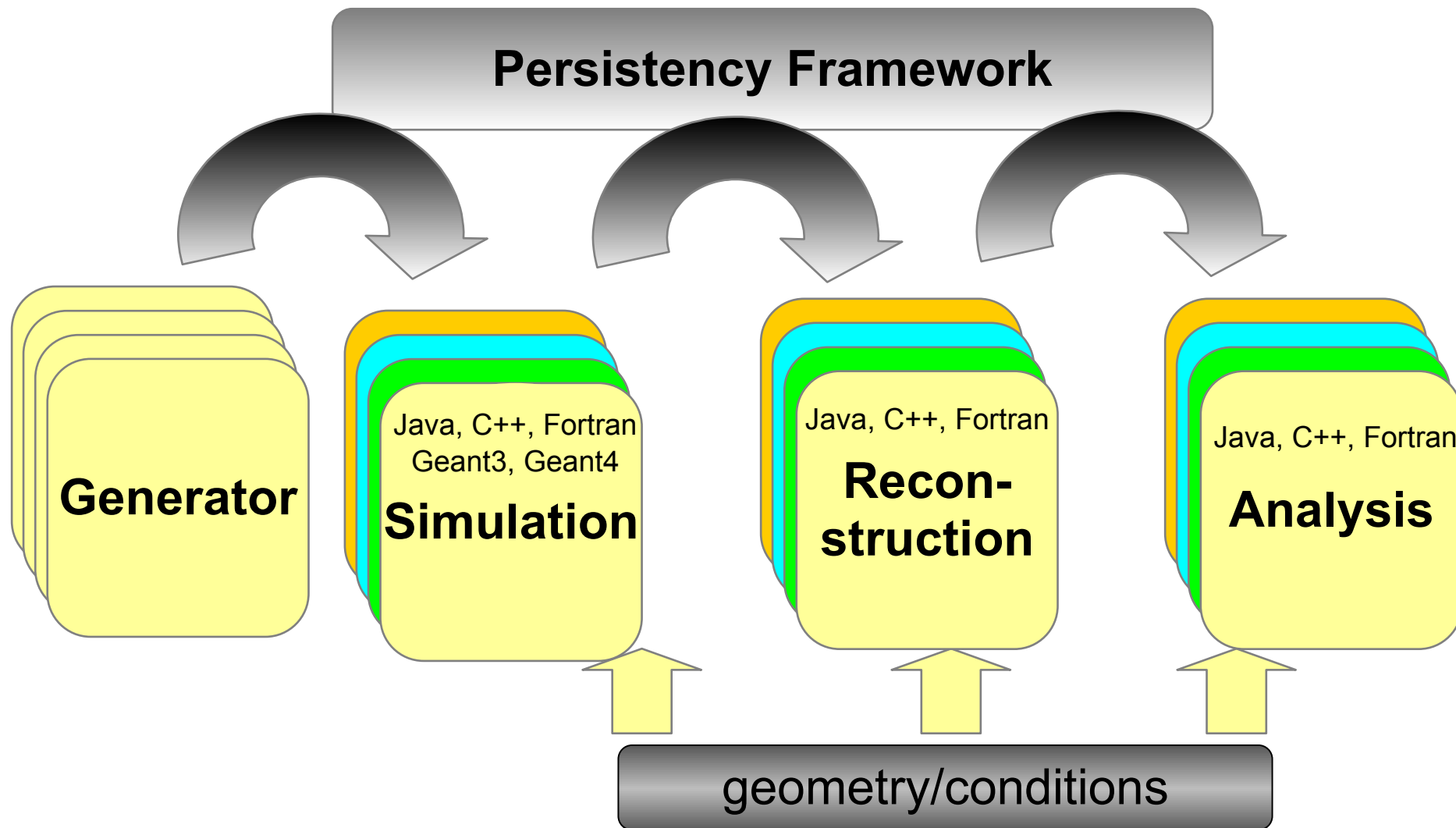
- vary detector parameters, eg:
- Hcal thickness
- Tracking radius
- B-field strength
- .....
- use Monte Carlo for (cost conscious) optimization of detector



# HEP Software Frameworks

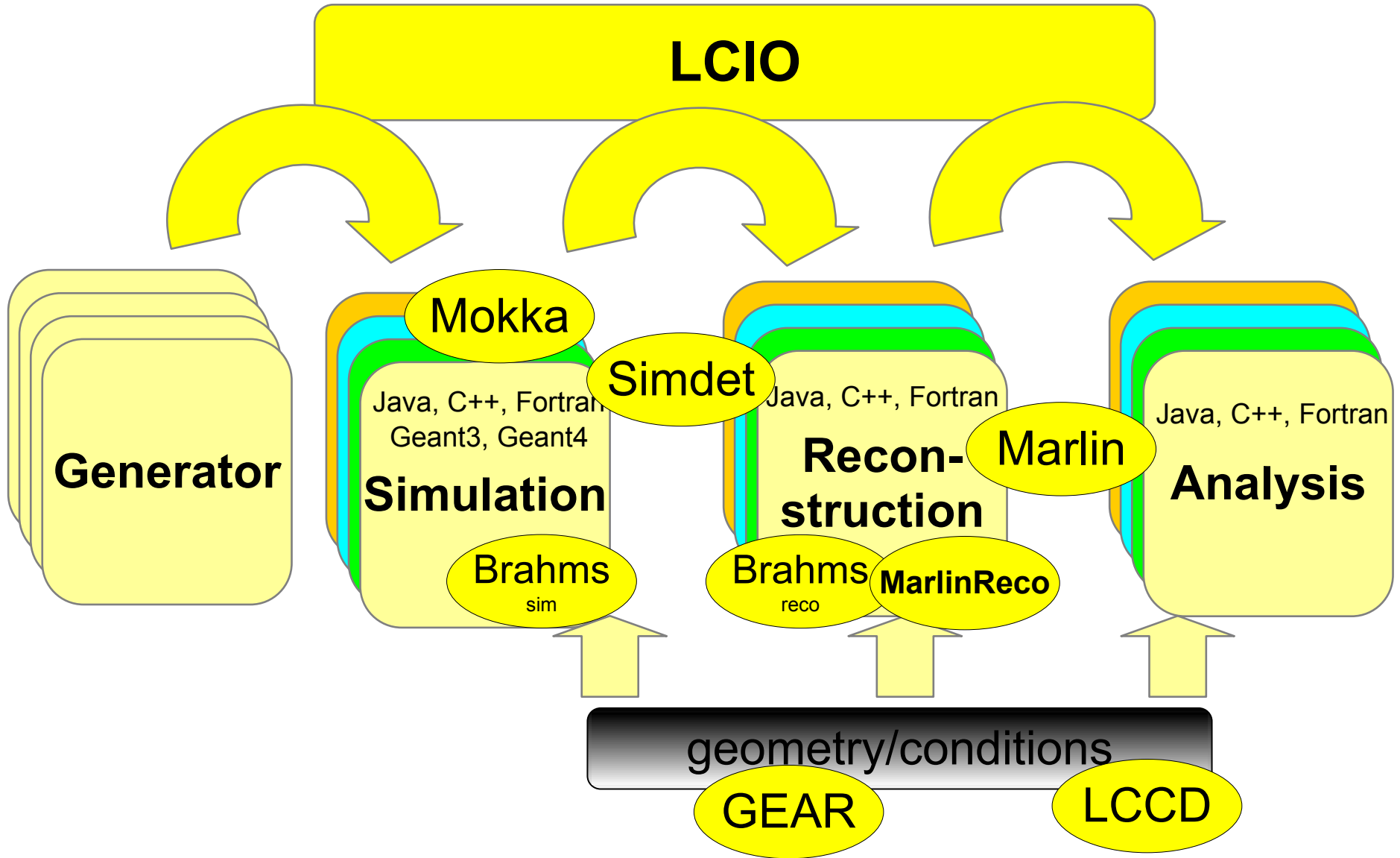
From generated 4-vectors and/or data  
to published histograms

# ILC Monte Carlo software chain





# ILC Monte Carlo software chain



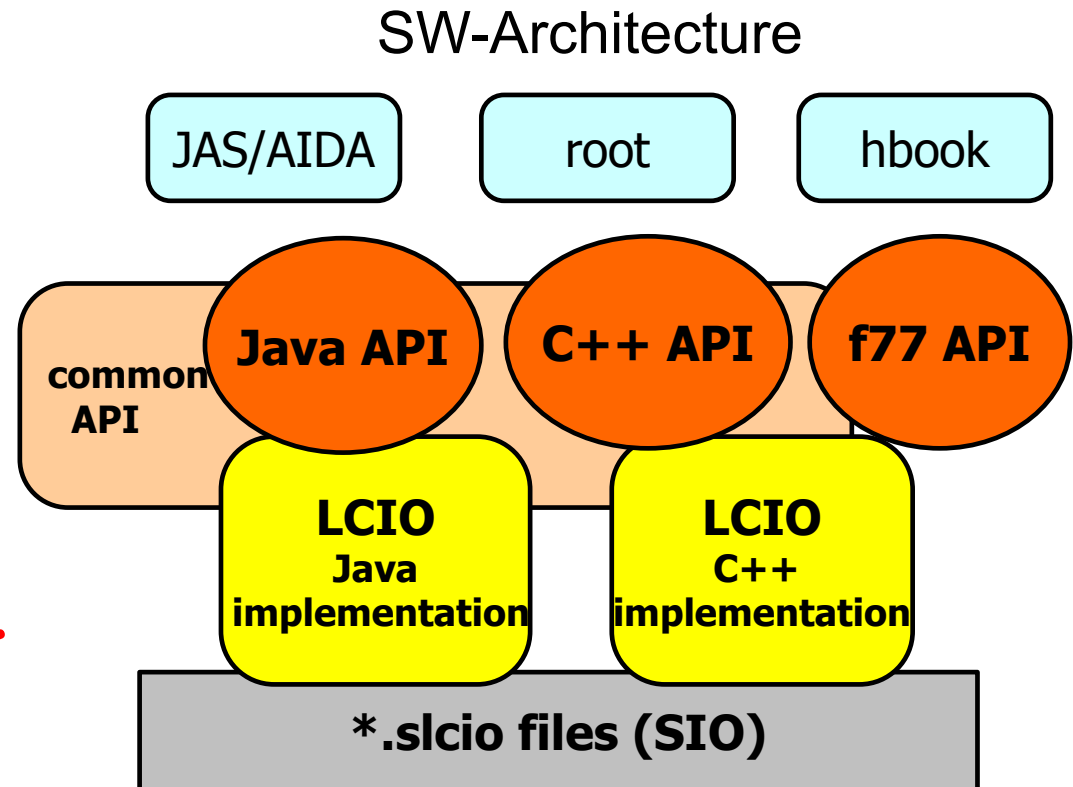
# LCIO overview

- DESY and SLAC joined project:
  - provide common basis for ILC software
- Features:
  - Java, C++ and f77 (!) API
  - extensible data model for current and future simulation and testbeam studies
  - user code separated from concrete data format
  - no dependency on other frameworks

**simple & lightweight**

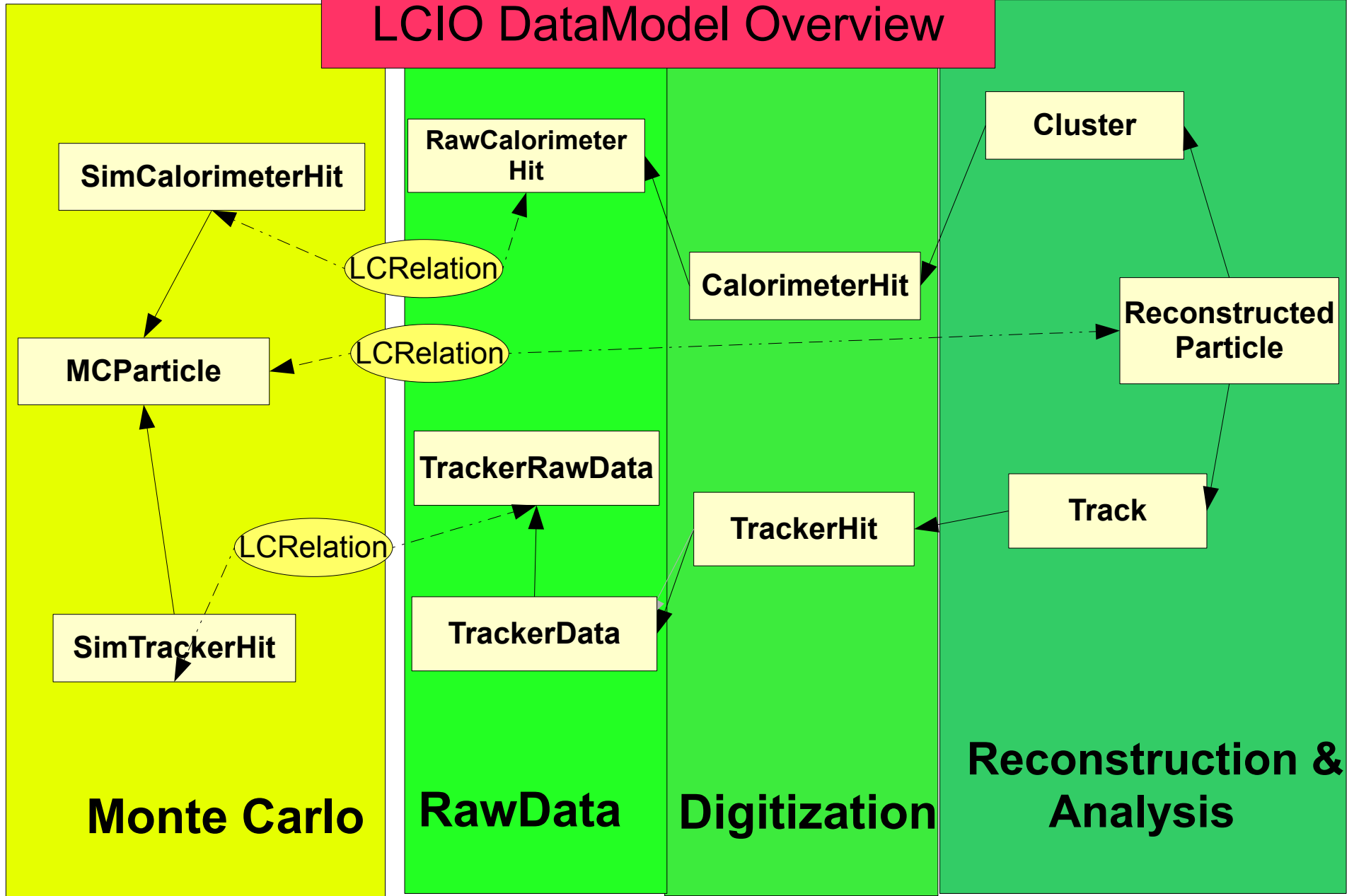
current release: v01-11

international standard  
persistency & datamodel  
for ILC software



# event data model

## LCIO DataModel Overview



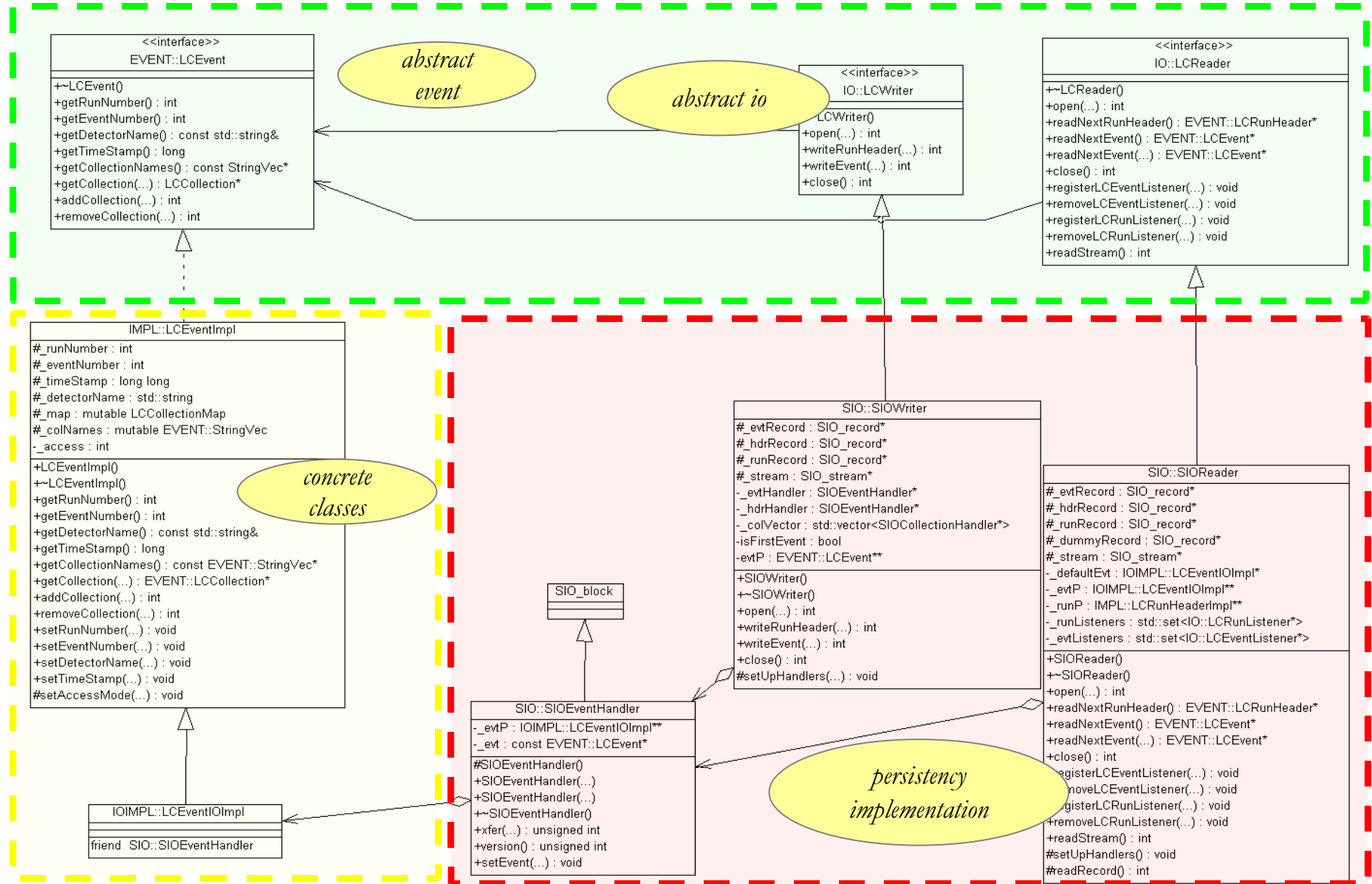
# example MCParticle data class

virtual <code>~MCParticle ()</code> <i>Destructor.</i>	virtual bool <code>isDecayedInTracker () const=0</code> <i>True if the particle decayed or interacted in a tracking region.</i>
virtual double <code>getEnergy () const=0</code> <i>Returns the energy of the particle (at the vertex) in [GeV] computed from particle's momentum and mass - only float used in files.</i>	virtual bool <code>isDecayedInCalorimeter () const=0</code> <i>True if the particle decayed or interacted (non-continuous interaction, particle terminated) in non-tracking region.</i>
virtual const <code>MCParticleVec &amp; getParents () const=0</code> <i>Returns the parents of this particle.</i>	virtual bool <code>hasLeftDetector () const=0</code> <i>True if the particle left the world volume undecayed.</i>
virtual const <code>MCParticleVec &amp; getDaughters () const=0</code> <i>Returns the daughters of this particle.</i>	virtual bool <code>isStopped () const=0</code> <i>True if the particle lost all kinetic energy inside the world volume and did not decay.</i>
virtual int <code>getNumberOfParents () const=0</code> <i>Returns the number of parents of this particle - 0 if mother.</i>	virtual const double * <code>getVertex () const=0</code> <i>Returns the production vertex of the particle in [mm].</i>
virtual <code>MCParticle * getParent (int i) const=0</code> <i>Returns the i-th parent of this particle.</i>	virtual float <code>getTime () const=0</code> <i>The creation time of the particle in [ns] wrt.</i>
virtual int <code>getPDG () const=0</code> <i>Returns the PDG code of the particle.</i>	virtual const double * <code>getEndpoint () const=0</code> <i>Returns the endpoint of the particle in [mm] if the endpoint has been set explicitly.</i>
virtual int <code>getGeneratorStatus () const=0</code> <i>Returns the status for particles as defined by the generator, typically 0 empty line 1 undecayed particle, stable in the generator 2 particle decayed in the generator 3 documentation line.</i>	virtual const double * <code>getMomentum () const=0</code> <i>Returns the particle's 3-momentum at the production vertex in [GeV] only float used in files.</i>
virtual int <code>getSimulatorStatus () const=0</code> <i>Returns the status for particles from the simulation, e.g.</i>	virtual double <code>getMass () const=0</code> <i>Returns the mass of the particle in [GeV] - only float used in files.</i>
virtual bool <code>isCreatedInSimulation () const=0</code> <i>True if the particle has been created by the simulation program (rather the generator).</i>	virtual float <code>getCharge () const=0</code> <i>Returns the particle's charge.</i>
virtual bool <code>isBackscatter () const=0</code> <i>True if the particle was created by the simulator as a result of an interaction decay in non-tracking region, e.g.</i>	virtual int <code>getNumberOfDaughters () const=0</code> <i>Returns the number of daughters of this particle.</i>
virtual bool <code>vertexIsNotEndpointOfParent () const=0</code> <i>True if the particle was created as a result of a continuous process where the parent particle continues, i.e.</i>	virtual <code>MCParticle * getDaughter (int i) const=0</code> <i>Returns the i-th daughter of this particle.</i>

# Persistency – file formats I

- most OO-languages such as C++ don't have a built in persistency mechanism
- typically experiments defines their own binary format, due to convenience and efficiency reasons
- -> need tools/code to 'persist' class contents to files
- preferences and requirements change

# LCIO class design



# Persistence – file formats II

- file formats need to be flexible:
- not always same information stored
- different tasks need different level of detail, ie. different amount of data, eg.
  - calibration needs all detector signals (Hits)
  - physics analyses need only output of reconstruction, ie. high level objects such as reconstructed particles or even jets
- -> typically several levels of data files exist with increased condensation of information and decreased file sizes

# ATLAS multilevel persistency scheme

EDM Level	Contents	Primary Intent	Size/ Event (KB)	Max Ideal Input rate (Hz)	Access-ibility
<b>Raw Data Objects</b>	Raw Channels	Reconstruction (calibration)	1600	N/A	Central Reco/Reprocessing: Tier 0/1
<b>Event Summary Data</b>	Cells, Hits, Clusters, Tracks, MET, Electron, Jet, Muon, Tau, Truth	Derive calibrations, Re-reconstruction, Re-calibration	500		CERN CAF (access limited), Tier 1 (on tape)
<b>Analysis Object Data</b>	Lepton Cells, Hits, Clusters, Tracks, MET, Electron, Jet, Muon, Tau, Slimmed Truth	Limited Re-reconstruction (eg Jets, b-tag), limited re-calibration, Analysis	100	1000	Full: Tier 1,2 (disk) Subset: Tier 3
<b>Derived Physics Data</b>	Any of the above + composites (eg top) + derived quantities (sphericity)	Interactive Analysis: Making plots, performing studies	Typically ~10	106	Tier 3: eg your laptop
<b>TAG</b>	Summary. Ex: $p_T, \eta$ of 4 best e, $\gamma, \mu, \tau, \text{jet}$	Selection Events for analysis	1	108	Everywhere



# conditions database

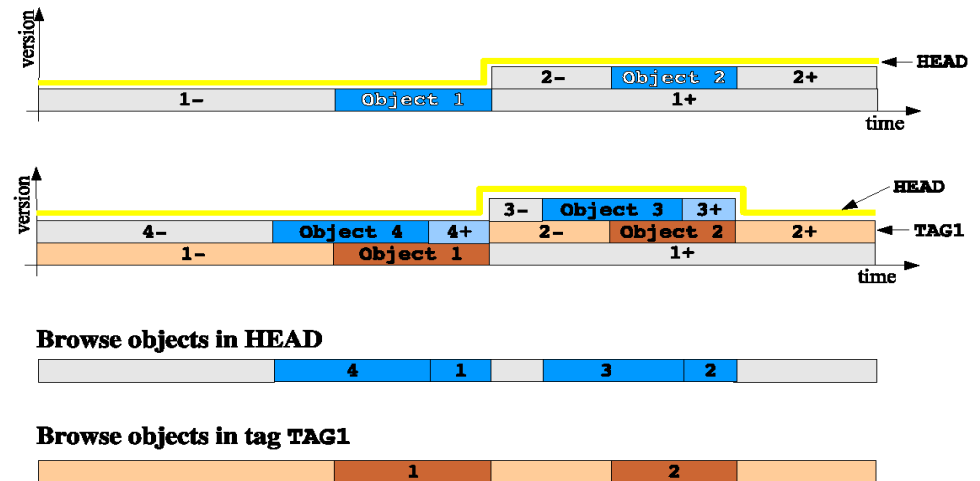
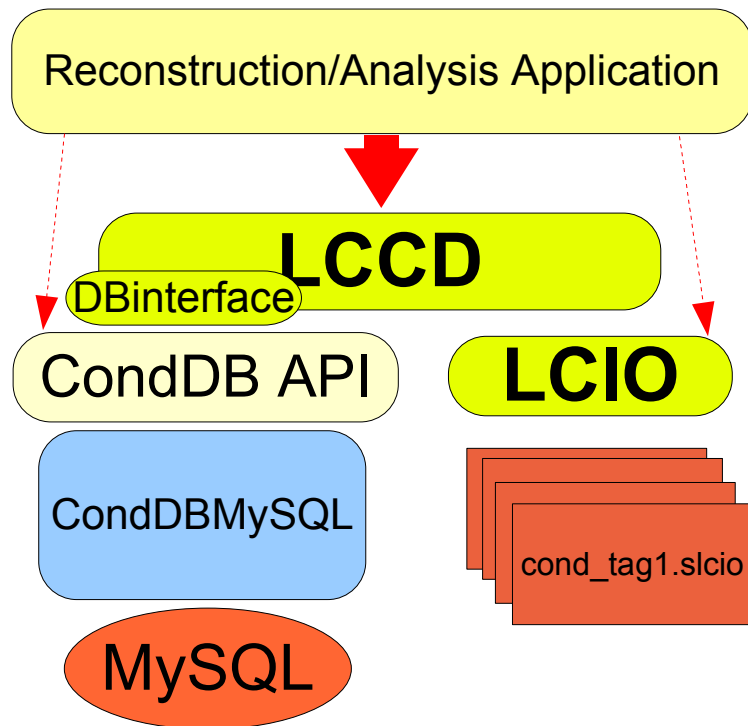


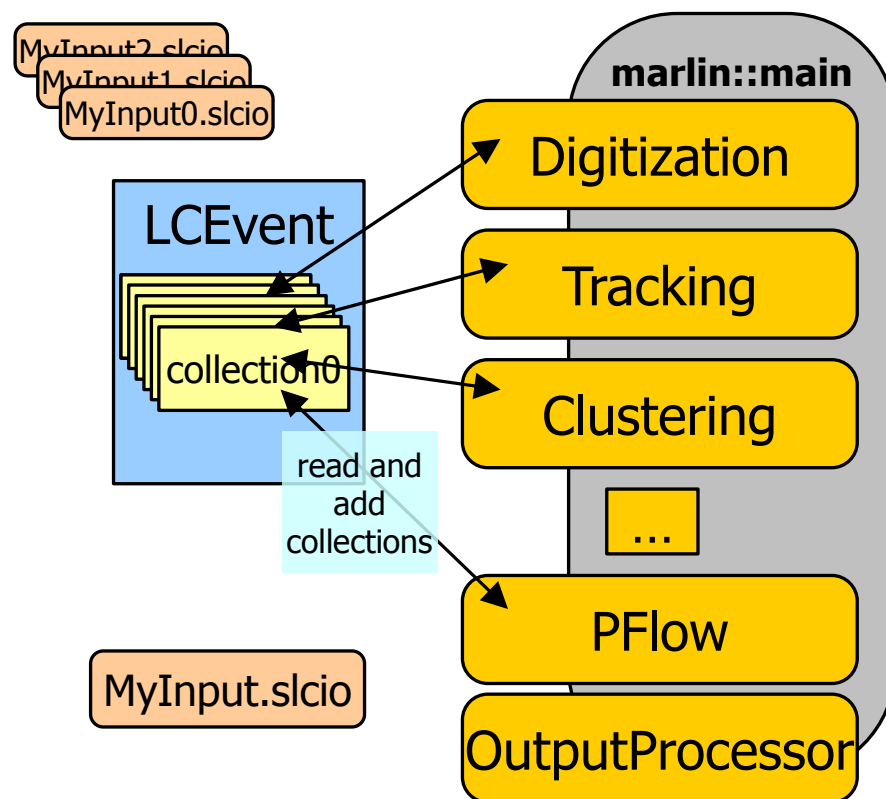
Figure 3: tagging and browsing example in the ConditionsDB mySQL's implementation.

- Conditions Data:
  - all data that is needed for analysis/reconstruction besides the actual event data
  - typically has lifetime (validity range) longer than one event
    - can change on various timescales, e.g. seconds to years
    - need for tagging mechanism, e.g. for calibration constants
  - example: trigger configuration, temperature readings, gas pressures, calibration constants, electronic channels mapping,...

# example analysis/reconstruction framework: Marlin

## Modular Analysis & Reconstruction for the L I N ear Collider

- modular C++ **application framework** for the analysis and reconstruction of LCIO data
- uses LCIO as transient data model
- software modules called Processors
- provides main program !
- provides simple user steering:
  - program flow (active processors)
  - user defined variables
    - per processor and global
  - input/output files
  - **Plug&Play** of processors



# Marlin Processor

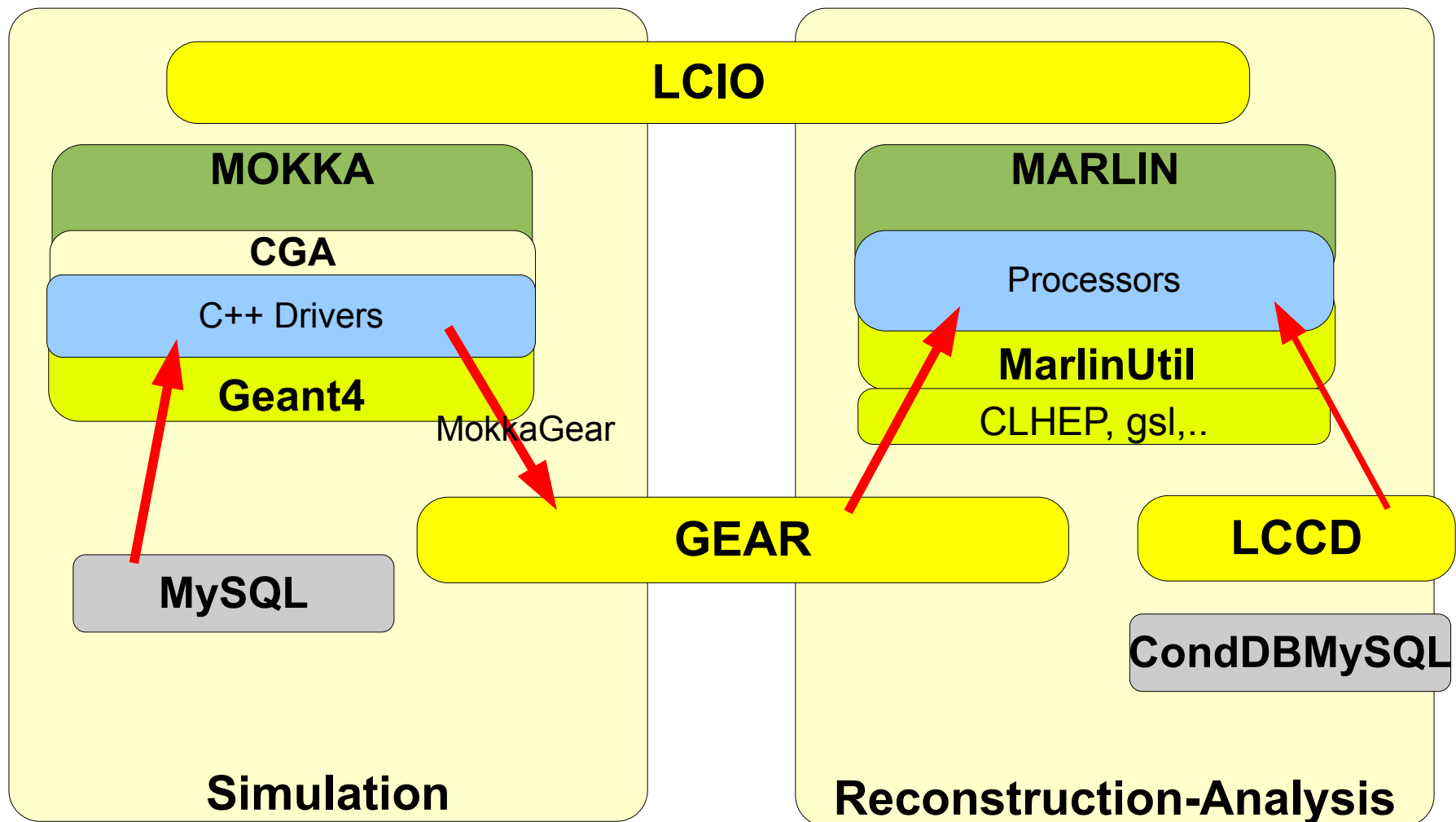
- provides main **user callbacks**
- has **own set of input parameters**
  - int, float, string (single and arrays)
  - parameter description
- naturally modularizes the application
- **order of processors is defined via steering file:**
  - easy to exchange one or several modules w/o recompiling
  - can run the same processor with different parameter set in one job
- **processor task can be as simple as creating one histogram or as complex as track finding and fitting in the central tracker**

```
marlin::Processor  
init()  
processRunHeader(LCRunHeader* run)  
processEvent( LCEvent* evt)  
check( LCEvent* evt)  
end()
```

```
UserProcessor  
processEvent( LCEvent* evt){  
    // your code goes here...  
}
```



# ILD simulation framework



# Analysis Tools - example root

Frank Gaede, DESY, Summerstudent Lecture, August 18, 2009

The screenshot shows the ROOT System Home Page in Mozilla Firefox. The browser window title is "The ROOT System Home Page - Mozilla Firefox" and the address bar shows "http://root.cern.ch/". The page content includes the ROOT logo, the text "An Object-Oriented Data Analysis Framework", and a classical-style illustration of a woman holding a globe. Below the browser window, several ROOT plots are displayed:

- Drawing options for one dimensional histograms:** Two histograms are shown. The first is a 2D histogram with a red fill, labeled "Default option". The second is a 3D histogram with a brown fill, labeled "option Level 1".
- Test random numbers:** A histogram showing a distribution of random numbers. The x-axis ranges from 0 to 10, and the y-axis ranges from 0 to 180. A blue curve is overlaid on the histogram. The statistics are: Nent = 10000, Mean = 3.58823, RMS = 1.85042.
- The Ntuple canvas:** A window titled "The Ntuple canvas" showing two plots. The left plot is a histogram with a blue fill, labeled "3\*px+2". The right plot is a histogram with a yellow fill, labeled "pZ.px".
- Rotating a view:** A 3D scatter plot showing a distribution of points in a cylindrical shape. A yellow circle is overlaid on the base of the cylinder. Text indicates: "You can interactively rotate this view in 2 ways: - With the RotateCube in clicking in this pad - Selecting View with x3d in the View menu".

Statistical data for the histograms:

Plot	Mean	RMS
px distribution (Default)	-0.462958	0.935192
px distribution (option Level 1)	-0.462958	0.935192
3*px+2	1.9571	3.6742
pZ.px	-0.007	0.9961
h1f (Test random numbers)	3.58823	1.85042

- analysis tools in HEP
- provide core features:
  - ntuples
  - histograms (1D/2D)
  - fitting
  - plotting /publishing

note: the root C++ framework provides much more functionality not discussed here

# Analysis Tools - example JAS3

<http://jas.freehep.org/jas3/index.html>

The screenshot displays the JAS3 software interface with several windows open:

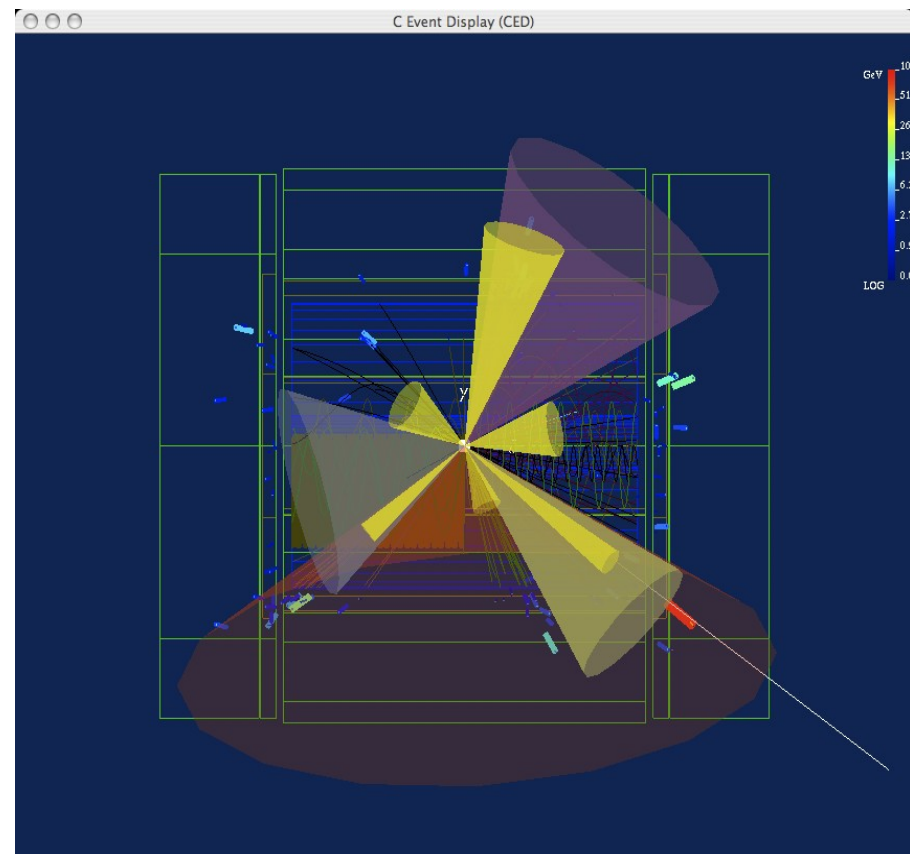
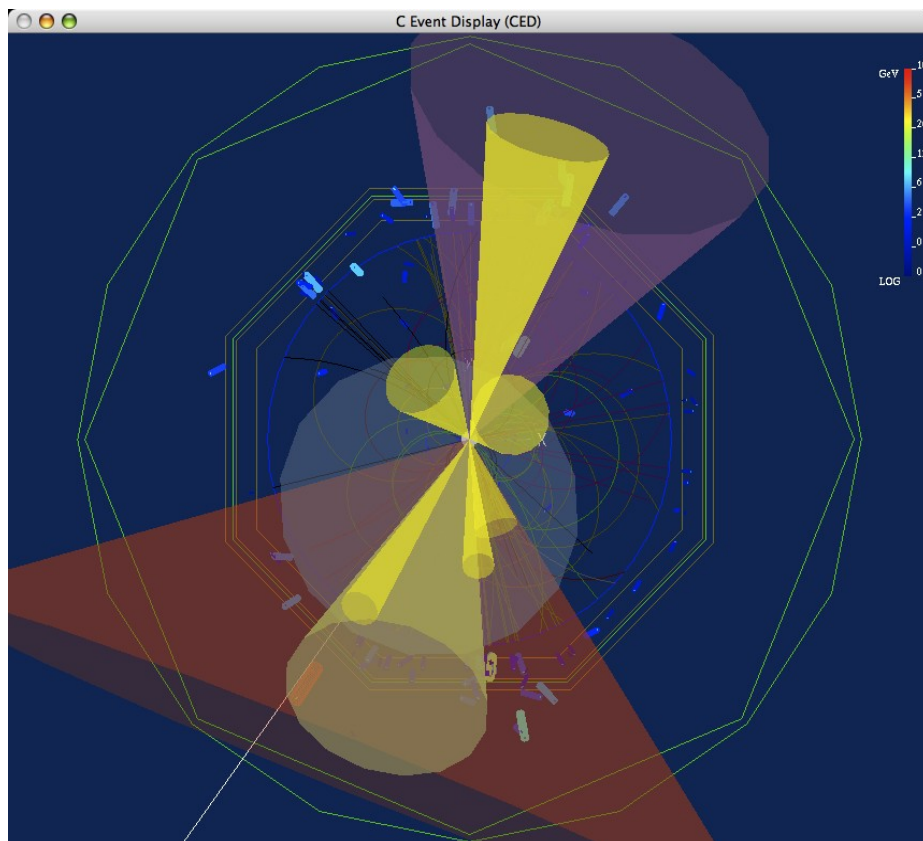
- Left Panel:** A tree view showing project folders like ETAPFV1, KAON, MM\_MMG, NICE, OMEGAF, PHI, PION, RO, STAFF, TEST OF N-TUPLES, and Programs.
- Top Center:** A histogram plot showing a distribution of data points, with the y-axis ranging from 100 to 800.
- Top Right:** A table titled "Collection: MCParticle type:MCParticle size:473 flags:0" showing event data for Run:9999 Event: 1. The table has columns for N, Type, Status, Parent, PX, PY, PZ, and Mass.
- Bottom Center:** A detailed diagram of a particle detector, showing a central interaction point with various detector components and particle tracks.
- Bottom Left:** A panel for "Attributes of picked object (18)" with a table of parameters and their values.

N	Type	Status	Parent	PX	PY	PZ	Mass
0	2212	Document...	0	0	7000.0	0.93827	
1	2212	Document...	0	0	-7000.0	0.93827	
2	21	Document...	0	0.25815	-0.27900	6.5793	0
3	-3	Document...	1	-0.45454	-0.36117	-1802.7	0
4	4	Document...	2	-0.40964	-1.0530	2.2164	0
5	-3	Document...	3	-13.179	1.9646	-717.51	0
6	22	Document...	4,5	0.78672	0.69178	-4.4768	0
7	24	Document...	4,5	-14.375	0.21979	-710.81	80.667
8	22	Final State	6	0.78672	0.69178	-4.4768	0
9	24	Intermediate	7	-14.375	0.21979	-710.81	80.667
10	3224	Intermediate	1	0.16978	0.20640	-1483.5	1.3846
11	-4	Intermediate	2	1.0287	0.84333	2.4188	1.3500
12	2	Intermediate	0	0.080131	0.087964	0.31987	5.6000E-3
13	-3	Intermediate	9	-11.920	16.413	-260.20	0.19900
14	21	Intermediate	9	-9.7052	16.270	-246.29	0
15	21	Intermediate	9	-0.18941	-0.12814	-6.3494	0
16	21	Intermediate	9	-0.47022	-0.21941	-2.9564	0
17	21	Intermediate	9	0.41252	0.36534	-2.3612	0
18	21	Intermediate	9	-0.11239	-0.075933	0.055171	0
19	21	Intermediate	9	1.3372	-4.4404	-32.038	0

Name	Value	Unit	Node
DrawAs	Line		<input type="checkbox"/>
Generator	HepRepPlugin - He...		<input type="checkbox"/>
Layer	70.0		<input type="checkbox"/>
P	0.70138 GeV: 3		<input checked="" type="checkbox"/>
PI	e+		<input checked="" type="checkbox"/>
PT	0.61707 GeV: 3		<input checked="" type="checkbox"/>
ViewScale	0.010000		<input checked="" type="checkbox"/>
color			<input checked="" type="checkbox"/>
d0	28.112 cm: 3		<input checked="" type="checkbox"/>
omega	-7.3368E-317 cm: 3		<input checked="" type="checkbox"/>
phi0	0.80486 rad: 3		<input checked="" type="checkbox"/>
tanDip	-0.54028 rad: 3		<input checked="" type="checkbox"/>
tkElectronID	Tight		<input checked="" type="checkbox"/>
tkKaonID	VeryTight		<input checked="" type="checkbox"/>
tkMuonID	Not A Muon		<input checked="" type="checkbox"/>
tkPionID	Not A Pion		<input checked="" type="checkbox"/>
tkProtonID	Not A Proton		<input checked="" type="checkbox"/>
z0	0.28231 cm: 3		<input checked="" type="checkbox"/>

- analysis tools in HEP
- typically also provide:
  - file I/O, file browsers
  - event displays
  - scripting environments
  - integration with experiment software

# example: ILD event display



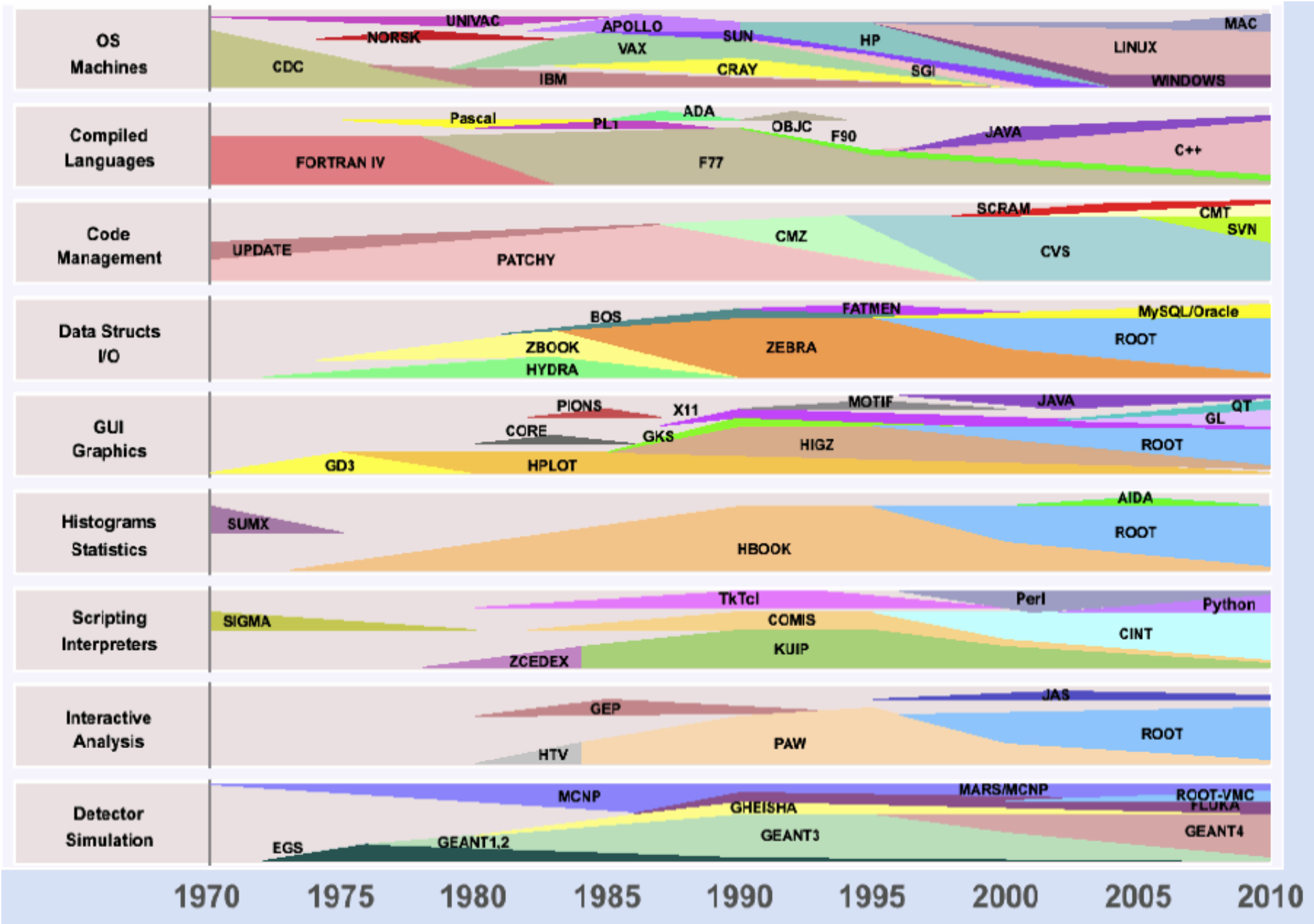
- summerstudent work 2008: extend event display to view DSTs (no hits available for visualization):

S.Daraszevic

- show Tracks as helices
- Clusters as lines & Cylinders (scaled w/ E)
- jets as cones ( $E, p_t$ ) + particles coloured
- 3 momentum of all particles in the event

# tools/languages used in HEP

Frank Gaede, DESY, Summerstudent Lecture, August 18, 2009

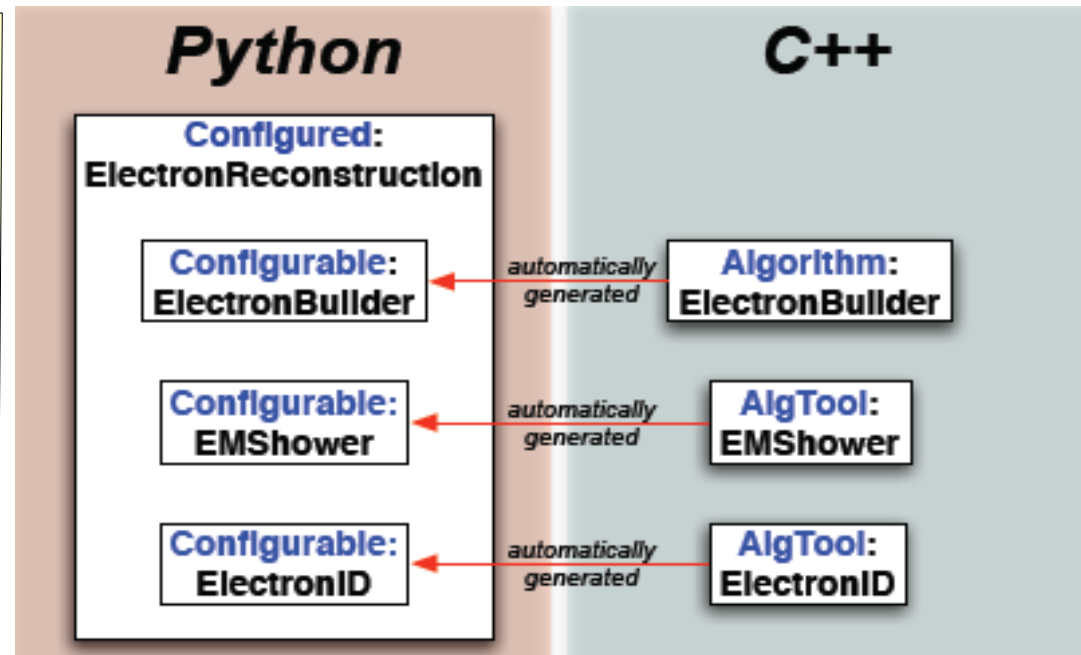




# multi language frameworks

- different languages have different advantages in different application domains
- programmers have different skills (and preferences !)
  - Atlas and CMS: more than 2000 physicists
- multi language frameworks are a possible way to 'keep everyone happy'

- example:
- Atlas sw framework:
  - core sw in C++
  - python binding for modules
  - main application incl. configuration and simple analyses in python



# Computing Infrastructure - Hardware

# Mass Storage



- mass storage of HEP data and Monte Carlo is typically done on tapes
- e.g. @ DESY we have (8/2006)
  - 22 000 tapes
  - 46 tape drives
  - 4 robots
  - ~ 1.6 PetaByte data (mostly HERA experiments and Monte Carlo)

access to data on tape fairly slow  
-> need smart disk caching system

# dCache

**SRM** for storage management

**nfs2/3** for name space



/pnfs/<site>/<VO>/...

**dCap, xRoot** for random LAN



**gsiFtp, http(g)** for random WAN



SRM provides interface to **grid**

Osm, Enstore,  
Tsm, Hpss, ...



- dCache developed @ DESY&FNAL
- transparent access to files on tape via pnfs file system
- various protocols for worldwide access and file transfer

# Computing platforms I



- the main working horse for HEP computing today are **large PC clusters or farms**, which are mostly operated with linux
- typically the high level trigger operates on a **dedicated experiment specific farms** in order to guarantee the throughput needed
- Monte Carlo production, reconstruction and analysis run on often on shared farms
  - shared between tasks
  - shared between experiments ( -> institute batch system )
  - shared between institutes (-> grid )

# Computing platforms II



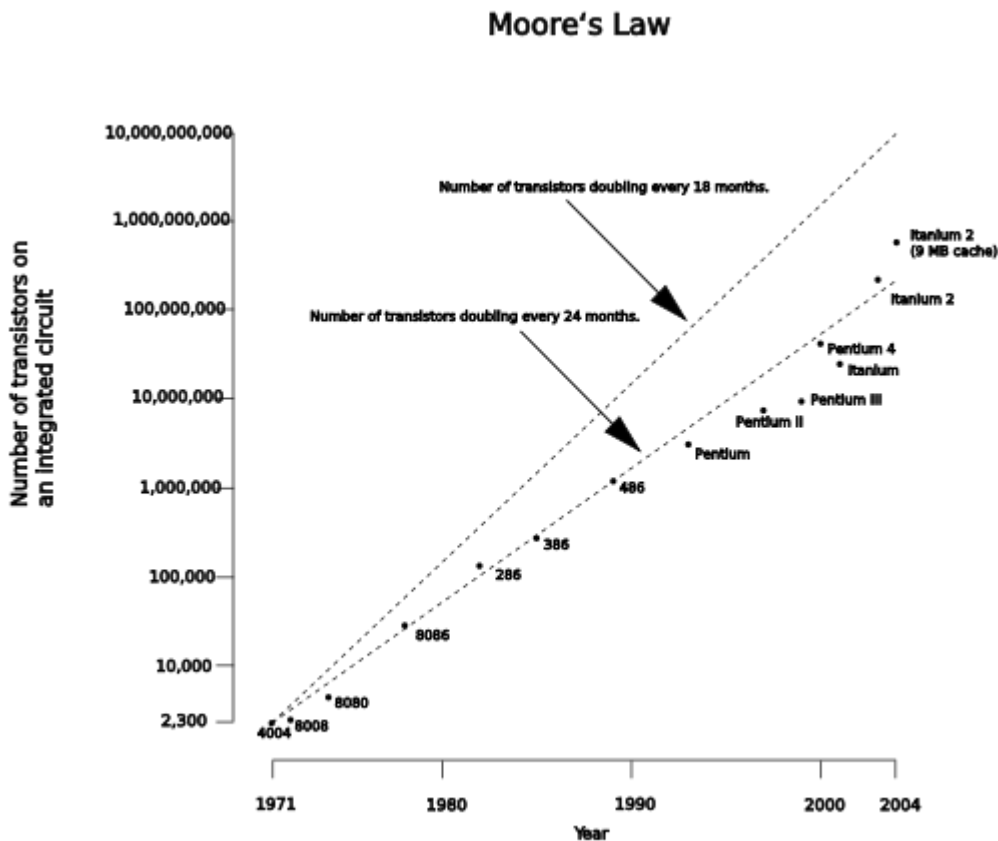
# Computing requirements

Application	Input	Output	CPU
MC generator	<i>none</i>	<i>small</i>	<i>little</i>
MC simulation	<i>small</i>	<i>large</i>	<i>huge</i>
MC digitization	<i>large/huge</i>	<i>large</i>	<i>little</i>
Reconstruction	<i>large</i>	<i>large</i>	<i>large</i>
Analysis	<i>large</i>	<i>small</i>	<i>small</i>

example:

- simulating (geant4) an ILC event takes ~200s on a standard PC (2007) (an Atlas event takes ~600s !)
- $O(10^6)$  events will take 10 CPU years !
- need hundreds (thousands) of CPUs to get events in reasonable time

# increase of computing power

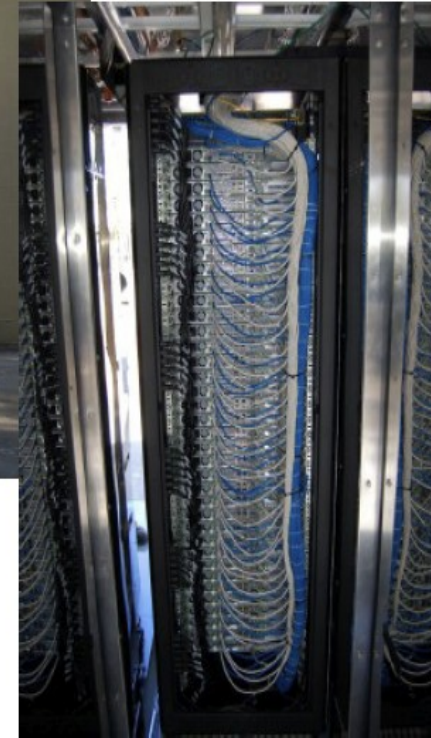
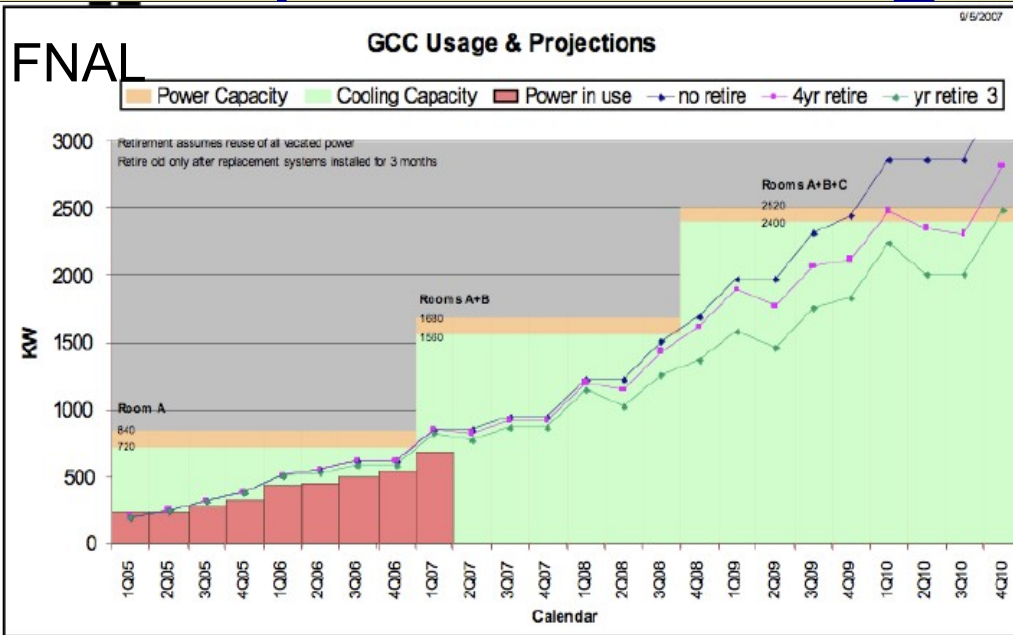


- **Moore's law:**
- number of transistors per chip doubles every  $\sim 2y$
- $\sim$ true for  $>40$  years !
- however:  
computing power does not grow at the same speed  
-> cache, memory, architecture
- requirements grow even faster:
  - more and larger data sets
  - complexity of algorithms
  - 'sloppiness' of programmers (why bother writing efficient code – the next generation of CPUs will run the slower/larger algorithm as fast ...)

- the ever increasing need for more
- computing power leads to larger and
- larger farms (thousands of CPUs)
- problem (cost):
  - **power and cooling**
  - need for new **facilities**



# power, cooling and facilities



SLAC

- replace old 'junk'
  - hardware older than 2-4 years is too expensive to operate -> replace w./ new hardware
- use water cooled racks (more efficient)
- increase density of CPUs per rack
  - -> multi core CPUs

electricity bills for computing will become a large fraction of labs' total budgets !

# multi core CPUs

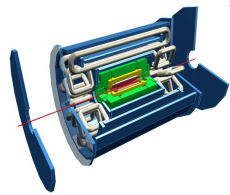
- HEP computing is 'embarrassingly parallel':
  - process one event at a time on one computer
  - -> total processing power scales w/ number of CPUs
  - HEP computing paradigm of 90s and 0s
- current trend: multi core CPUs
  - memory (RAM and L2 cache) does not scale w. # cores
  - processing one event per core is inefficient
  - -> need new computing paradigms and tools
    - multithreading (non trivial)
    - dedicated compilers that do multithreading for you
    - smaller memory footprint of programs

# Grid Computing

*“Sharing resources within Virtual Organizations in a global world.”*



# LHC Computing model



~PBytes/sec

Online System

~100 MBytes/sec

1 TIPS = 25,000 SpecInt95  
PC (1999) = ~15 SpecInt95

- One bunch crossing per 25 ns
- 100 triggers per second
- Each event is ~1 Mbyte

Offline Farm  
~20 TIPS

~100 MBytes/sec

~ Gbits/sec

**Tier 0**

CERN Computer Centre >20 TIPS



or Air Freight

**Tier 1**

US Regional Centre

Italian Regional Centre

French Regional Centre

RAL Regional Centre

**Tier 2**

ScotGRID++ ~1 TIPS

Tier2 Centre ~1 TIPS

Centre TIPS

**Tier 3**

Institute ~0.25TIPS

Institute

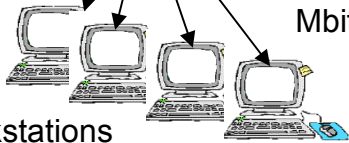
Institute

Physicists work on analysis "channels"  
Each institute has ~10 physicists working on one or more channels  
Data for these channels should be cached by the institute server

Physics data cache

100 - 1000 Mbits/sec

Workstations



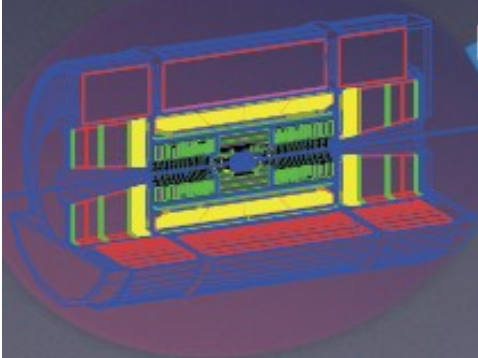
# ATLAS grid computing model

- Resources Spread Around the GRID

- Derive 1st pass calibrations within 24 hours.
- Reconstruct rest of the data keeping up with data taking.

- Reprocessing of full data with improved calibrations 2 months after data taking.
- Managed Tape Access: RAW, ESD
- Disk Access: AOD, fraction of ESD

- Interactive Analysis
- Plots, Fits, Toy MC, Studies, ...



**Tier 0**

RAW/  
AOD/  
ESD

**Tier 1**

10 Sites Worldwide

AOD

**Tier 2**

30 Sites Worldwide

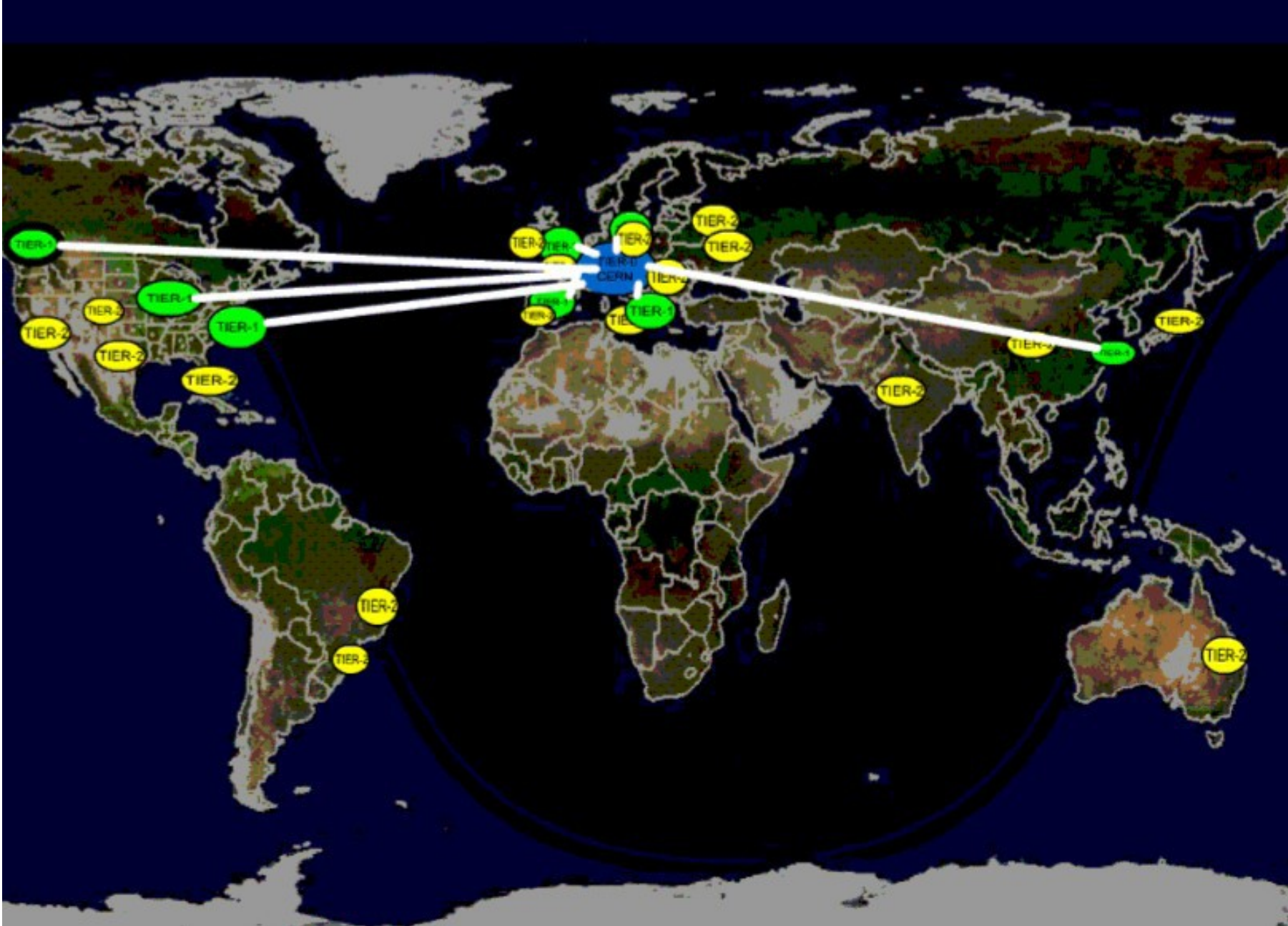
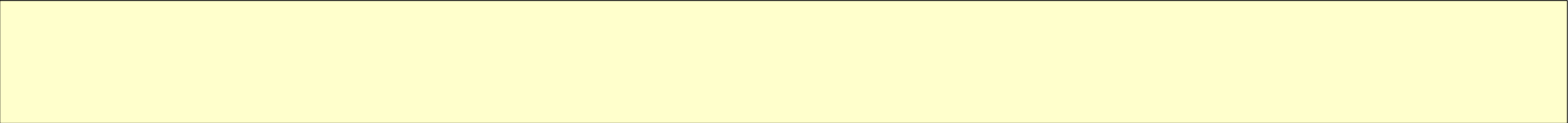
DPD

**Tier 3**

**CERN  
Analysis  
Facility**

- Primary purpose: calibrations
- Small subset of collaboration will have access to full ESD.
- Limited Access to RAW Data.

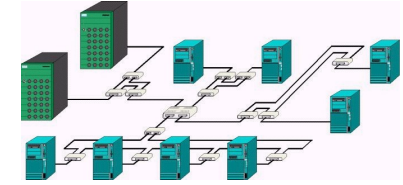
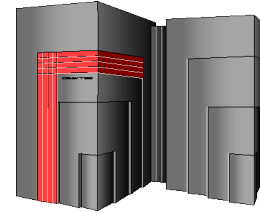
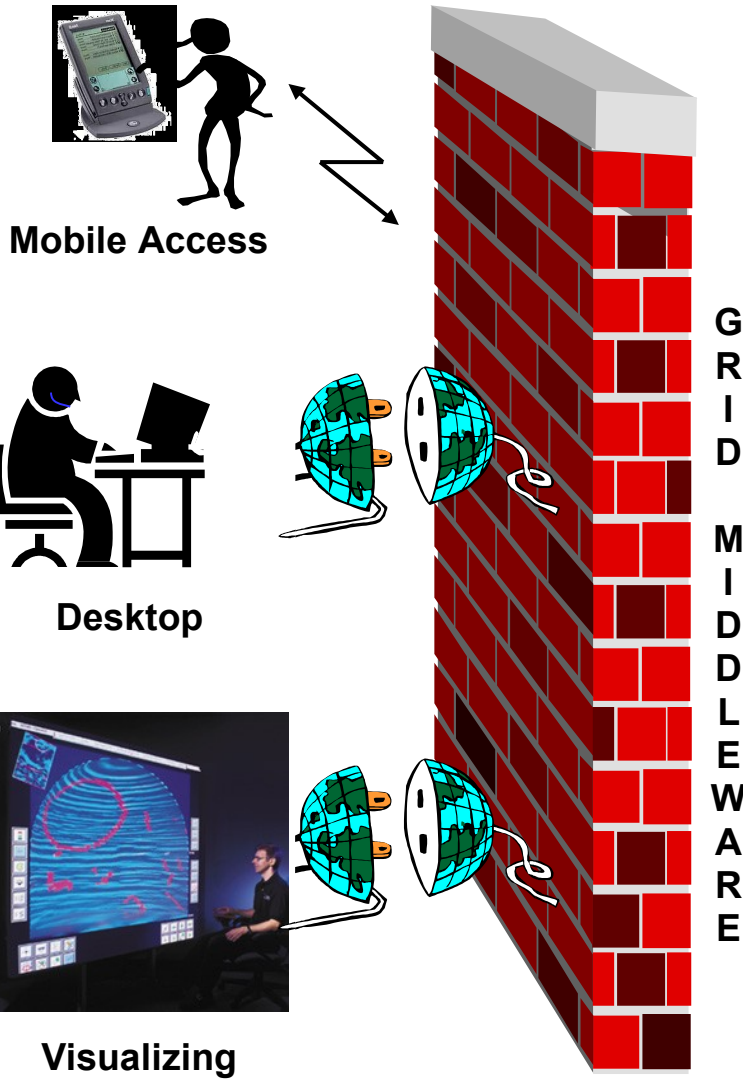
- Production of simulated events.
- User Analysis: 12 CPU/ Analyzer
- Disk Store: AOD



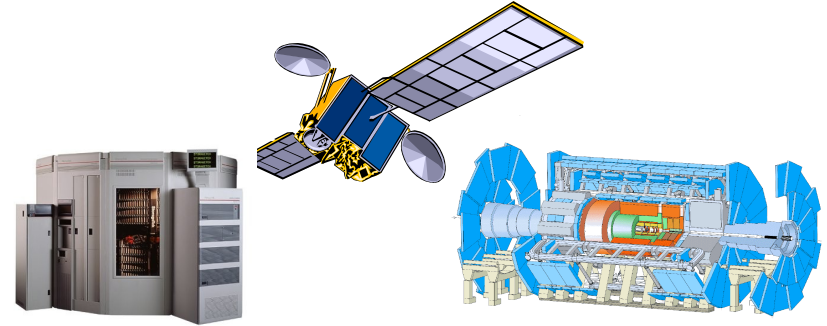
# Grid Definition

- I. Foster: **What is the Grid? A Three Point Checklist (2002)**
- **"A Grid is a system that:**
- **coordinates resources which are not subject to centralized controls ...**
  - integration and coordination of resources and users of different domains vs. local management systems (batch systems)
- **... using standard, open, general-purpose protocols and interfaces ...**
  - standard and open multi-purpose protocols vs. application specific system
- **... to deliver nontrivial qualities of services."**

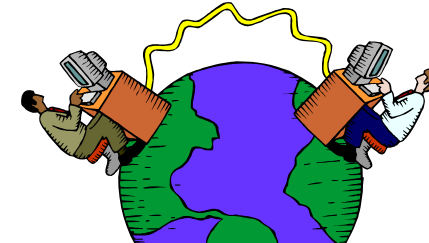
# The Grid dream



Supercomputer, PC-Cluster



Data Storage, Sensors, Experiments



Internet, Networks



# Grid Types

- **Data Grids:**
  - Provisioning of transparent access to data which can be physically distributed within **Virtual Organizations (VO)**
- **Computational Grids:**
  - allow for large-scale **compute resource sharing** within Virtual Organizations (VO)
- **Information Grids:**
  - Provisioning of **information** and data exchange, using well defined standards and web services

# Grid Ingredients

- **Authorization:**
  - Users must be registered in a **Virtual Organization (VO)**
- **Information Service:**
  - Provide a system which keeps track of the available resources
- **Resource Management:**
  - Manage and exploit the **available computing resources**
- **Data Management:**
  - Manage and exploit the data

# Grid: Authentication & Authorization

- a user is uniquely identified through a **certificate**
  - an encrypted electronic document, digitally signed by a Certification Authority (CA)
  - a certificate is your passport to enter the grid world
  - example: `/O=GermanGrid/OU=DESY/CN=Frank Gaede`
- access to resources is provided (controlled) via membership in a **Virtual Organization**
  - a dynamic collection of individuals, institutions, and resources which is defined by certain sharing rules
  - the VO a user belongs to is not part of the certificate.
  - a VO is defined in a central list, e.g. a LDAP tree.

# Grid Middleware

## Globus:

- Toolkit
- Argonne, U Chicago



## EDG (EU DataGrid):

- Project to develop Grid middleware
- Uses parts of Globus
- Funded for 3 years (01.04. 2001 - 31.03.2004)



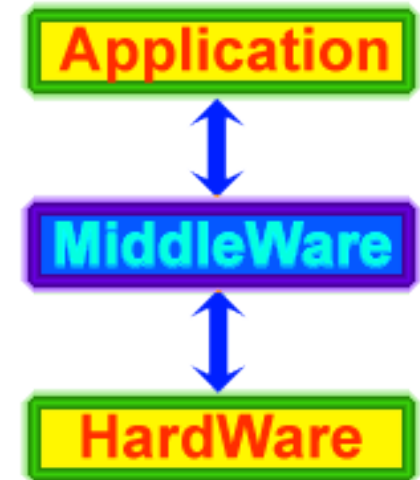
## LCG (LHC Computing Grid):

- Grid infrastructure for LHC production
- Based on stable EDG versions plus VDT etc.
- LCG-2 for Data Challenges



## EGEE (Enabling Grids for E-Science in Europe)

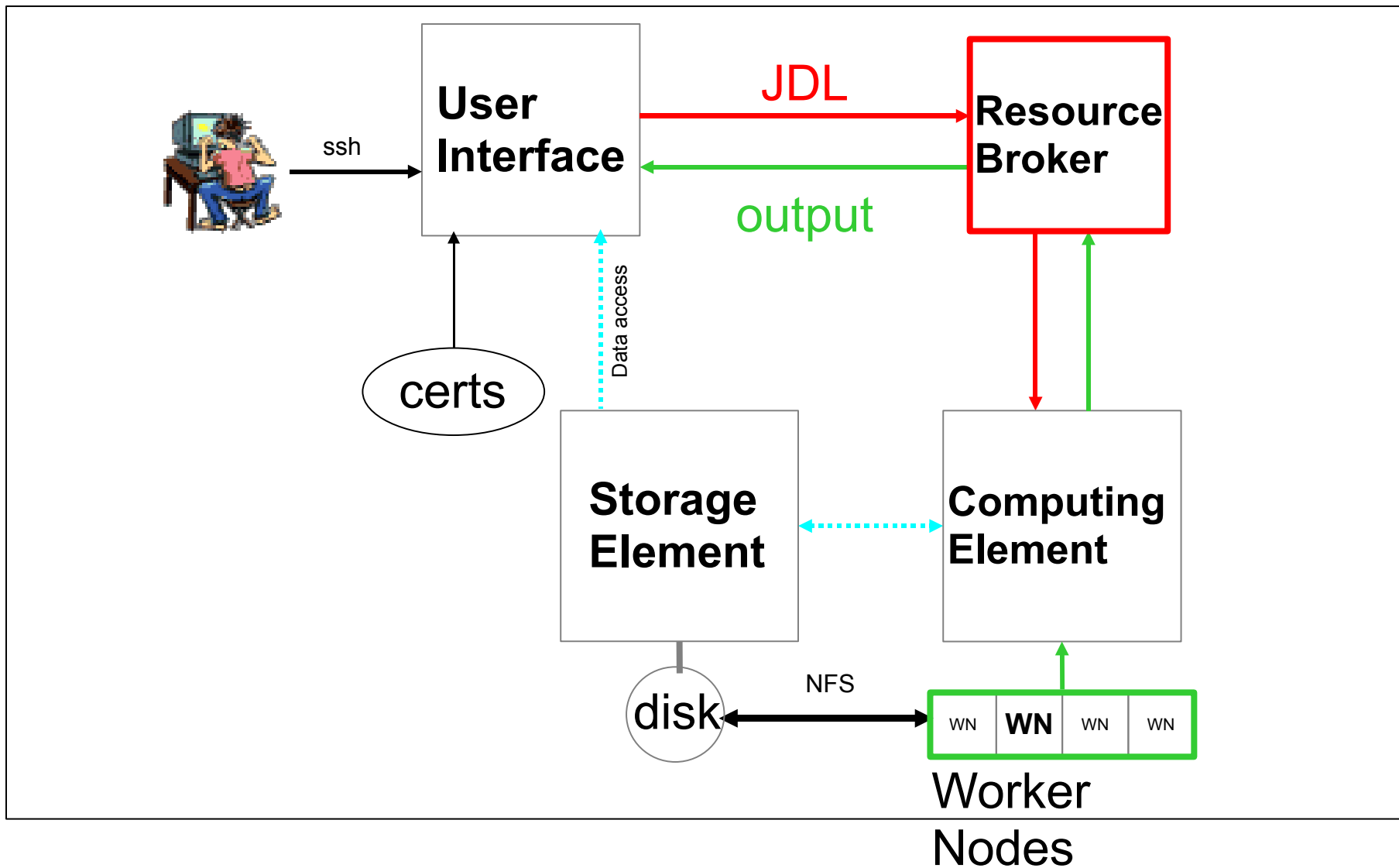
- Started 01.04.2004 for 2 + 2 years
- developed **gLite** as successor of LCG middleware



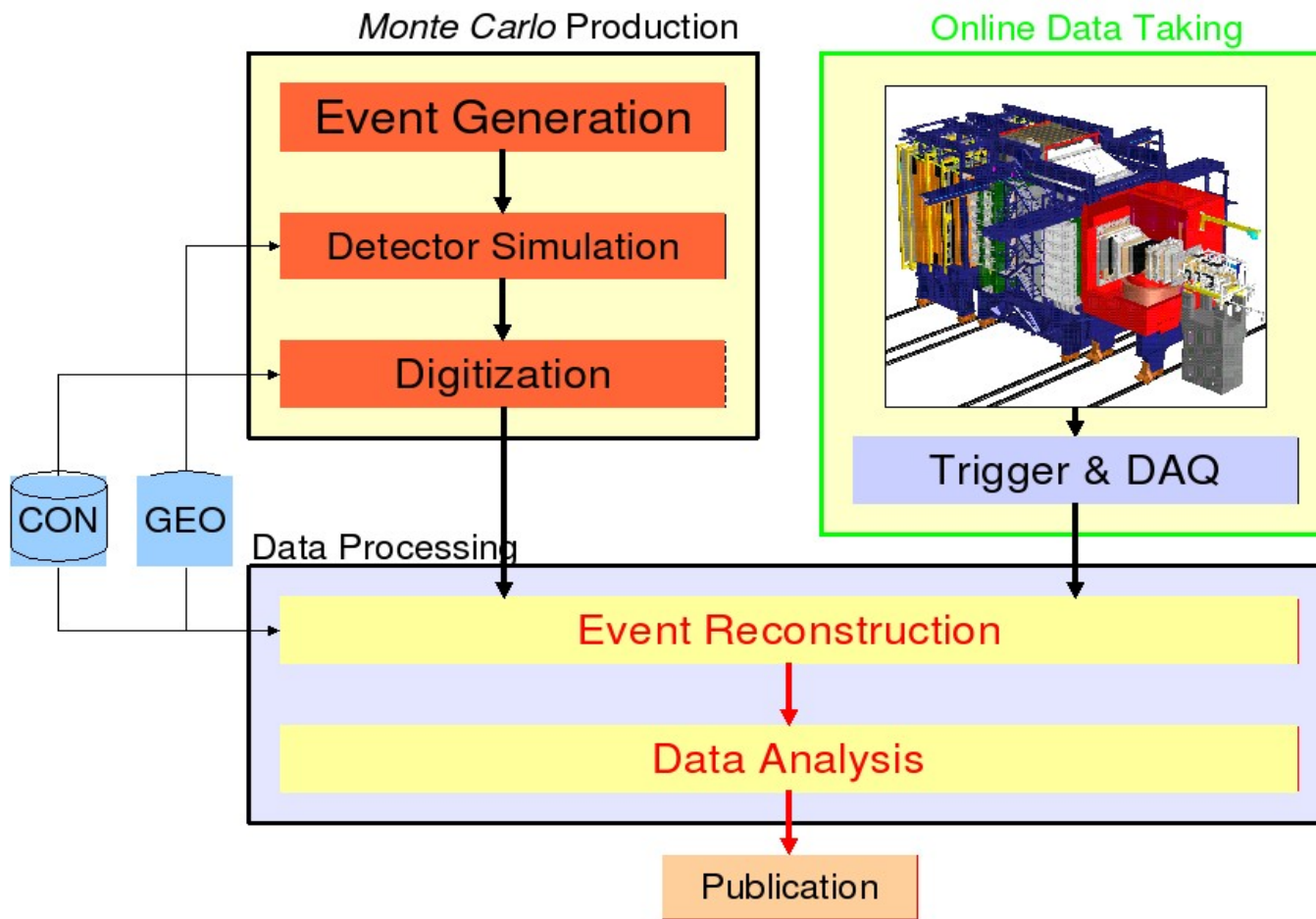
# Job submission to the Grid

- requirements:
  - grid certificate
  - VO membership
  - all files (input, binary, libraries,...) on SE
  - jobscript (JDL) that:
    - retrieves all needed files from SE onto WN
    - sets custom environment
    - executes binary/script
    - stores output on SE
- submission:
  - start your proxy ( secured interface)
  - put all job depended files on SE
  - write your JDL script specifying:
    - name of jobscript
    - arguments
    - input/output Sandbox
    - VO
  - `edg-job-submit *your-jdl-file*`
  - check status via job-ID

# Grid schematic view



# 'summary'



Questions ?

thanks to A.Gellrich for Grid material