SEAL and POOL in LHCb

Strategy

Integration of SEAL

Integration of POOL and Status

M. Frank

Applications Area Meeting, October 8th, 2003

Motivation

- **LHCb has invested significantly in LCG**
- **It's payback time!**
- **Retire parts of the Gaudi framework**
  - Decrease long term maintenance load
- **Test SEAL and POOL in the LHCb environment**
  - Give feedback to SEAL and POOL

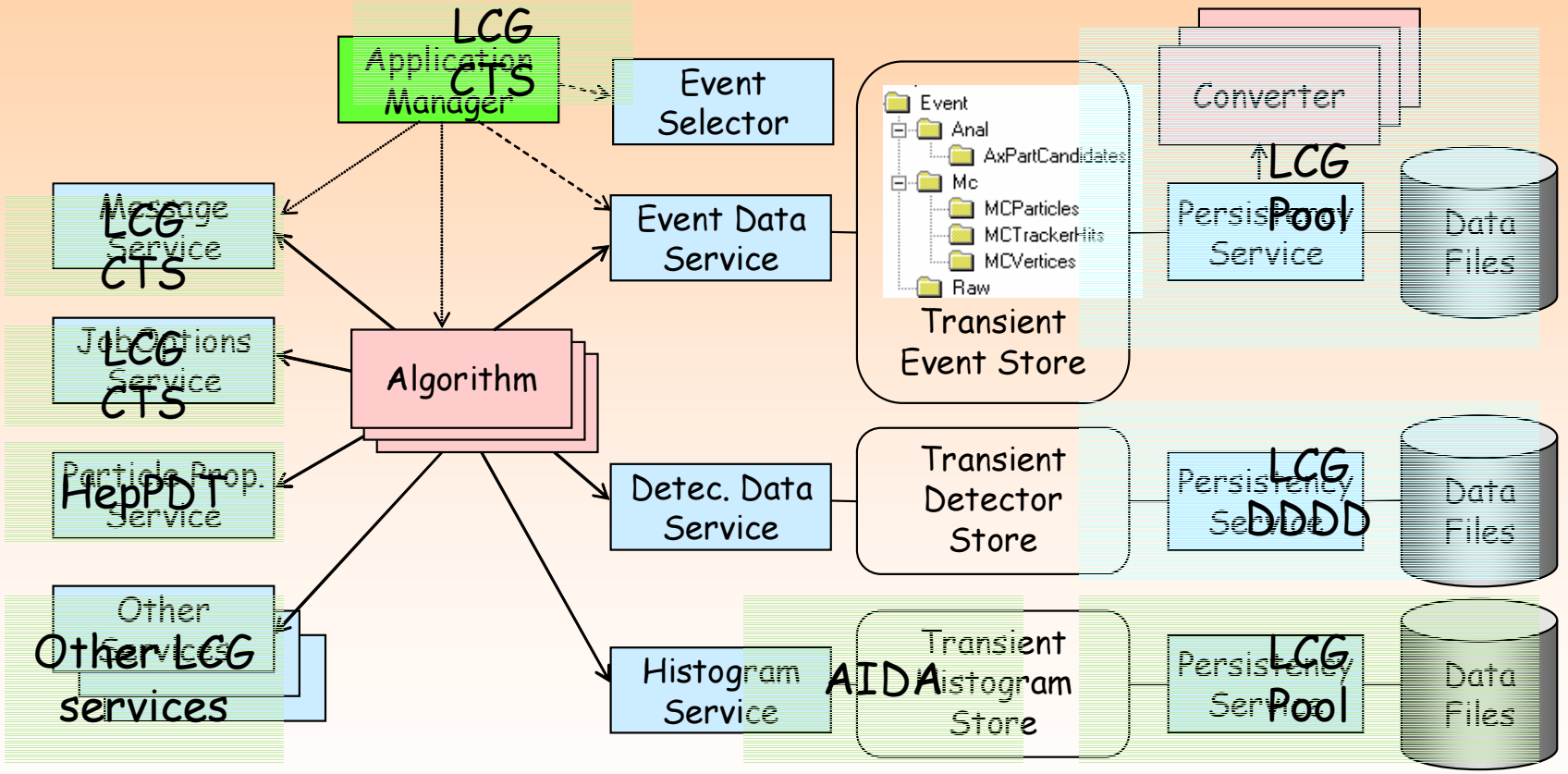
Strategy

- **Adiabatic adaptation of Gaudi to SEAL/POOL**
  - Slow integration according to available manpower
  - Time estimate roughly 1 year for full migration
  - Take advantage for face-lifting of “bad” interfaces and implementations
- **Minimal change to interfaces visible to physicists**
- **Integration of SEAL in steps**
  - Dictionary integration and plugin manager
  - Use of SEAL services later

Other Constraints

- **Integration of POOL earlier than SEAL**
  - Plan to have a fully working prototype by end of the year
- **Necessity to read "old" ROOT data**
  - For roughly 1 year
  - Consider data reformatting from Gaudi ROOT to POOL
- **Keep the LHCb event model as is**
  - Event model classes/objects are transient
  - **May, but not must** have a 1:1 persistent correspondence

Standard View of LHCb Software



Seal Integration Status

- Dictionary
- Plugin Manager
- Component model

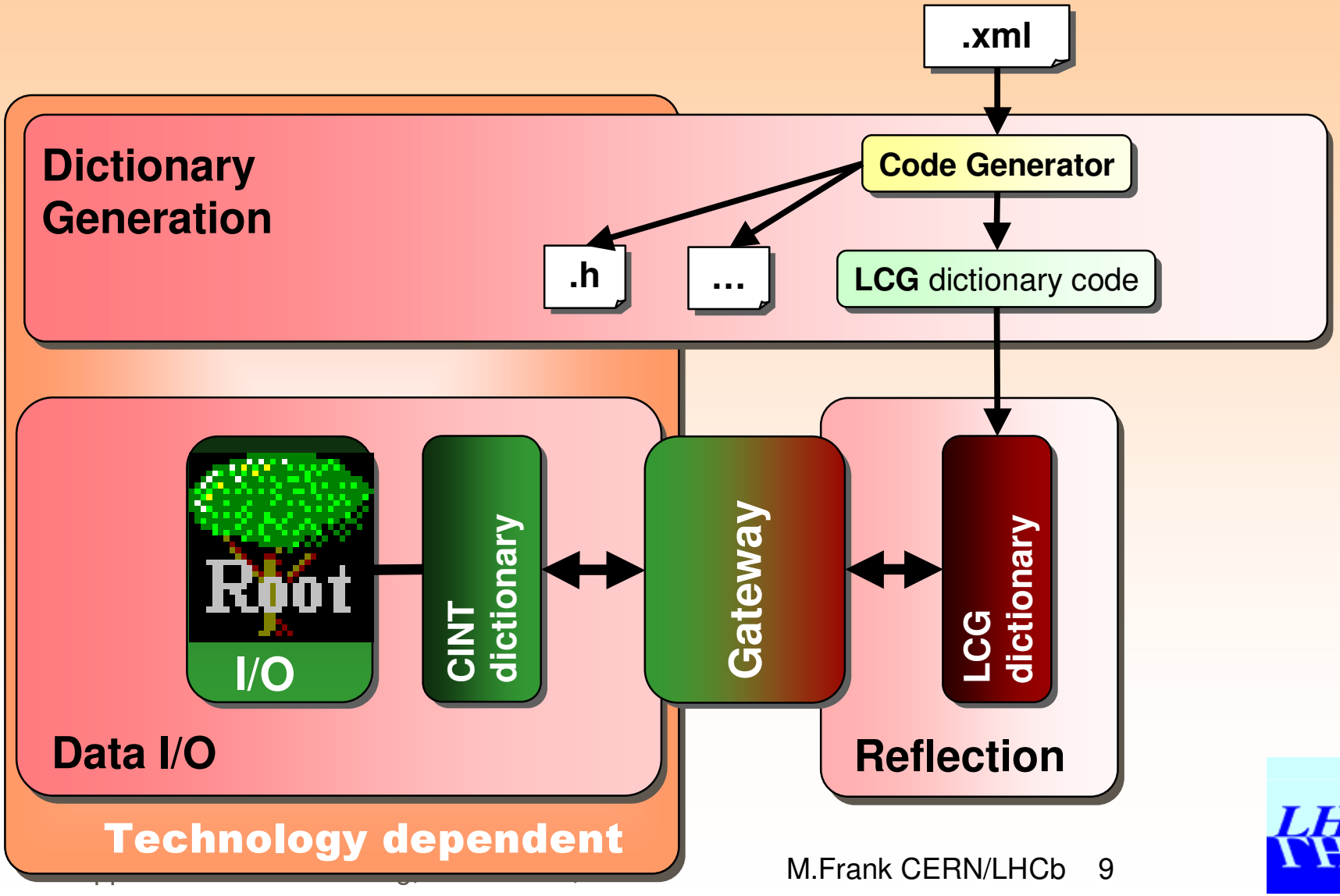
Seal Dictionary

- **LHCb has it's own approach**
  - Generate code from object model described with XML
- **SEAL dictionary is mandatory for POOL integration prototype**
- **Code generator is missing**
  - Work will start [very] soon

Dictionary: Population/Conversion

- **Event model is described in XML**
- **Generator delivers**
  - LCG dictionary
  - C++ Header files
  - …Whatever the future brings
  - Limited need GCCXML: external libraries like CLHEP
- **Much simpler/shorter description**
- **Homogeneous header files with one style**
- **Not all is good what C++ offers**

Dictionary: Population/Conversion



Seal Component Model

- **We will not jump immediately on what is present**
  - SEAL is evolving and we cannot constantly follow
- **SEAL is used through POOL**
  - Well encapsulated
  - Limited visibility to end users
- **No replacement of Gaudi services foreseen before component model is not mature**

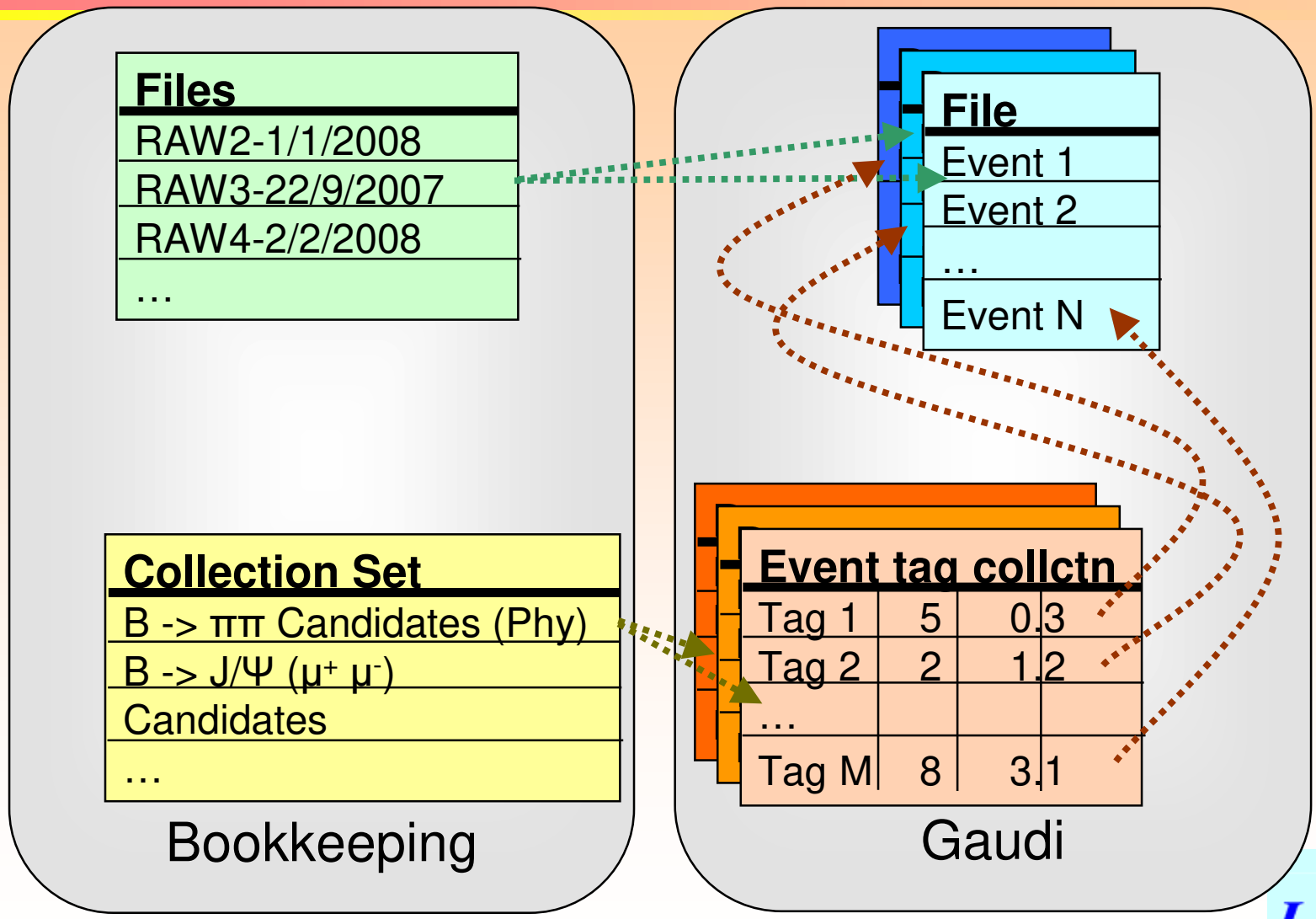
Seal Foundation Libraries

- **Concerns regarding modularity**
  - SEAL foundation libraries depend on many external libraries
  - Example:
    To use the plugin manager the regular expression library (libpcre) must be loaded, which is needed by the SEAL regular expression wrapper
    e.g. also boost has a regular expression wrapper

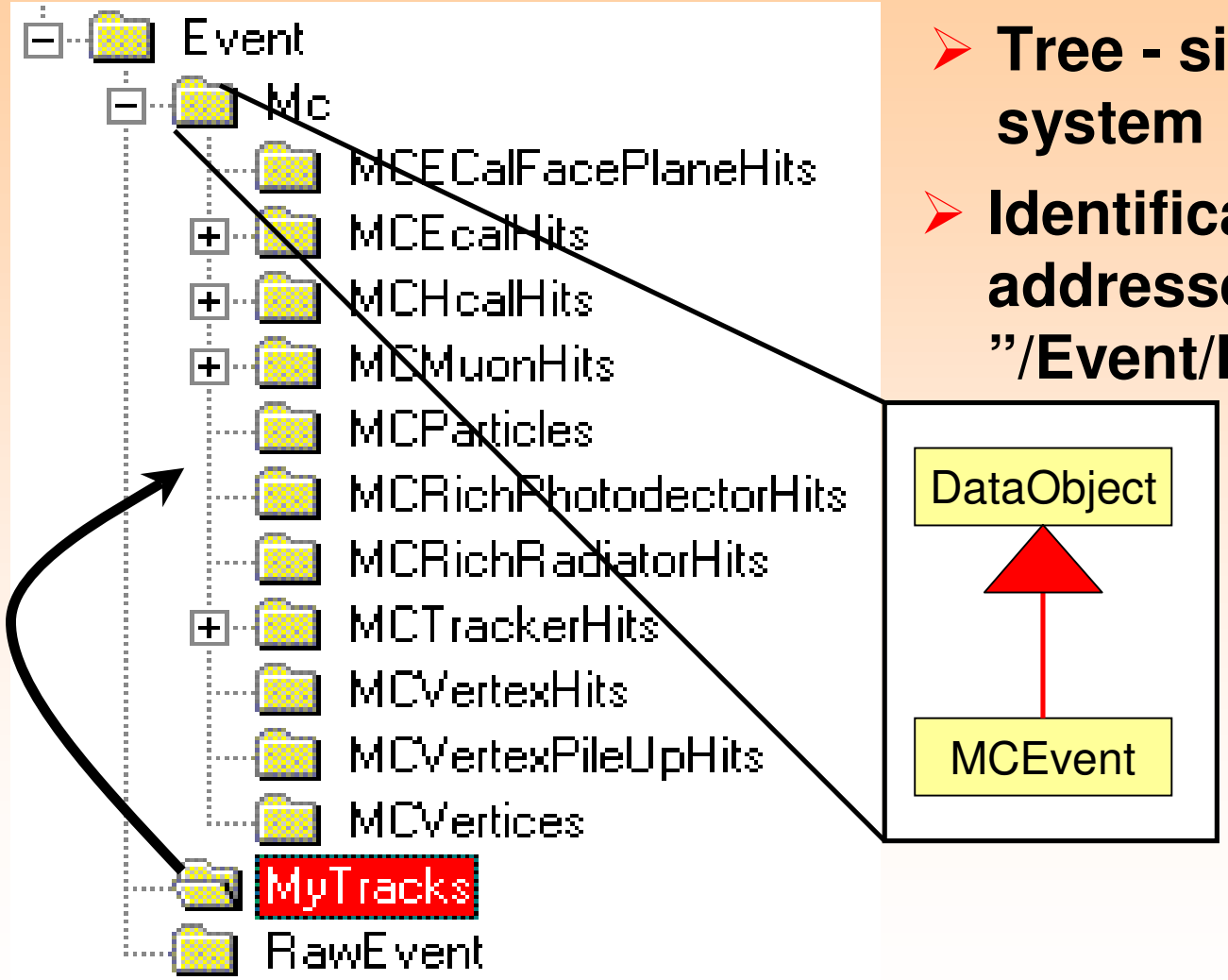
POOL Integration: Goals

- **Save and read the LHCb event model**
- **No more serializers, let ROOT do the job**
  - Object can keep personality in ROOT
- **Emphasis on some aspects of the LHCb event model**
  - Stress on object relationships
- **Dictionary generation for the LHCb event model**
- **Event tag collections: POOL & Gaudi**
- **Integration of POOL event tag collections**

Event Data Access

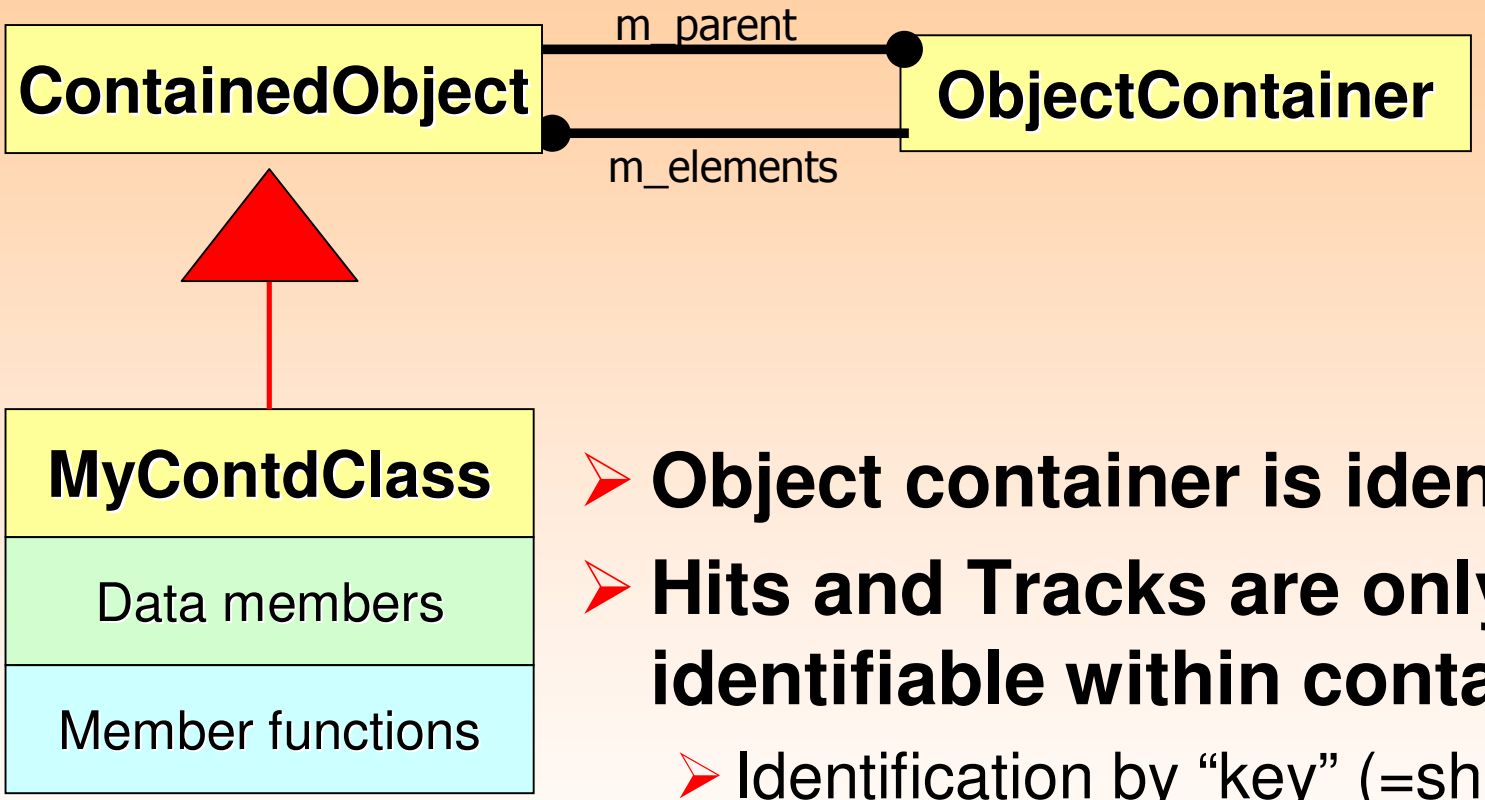


Events in the Gaudi Data Store



- Tree - similar to file system
- Identification by logical addresses: ”/Event/MC/MCEcalHits”
- Store item= directory + attributes
- Browse capability

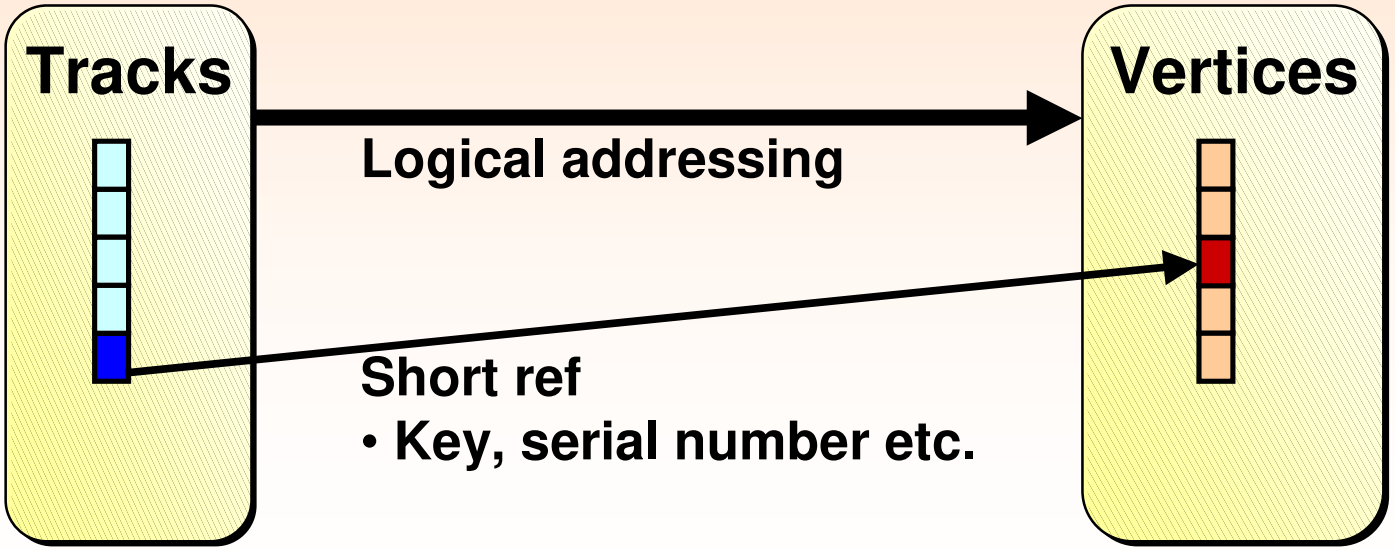
Object Collections (Hits, Tracks etc.)



- **Object container is identifiable**
- **Hits and Tracks are only identifiable within container**
  - Identification by “key” (=short ref)

References between Contained Objs

- Most objects are aggregated in containers
- Identify containers using “logical addressing”
- Internal links between contained objects (Short refs)

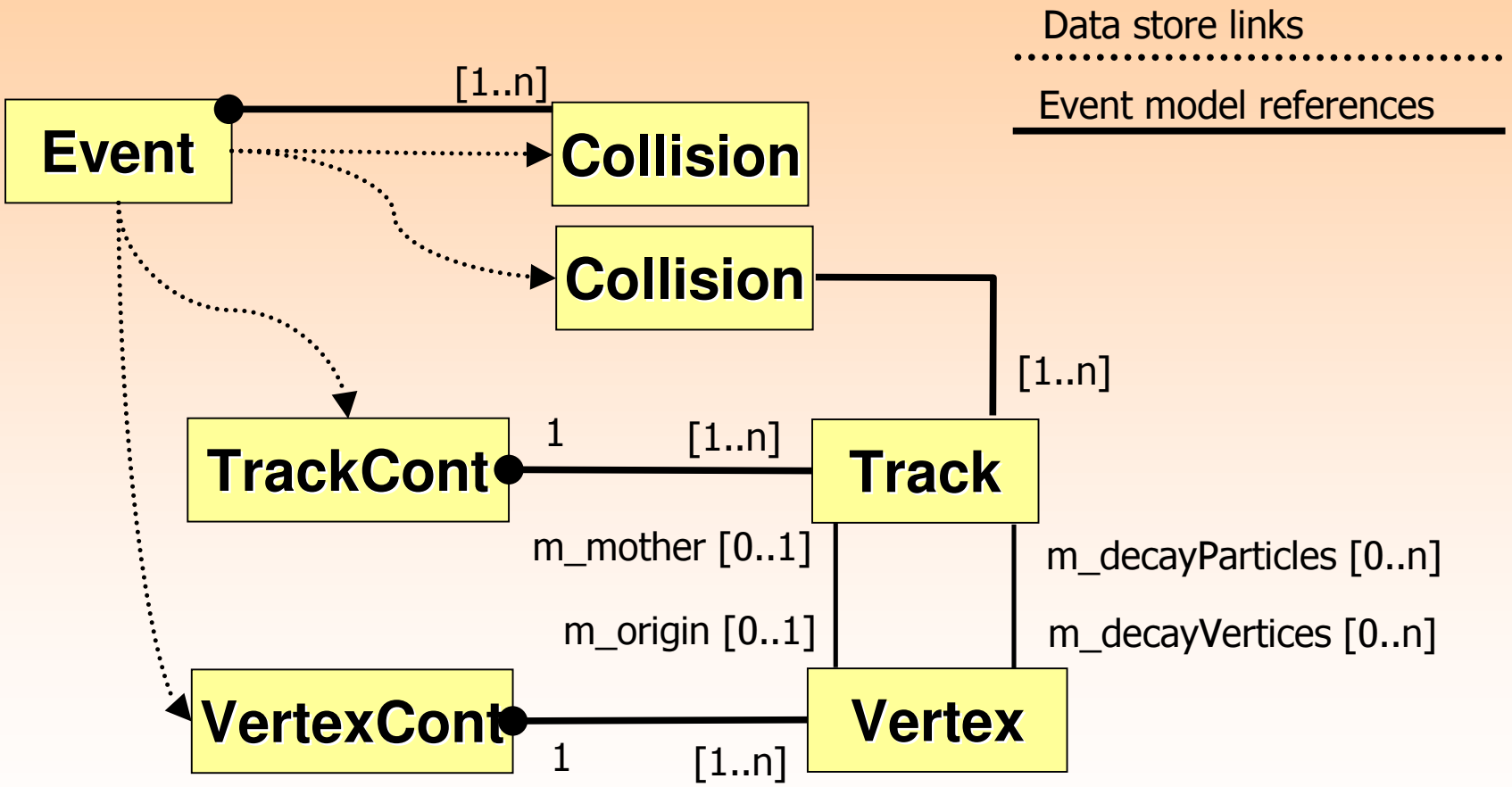


Toy Event Model

- Identifiable objects (DataObject)
- Non-identifiable objects (ContainedObject) hosted by “Keyed Containers”
- 0..1 Associations
- 0..n Associations

- All combinations of these relationships must work

Toy Event Model



Status of Event Data Storage

- **Mechanics for data persistency of event data is partially implemented**
  - New Gaudi Plugin created: package GaudiPoolDb
  - New Gaudi “conversion service” capable of dealing with POOL technologies: Root Tree
  - New EventSelector service to access implicit POOL collections for reading files
  - One converter class for all object types

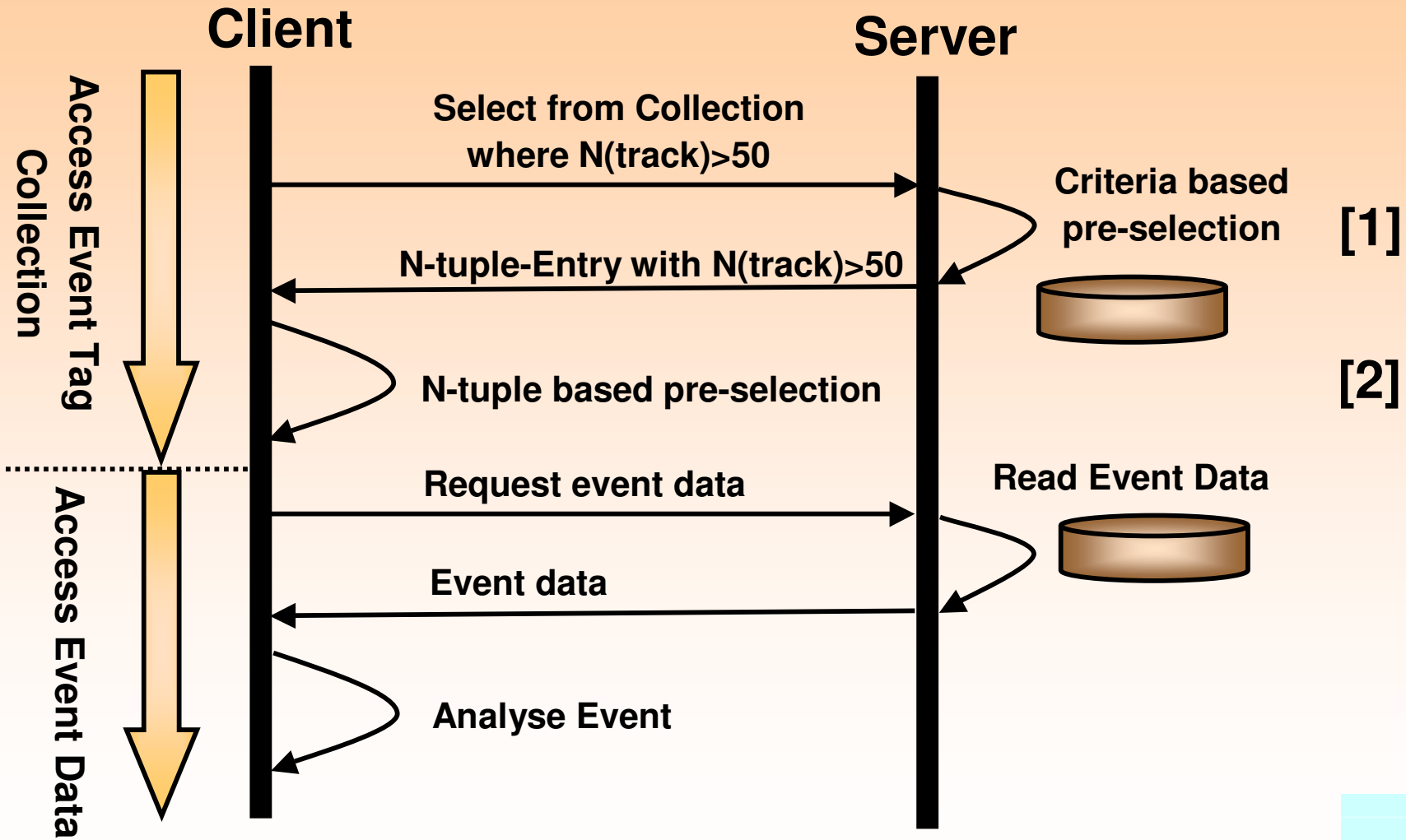
Status of Event Data Storage

- **Dictionary generation is missing**
  - Use generated dictionaries for the time being
- **MC truth relationships still missing**
  - Remove MC truth information completely from data
  - Objects containing arrays of reference pairs

Event Tag Collections: LHCb view

- **N-tuples, but with references to external objects**
- **Data content as for column wise N-tuples**
  - Scalars, arrays, matrices
- **Writing is equivalent as for N-tuples**
  - Simple, intuitive, fast
  - Interface identical to N-tuples
  - The physicists, which used them are enthusiastic

Data Access For Event Processing



Gaudi Event Tag Collections

- **Note the subtle difference:**
  - Gaudi event tag collections using POOL
  - POOL event tag collections in Gaudi
- **Implemented Gaudi Event Tag Collections**
  - Backwards compatibility with existing implementation
  - Use POOL as a mechanism to populate Gaudi N-tuple structures with data
  - Usage of lower level POOL interface
  - Can also be used to store pure N-tuples
- **Required some interaction with low level interfaces**
  - Access to pool::IPersistencySvc interface

POOL Event Tag Collections

[Carmine Cioffi LHCb/Oxford]

- **To be used by the "EventSelector" component**
- **Take advantage of interface redesign**
  - Interface proposal is out
  - Waiting for implementation
- **Use POOL explicit collections for the implementation**
  - No major technical obstacles foreseen

Conclusions

- **SEAL dictionary will have to be fully integrated**
- **SEAL services can only be integrated when manpower is present**
- **POOL will soon be integrated in Gaudi**
  - Event data storage *~working*
    [dictionary generator missing]
    [MC truth relationships missing]
  - Gaudi event tag collections *working*
  - POOL event tag collections *coming*

Additional Personal Remarks

- **Integrated LCG software development environment is suboptimal**
  - In clear text: it is close to not existing
  - Situation is only marginally better than 12 years ago on CERNVM
- **Debugger does not work properly on public platforms**
  - Cannot save state, hangs, spurious error messages about RTTI missing, problems with dynamic loading….
  - Ask whom you want: Rado, Pere, me…
  - Something is wrong with Lxplus installation of gcc 3.2, gdb or other components
- **No windows built available: Cannot use VC++**
  - Could maintain a strip down version until this summer: Gone by now…
- **I have 2 hats: This will also affect the POOL development Root implementation of POOL was developed with VC++**