Reconstruction and Analysis Software Environment of LHCb

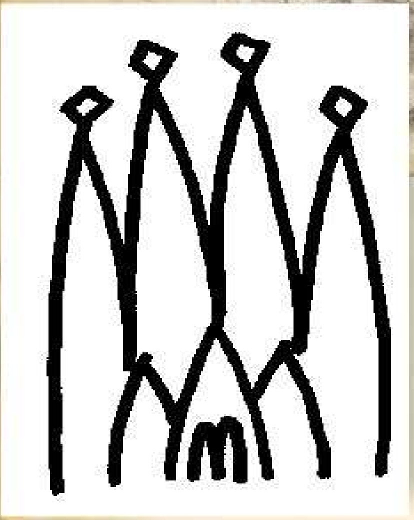
- LHCb reconstruction and analysis
- Software Architecture “**Gaudi**”
- Some Examples:
  - High-Level Trigger
  - Event Display
  - Interactive Analysis

Patrick Koppenburg

CERN / PH

On behalf of the LHCb collaboration

Software Environment of LHCb



- LHCb reconstruction and analysis
- Software Architecture “**Gaudi**”
- Some Examples:
  - High-Level Trigger
  - Event Display
  - Interactive Analysis

Patrick Koppenburg

CERN / PH

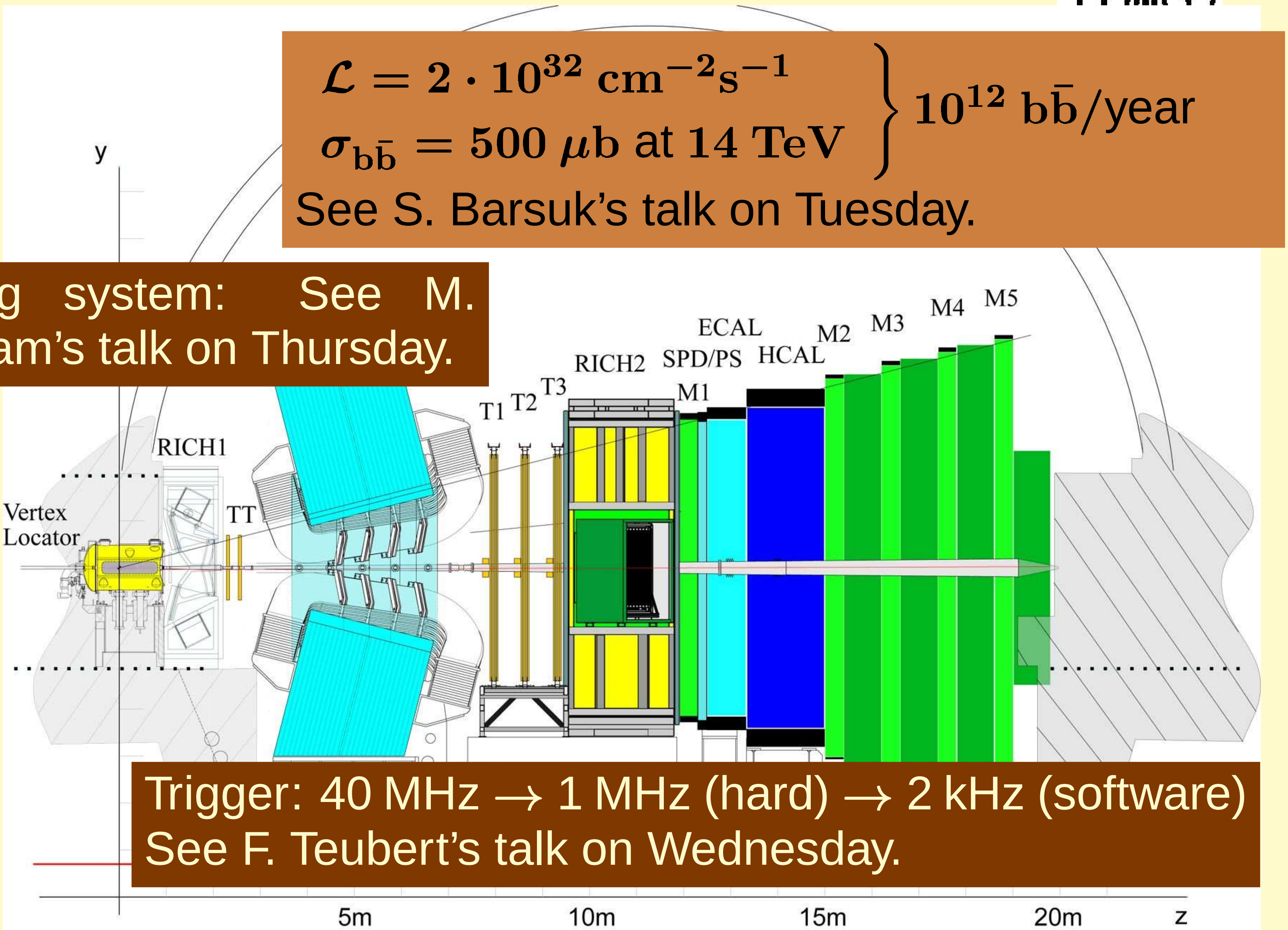
On behalf of the LHCb collaboration

LHCb

$$\left.\begin{array}{l}\mathcal{L} = 2 \cdot 10^{32}\ \mathrm{cm}^{-2}\mathrm{s}^{-1} \\ \sigma_{\mathrm{b\bar{b}}} = 500\ \mu\mathrm{b}\ \text{at}\ 14\ \mathrm{TeV}\end{array}\right\} 10^{12}\ \mathrm{b\bar{b}}/\text{year}$$

See S. Barsuk’s talk on Tuesday.

Tracking system: See M. Needham’s talk on Thursday.



Trigger: 40 MHz → 1 MHz (hard) → 2 kHz (software)
See F. Teubert’s talk on Wednesday.

LHCb Software strategy

Structure: Develop an *Architecture* ("blueprint") and a *Framework* (real code) to be used at all stages of LHCb data processing

- Software triggers, simulation, reconstruction, analysis, visualization. . .
- One single framework used by all members of the collaboration for all code

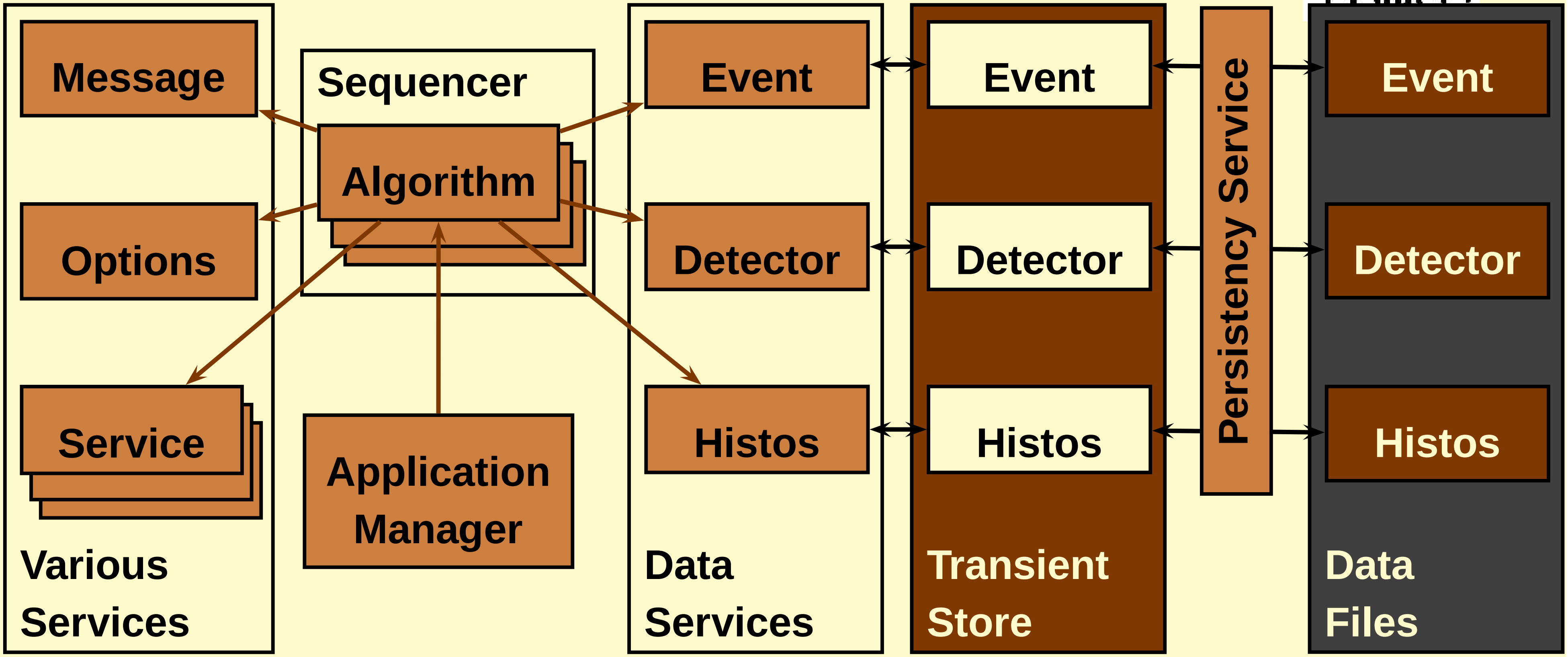
→ **Gaudi**

Development: Avoid duplication of computing effort

- Develop simple components that can be used in any environment
- Use common interfaces to allow easy "plug-and-play" switching of tools. Ex: Vertex Fitters, Cuts. . .

Applications are developed by customizing the framework.

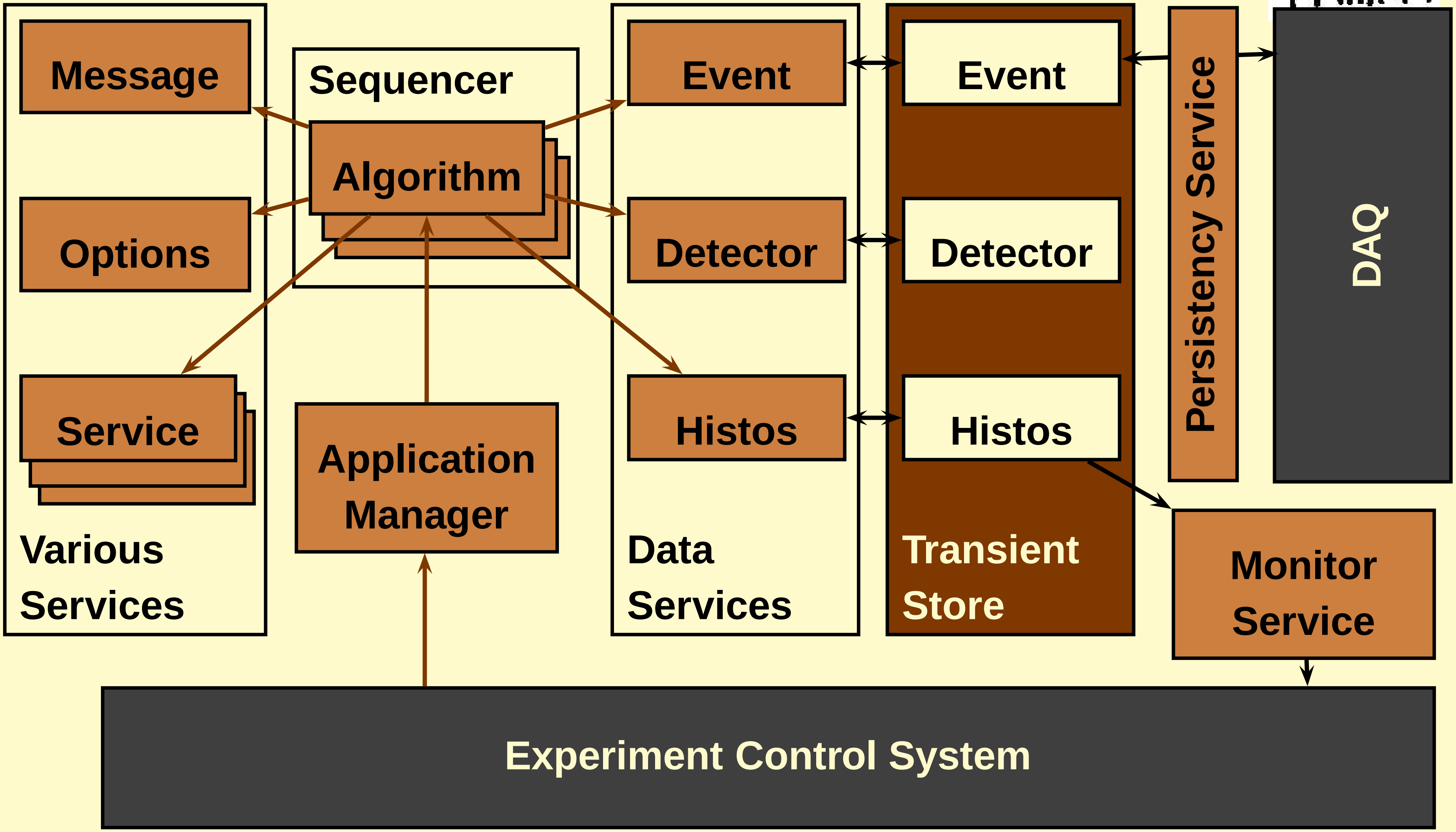
Gaudi Architecture



- Separation of “data” and “algorithms”: Write 00 code, but profit from decades of HEP experience
- Separation of “transient” and “persistent” data
- Physicist code encapsulated in specific places (Algorithms)

P. Koppenburg

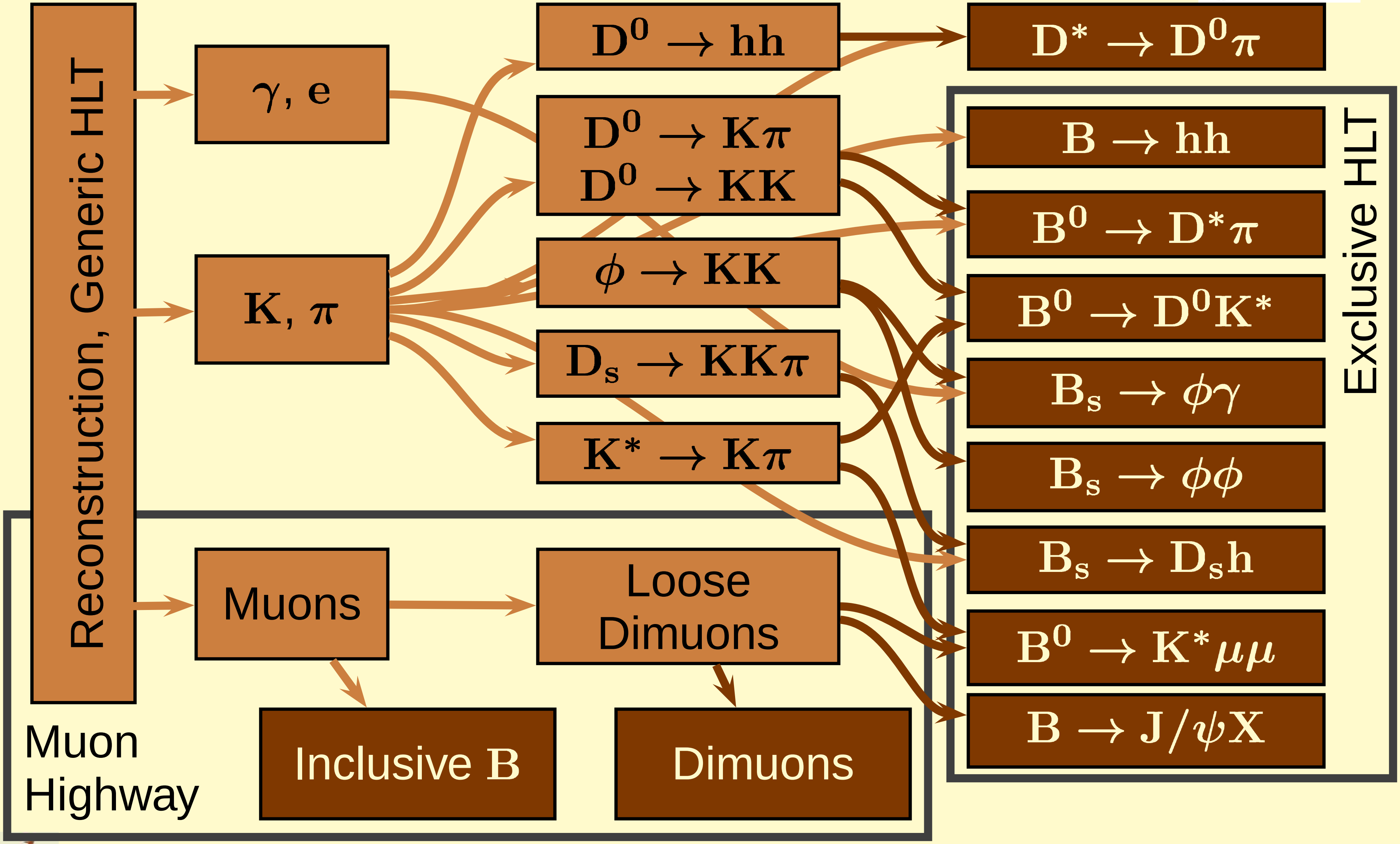
Gaudi Architecture



The same architecture is used online in the software triggers
→ The same algorithms are used

P. Koppenburg

Example: High-Level Trigger



Reconstruction, Generic HLT

γ, e

K, π

Muons

$D^0 \rightarrow hh$

$D^0 \rightarrow K\pi$

$D^0 \rightarrow KK$

$\phi \rightarrow KK$

$D_s \rightarrow KK\pi$

$K^* \rightarrow K\pi$

Loose Dimuons

$D^* \rightarrow D^0\pi$

Exclusive HLT

$B \rightarrow hh$

$B^0 \rightarrow D^*\pi$

$B^0 \rightarrow D^0K^*$

$B_s \rightarrow \phi\gamma$

$B_s \rightarrow \phi\phi$

$B_s \rightarrow D_sh$

$B^0 \rightarrow K^*\mu\mu$

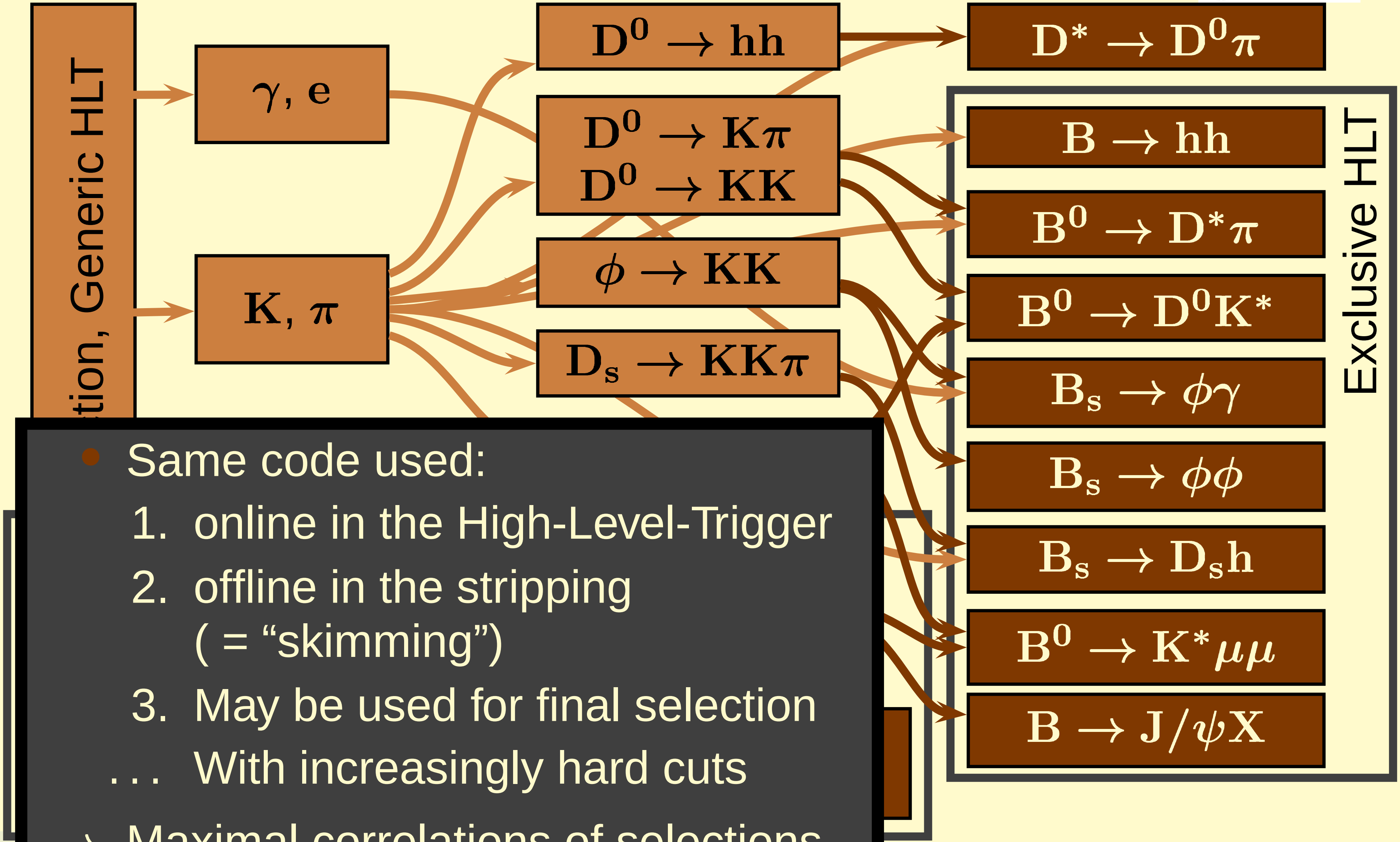
$B \rightarrow J/\psi X$

Muon Highway

Inclusive B

Dimuons

Example: High-Level Trigger



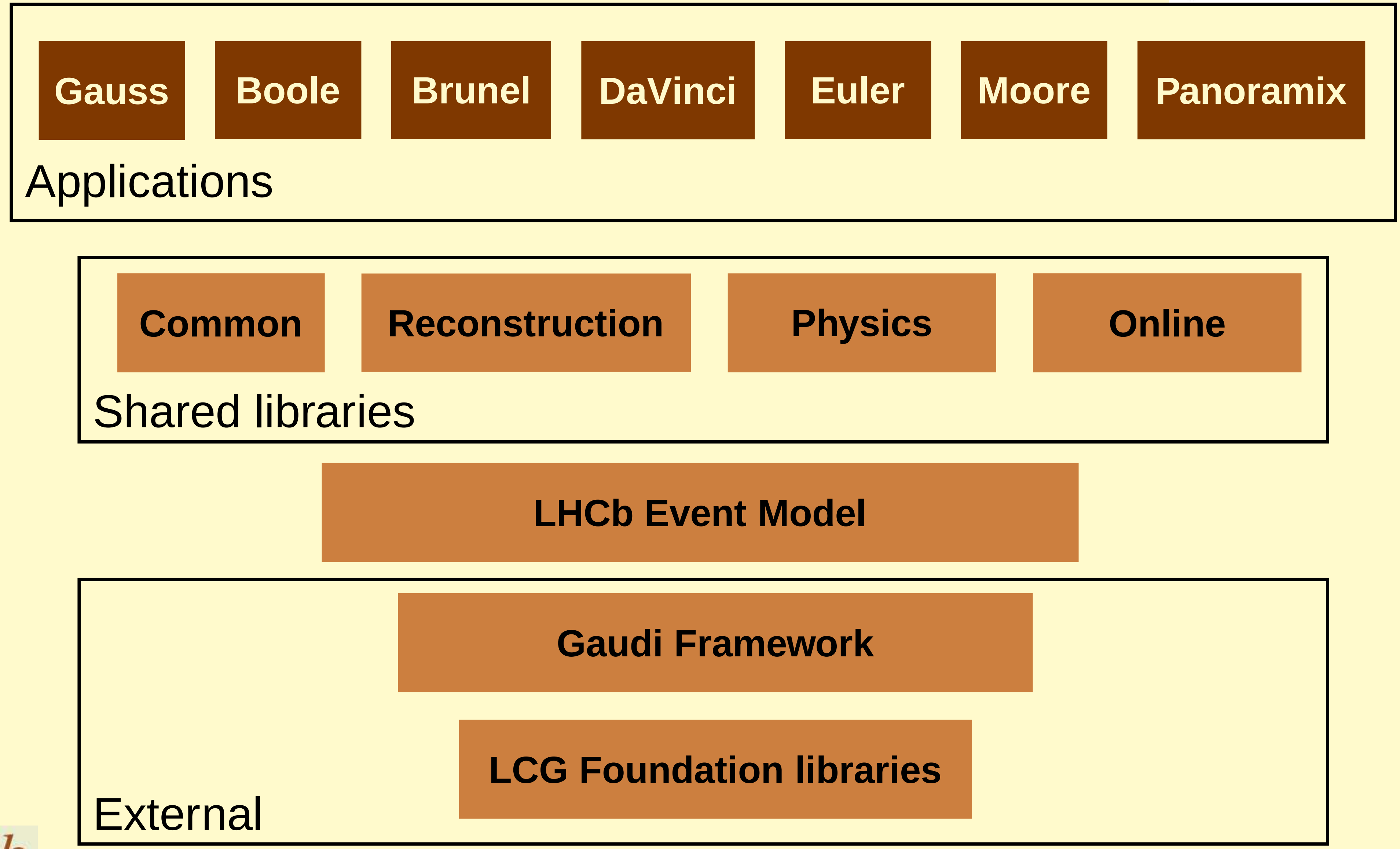
- Same code used:
  1. online in the High-Level-Trigger
  2. offline in the stripping (= “skimming”)
  3. May be used for final selection

  . . . With increasingly hard cuts

→ Maximal correlations of selections

P. Koppenburg

The software pyramid



Gauss Boole Brunel DaVinci Euler Moore Panoramix

Applications

Common Reconstruction Physics Online

Shared libraries

LHCb Event Model

Gaudi Framework

LCG Foundation libraries

External

Applications

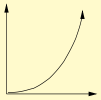
Gauss: Simulation.
Uses generators and **Geant 4**.

Boole: Digitization. Simulates detector response and transforms to “raw” data format.

Brunel: Reconstruction.
Full pattern recognition and PID.

DaVinci: Analysis. Deals with “Particles” and “Vertices”. Final event selection.

Euler: L1 Trigger. For on- and offline use.

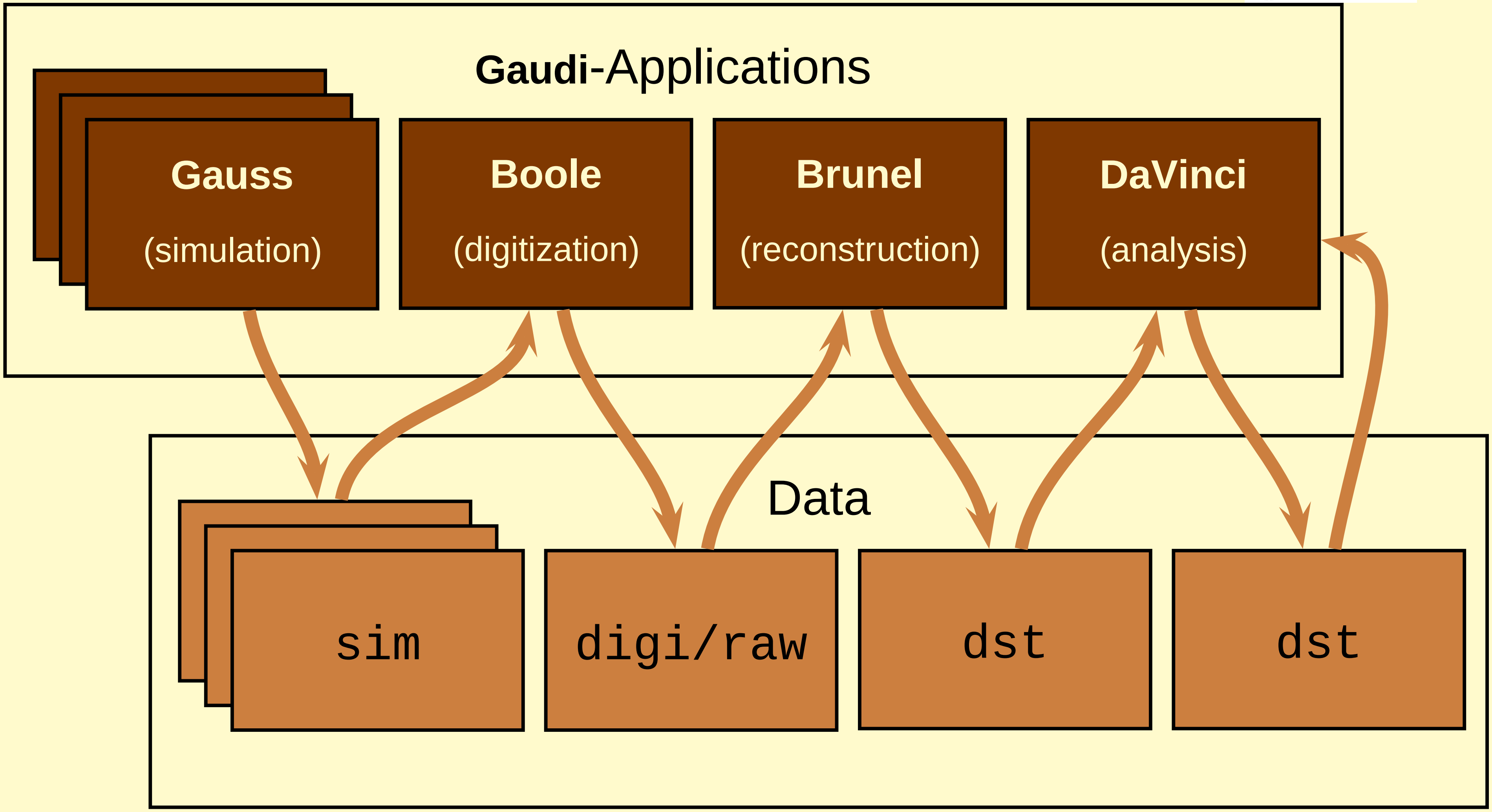


Moore: High-Level Trigger. For on- and offline use.

Panoramix: Visualization.

Bender: Interactive Analysis in `python`.

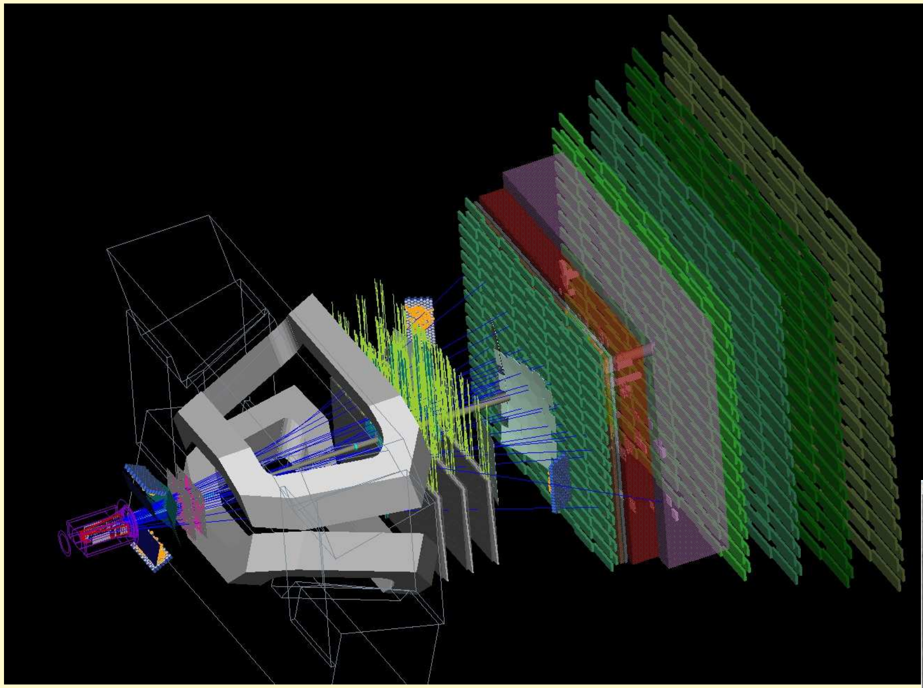
Data Production Flow

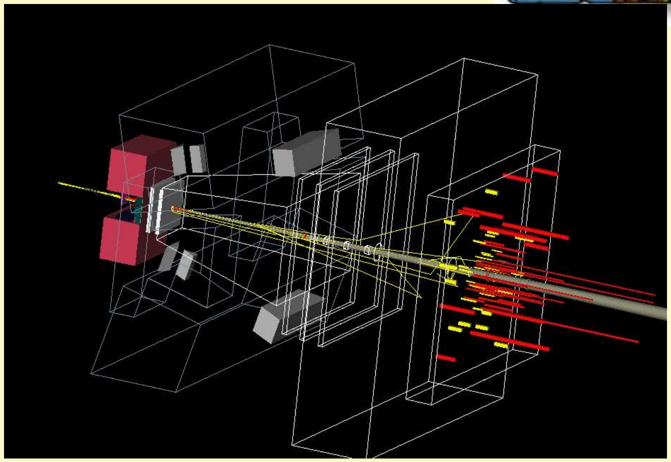


~ 400 million events produced during 2004 data challenge

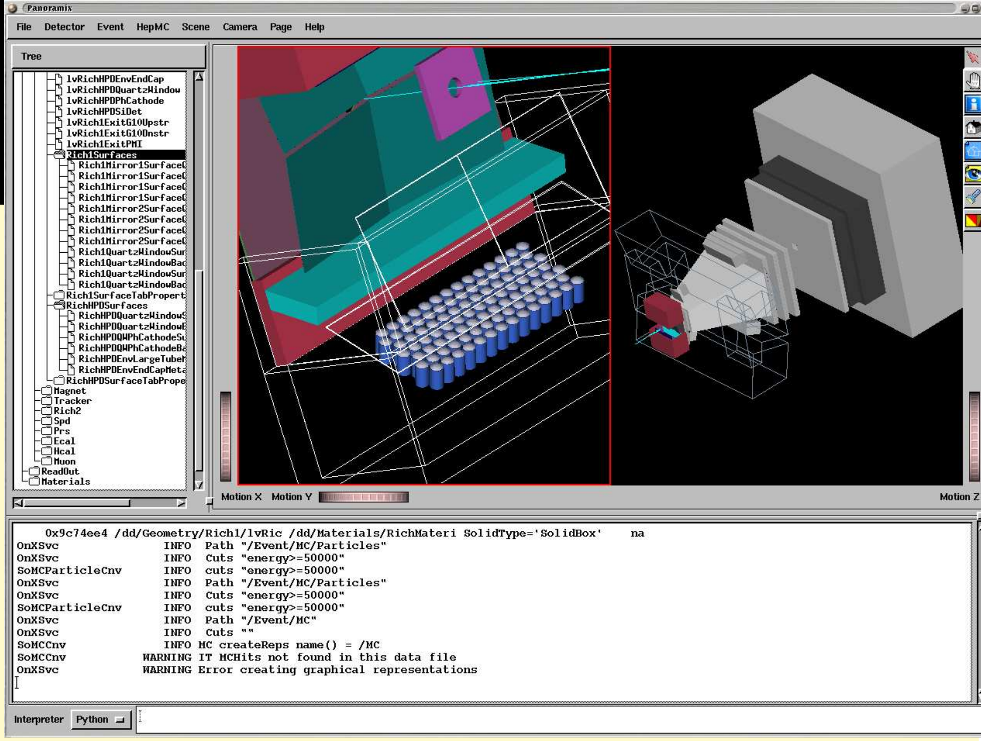
P. Koppenburg

Example: Panoramix Display





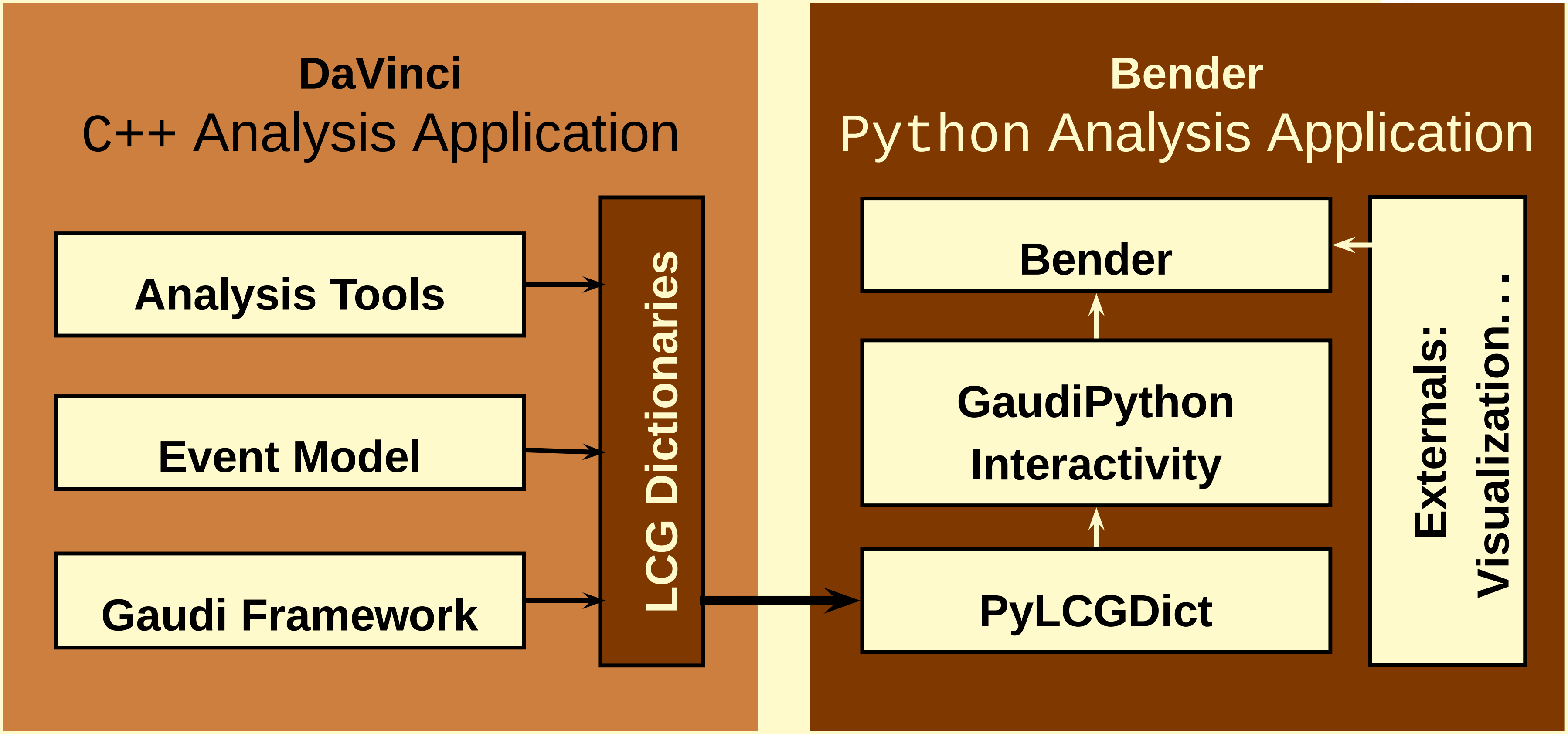
- Visualization application based on **Gaudi**, **OnX** and **OpenInventor**
- Scripting based on `python`
- Allows to (re-)process everything



```
    0x9c74ee4 /dd/Geometry/Rich1/lvRic /dd/Materials/RichMateri SolidType='SolidBox'    na
OnXSvc                 INFO  Path "/Event/MC/Particles"
OnXSvc                 INFO  Cuts "energy>=50000"
SoMCParticleCnv        INFO  cuts "energy>=50000"
OnXSvc                 INFO  Path "/Event/MC/Particles"
OnXSvc                 INFO  Cuts "energy>=50000"
SoMCParticleCnv        INFO  cuts "energy>=50000"
OnXSvc                 INFO  Path "/Event/MC"
OnXSvc                 INFO  Cuts ""
SoMCCnv                INFO MC createReps name() = /MC
SoMCCnv             WARNING IT MCHits not found in this data file
OnXSvc              WARNING Error creating graphical representations
```

P. Koppenburg

Example: Interactive analysis



DaVinci
C++ Analysis Application

Analysis Tools

Event Model

Gaudi Framework

LCG Dictionaries

Bender
Python Analysis Application

Bender

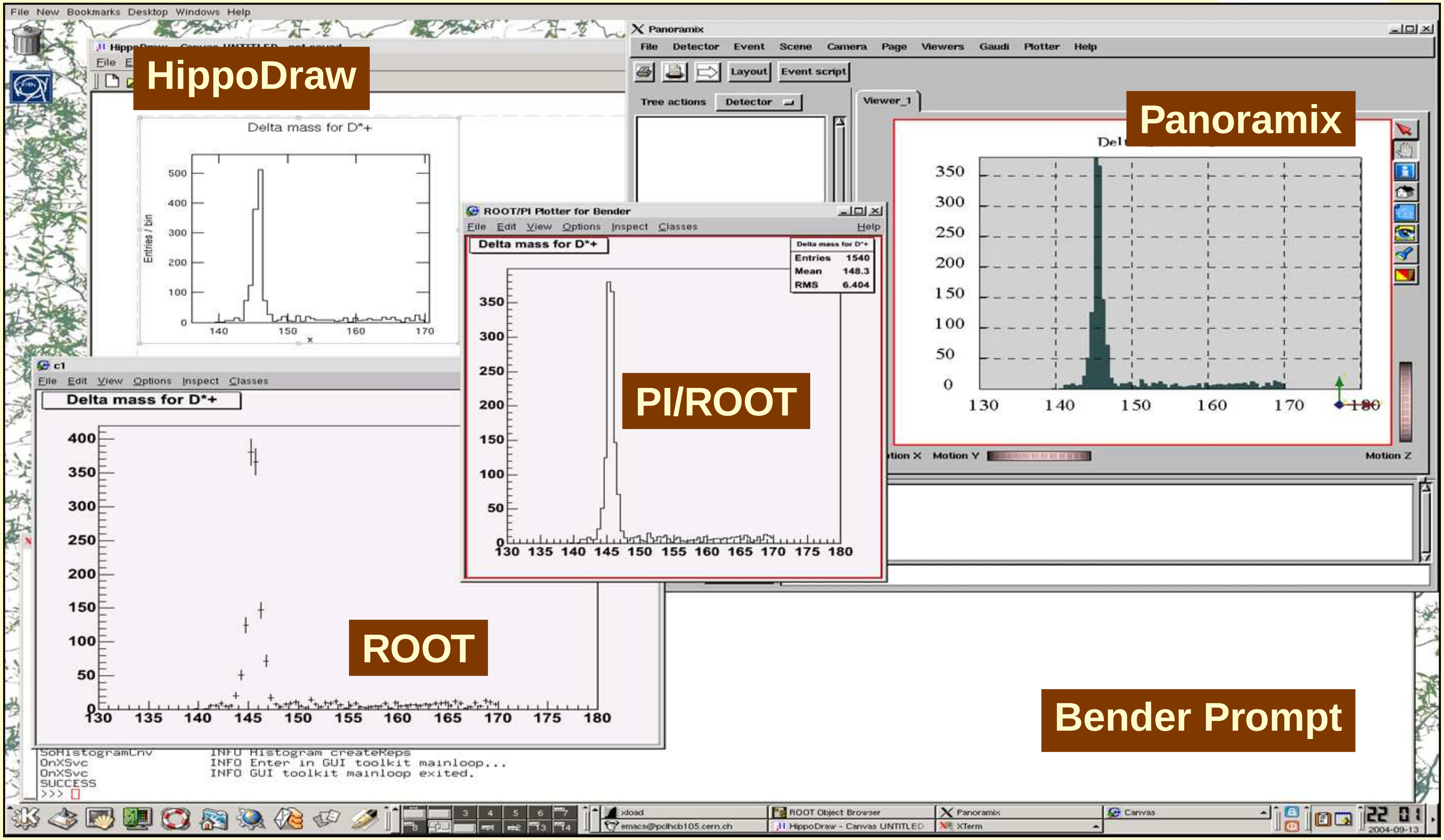
GaudiPython
Interactivity

PyLCGDict

Externals:
Visualization. . .

- **Bender** combines a `python`-wrapping of the **DaVinci** tools, the LHCb Event Model and the **Gaudi** framework
- It allows to perform interactive physics analysis. . .
- . . . and access to many external tools.

Example: Interactive analysis


File New Bookmarks Desktop Windows Help
HippoDraw - Canvas UNTITLED - not saved
Delta mass for D*+
Entries / bin
x
c1
File Edit View Options Inspect Classes
Delta mass for D*+
ROOT/PI Plotter for Bender
File Edit View Options Inspect Classes Help
Delta mass for D*+
Panoramix
File Detector Event Scene Camera Page Viewers Gaudi Plotter Help
Layout
Event script
Tree actions
Detector
Viewer_1
Motion X
Motion Y
Motion Z
xload
emacs@pclhcb105.cern.ch
ROOT Object Browser
HippoDraw - Canvas UNTITLED
Panoramix
XTerm
Canvas
2004-09-13

Delta mass for D*+
Entries 1540
Mean 148.3
RMS 6.404

```
SoHistogramCnv          INFO Histogram createReps
OnXSvc                  INFO Enter in GUI toolkit mainloop...
OnXSvc                  INFO GUI toolkit mainloop exited.
SUCCESS
>>> 
```

HippoDraw

Panoramix

PI/ROOT

ROOT

Bender Prompt

Conclusion

- A well defined structure for 500 users
- One framework for 8 applications:
  - Data production
  - Analysis
  - Trigger
  - Interactivity
- Facilitates migration of algorithms between applications
- Encourages optimal usage of the code
- Helps to minimize inefficiencies
- Allows analysis in `C++` or `python`