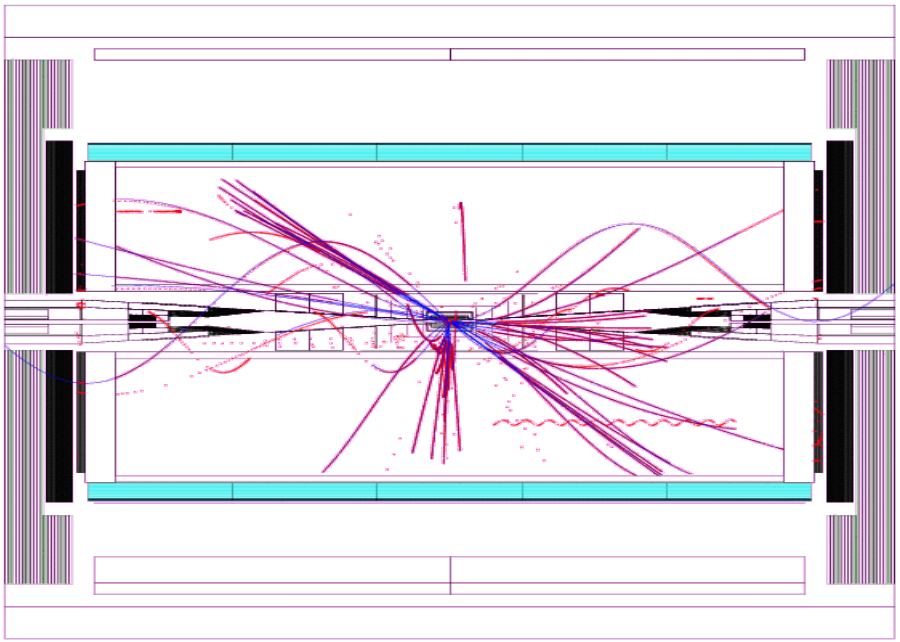
Simulation Framework

- Full simulation systems: status report
- The next steps: where do we go from here
- The data challenge: status report



The LC simulation groups:

Europe:	David Ward, Ties Behnke	:http://www-zeuthen.de/linear_coll
US:	Norman Graf	:http://blueox.uoregon.edu/~lc/alcpg/
Asia:		:http://acfahep.kek.jp

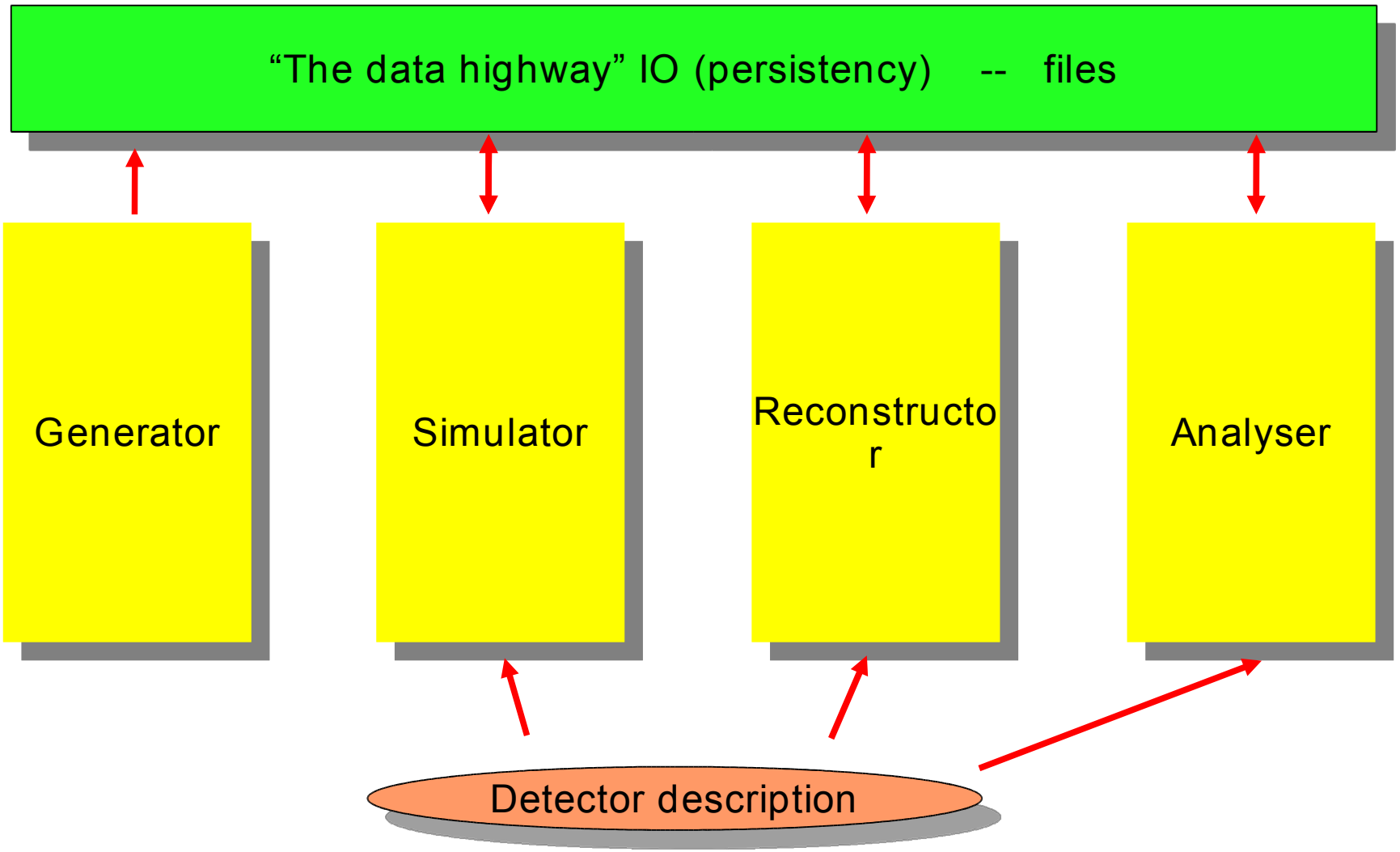
The mission

- Coordinate the creation of appropriate simulation tools
- Coordinate the creation of appropriate reconstruction tools
- Provide a frame for the creation of appropriate analysis tools

The boundary conditions:

- System should be long lived
- Should be lightweight
- We are not a collaboration: commercial software is difficult (funding)
- Share across regions as much as possible
- Little to no person power available

The ingredients



Full simulation

- BRAHMS:
  - GEANT3/ f77 based
  - TESLA TDR detector updated for recent changes
  - Complete simulation (and reconstruction)

- MOKKA:
  - GEANT4/ C++ based
  - TESLA TDR detector “more or less”
  - No reconstruction framework

- BRAHMS is still the most complete package, but
- MOKKA is quickly getting there, will soon replace BRAHMS SIM

The future of the simulator(s)

MOKKA is our main GEANT4 based full simulation environment

- GEANT4 based
- light weight, C++ code
- geometry via mysql and C++ drivers

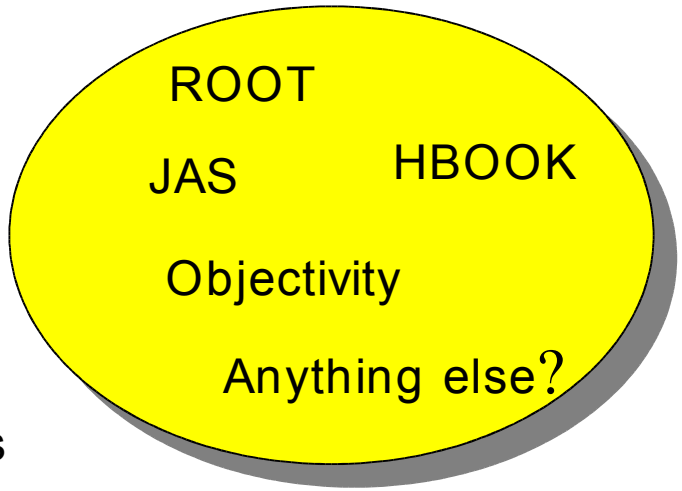
Our US collegues are evaluating MOKKA to use it as a base for their GEANT4 simulation environment.

MOKKA developments:

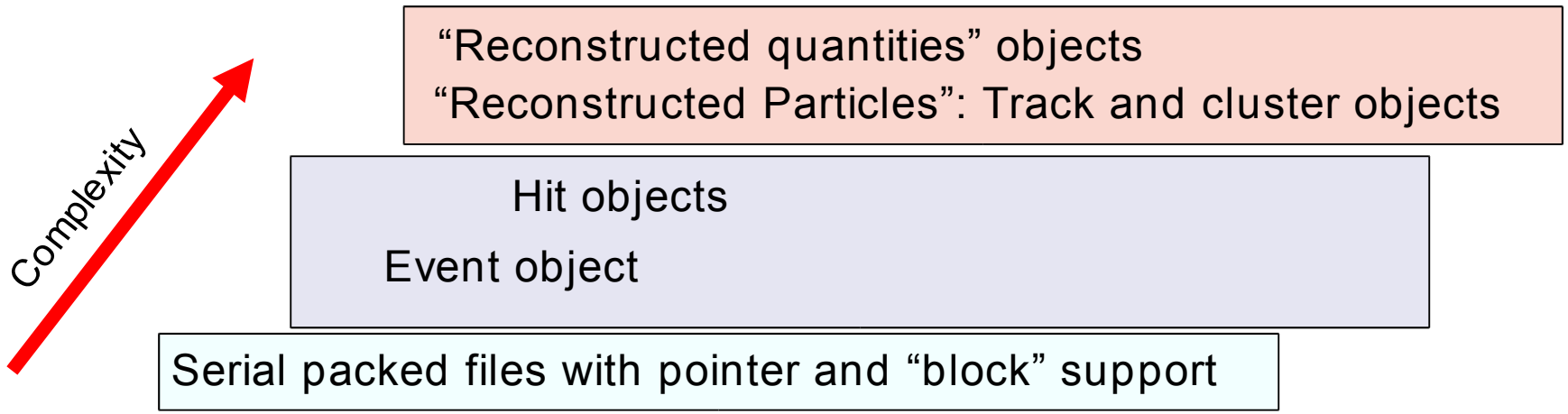
- Improve the detector models
- allow for more flexible system: concept of user plugins
- Watch closely CERN geometry developments (GDML)

The data highway: LCIO

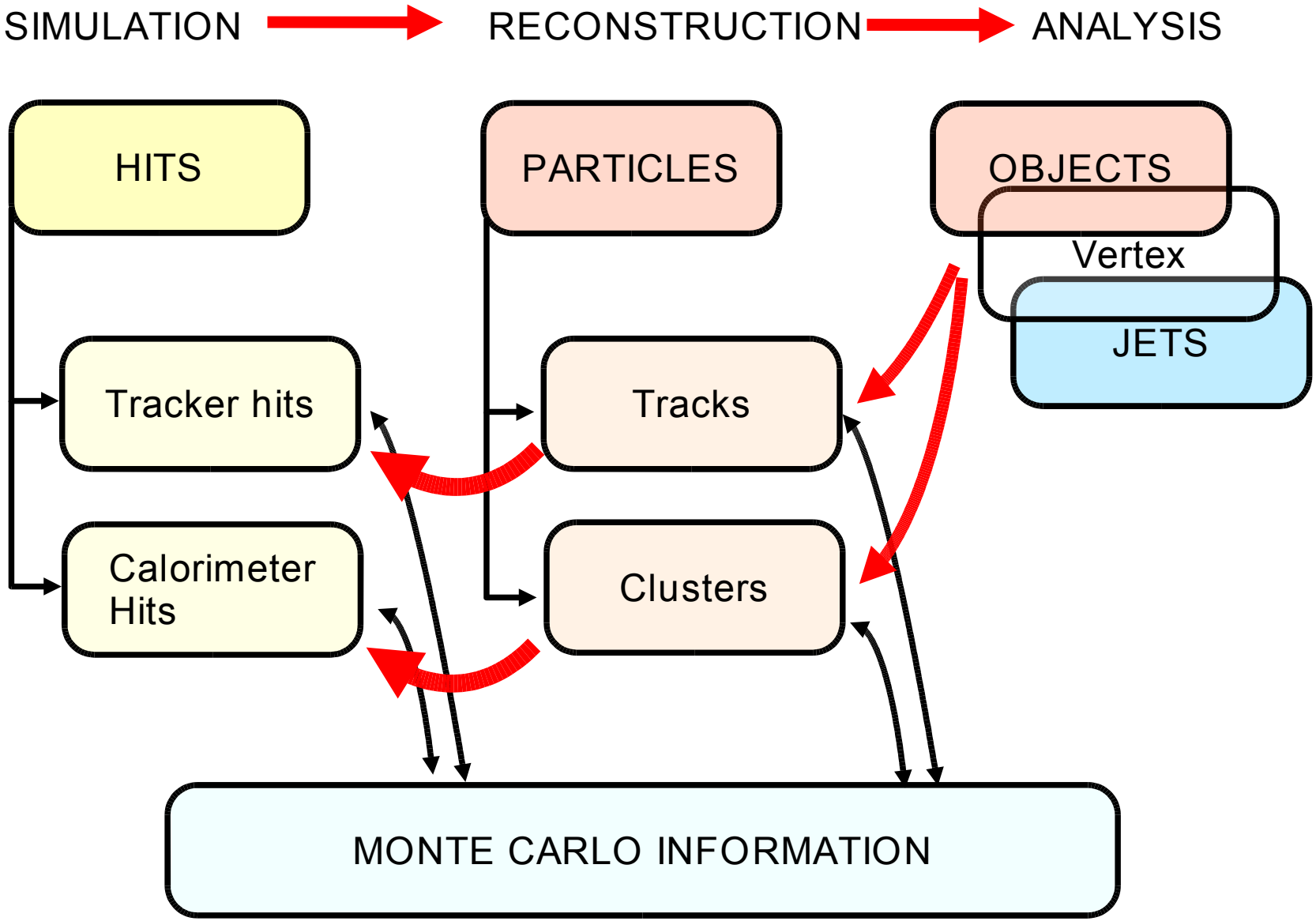
- Need definition of a data model:
  - What is a hit
  - What is a track
  - What is a calorimeter cluster
  - etc.
- Need a common way to store ("persist") objects



- **LCIO**: a simple data model with underlying persistency system:



LCIO: LC persistency framework



LCIO: Status

- Developed in collaboration between SLAC / DESY / LLR (France)
- Common data model / persistency framework for LC studies
- Simple API to store and retrieve data
  - Same API in C++, Java (and Fortran)
  - Simple underlying IO format (SIO): can be changed easily at a later time

Status:

- Development started December 2002
- First public pre-release in March
- First public production release for Montpellier: LCIO 1.0
  - Full C++ implementation
  - Full Java implementation
  - Full Fortran implementation
  - Hit based data model “fully” defined, reconstructed objetcs under discussion

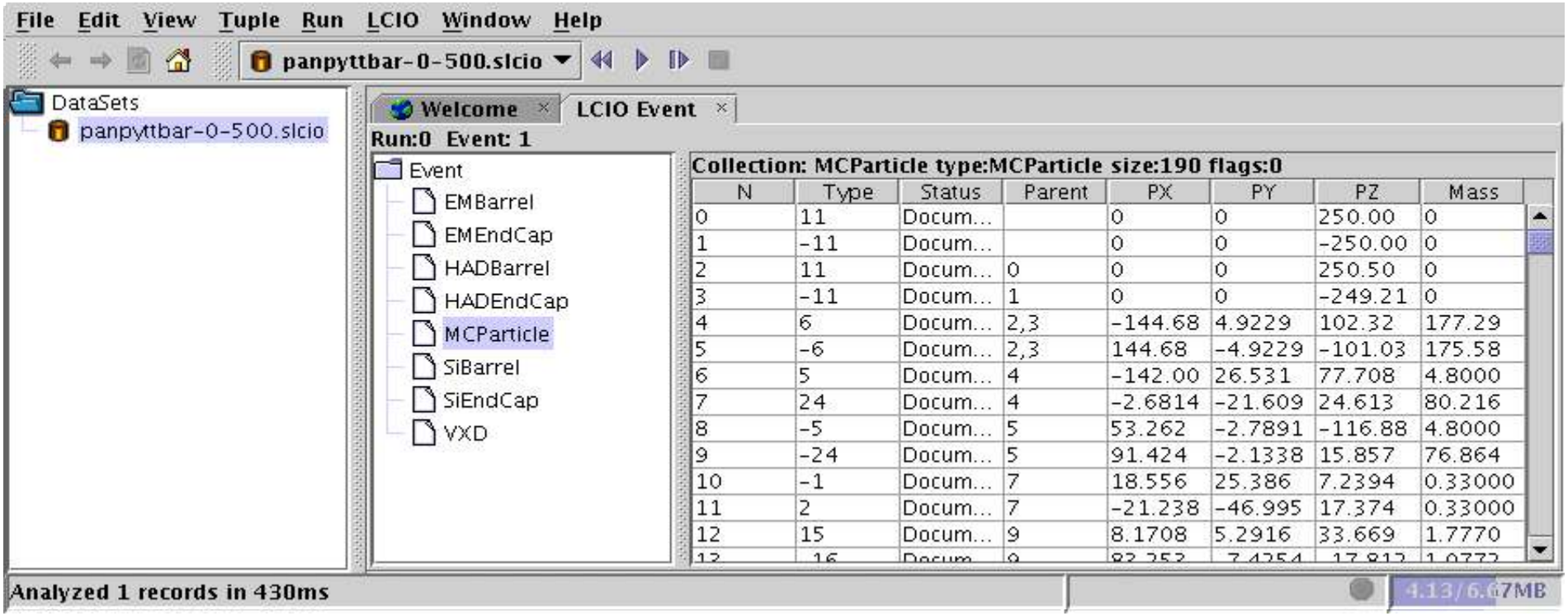
http://www-it.desy.de/physics/projects/simsoft/lcio

LCIO: Implementation

- LCIO interface exists for
  - MOKKA
  - BRAHMS

Write *.slcio files

Simple lcio browser in JAS3:



Similar functionality exists in C++ (examples in distribution)

Reconstruction

- BRAHMS/ f77 based reconstruction:
  - Complete and sophisticated tracking
  - Particle flow reconstruction package
  - Tools (n-tuple for analysis, interface to main packages like ZVTOP, ...)
- OO world:
  - No coherent reconstruction frame yet
  - LCIO allows using f77 based reconstruction package on MOKKA output (REPLIC, BRAHMS reco to be released this week)

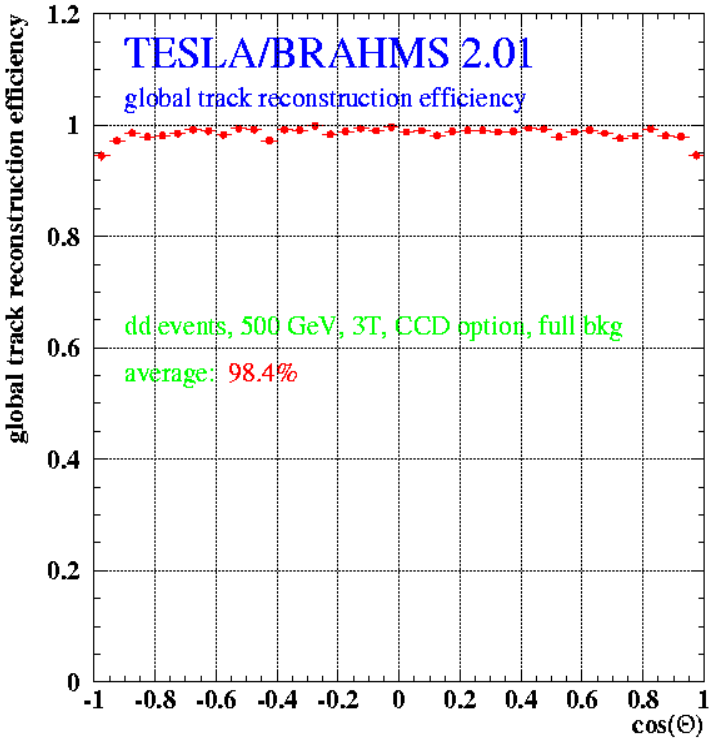
We need to move fast towards a OO based reconstruction framework, to enable reconstruction program developments!

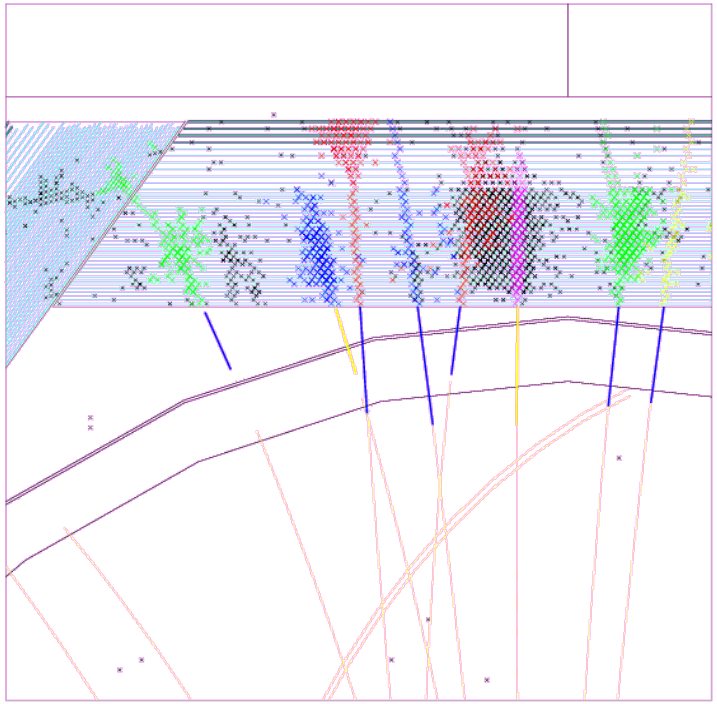
Reconstruction

Status of reconstruction packages:

BRAHMS:

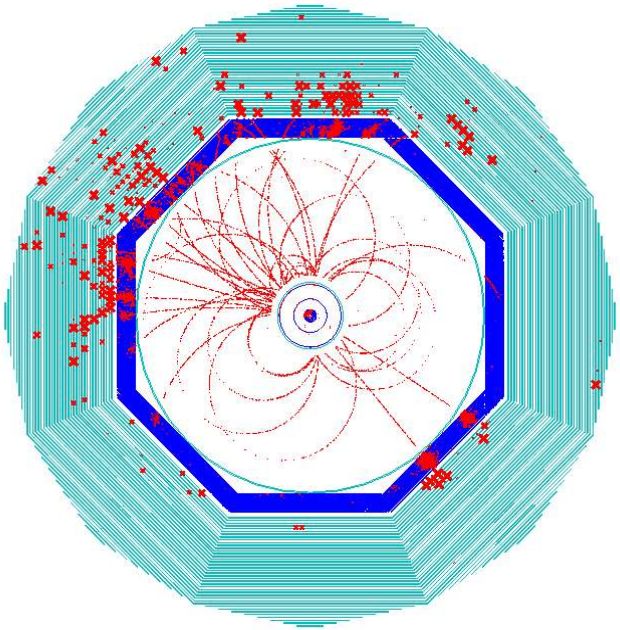
- sophisticated and complete reconstruction software available
- OO based: no reconstruction tools available at the moment



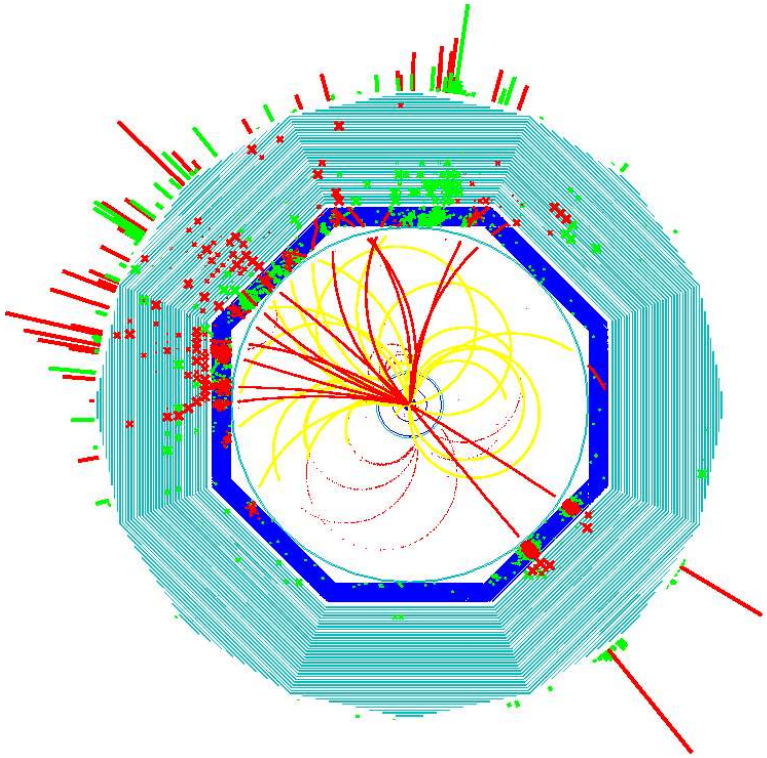


Reconstruction

Hits simulated in detector (BRAHMS)



Tracks and Pflow objects reconstructed (BRAHMS reco)



Plots Karsten Buesser

The Plan

- Develop a plain C++ reconstruction framework
  - LCIO as data model
  - Possibly CGA as geometry access model
  - Very simple interface, no fancy interactive environment for a start
  - Multi language support is no problem (JAVA ...)

Use LCIO ROOT
LCIO JAS
to provide a simple user interaction with the reconstruc

At this stage:
remain independent of any particular environment (root, JAS, PAW, ...)
be open for future developments
be light-weight (minimise the number of libraries the user needs to install)

.... but we are still far from this goal...

Fast Simulation

- SIMDET: fast, parametrised Monte Carlo

  NEW: interface to used track based ZVTO P in SIMDET
  improvements to parametrisations

- SGV: fast, semi-parametrised Monte Carlo (produces hits)

  Our US collegues are developing a similar system

  NEW: interface to ZVTOP in SGV

At the moment, fast simulations work, are producing physics results but
we have no project for a modern fast simulator

Software: Outlook

Simulation:

- Converge towards GEANT4 based MOKKA simulation
- program should become more flexible and user friendly
- Close collaboration with our US collegues on simulation establised

IO:

- LC specific persistency scheme and data model established
- Interface to C++, JAVA and Fortran exists
- Common development with our US collegues

Reconstruction:

- Next big project
- Digitisation and reconstruction “complete” in BRAHMS
- Much effort needed in OO framework: the next project

Towards a data challenge

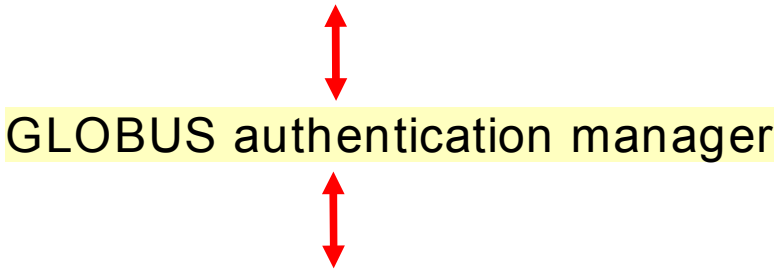
Software environment: see previous slides

Access to data:

- Need transparent and flexible access system
- Data volume small compared to LHC, but still significant

The plan:

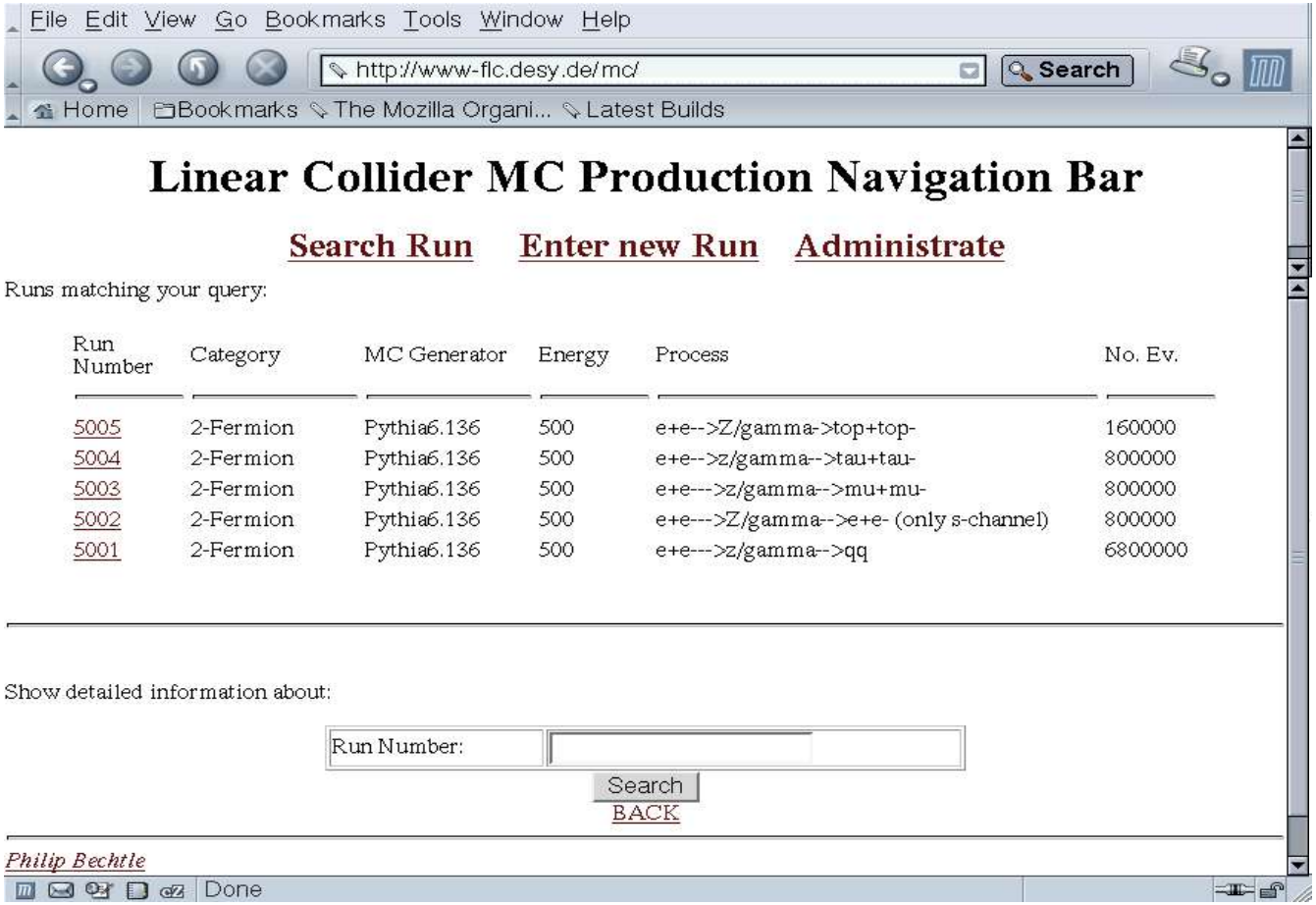
- User interface/ interaction: use the GRID computing model and software



- Storage manager: dCache (FNAL and DESY, integrated into GRID)

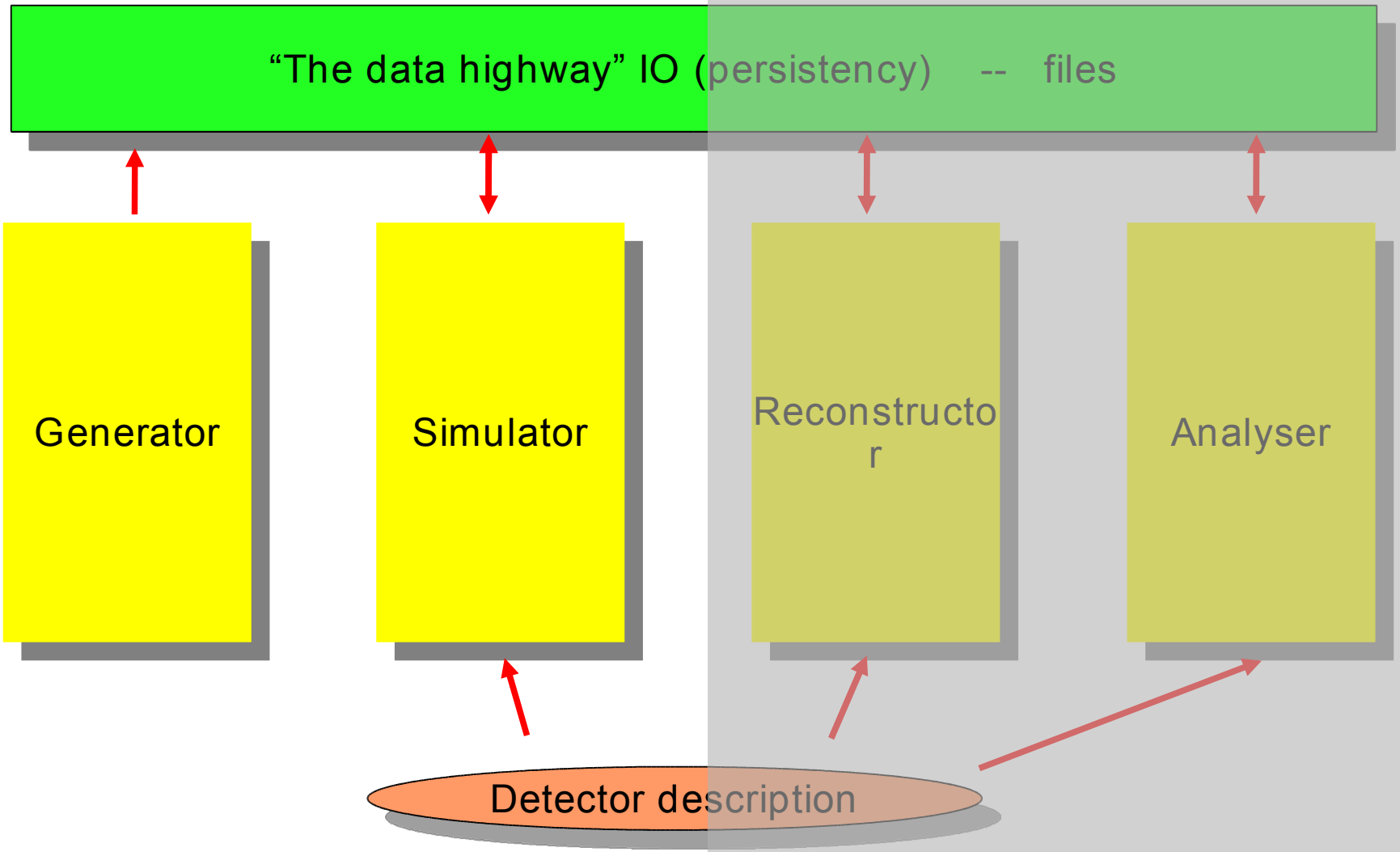
LC Data depository

- Database interface to data stored under dCache at DESY



- Current problems: access from outside DESY

Summary


"The data highway" IO (persistency) -- files
Generator
Simulator
Reconstructo
r
Analyser
Detector description

Summary

- Software tools present a significant challenge
- Progress on the simulation:
  - GEANT3 based simulation more stable and complete
  - GEANT4 based simulation becoming available
  - Common persistency scheme is reality
- Progress on the reconstruction:
  - BRAHMS based reconstruction exists
  - OO reconstruction still in the future
  - LCIO input exists
  - LCIO output being defined
- Goals:
  - GEANT4 based simulator
  - Simple reconstruction environment independent of particular framework
  - LCIO as basic data format and model to facilitate data exchange
  - Management of data sets through GRID like system