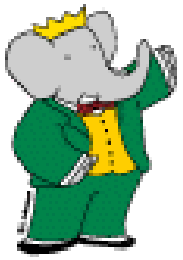
The new BaBar Computing and Analysis Model

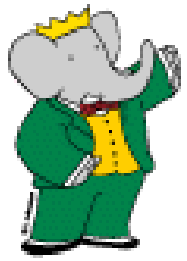


Peter Elmer (Princeton University)

For the BaBar Computing Group

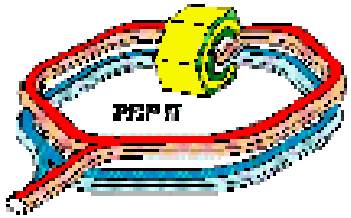
ACAT03

2 December, 2003



The BaBar Experiment

- BaBar is a colliding beam experiment whose primary objectives include B-physics and the study of CP-violation.
- It studies (asymmetric) electron-positron collisions at the Upsilon(4s) resonance using PEP-II at the Stanford Linear Accelerator Center (SLAC)



- BaBar is an international collaboration involving ~80 institutes in 10 countries
- We have been taking data since May, 1999. We are now several months into “Run4”, which will last through summer, 2004.

Overview

- BaBar last formally wrote down a computing model (CM1) in summer 2000
- During 2002, it became clear that many things had changed:
  - Two additional years of experience
  - BaBar had ever larger data samples and needed to support more numerous and detailed analyses
  - Distributed computing: 5 “Tier A” sites at end of 2002 (only 2 during formulation of CM1)
  - Continued use of two different eventstore technologies
  - The development of a new data *content* (the “mini”)
- In addition, making analysis easier, more flexible and reducing the time from “idea” to “result” is key to exploiting the very large dataset we are accumulating.

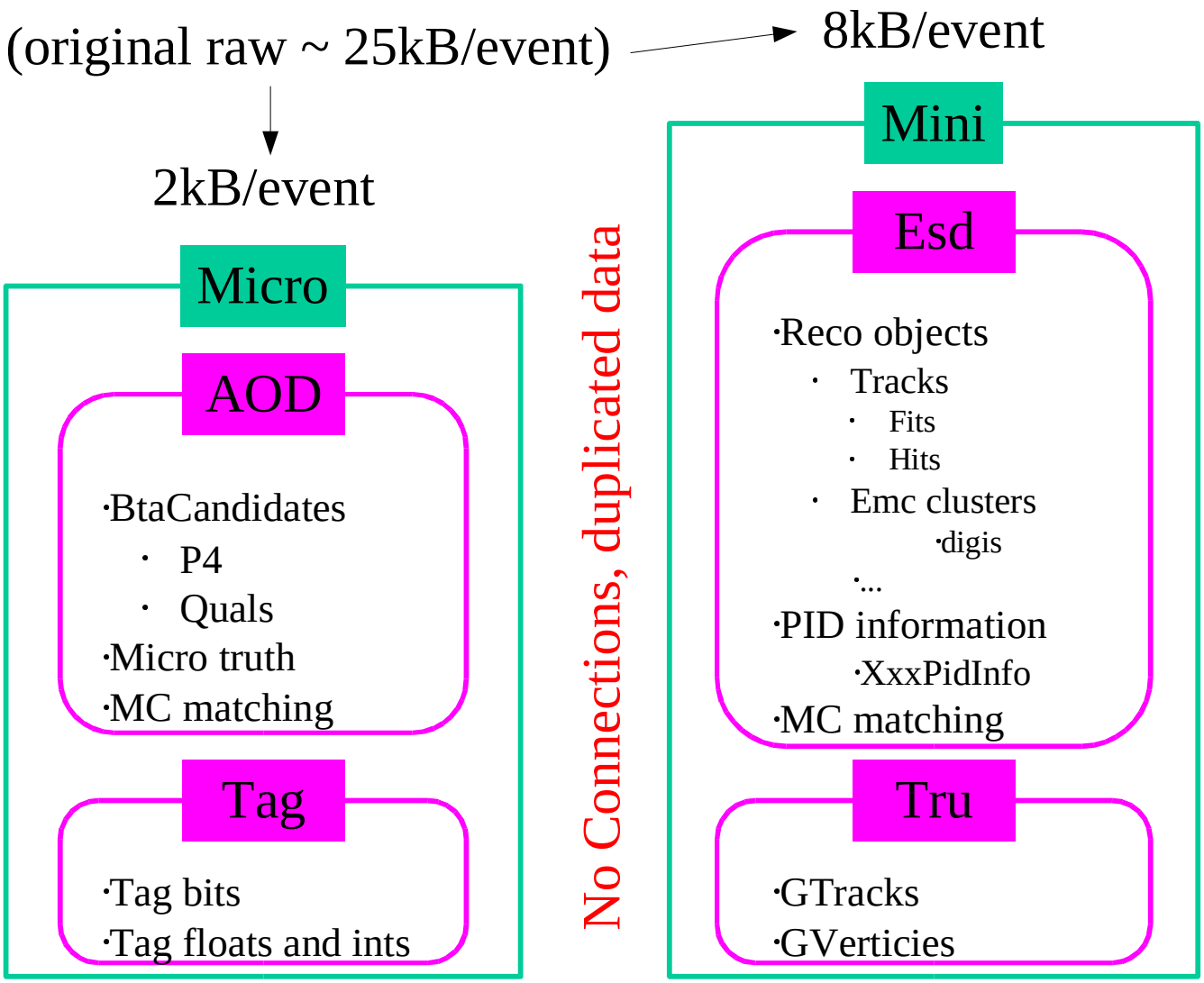
Overview (2)

- BaBar thus revised its computing model in fall 2002 (Computing Model 2 or “CM2”). The implementation of the new model has taken place during 2003 and it is currently being deployed.
- In this presentation I will discuss several key aspects of the new computing and analysis model:
  - The “mini” and data content
  - Eventstore
  - Skimming and user/analysis-specific customization
  - Data access
  - Distributed computing

Mini

- BaBar originally planned for a hierarchical eventstore with the possibility to “drill down”, for sub-samples of events, to more detailed (and hence typically larger) representations of the data:
  - Tag, Aod, (Tru), Esd, Rec, (Sim), Raw
- In practice, several issues have prevented users from using more than the Tag/Aod (usually called the “micro”):
  - Technical access problems with the eventstore
  - Large size/event of data levels other than the micro
  - Content not thought through in terms of actual use cases
- Users typically found themselves choosing between the “micro” and the original raw (a custom flat-file format outside of the eventstore)
- Driven initially in part by the needs of calibrations, a new and compact “mini” format was developed in 2001/2002 and stored in the Esd component of the eventstore

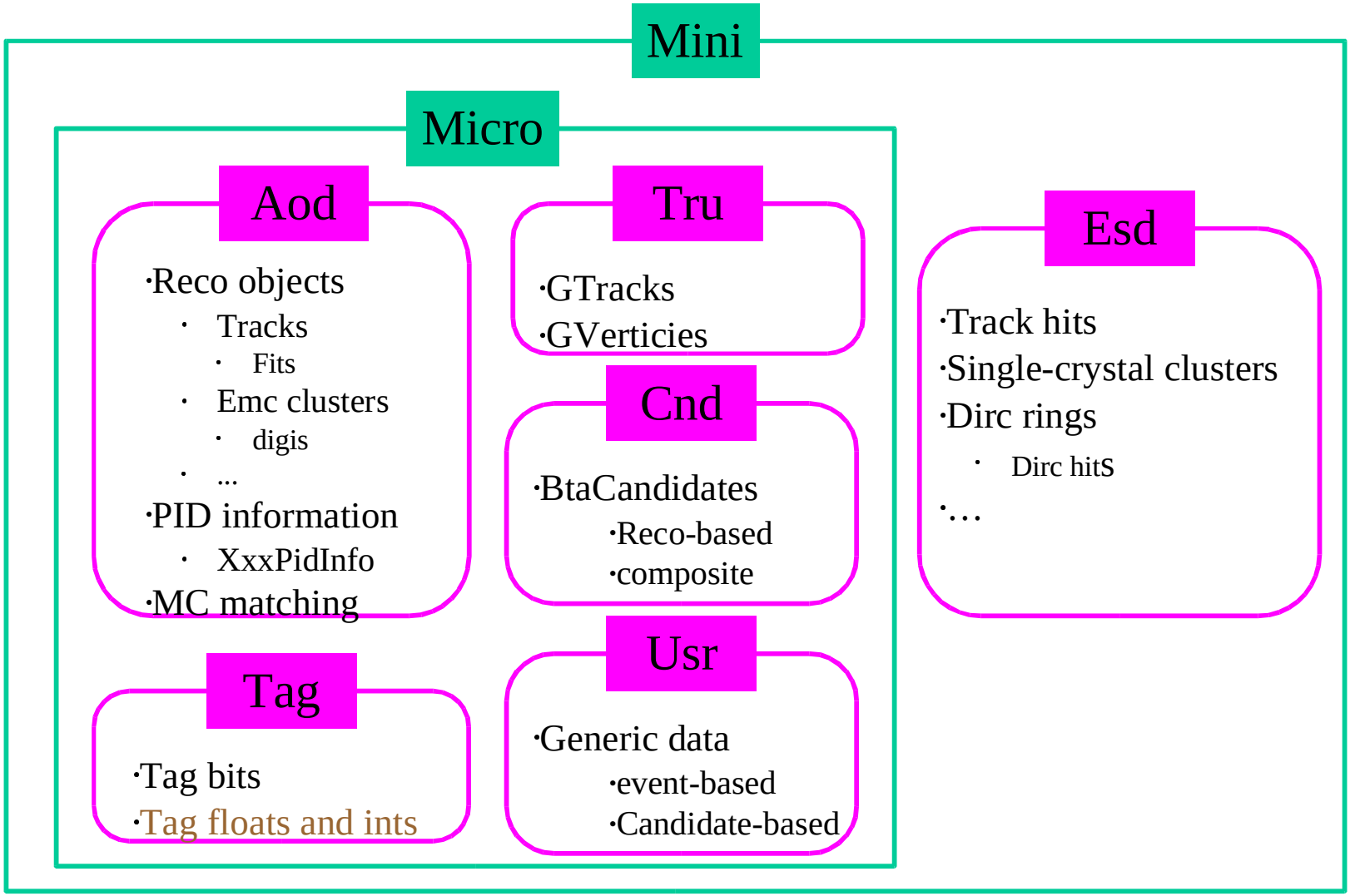
Original data organization with new Mini



- This new compact format solved the needs of calibrations, but still left a fundamental gap between the "micro" being used by the average analysis user and the "esd"

- As part of the CM2 implementation, we decided to improve on the Micro implementation

New CM2 data organization



- This change was happening in parallel to some of the technology changes I'll describe in upcoming slides

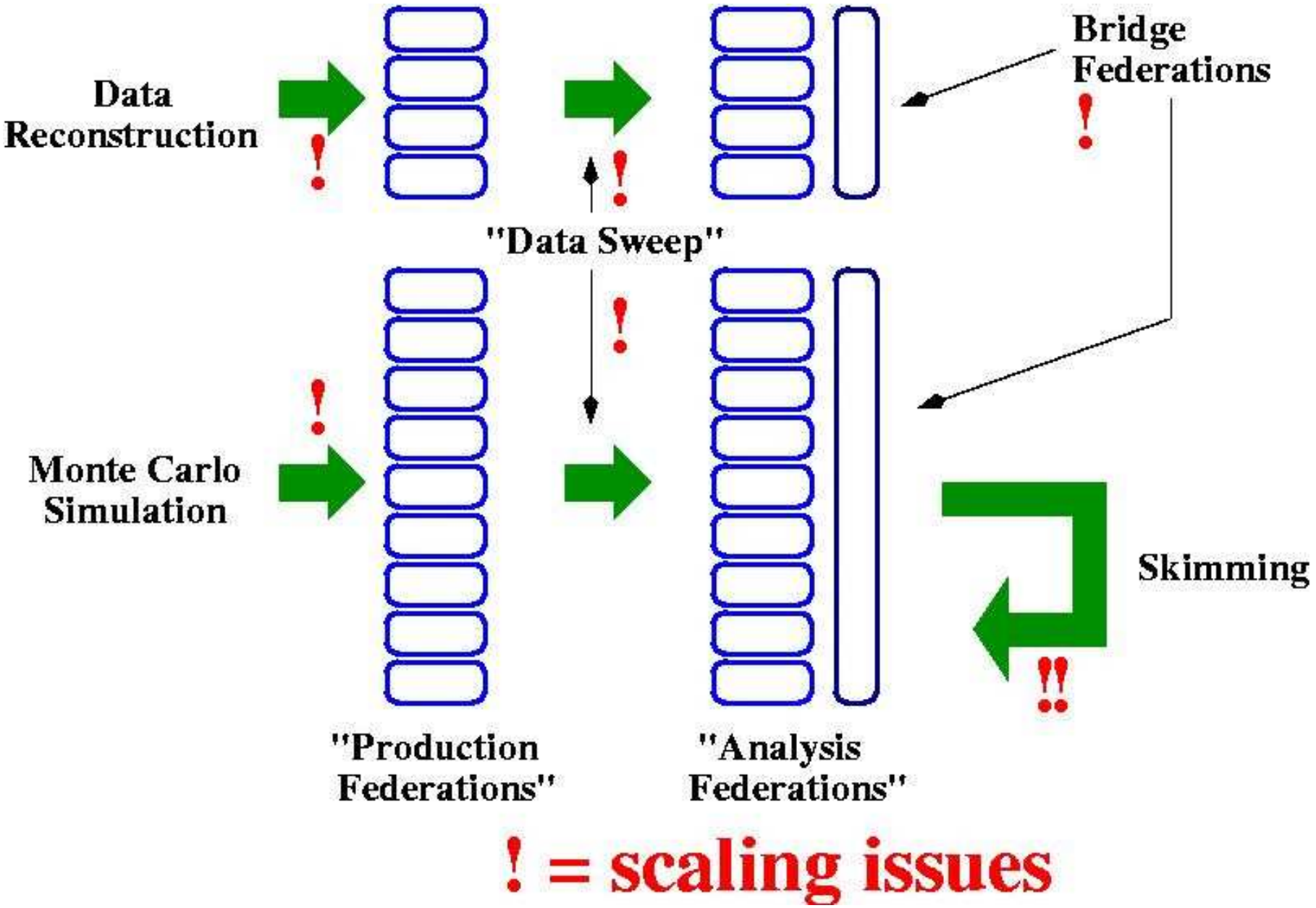
Eventstore

- BaBar's primary eventstore (Bdb) has been based on a commercial OO database technology (Objectivity). All data and MC production wrote into Objectivity and analysis jobs read from there (in particular at SLAC and CCIn2p3).
- The Bdb eventstore supported a hierarchy of data components of (theoretically) increasing detail (tag, aod, esd, rec, raw).
- Early difficulties with the Objy eventstore led to the development of an “analysis-only” format (Kanga) in late 1999. This was based on ROOT I/O: a single TTree contains the “micro” (tag/aod) data. The Kanga data was produced by a dedicated conversion application from the Bdb/Objy eventstore. This data format was used at RAL and Karlsruhe, and was the only data format used for analysis at universities.

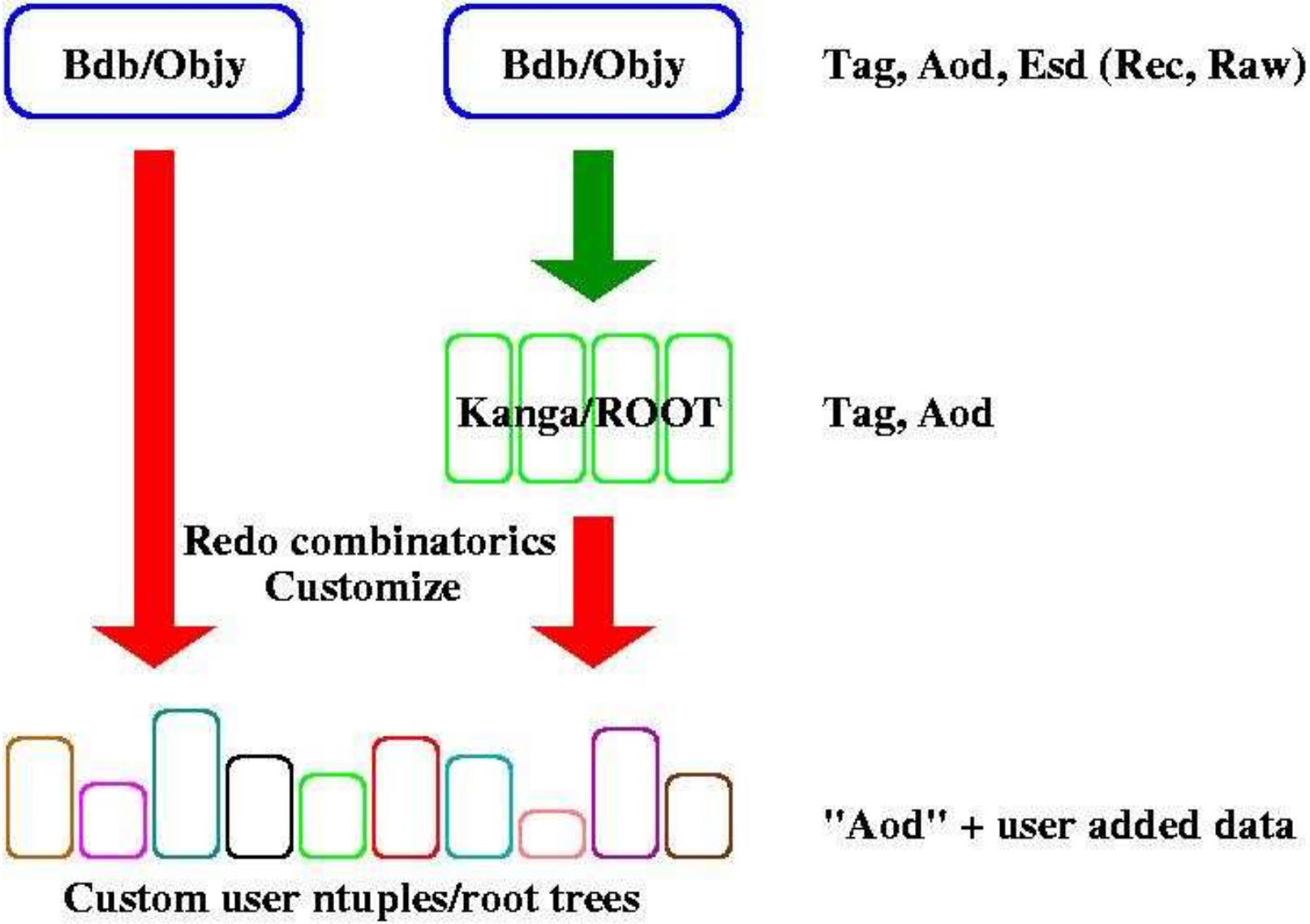
Bdb/Objy eventstore problems

- Many, many, many scaling issues for both production and analysis. (When you have 100's or 1000's of jobs the keyword is “parallel”.)
- High maintenance costs (unacceptably high for small sites to use for analysis)
- Data volume was larger than desired due to:
  - Navigational overheads from the eventstore design, poor packing
  - Lack of useful compression (until relatively recently)
- Difficulty to access and distribute the data
- Of particular difficulty was the transition to the use of “multiple federations” to deal with limited number of DBID's per Objy federation
- Concerns about relying on a proprietary technology from a small company while HEP moves in another direction
- But we did produce physics results since 4 years using this technology!

Data Production with Bdb/Objy



Old analysis method



New CM2 Kanga/ROOT Eventstore

- Extended version of our Kanga/ROOT eventstore
- Multiple TTrees spread across several files, one tree per data “component” (tag, aod, esd, ...). New components (usr, cnd) where users may add their own customized data.
- A simple event header knows where components for that particular event are located (allows for sparse “borrowing” and “pointer” collections). The TTree containing the event headers defines the “collection”.
- Do not require that a job access a central catalog to run
- In production the output of each job is separate from others (simplifying parallelization, writing to local disk on farm nodes, etc.). If necessary the data from multiple jobs is merged in a subsequent process.

New CM2 “Kanga” Eventstore

pcuw01

```
noric10> KanCollUtil -L /work/users/elmer/031123/elfevents4
/work/users/elmer/031123/elfevents4 (729 events)
   LFN 000  /work/users/elmer/031123/elfevents4.01.root (owned)
   LFN 001  /work/users/elmer/031123/elfevents4.02E.root (owned)
noric10>
noric10> KanCollUtil -P /work/users/elmer/031123/elfevents4
/work/users/elmer/031123/elfevents4 (729 events)
   PFN 000  /nfs/kan001/vol6//work/users/elmer/031123/elfevents4.01.root
   PFN 001  /nfs/kan001/vol6//work/users/elmer/031123/elfevents4.02E.root
noric10>
noric10> KanCollUtil -n 5 -f /work/users/elmer/031123/elfevents4
/work/users/elmer/031123/elfevents4 (729 events)
   EVT 000001  hdr=000:0 tag=000:0 cnd=000:0 aod=000:0 esd=001:0
   EVT 000002  hdr=000:1 tag=000:1 cnd=000:1 aod=000:1 esd=001:1
   EVT 000003  hdr=000:2 tag=000:2 cnd=000:2 aod=000:2 esd=001:2
   EVT 000004  hdr=000:3 tag=000:3 cnd=000:3 aod=000:3 esd=001:3
   EVT 000005  hdr=000:4 tag=000:4 cnd=000:4 aod=000:4 esd=001:4
noric10>
noric10> KanFileUtil -t all /nfs/kan001/vol6//work/users/elmer/031123/elfevents4
.01.root
/nfs/kan001/vol6//work/users/elmer/031123/elfevents4.01.root
   TREE        aod  tid=  4  cycle=  1  entries= 729
   TREE  aod__Meta  tid=  4  cycle=  2  entries= 1
   TREE        cnd  tid=  3  cycle=  1  entries= 729
   TREE  cnd__Meta  tid=  3  cycle=  2  entries= 1
   TREE        hdr  tid=  0  cycle=  1  entries= 729
   TREE  hdr__Meta  tid=  0  cycle=  2  entries= 1
   TREE        tag  tid=  2  cycle=  1  entries= 729
   TREE  tag__Meta  tid=  2  cycle=  2  entries= 1
noric10>
```

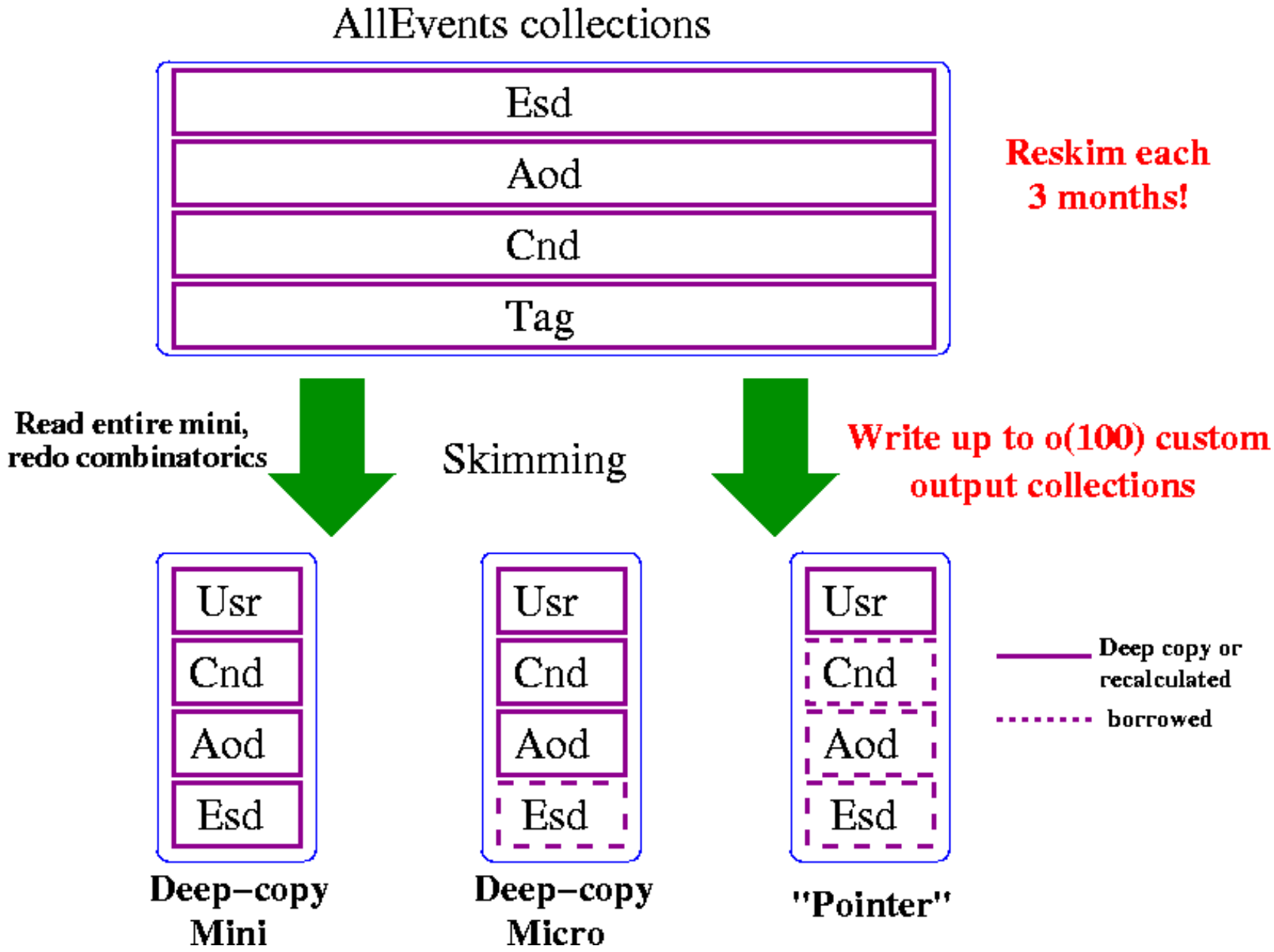
Interactive use of (new) Kanga

- In the original Kanga/ROOT implementation we used ROOT I/O as a sort of database. (BaBar's event model has a transient/persistent split.)
- It is not practical to use the original Kanga directly in interactive ROOT, but only in the context of the full BaBar Framework after the objects have been read in and converted to fully transient objects.
- For some purposes, it was considerable desirable for CM2 to allow “interactive” access (i.e. at the ROOT CINT prompt or in small standalone applictions) directly to the ROOT trees.
- This new Kanga version fully supports this interactive use, which is ideal for both adminstrative needs and initial exploratory analysis efforts.

Skim production

- Building on the new eventstore we decided to create a centralized “skim” production which would run frequently (nominally each 3 months)
- Each analysis group or user may define a “skim” output for this production in which they may chose to:
  - Deep-copy micro or mini data
  - Add their own customized data (e.g. composite candidate lists or associate calculated quantities with either candidates or the event)
- In the past “skimming” in BaBar has meant redoing (and adding new) event selections and rewriting the tag bits only!
- There are currently o(100) skims defined, users “opt-in” each round
- This is intended to replace the “ntuple productions” that the analysis groups were doing in the past, avoiding the pointless duplication of cpu and I/O and (perhaps) incoherent access to data in mass storage. Output is stored (and managed!) as eventstore data.

Skim production



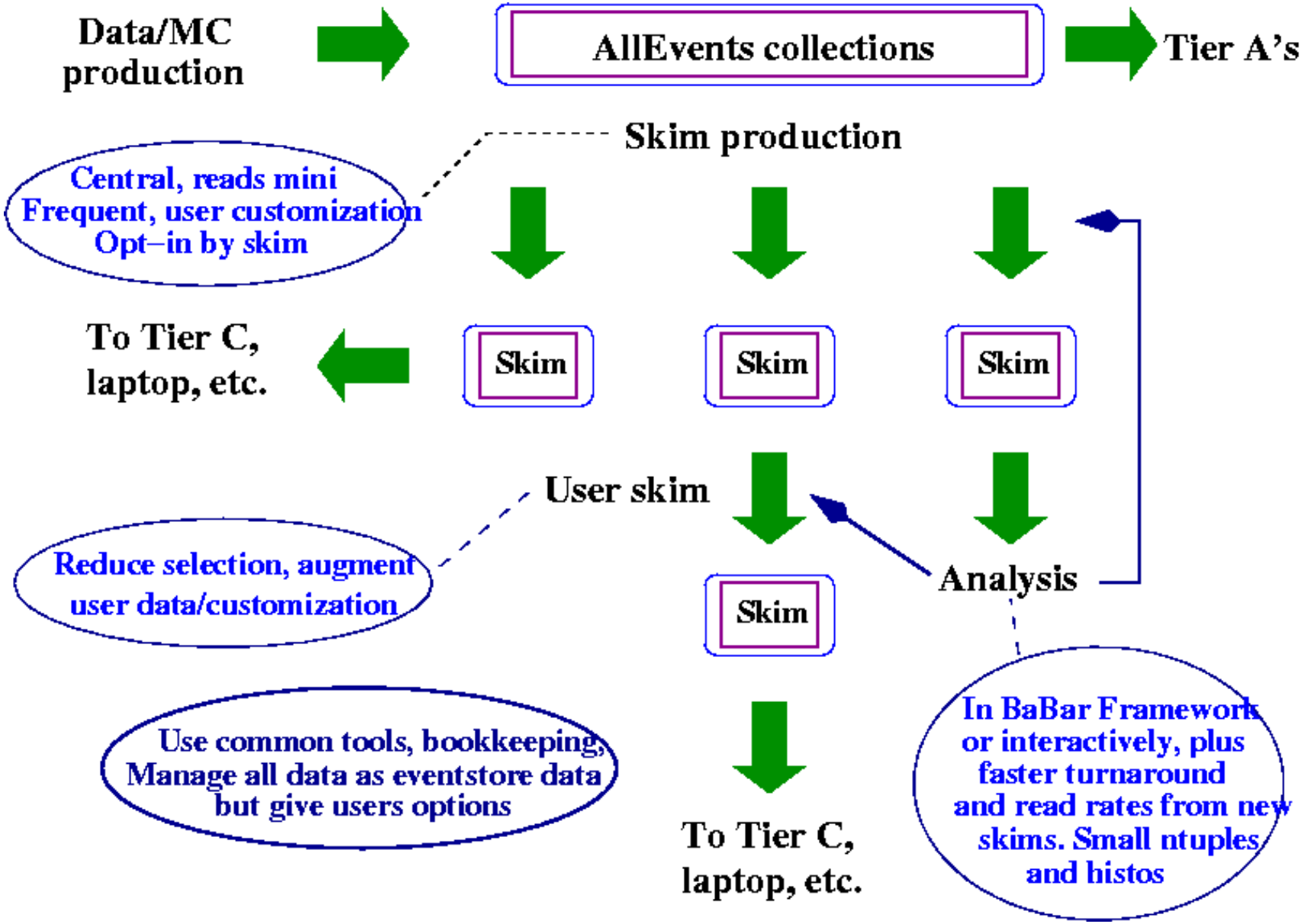
- Deep copy only the micro, reference original esd:

```
pcuw01
noric10> KanCollUtil /store/test/reco/recoevents
/store/test/reco/recoevents (729 events)
noric10>
noric10> KanCollUtil -P /work/users/elmer/skims/DileptonKan
/work/users/elmer/skims/DileptonKan (16 events)
   PFN 000  /nfs/kan001/vol6//work/users/elmer/skims/DileptonKan.01.root
   PFN 001  root://kan002.slac.stanford.edu:1094/kanga//store/test/reco/recoeven
ts.02E.root
noric10> KanCollUtil -n 5 -f /work/users/elmer/skims/DileptonKan
/work/users/elmer/skims/DileptonKan (16 events)
   EVT 000001  hdr=000:0 tag=000:0 cnd=000:0 aod=000:0 esd=001:1
   EVT 000002  hdr=000:1 tag=000:1 cnd=000:1 aod=000:1 esd=001:18
   EVT 000003  hdr=000:2 tag=000:2 cnd=000:2 aod=000:2 esd=001:31
   EVT 000004  hdr=000:3 tag=000:3 cnd=000:3 aod=000:3 esd=001:177
   EVT 000005  hdr=000:4 tag=000:4 cnd=000:4 aod=000:4 esd=001:396
noric10>
```

- Deep copy the full mini:

```
pcuw01
noric10> KanCollUtil -P /work/users/elmer/skims/AlignCalKan
/work/users/elmer/skims/AlignCalKan (64 events)
   PFN 000  /nfs/kan001/vol6//work/users/elmer/skims/AlignCalKan.01.root
noric10> KanCollUtil -n 5 -f /work/users/elmer/skims/AlignCalKan
/work/users/elmer/skims/AlignCalKan (64 events)
   EVT 000001  hdr=000:0 tag=000:0 cnd=000:0 aod=000:0 esd=000:0
   EVT 000002  hdr=000:1 tag=000:1 cnd=000:1 aod=000:1 esd=000:1
   EVT 000003  hdr=000:2 tag=000:2 cnd=000:2 aod=000:2 esd=000:2
   EVT 000004  hdr=000:3 tag=000:3 cnd=000:3 aod=000:3 esd=000:3
   EVT 000005  hdr=000:4 tag=000:4 cnd=000:4 aod=000:4 esd=000:4
noric10>
```

New analysis model



Xrootd

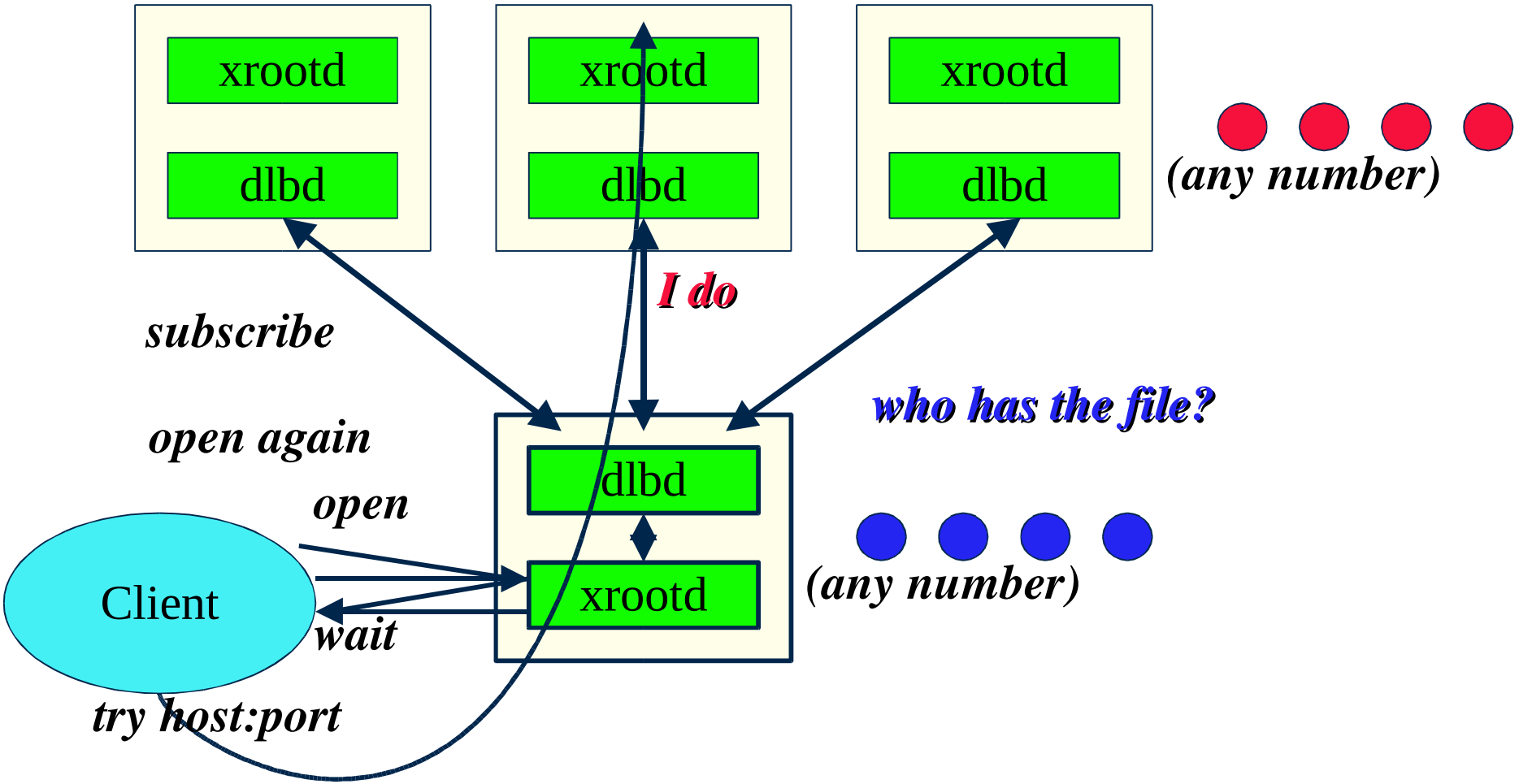
- Bdb/Objy eventstore used AMS for daemon based file access and as hook for dynamic file staging, load balancing, etc.
- We want the equivalent functionality for the new eventstore as an alternative to NFS file access
- The existing rootd server serves files, but lacks many features that we had with the AMS (or wish we had with the AMS)
- Want fault-tolerant, scalable, high performance file access
- Build on experience with AMS and reuse related code (e.g. load balancer) when possible

➔ BaBar has developed a new general replacement for rootd (and TNetFile) with an extended feature set

Xrootd/XTNetFile features

- Multi-threaded daemon, normally one per data server
- Connection multiplexing
- Request redirection (e.g. for use in load-balancing, fault tolerance)
- Request deferral
- Eventual asynchronous mode (including client pre-read)
- Unsolicited reverse request (server manages client)
- See web page: http://www.slac.stanford.edu/~abh/xrootd/
- Compatible with old rootd/TNetFile, plan to merge back into ROOT distribution as a replacement for existing rootd/TNetFile on the time scale of the ROOT2004 workshop at SLAC (25-27 Feb., 2004)

Data access model with xrootd and dynamic load balancer (dlb)



Distributed Computing

- Having a single eventstore format simplifies things, with CM2 we no longer have such a “balkanized” situation (Tier A's and universities using different formats)
- BaBar now has 5 “Tier A” sites (SLAC, CCIn2p3, Karlsruhe, Padova, RAL) plus 20+ “Tier C” university sites
- Significant step forward in the past months is that all new data is processed off-site, in Padova, and output can be back at SLAC with an average latency under 24 hours. Padova and GridKa are also setting up to run production skims. (Previously done entirely at SLAC.)

Distributed computing (2)

- Non-SLAC Tier A sites (Padova and Karlsruhe) will contribute significantly to the skim productions
- RAL and In2p3 provide login access to collaborators for analysis work and generate MC and will continue as such (providing also transitional access to “classic” Kanga and Objy data)
- Tier C (University) sites will continue to produce MC, but the new model will provide new opportunities and flexibility to “take a skim home” for analysis
- Expect evolution in resource utilization over time, but flexibility of new model/eventstore should only improve our ability to use resources (even those only available transiently).

CM2 Deployment Status

- Users updated/developed their skims in Aug/Sep2003
- PromptReco (data production) began production testing in Jul2003 and began to write (new) data in the new Kanga format from Sep2003
- Conversion (+ repair of data using mini and new calibrations!) of existing tag/aod/esd data from Bdb/Objy began 1 month ago
- The first skim production has been ramping up over the past month, should finish in Jan/Feb2003
- Simulation Production has been testing continously since very early this year, will start production soon.
- We expect that the new computing and analysis model will begin to make its impact on analysis from early this spring

Summary

- BaBar has reworked a number of pieces of its computing model over the past year, including:
  - a new eventstore, extending our original Kanga/ROOT eventstore in a scalable way and providing for “interactive” access
  - new data content (the “mini”) and a reworked “micro”
  - a new skim production (with customizable content) which runs very frequently, allowing new and updated analyses to be introduced
  - A new fault-tolerant and scalable means for data access (xrootd)
  - Improved bookkeeping and a new Task Management system
  - Continued emphasis on and improved use of distributed computing
- All of these things are expected to simplify and improve analysis in BaBar as we move to ever larger data samples

Backup slides

Bookkeeping/Task Management (2)

- Ever larger datasets also means ever larger numbers of jobs that a typical user needs to run as part of their analysis
- Basic tools were provided to help users determine which collections they could run over and to submit the jobs, but they were largely left on their own to determine whether the jobs succeeded, resubmit any which failed, etc.
- As part of the new computing model, we also have developed a new “Task Management” system which:
  - Will be more integrated with the dataset bookkeeping
  - Take much of the burden off the user in managing so many jobs
  - Generically apply a “task” to a “dataset”
- This is currently being tested to do our skim productions

Bookkeeping/Task Management

- BaBar uses a set of custom bookkeeping tools implemented in perl and using a relational database
- Users can query the bookkeeping for lists of collections meeting particular criteria (software release used, etc.), but are left on their own to manage these lists. In addition updates (adding, removing good/bad runs) are a bit ad-hoc.
- For historical reasons some information needed by users was not in the central collection database, but in various “production” databases and hence users needed to look in multiple places.
- In parallel to the other CM2 changes we decided to rework the bookkeeping in order to introduce more explicitly the concept of a dataset and to centralize the API's for accessing various pieces of information.