ILC Software: Where are we?

Ties Behnke, DESY

Software: where are we

where are we going in the future

Software is always difficult to discuss

many personal likes and dis-likes

many clear and often very vocal opinions

What does “core software” mean for a project which is not yet a project

“Central” ILC Software: Why now?

Ties Behnke, DESY

Software becomes an integral part of the “detector development”

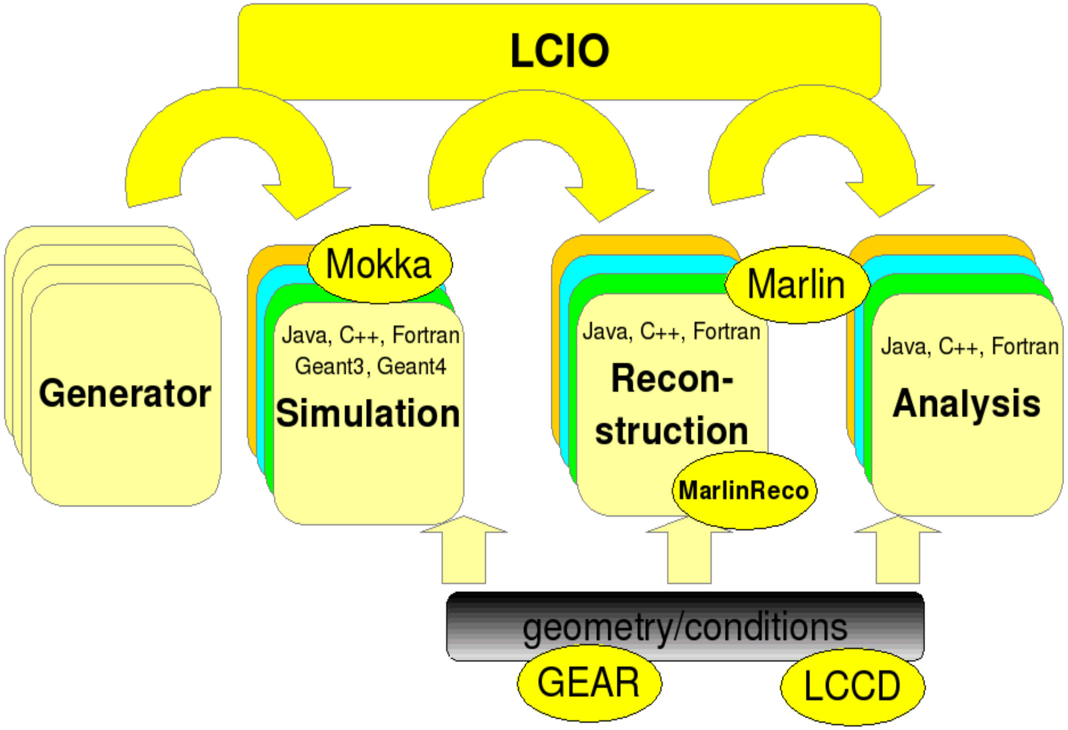
- close relation between optimization and software
- algorithm development becomes centrally important
- data management becomes an important topic early on

Requirements

- easy to use, easy to install, easy to run
- has to be done in “my own” preferred system and language

Where are we?

- Core software packages - status and latest developments
  - LCIO
    - data model & persistency
  - MARLIN
    - C++ application framework
  - LCCD
    - conditions data toolkit
  - GEAR
    - geometry description
  - LCGEO
    - a proposal for a new geometry package

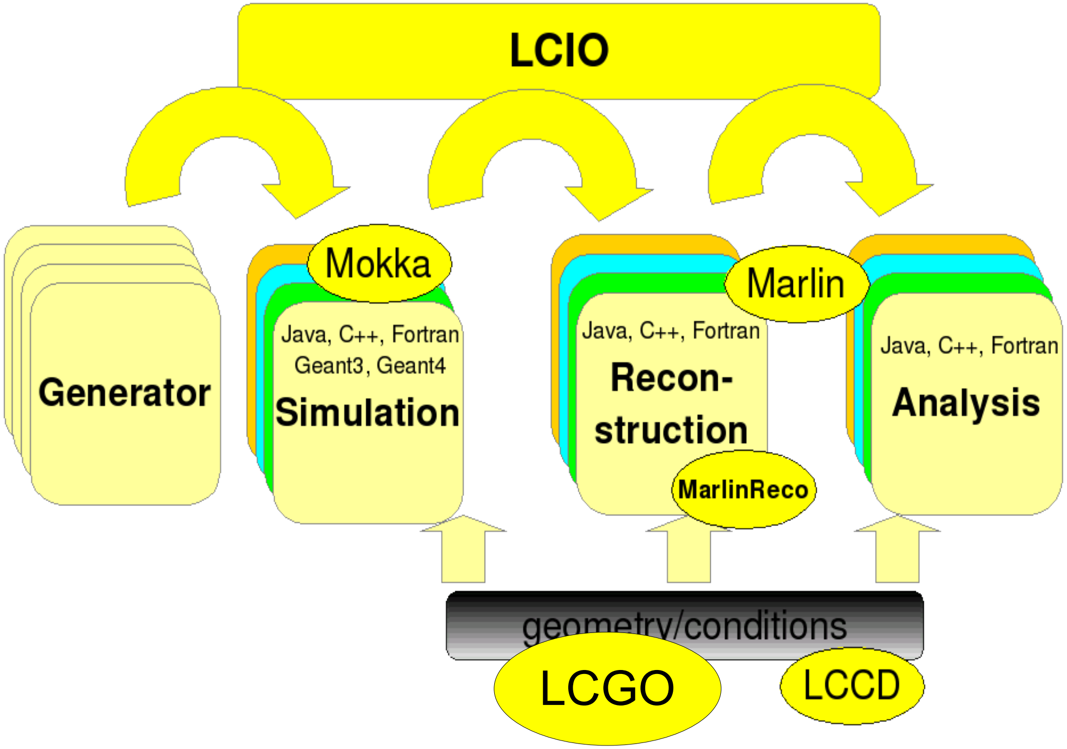


F.Gaede

+ GEANT4, stdhep, clhep, gsl, ... (external packages)

Where are we?

- Core software packages - status and latest developments
  - LCIO
    - data model & persistency
  - MARLIN
    - C++ application framework
  - LCCD
    - conditions data toolkit
  - GEAR
    - geometry description
  - LCGO
    - a proposal for a new geometry package

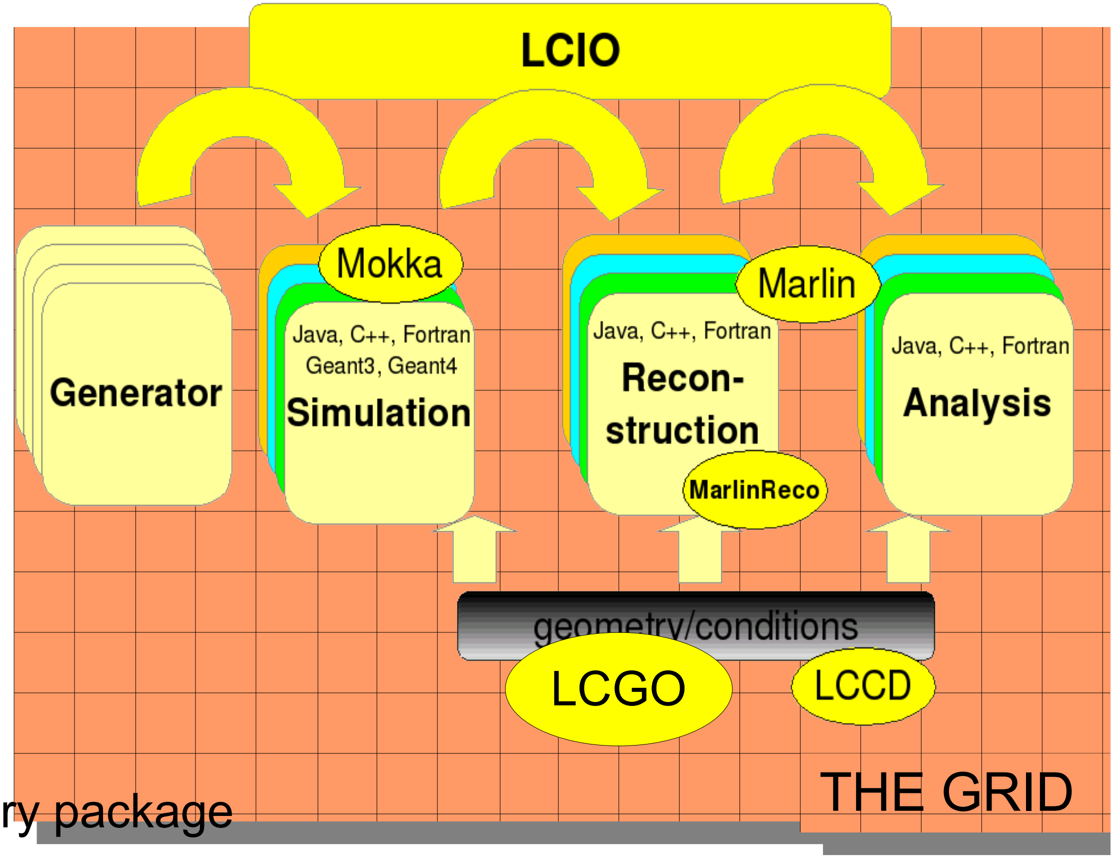


F.Gaede

+ GEANT4, stdhep, clhep, gsl, ... (external packages)

Where are we?

- Core software packages - status and latest developments
  - LCIO
    - data model & persistency
  - MARLIN
    - C++ application framework
  - LCCD
    - conditions data toolkit
  - GEAR
    - geometry description
  - LCGO
    - a proposal for a new geometry package



F.Gaede

+ GEANT4, stdhep, clhep, gsl, ... (external packages)

The basic concept

Based on C++ as core language (widely accepted, though not easy, large common base with LHC)

Lightweight tools, well defined functionality, no large overhead

re-use existing work as much as possible

leave maximum space for personal freedom in development

follow very much the toolbox approach (as opposed to the complete package approach)

Integrate with the GRID for data management and processing

Core packages provide the user with a frame
functionality needs to be filled in by the community / user

Interlude: Interfaces

The definition of interfaces is (one of) the main task(s) of core software:

Examples: LCIO is primarily an interface
AIDA is an histogramming interface
GEAR is a geometry interface
LCGO will be a geometry interface
....

- Independent from the implementation (SIO in LCIO, ROOT in AIDA, ...)
- Software remains portable and adaptable
- Scales with the number of systems and complexities
- Allows cooperative developments
- Might eventually allow multi-language support

Practical implications

Interfaces need to be defined:

Significant amount of work
usually defined interfaces do not answer your immediate needs...

Interfaces have to be accepted by the developers and users

Saving are not immediatly apparent
often it is seen as restrictive and slowing down the work

“I need to get this information from processor A to processor B, therefore I created a static class .....”
“I cannot be bothered with dealing with the interface, is slows down my work...”

Practical implications

Interfaces need to be defined:

Significant amount of work
usually defined interfaces do not answer your immediate needs...

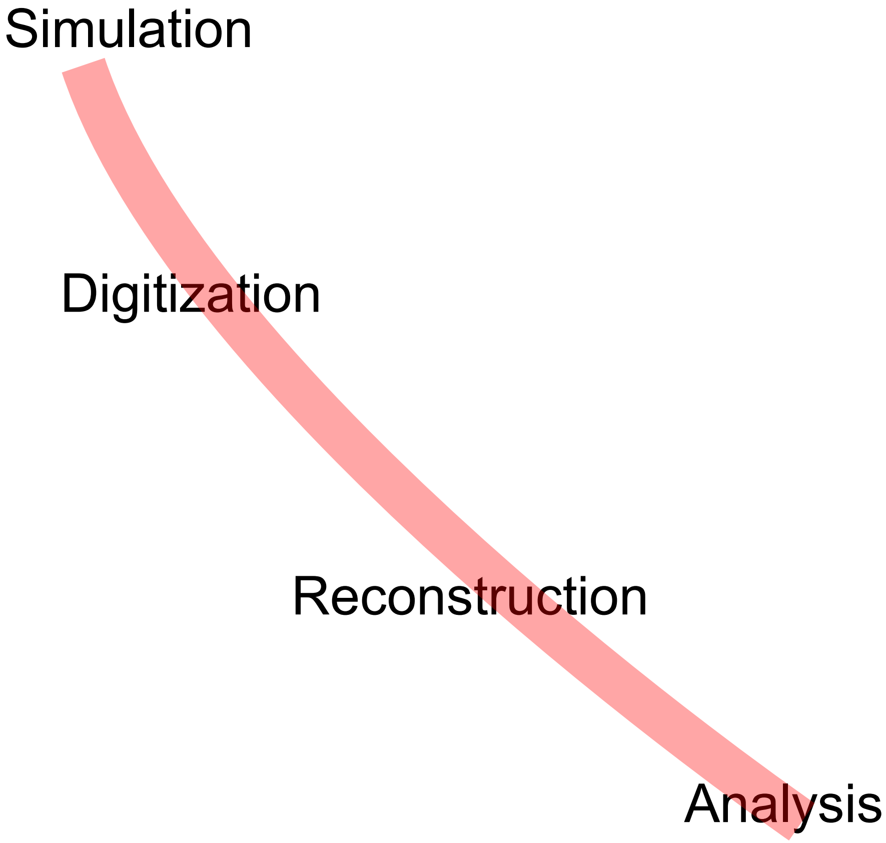
Interfaces have to be accepted by the developers and users

Saving are not immediatly apparent
often it is seen as restrictive and slowing down the work

“I need to get this information fro ssor A to processor B,
therefore I created a static cl
“I cannot be bothered wit ng with the interface, is slows
down my work...”

Its ALL or NOTHING!!

What is there?



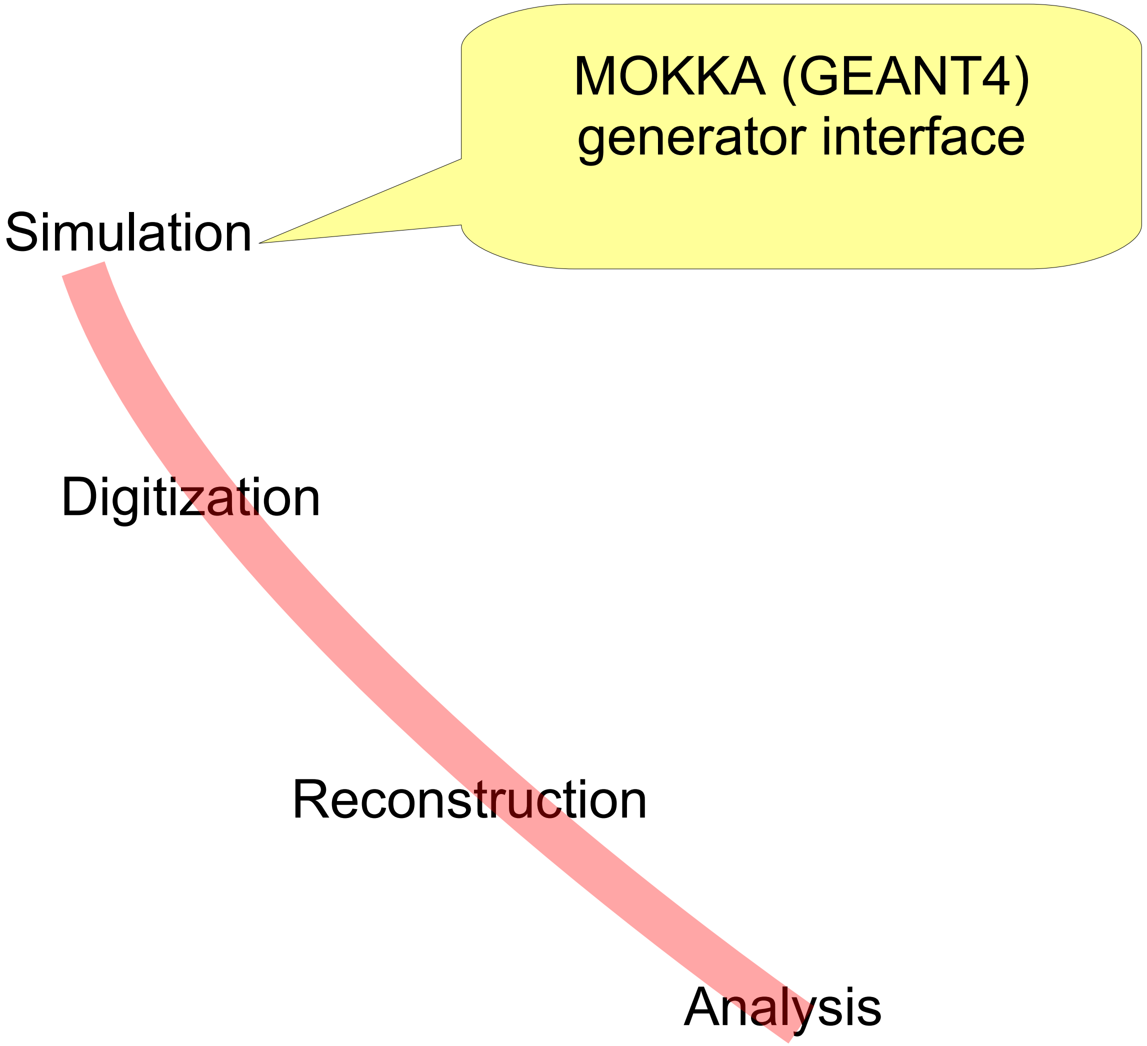
Simulation

Digitization

Reconstruction

Analysis

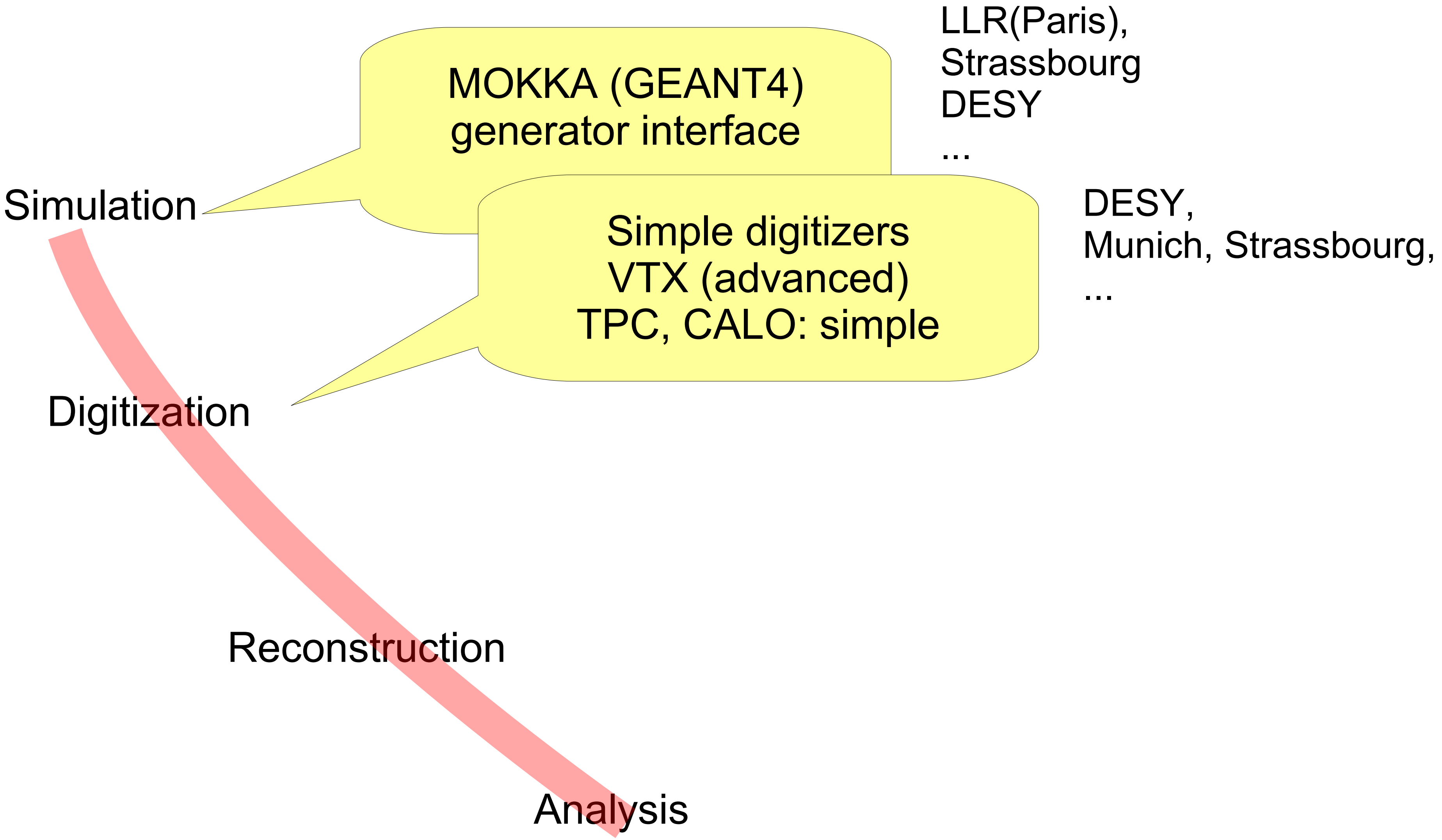
What is there?



MOKKA (GEANT4)
generator interface

LLR(Paris),
Strassbourg
DESY
...

What is there?



MOKKA (GEANT4)
generator interface

LLR(Paris),
Strassbourg
DESY
...

Simulation

Simple digitizers
VTX (advanced)
TPC, CALO: simple

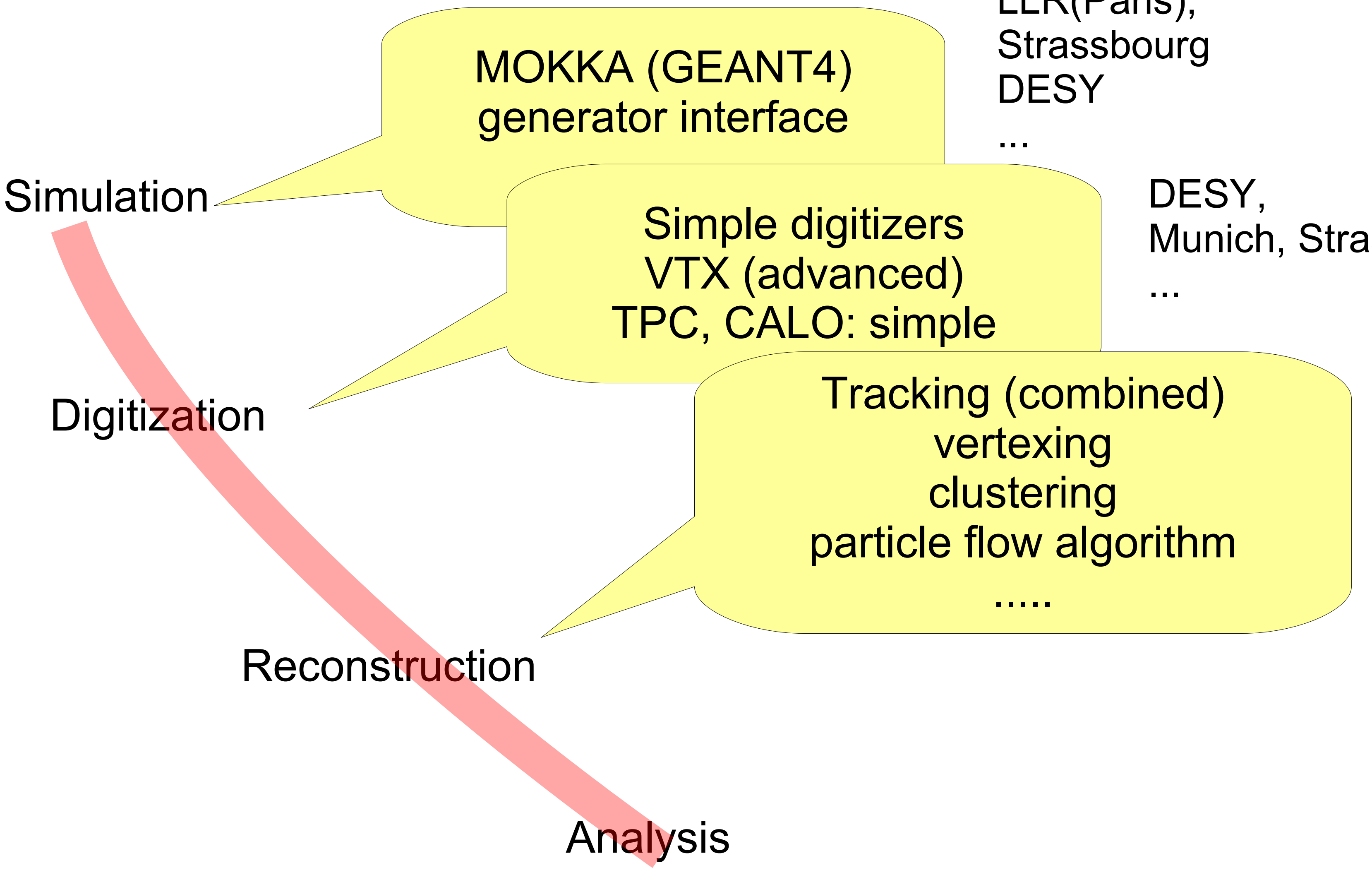
DESY,
Munich, Strassbourg,
...

Digitization

Reconstruction

Analysis

What is there?



Simulation

MOKKA (GEANT4)
generator interface

LLR(Paris),
Strassbourg
DESY
...

Digitization

Simple digitizers
VTX (advanced)
TPC, CALO: simple

DESY,
Munich, Strassbourg,
...

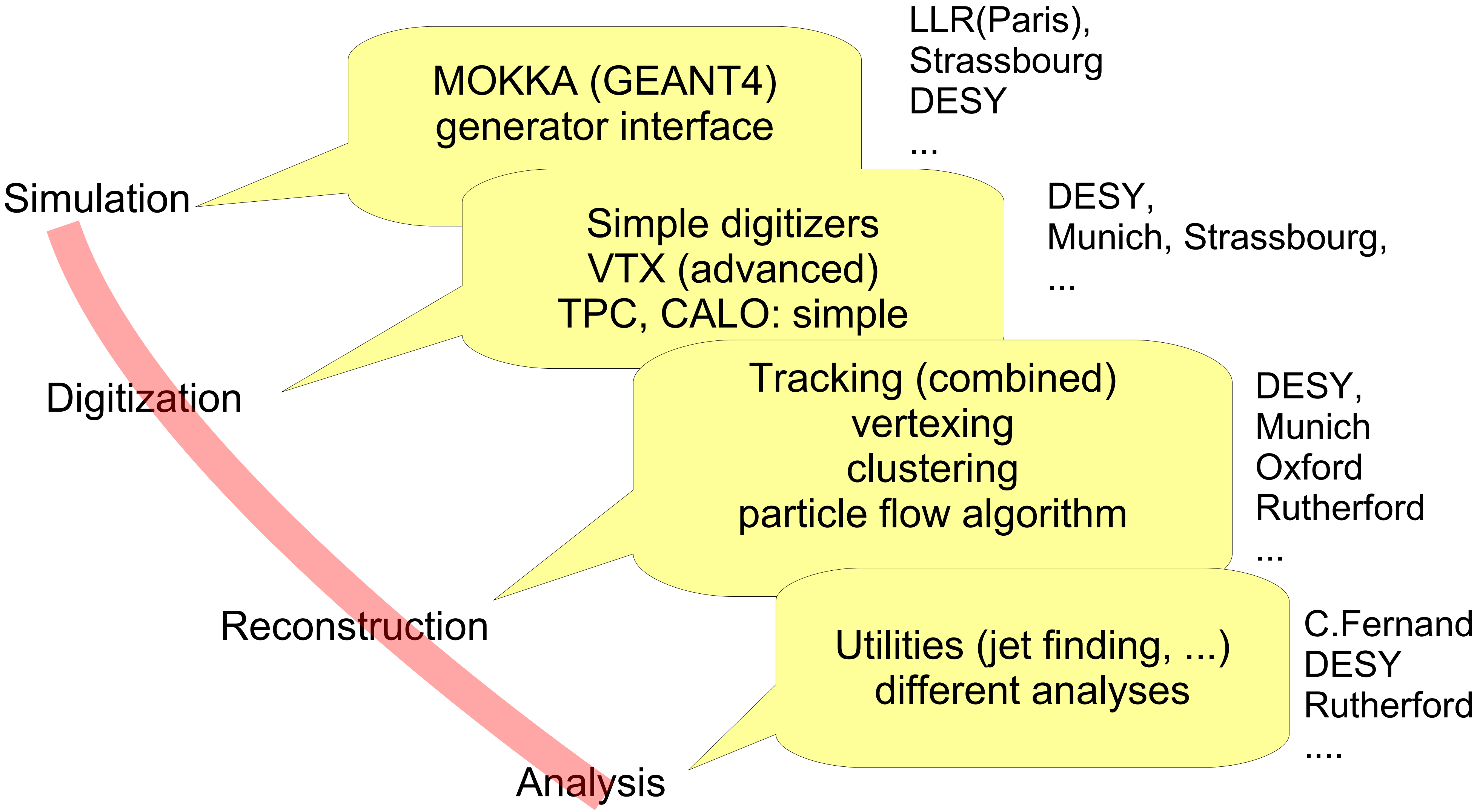
Reconstruction

Tracking (combined)
vertexing
clustering
particle flow algorithm
.....

DESY,
Munich
Oxford
Rutherford
...

Analysis

What is there?



Simulation

MOKKA (GEANT4)
generator interface

LLR(Paris),
Strassbourg
DESY
...

Digitization

Simple digitizers
VTX (advanced)
TPC, CALO: simple

DESY,
Munich, Strassbourg,
...

Reconstruction

Tracking (combined)
vertexing
clustering
particle flow algorithm

DESY,
Munich
Oxford
Rutherford
...

Analysis

Utilities (jet finding, ...)
different analyses

C.Fernand
DESY
Rutherford
....

MARLIN-RECO

Suite of reconstruction tools based on the MARLIN framework

In this sense MARLIN-Reco is one implementation of reconstruction code within a given software model

Provides in one package

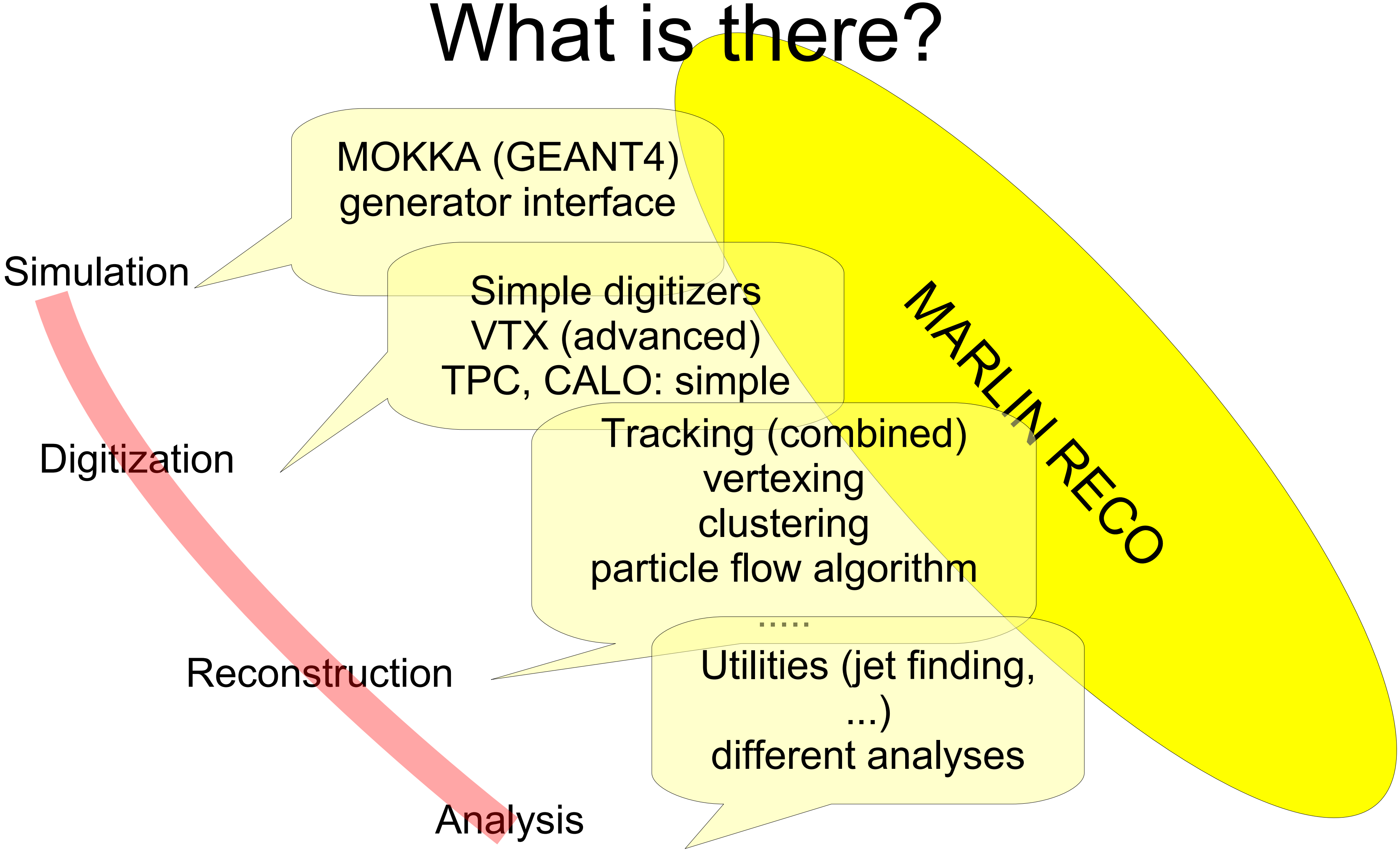
- Tracking
- Vertexing
- Calorimeter reconstruction
- Particle flow
- Helper applications

Fairly complete,

modularity allows the user to built its own if he / she so wants to do that

but far from finished

What is there?



Simulation

MOKKA (GEANT4)
generator interface

Digitization

Simple digitizers
VTX (advanced)
TPC, CALO: simple

Reconstruction

Tracking (combined)
vertexing
clustering
particle flow algorithm
.....

Analysis

Utilities (jet finding, ...)
different analyses

MARLIN RECO

What is not there

Main problem: coherent and easy to use geometry interface

LCGO is designed to close this gap, but it is not yet there
(common DESY/ SLAC development)

State of Algorithms:

- Many are there, but need optimization, debugging, and your ideas
- This is particularly true for particle flow:

  WOLF, PandoraPFA, track based PFA
  many systems, but do not expect something which is “ready”

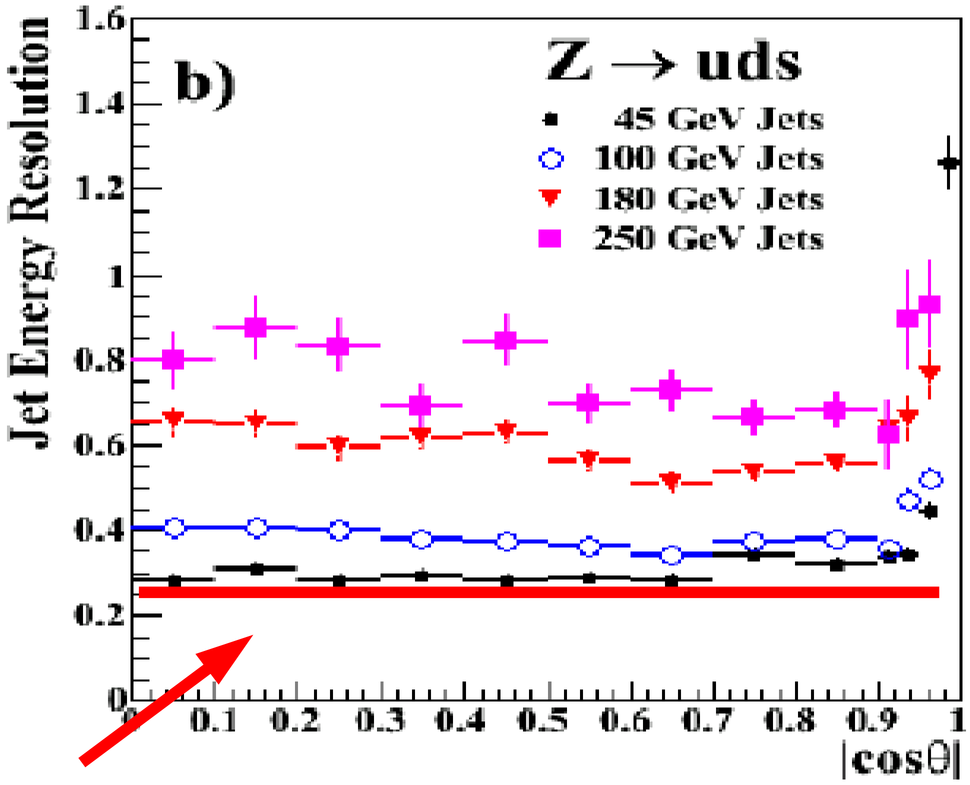
- Many useful tools are missing

State of Particle Flow

2 major packages

WOLF

PandoraPFA

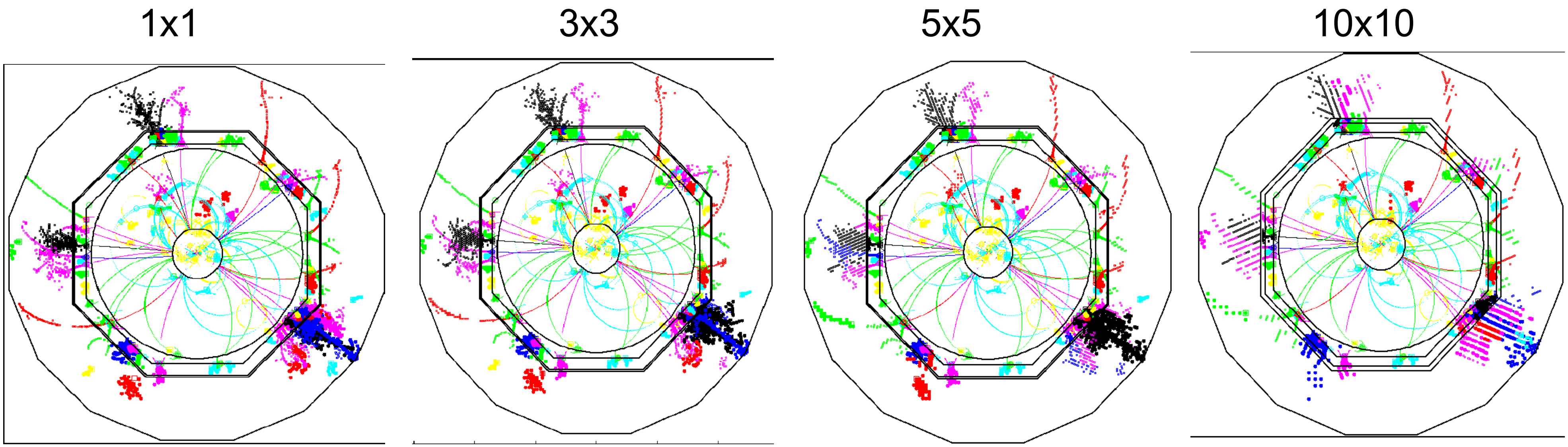


Performance of PandoraPFA

for different energies

ILC goal: 30%/√E

Detector Optimization



HCAL cell	Z peak	Ttbar (500)
1x1	31.4+-0.3%	42+-1%
3x3	30.6+-0.3%	45+-1%
5x5	31.1+-0.3%	48+-1%
10x10	33.7+-0.3%	56+-1%

Visible energy resolution

10x10 clearly worse, gain when going from 5x5 to 1x1 less obvious

M. Thomson

Real Life applications

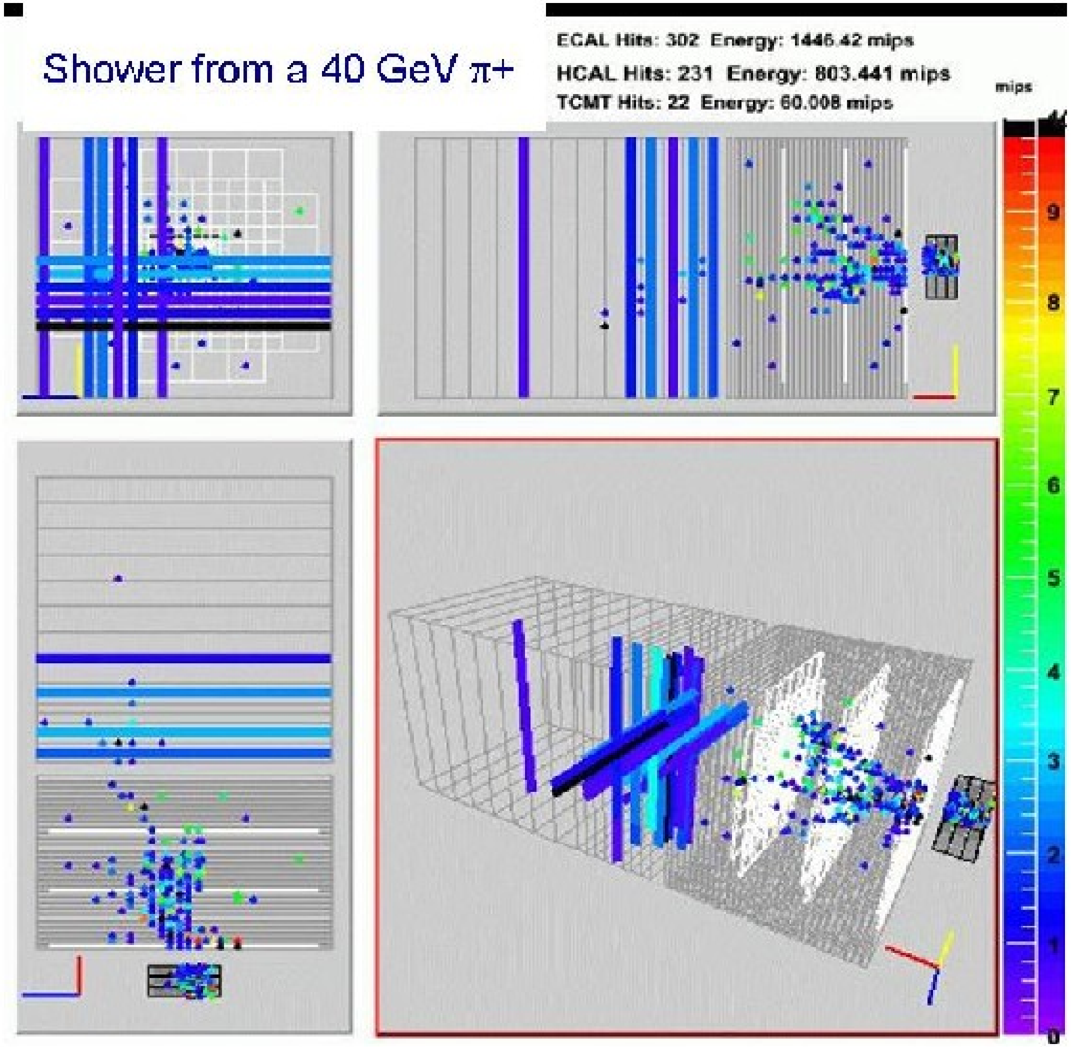
Tools are used in test beam experiments:

- CALICE: see Romans talks in this session
- TPC
- SILC: Silicon Tracking effort
- FCAL: forward instrumentation

EUDET provides a bracket to enforce common developments

Most important: LCIO as a common data model ensures the compatibility of different components

A Calice Test beam event


Shower from a 40 GeV π+
ECAL Hits: 302 Energy: 1446.42 mips
HCAL Hits: 231 Energy: 803.441 mips
TCMT Hits: 22 Energy: 60.008 mips
mips
9
8
7
6
5
4
3
2
1
0

Large data sample collected

Offline data are available as LCIO files on the GRID for analysis

Relation to root

Core system does not use ROOT:

- no dependence on a very complex software system
- well defined (and restricted) functionality makes common developments easier
- direct control over releases, compatibility, etc

User is free to use ROOT at any stages

- RAIDA provides interface to histogramming package

Future direction?

- Investigate replacement of IO package by RIO?

Running the Software

Supported systems:

primary system at the moment is linux

The software has been shown to run under windows, but this is not actively supported (lack of resources)

How to install and run the software:

use the software portal http://ilcsoft.desy.de

In the future: if you have access to afs, use centrally installed afs software under linux and download / install/ link against afs

Conclusion

- A coherent software Ansatz exists for ILC studies
- MARLIN/ MARLINRECO is a particular implementation based on a small number of core packages (LCIO in particular)
- It is fairly lightweight, fairly easy to use
- It is flexible and scalable
- It can be used (though it is far from perfect)
  - But it can only improve if it is used
  - And if everyone contributes to the “tool box” with new tools and gadgets

We should stop to develop only for ourselves,
and we improve the methods to develop towards a common goal:

demonstrate the great potential of the ILC and the ILC detector!