GAUDI Architecture

SW Architecture SG meeting

22 July 1999

P. Mato, CERN

GAUDI: Scope

- We want to develop a Framework to be used in ALL the **LHCb event data processing applications** in all stages: high level trigger, simulation, reconstruction, analysis.
- Control applications are not included (slow control, run control)
- The bulk of the LHCb data processing software will be developed by physicists.
- Components will be developed using other frameworks (GUI, object persistency, simulation, …)

GAUDI: Goals

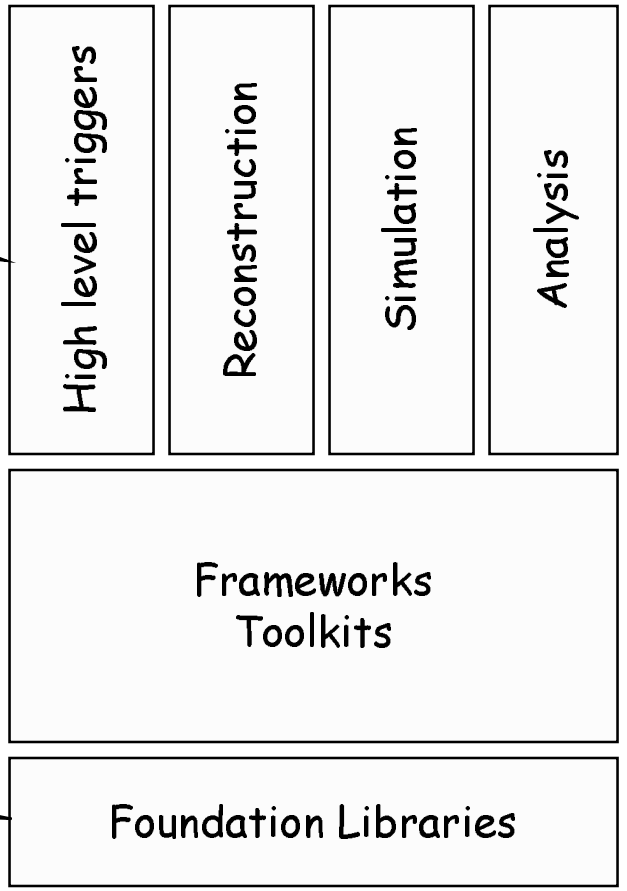
- To develop a framework
- The framework should be customizable to provide solutions to the event data processing needs.
- It should:
  - Allow physicists to focus on solving the physics problem.
    - » Easy to use
    - » Non physics-related functionality implemented by the framework
  - Ensure low coupling between concurrent developments.
  - Guarantee a later smooth integration of developments.
  - Facilitate software re-use

Software Structure

A series of data processing applications built on top of the frameworks and implementing the required physics algorithms.

A series of Frameworks and Toolkits. One main framework: **GAUDI**, various specialized frameworks: visualization, persistency, interactivity, simulation (Geant4), etc.

A series of basic libraries widely used: STL, CLHEP, etc.



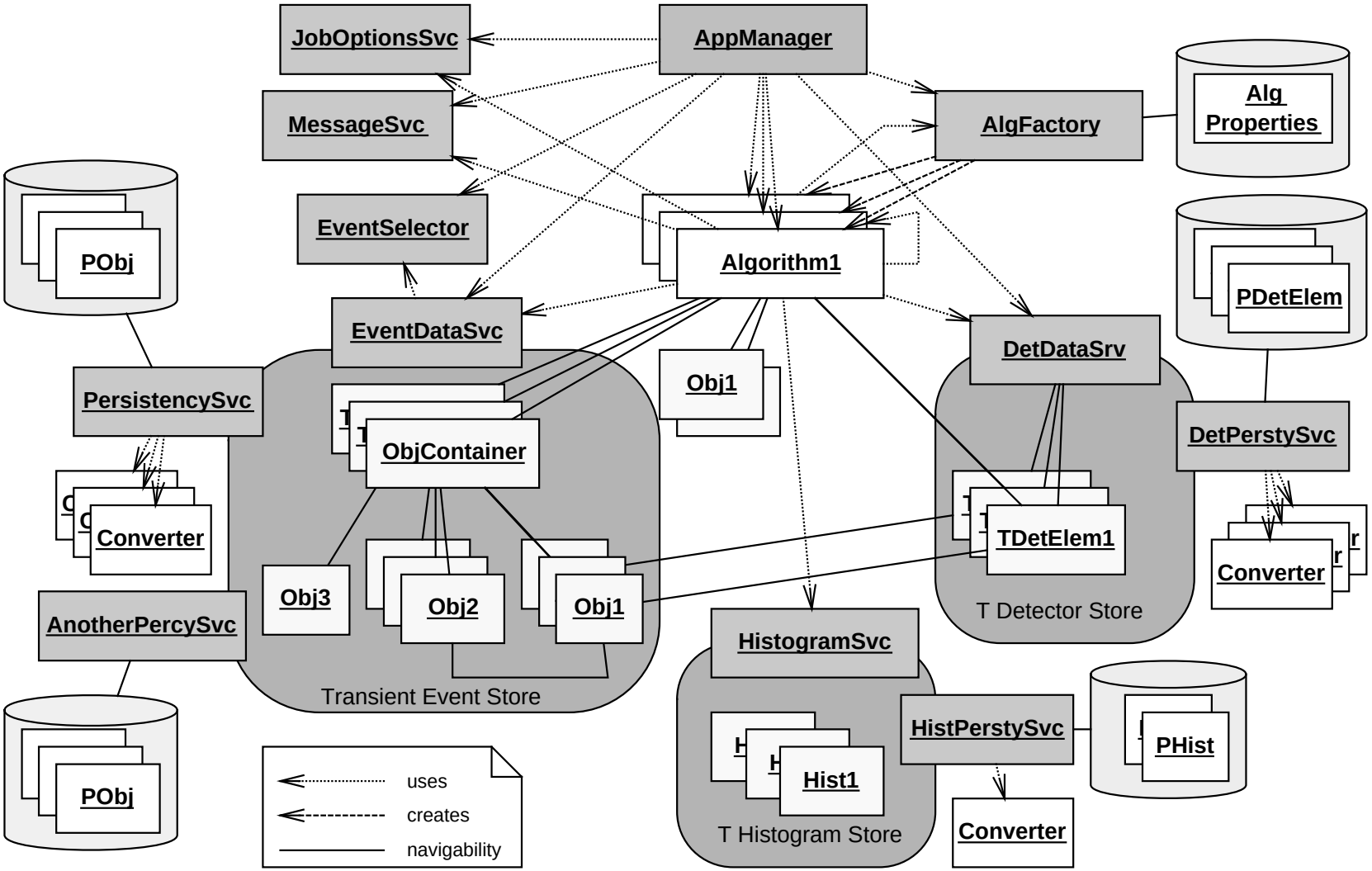
Major design criteria

- Clear separation between “data” and “algorithms”
- Three basic types of data:
  - **event data** (data obtained from the particle collisions)
  - **detector data** (structure, geometry, calibration, alignment, environmental parameters,..)
  - **statistical data**: (histograms, …)
- Clear separation between “persistent data” and “transient data”.
  - Isolation of user’s code.
  - Different/incompatible optimization criteria.
  - Transient as a bridge between various representations.

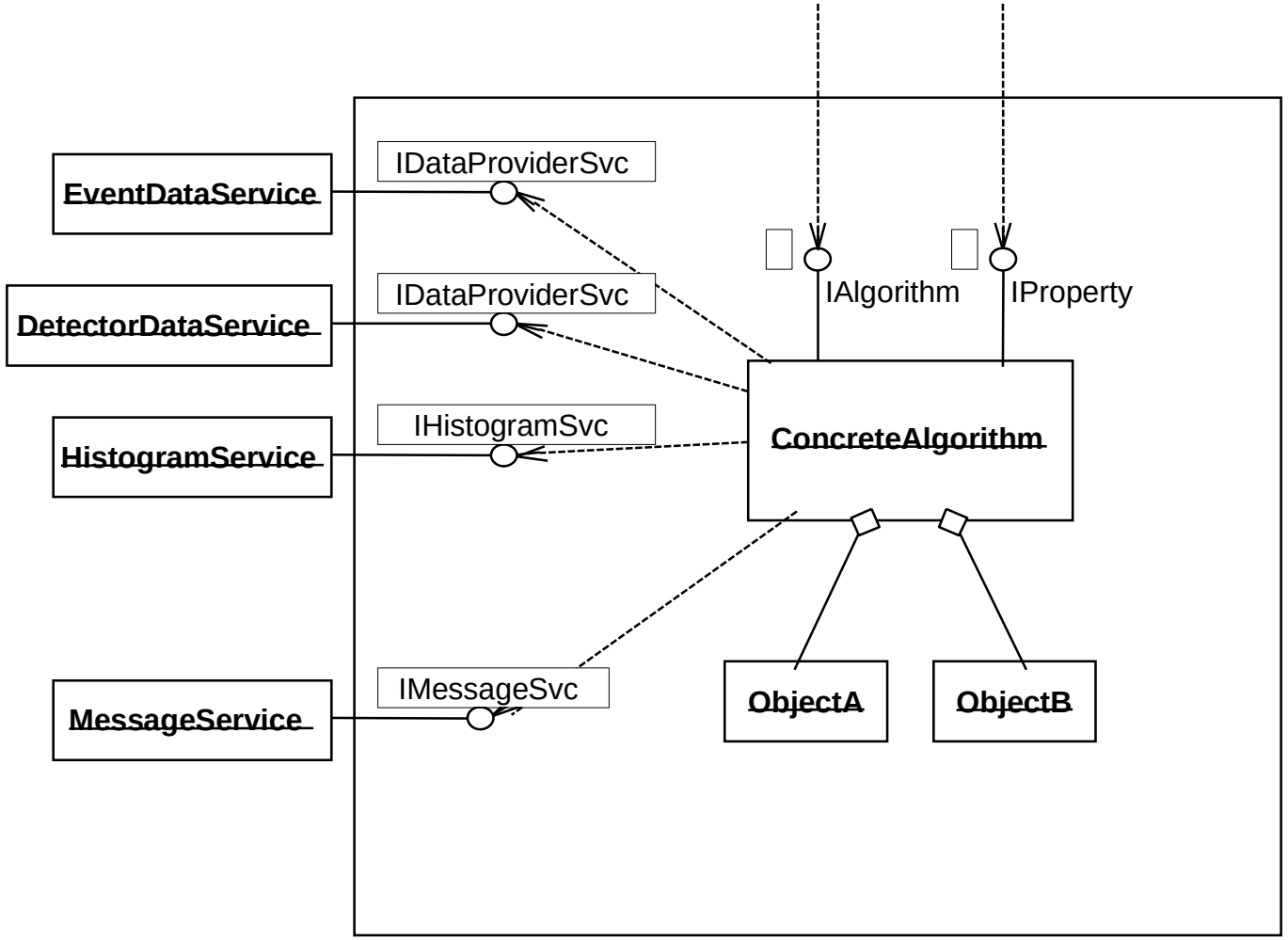
Major design criteria (2)

- Data store centered architectural style.
  - Algorithms as data producers and consumers.
- *User code* encapsulated in few specific places:
  - "Algorithms": Physics code
  - "Converters": Converting data objects into other representations
- All components with well defined "interfaces" and as "generic" as possible.
- Design principles
  - Coupling, inheritance, static storage, ...

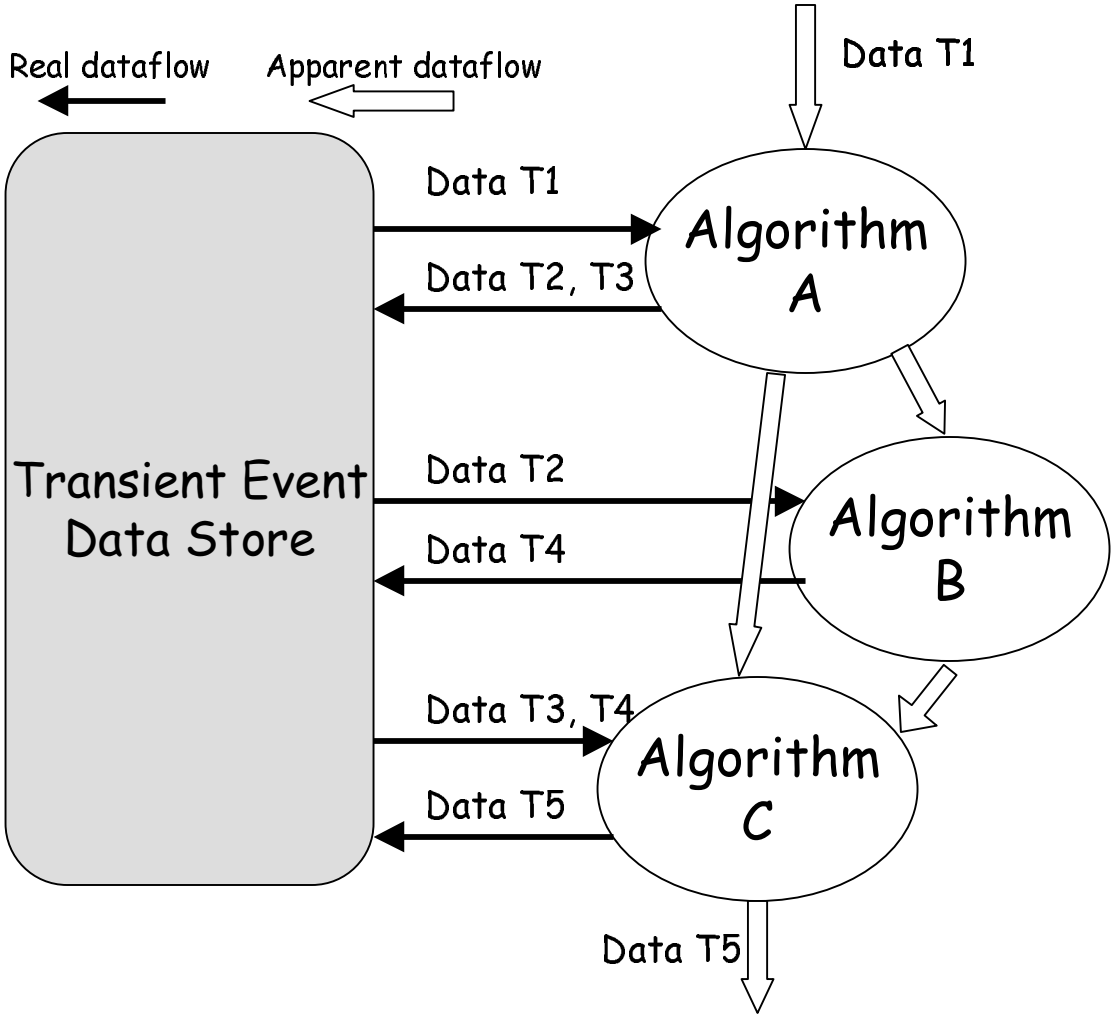
Building blocks



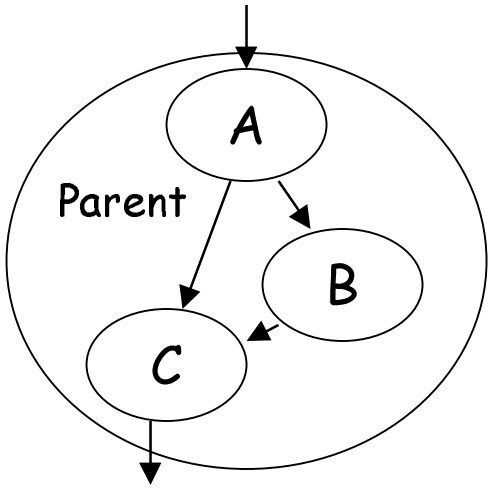
Interfaces


EventDataService
DetectorDataService
HistogramService
MessageService
IDataProviderSvc
IDataProviderSvc
IHistogramSvc
IMessageSvc
IAlgorithm
IProperty
ConcreteAlgorithm
ObjectA
ObjectB

Algorithms

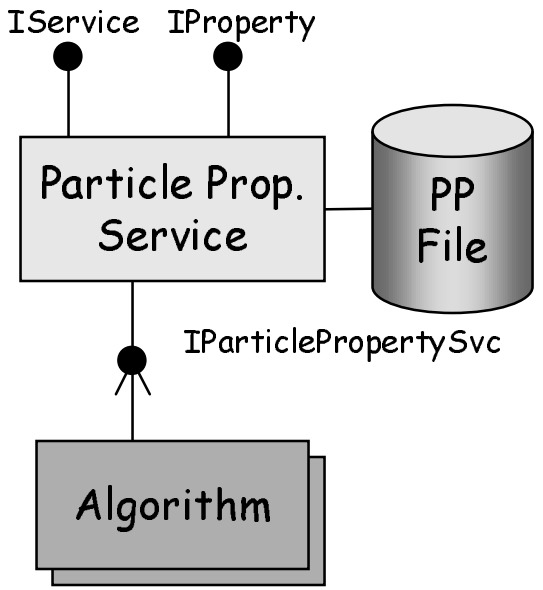


- Each Algorithm only knows what data (type and name) is expecting as input and creating as output.
- The only coupling is through the data.
- **Scheduling of sub-algorithms** is responsibility of the parent algorithm.

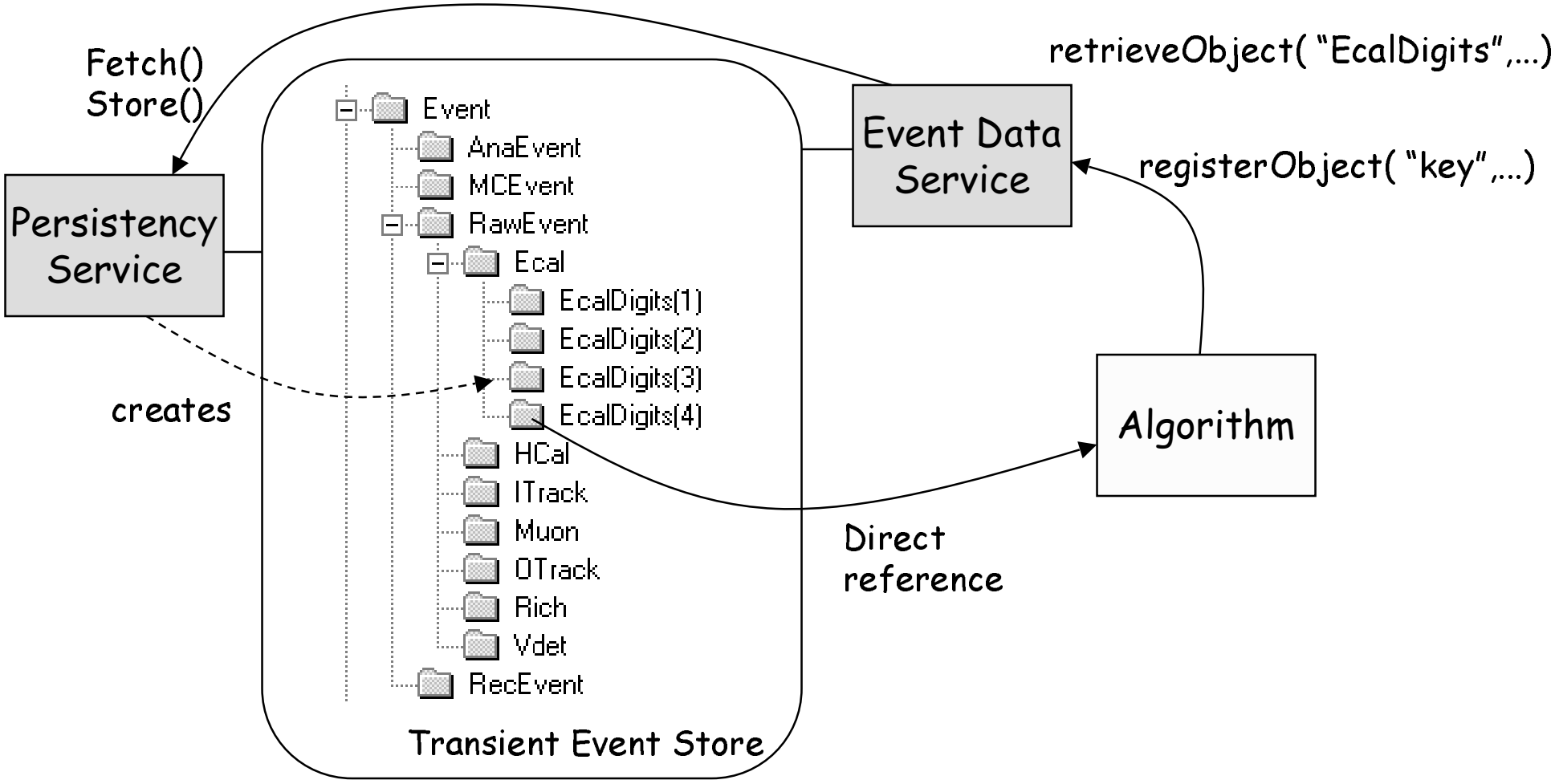


Services

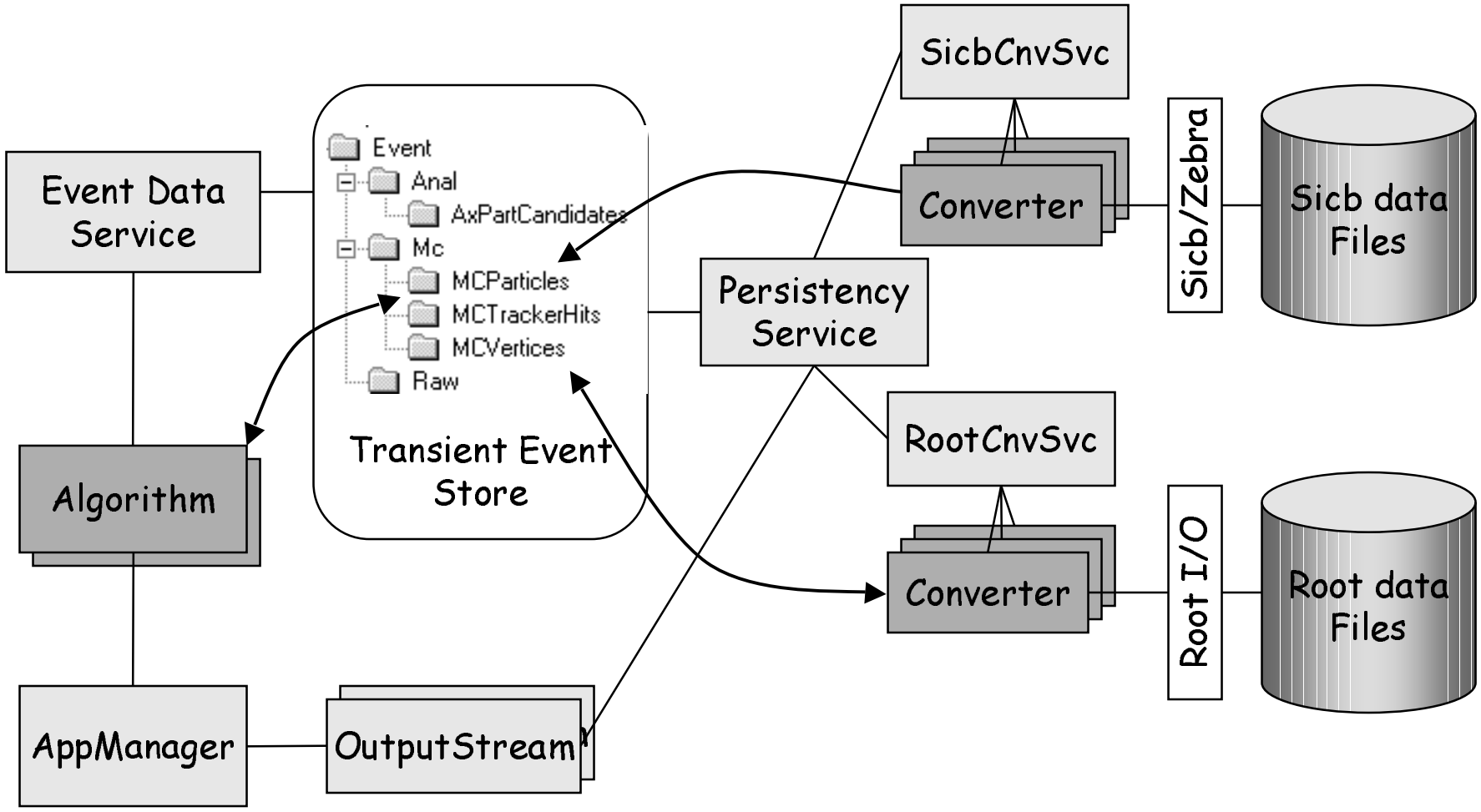
- Services are provided to *Algorithms*
- Examples:
  - Job Options service (card files)
  - Message reporting service
  - Event/Detector/Histogram data service
  - Event Selector
  - Persistency and Conversion services
  - User Interface (GUI)
  - Particle property service
  - ...



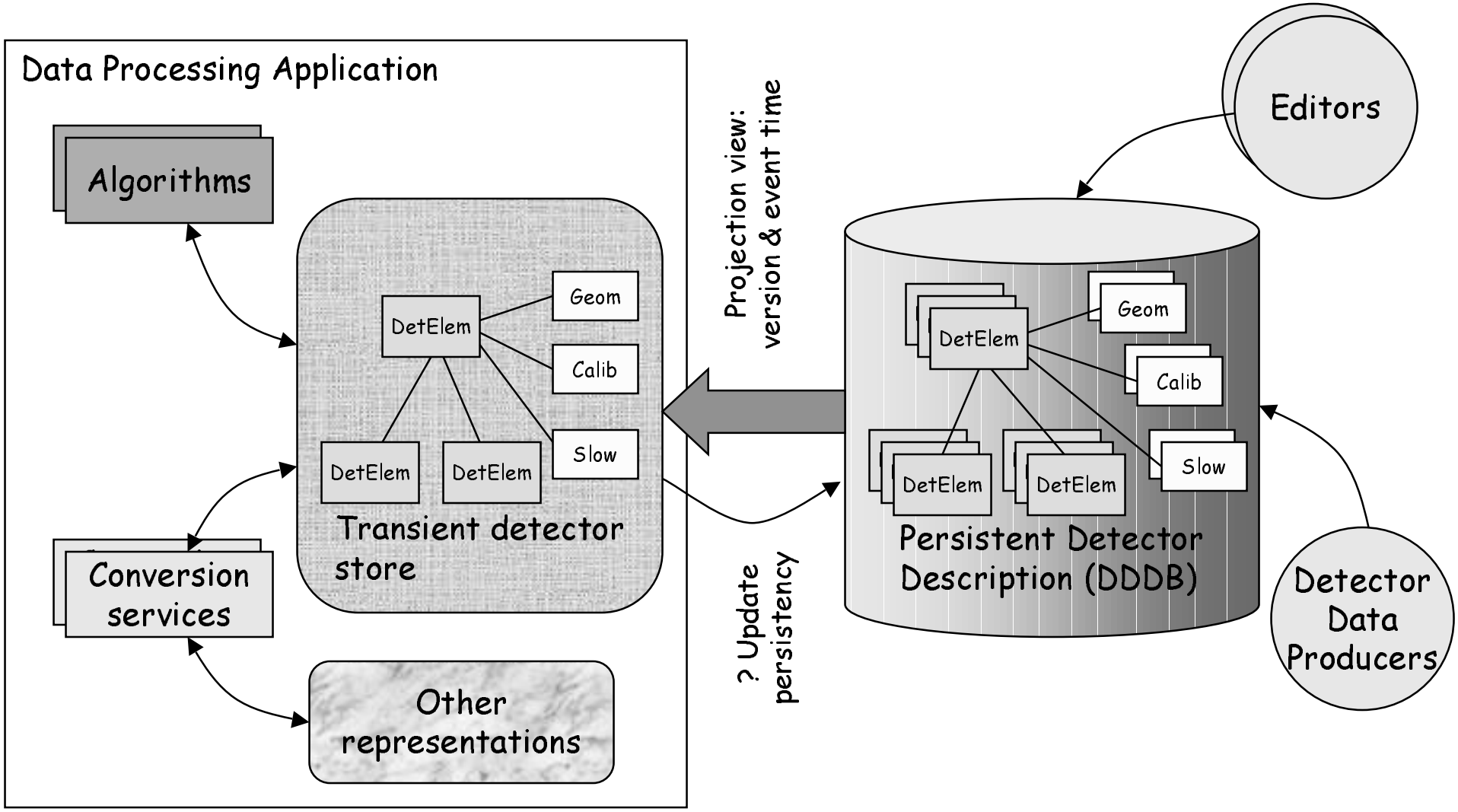
Event Data Store



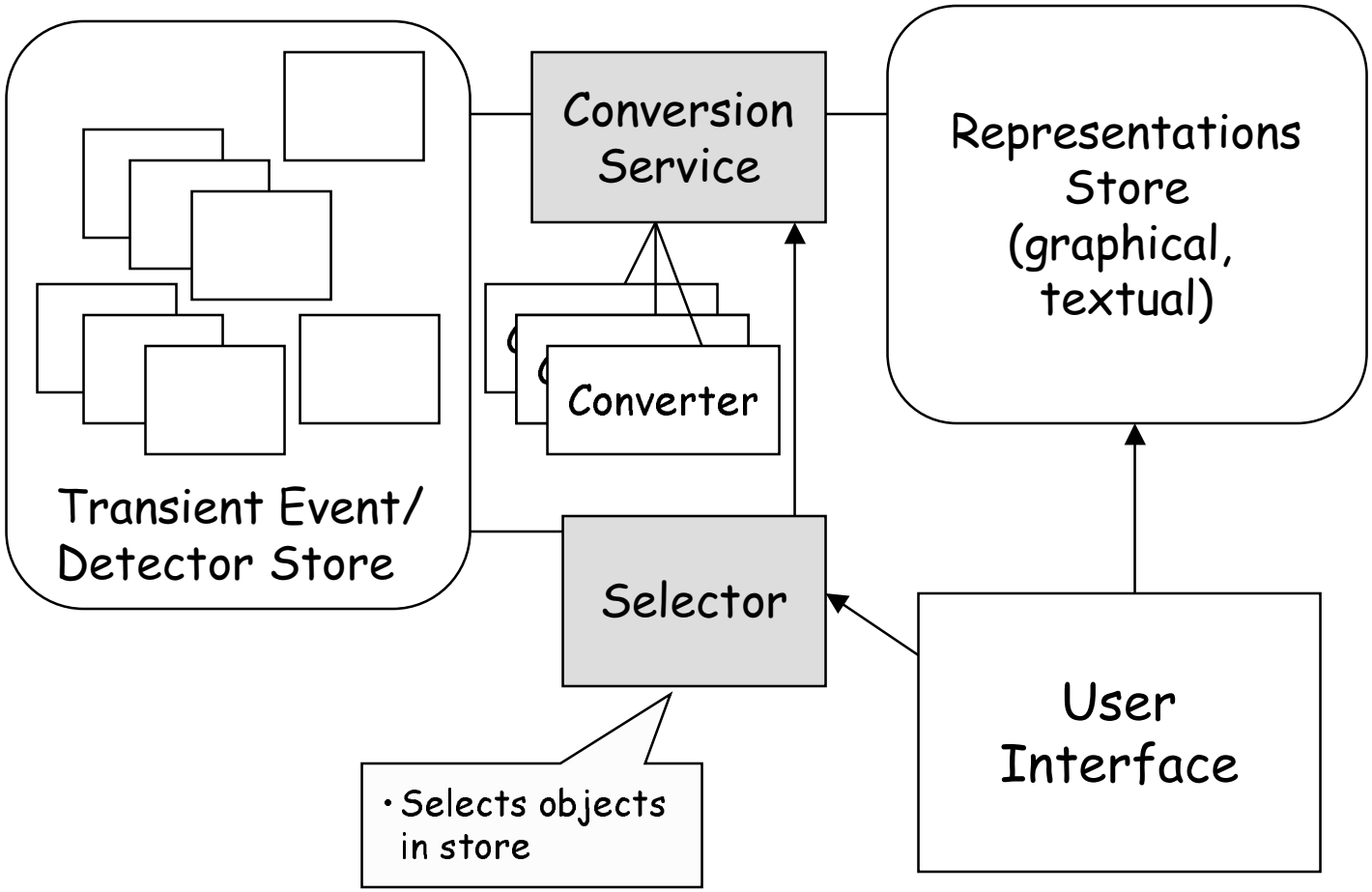
Event Persistency



Detector Description



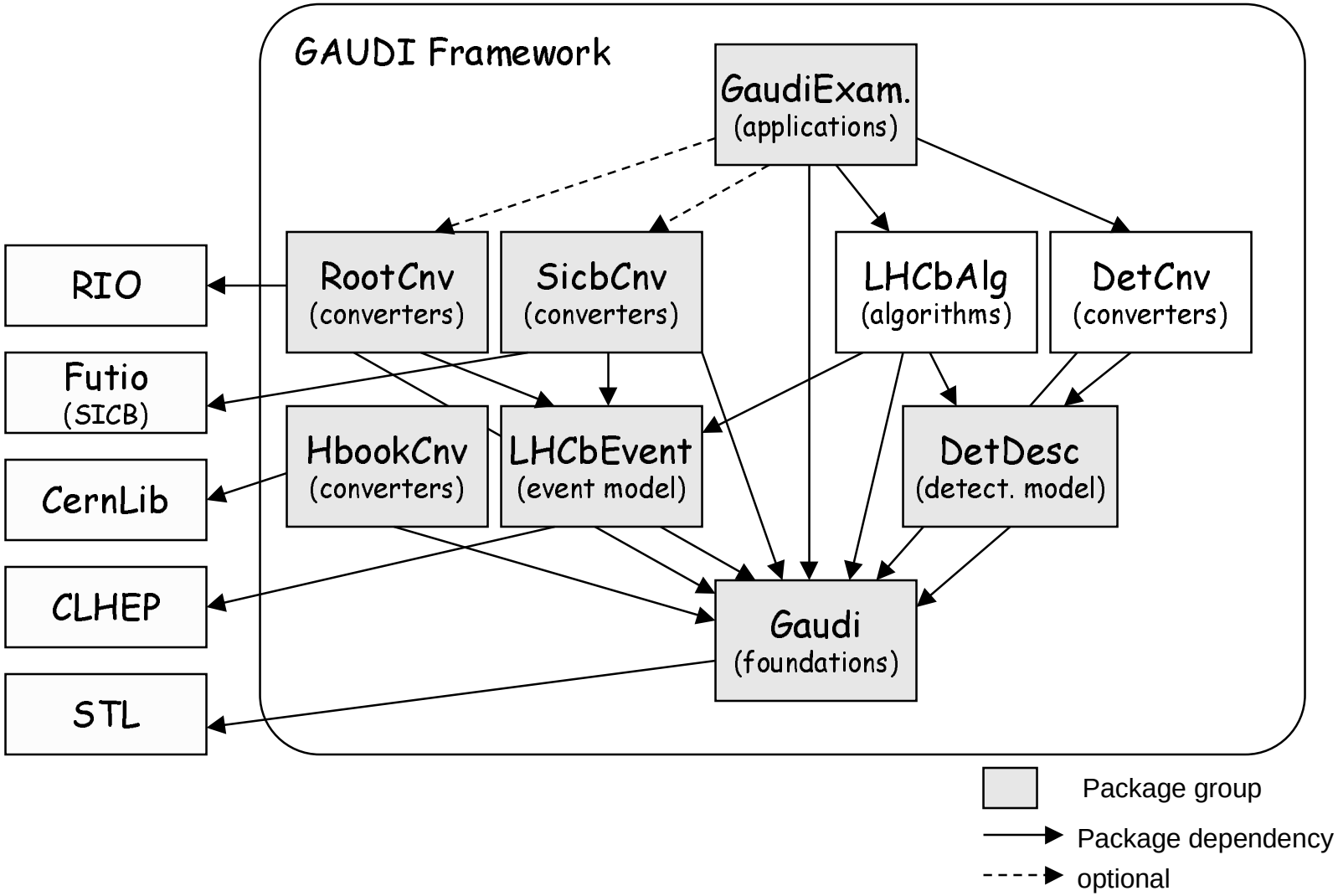
Data Visualization



Application Configuration

- What are the knobs at our disposal?
  - **JobOptions.** Simple usage. It allows the end-user to overwrite any property of any algorithm or service.
  - **Algorithm/Service properties database.** A more sophisticated way to modify the properties of the algorithms and services.
  - **Detector database edition** to create new versions or releases.
  - **Write specific code**. Configure your application by setting it at runtime.
  - **User interface component**. Graphical (a la Visual Basic), command line (scripting language), etc.

Packages


GAUDI Framework
GaudiExam.
(applications)
RIO
Futio
(SICB)
CernLib
CLHEP
STL
RootCnv
(converters)
SicbCnv
(converters)
LHCbAlg
(algorithms)
DetCnv
(converters)
HbookCnv
(converters)
LHCbEvent
(event model)
DetDesc
(detect. model)
Gaudi
(foundations)
Package group
Package dependency
optional