Persistency Framework RTAG: Interim Report

David Malon
malon@anl.gov

LHC Computing Grid Launch Week
12 March 2002

SC2 mandate to the RTAG

- Write the product specification for the Persistency Framework for Physics Applications at LHC
- Construct a component breakdown for the management of all types of LHC data
- Identify the responsibilities of Experiment Frameworks, existing products (such as ROOT) and as yet to be developed products
- Develop requirements/use cases to specify (at least) the metadata /navigation component(s)
- Estimate resources (manpower) needed to prototype missing components

Guidance from the SC2

- The RTAG may decide to address all types of data, or may decide to postpone some topics for other RTAGS, once the components have been identified.
- The RTAG should develop a detailed description at least for the event data management.
- Issues of schema evolution, dictionary construction and storage, object and data models should be addressed.

RTAG Composition

- One member from each experiment, one from IT/DB, one from ROOT team:
  - Fons Rademakers *(Alice)*
  - David Malon *(ATLAS)*
  - Vincenzo Innocente *(CMS)*
  - Pere Mato *(LHCb)*
  - Dirk Duellman *(IT/DB)*
  - Rene Brun *(ROOT)*

Quoting Vincenzo's report at CMS Week (6 March 02)

"Collaborative, friendly atmosphere"

"Real effort to define a common product"

This is already an accomplishment.

Response of RTAG to mandate and guidance (excerpted from report)

- Intent of this RTAG is to assume an optimistic posture regarding the potential for commonality among the LHC experiments in all areas related to data management
- Limited time available to the RTAG precludes treatment of all components of a data management architecture at equal depth
  - will propose areas in which further work, and perhaps additional RTAGs, will be needed

Response of RTAG to mandate and guidance (excerpted from report)

- Consonant with SC2 guidance, the RTAG has chosen to focus its initial discussions on the architecture of a persistence management service based upon a common streaming layer, and on the associated services needed to support it
  - Even if we cannot accomplish everything we aspire to, we want to ensure that we have provided a solid foundation for a near-term common project
- While our aim is to define components and their interactions in terms of abstract interfaces that any implementation must respect, it is not our intention to produce a design that requires a clean-slate implementation

Response of RTAG to mandate and guidance (excerpted from report)

- For the streaming layer and related services, we plan to provide a foundation for an initial common project that can be based upon the capabilities of existing implementations, and upon ROOT's I/O capabilities in particular
- While new capabilities required of an initial implementation should not be daunting, we do not wish at this point to underestimate the amount of repackaging and refactoring work required to support common project requirements

RTAG timetable

- RTAG met for the first time on 28 January
- Further meetings on 29, 30, 31 January and 18, 19 February
- Aside: my own availability turned out not to be a significant constraining factor
- Interim report to SC2 on 8 March
- Use LCG launch week (11-15 March) to solicit additional input and feedback
- Deliver a final report by 29 March (prior to 5 April SC2 meeting in any case)

Status

- Reasonable agreement on design criteria, e.g.,
  - Component oriented, communication through abstract interfaces, no back channels, components make no assumptions about implementation technology of components with which they communicate
  - Persistence for C++ data models is the principal target, but our environments are already multilingual; should avoid constructions that make language migration and multi-language support difficult
  - Architecture should not preclude multiple persistence technologies
  - Experiments' transient data models should not need compile-time/link-time dependencies on persistence technology in order to use persistence services

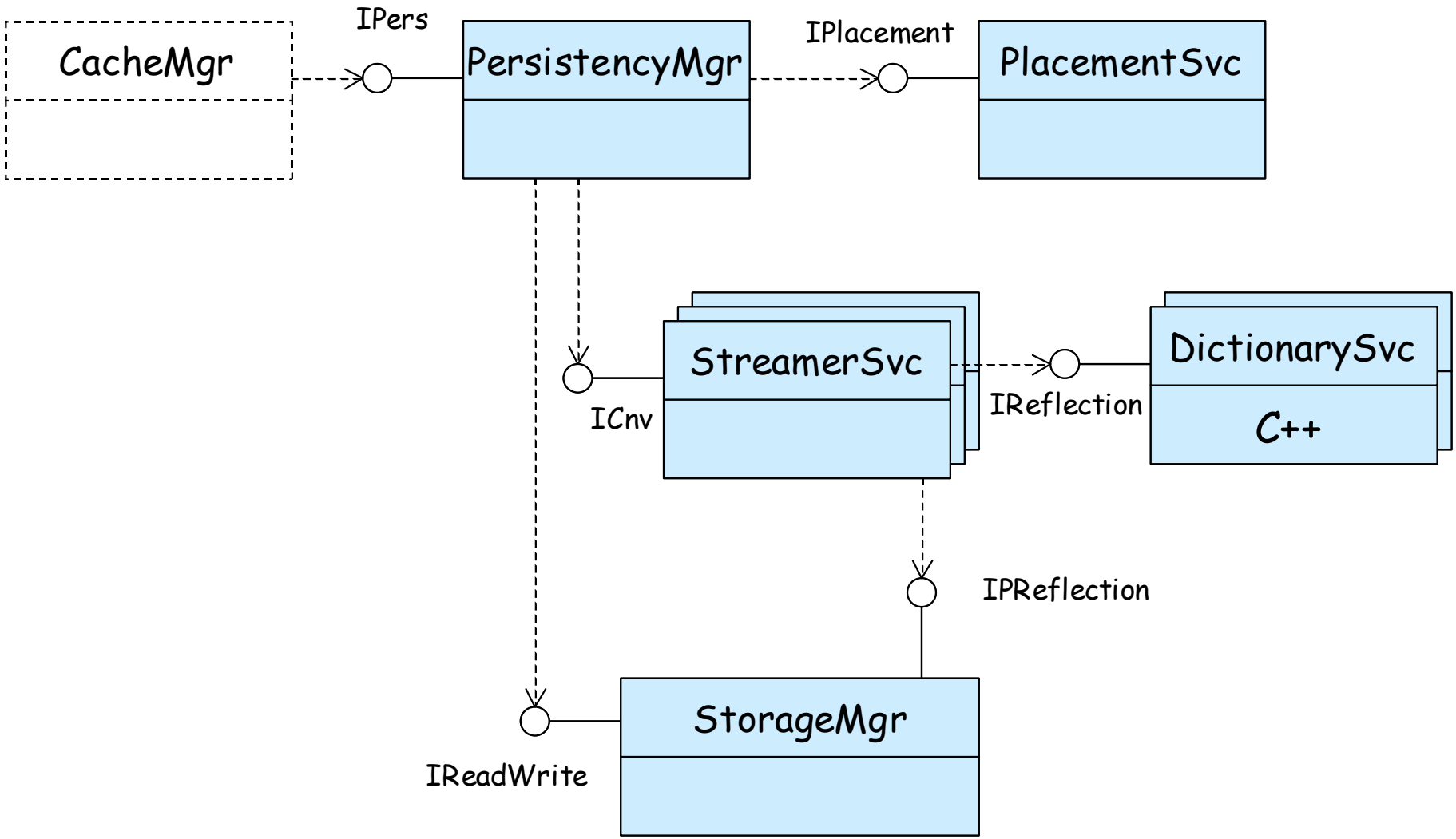
Status II

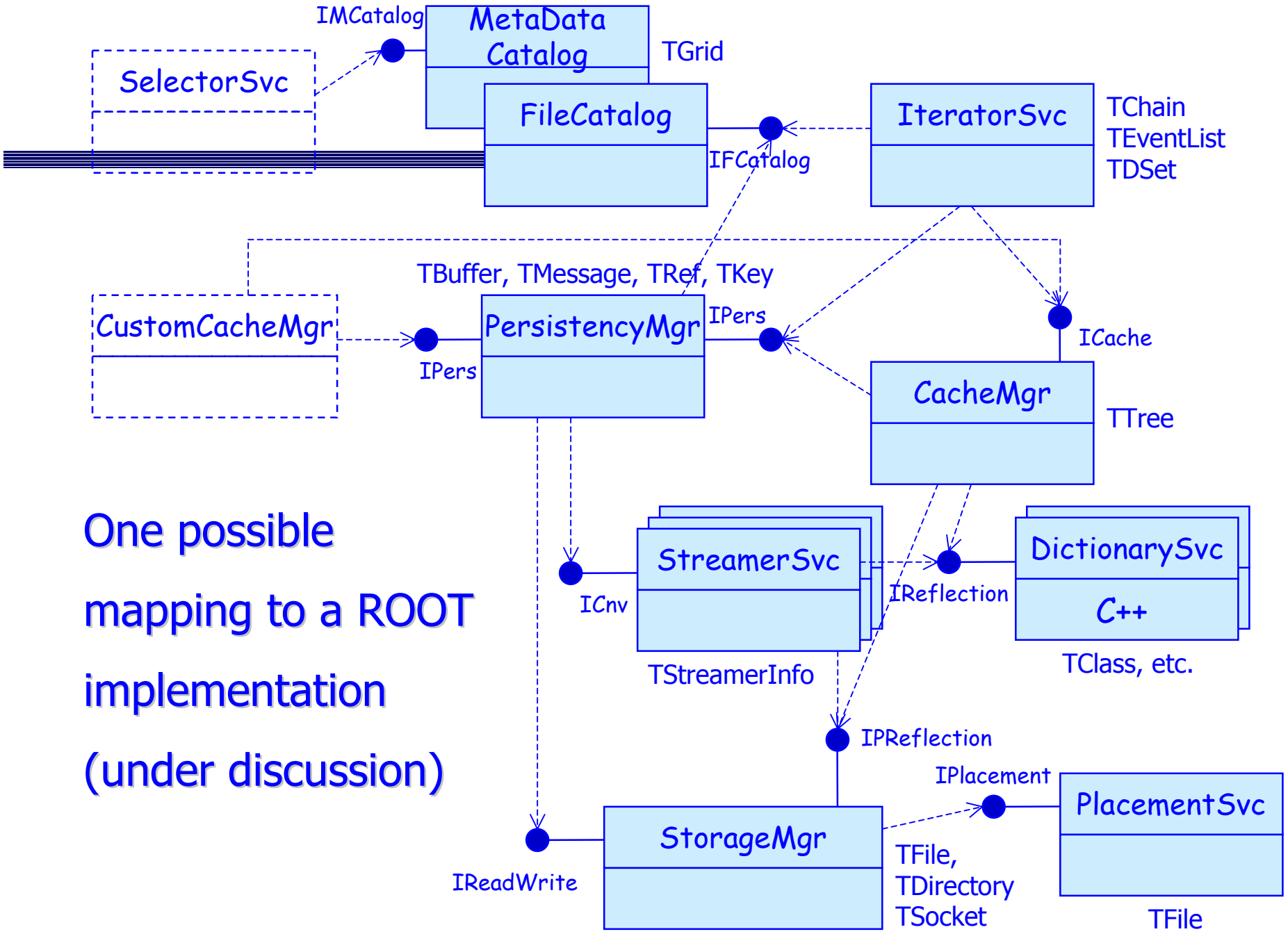
- Reasonable agreement on design criteria, e.g.,
  - Transient object types may have several persistent representations, the type of a transient object restored from a persistent one may be different than the type of the object that was saved, a persistent object cannot assume it "knows" what type of transient object will be built from it
  - …more…
- Component discussions and requirement discussions have been uneven—extremely detailed and highly technical in some areas, with other areas neglected thus far for lack of time
- Primary focus has been on issues and components involved in defining a common persistence service
  - Cache manager, persistence manager, storage manager, streamer service, placement service, dictionary service(s), …
  - Object identification, navigation, …

Caveat

- Do not take the following diagrams literally: their inclusion in this report is intended to illustrate that we are working both to describe a common architecture, its components, and its abstract interfaces, and simultaneously to ensure that we producing something reasonably matched to the capabilities of existing technologies

RTAG's First Component Diagram (under discussion)




IMCatalog
MetaData Catalog
TGrid
SelectorSvc
FileCatalog
IFCatalog
IteratorSvc
TChain
TEventList
TDSet
TBuffer, TMessage, TRef, TKey
CustomCacheMgr
IPers
PersistencyMgr
IPers
ICache
CacheMgr
TTree
One possible mapping to a ROOT implementation (under discussion)
StreamerSvc
ICnv
IReflection
DictionarySvc
C++
TClass, etc.
TStreamerInfo
IPReflection
IPlacement
PlacementSvc
StorageMgr
IReadWrite
TFile,
TDirectory
TSocket
TFile

Status III

- Much discussion, some text, and a good start on requirements for navigation, object identity, externalizable Reference classes, transient and persistent representation dictionaries, control of physical placement,…
- Overall component breakdown: underway
- Use cases: weak thus far
- Resource estimates: not yet addressed

Interim conclusions

- RTAG's highest priority is to provide foundation for a near-term common project reasonably matched to current capabilities of ROOT, with a relational layer above it
- Optimistic about prospects to accomplish this—significant progress to date
- Additional work (further RTAGs) in other areas will almost certainly be necessary—we will make recommendations
- While the limited time available for this RTAG forces us to be a bit myopic, we do believe that LCG should not be too short-sighted—some amount of R&D should be part of the project effort profile