STAR Offline Computing and Software

Torre Wenaus
STAR Computing and Software Leader
Brookhaven National Laboratory, USA

ALICE / STAR Computing Meeting
April 8, 2000

Outline

STAR and STAR Computing

Offline software

- Organization, environment, QA
- Framework
- Event model and data management
  - Technology choices
    - ROOT based event store
    - MySQL databases

Analysis operations

- Grand Challenge Architecture

Current status

STAR at RHIC

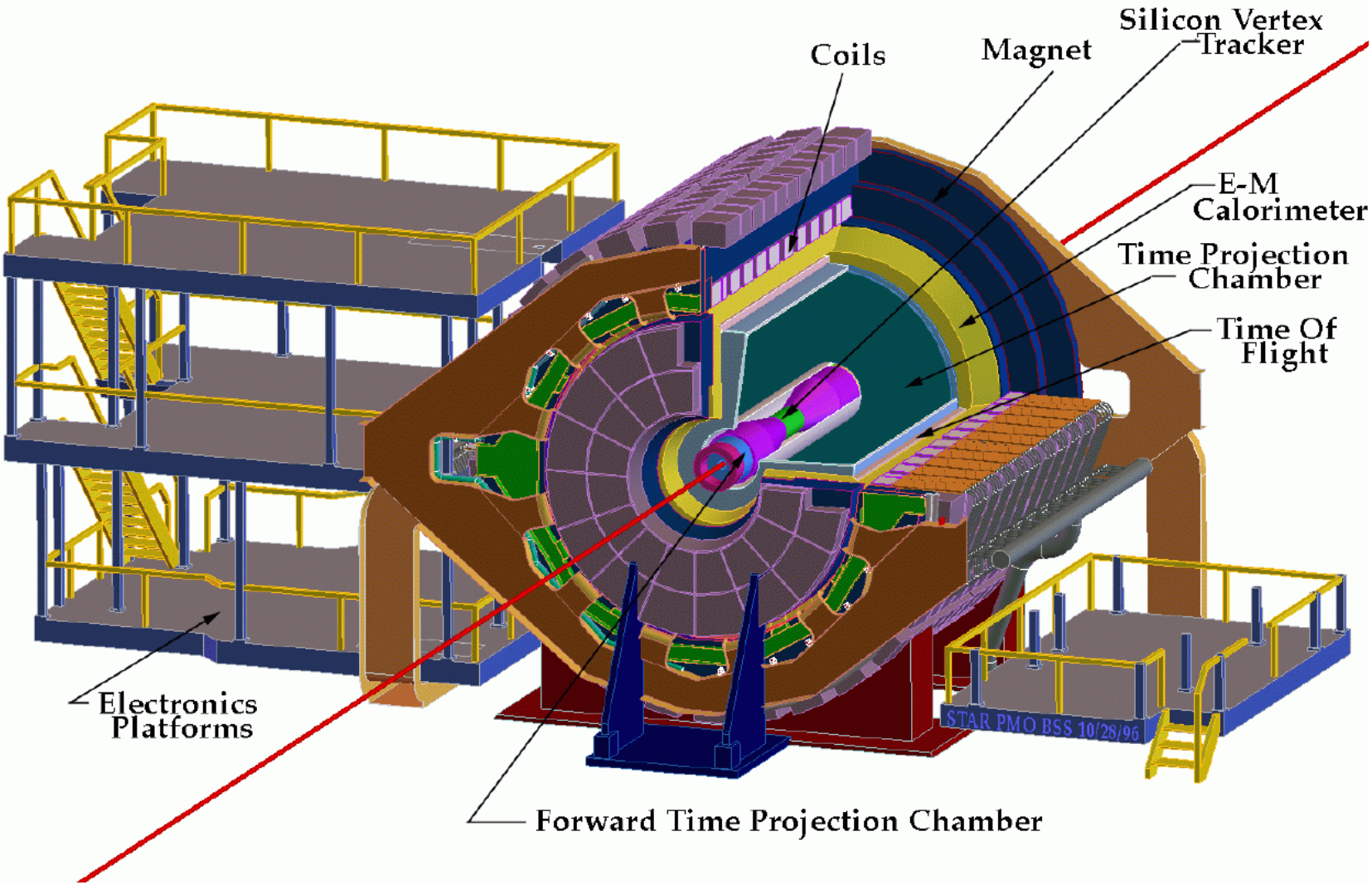
RHIC: Relativistic Heavy Ion Collider at Brookhaven National Laboratory

- Colliding Au - Au nuclei at 100GeV/nucleon
- Principal objective: Discovery and characterization of the Quark Gluon Plasma (QGP)
- First year physics run April-August 2000

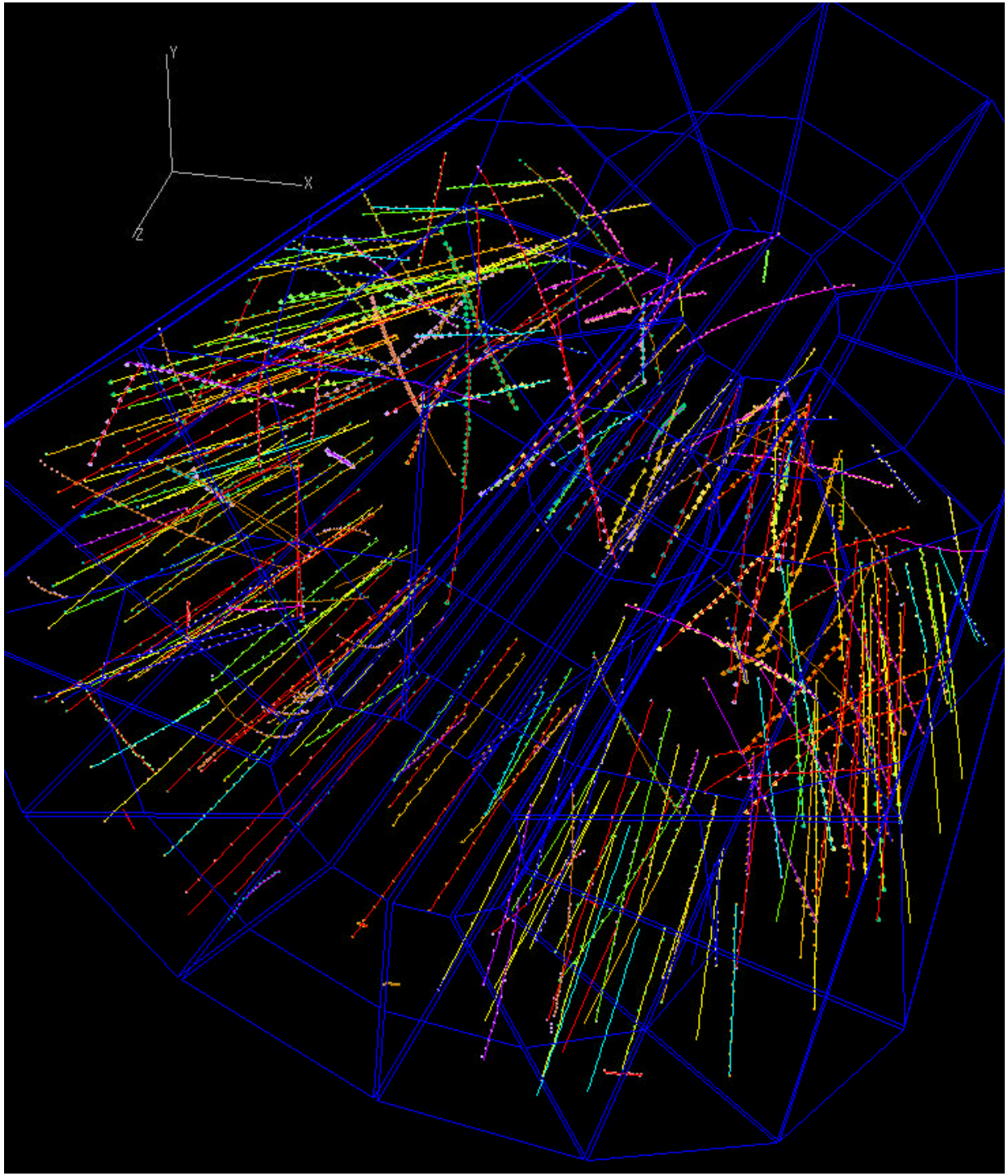
STAR experiment

- One of two large experiments at RHIC, >400 collaborators each
  - PHENIX is the other
- Hadrons, jets, electrons and photons over large solid angle
  - Principal detector: 4m TPC drift chamber
- ~4000+ tracks/event recorded in tracking detectors
- High statistics *per event* permit event by event measurement and correlation of QGP signals

STAR Detector



Summer ’99
Engineering run
Beam gas event



Computing at STAR

Data recording rate of 20MB/sec; 15-20MB raw data per event (~1Hz)

- 17M Au-Au events (equivalent) recorded in nominal year
- Relatively few but highly complex events

Requirements:

- 200TB raw data/year; 270TB total for all processing stages
- 10,000 Si95 CPU/year
- Wide range of physics studies: ~100 concurrent analyses in ~7 physics working groups

Principal facility: RHIC Computing Facility (RCF) at Brookhaven

- 20,000 Si95 CPU, 50TB disk, 270TB robotic (HPSS) in ’01

Secondary STAR facility: NERSC (LBNL)

- Scale similar to STAR component of RCF

Platforms: Red Hat Linux/Intel and Sun Solaris

Computing Requirements

Nominal year processing and data volume requirements:

Raw data volume: 200TB

Reconstruction: 2800 Si95 total CPU, 30TB DST data

- 10x event size reduction from raw to reco
- 1.5 reconstruction passes/event assumed

Analysis: 4000 Si95 total analysis CPU, 15TB micro-DST data

- 1-1000 Si95-sec/event per MB of DST depending on analysis
  - Wide range, from CPU-limited to I/O limited
- 7 physics groups, ~100 active analyses, 5 passes per analysis
- micro-DST volumes from .1 to several TB

Simulation: 3300 Si95 total including reconstruction, 24TB

Total nominal year data volume: 270TB

Total nominal year CPU: 10,000 Si95

RHIC/STAR Computing Facilities

Dedicated RHIC computing center at BNL, the RHIC Computing Facility

- Data archiving and processing for reconstruction and analysis
- Three production components: Reconstruction (CRS) and analysis (CAS) services and managed data store (MDS)
- 10,000 (CRS) + 7,500 (CAS) SpecInt95 CPU
- ~50TB disk, 270TB robotic tape, 200MB/s I/O bandwidth, managed by High Performance Storage System (HPSS) developed by DOE/commercial consortium (IBM et al)
- Current scale: ~2500 Si95 CPU, 3TB disk for STAR

Limited resources require the most cost-effective computing possible

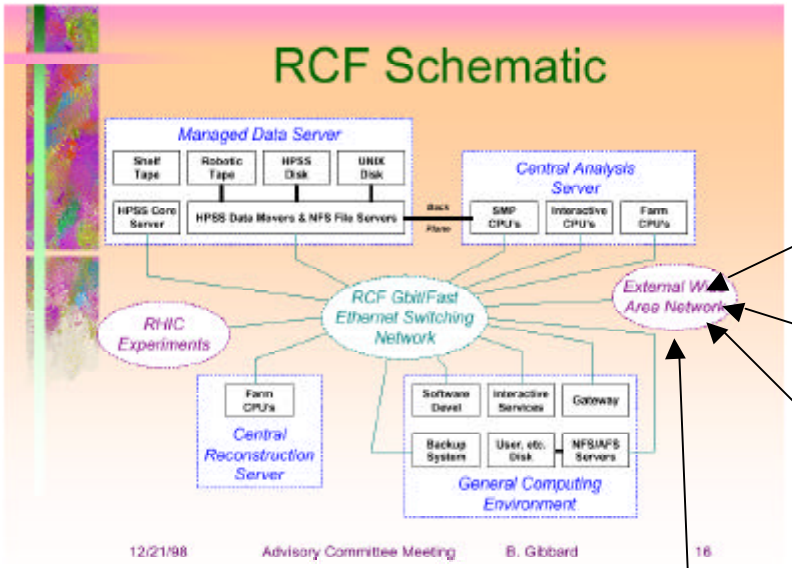
- Commodity Intel farms (running Linux) for all but I/O intensive analysis (Sun SMPs)

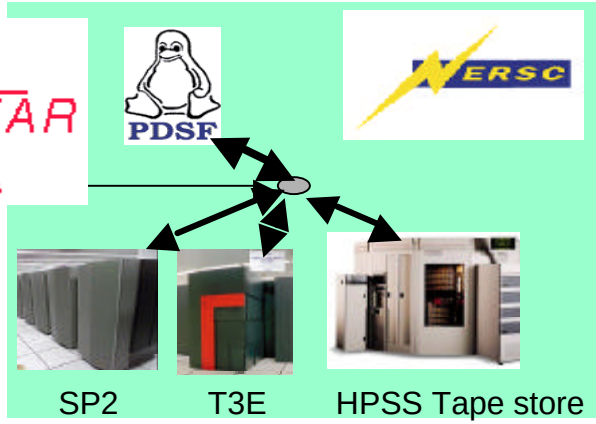
Smaller outside resources:

- Simulation, analysis facilities at outside computing centers
- Limited physics analysis computing at home institutions

Implementation of RHIC Computing Model

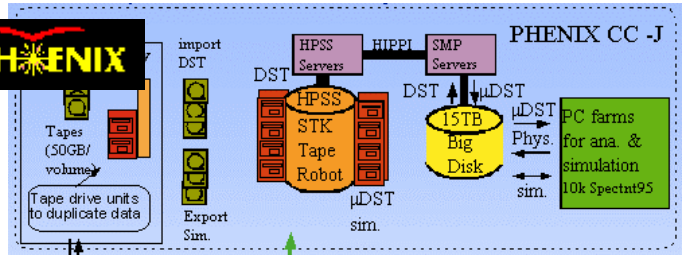
Incorporation of Offsite Facilities





Berkeley

Japan



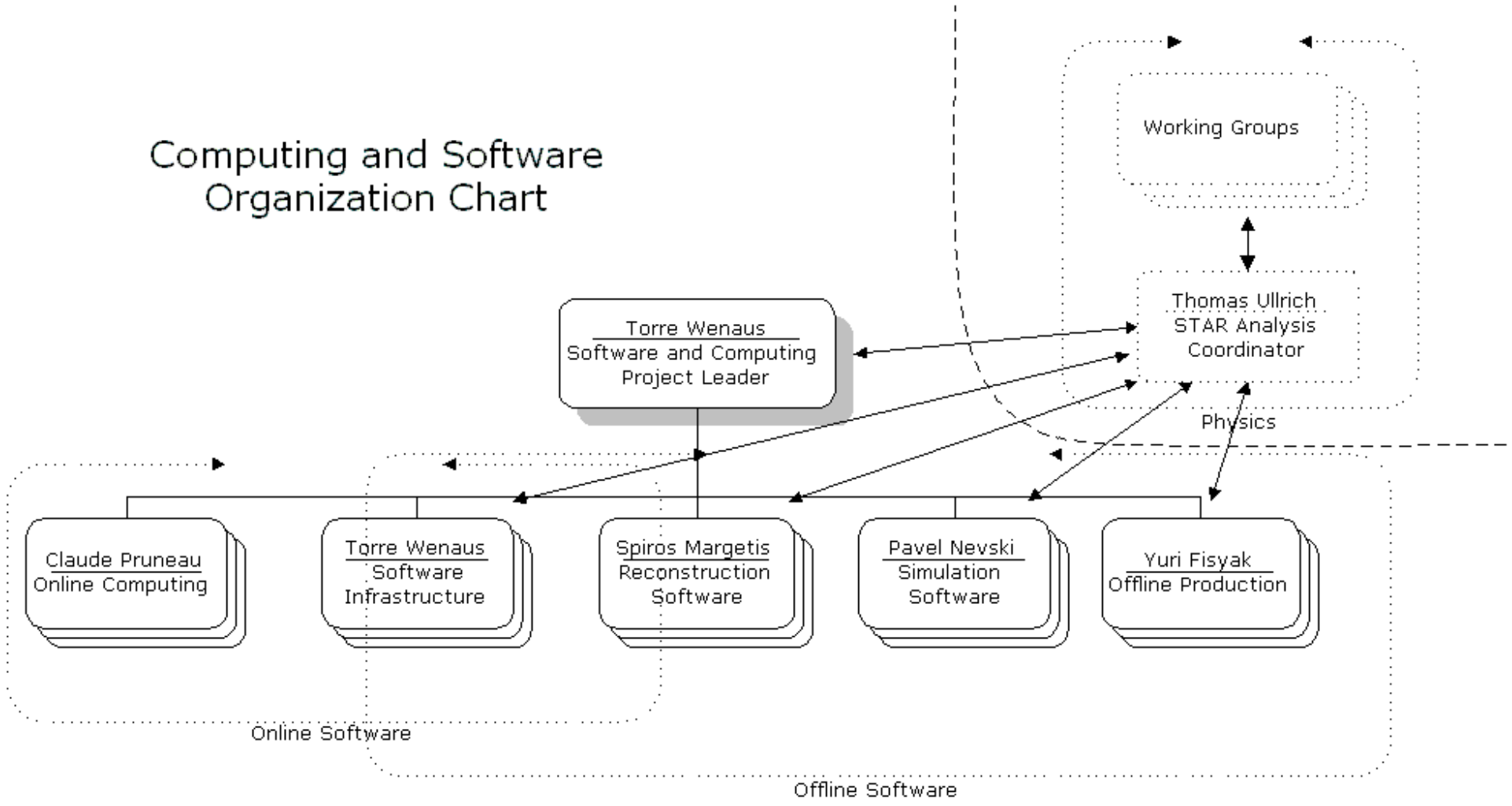
MIT

PHOBOS

The Scientific Computing Facility
Abacus @ mit-lns

Many universities, etc.

Computing and Software Organization



Some of our Youthful Participants

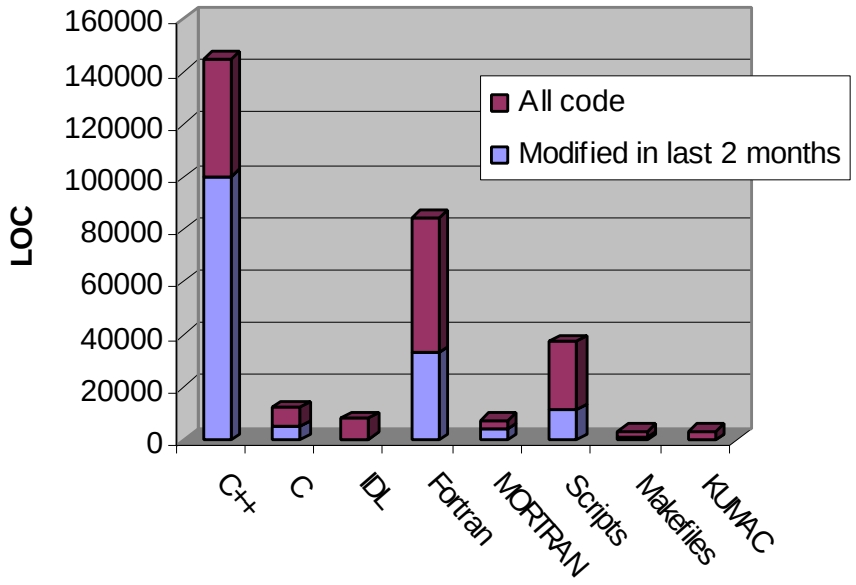
A **partial** list of young students and postdocs now active in aspects of software (as of last summer)

Dave Alvarez, Wayne, SVT
Lee Barnby, Kent, QA and production
Jerome Baudot, Strasbourg, SSD
Selemon Bekele, OSU, SVT
Marguerite Belt Tonjes, Michigan, EMC
Helen Caines, Ohio State, SVT
Manuel Calderon, Yale, StMcEvent
Gary Cheung, UT, QA
Laurent Conin, Nantes, database
Wensheng Deng, Kent, production
Jamie Dunlop, Yale, RICH
Patricia Fachini, Sao Paolo/Wayne, SVT
Dominik Flierl, Frankfurt, L3 DST
Marcelo Gameiro, Sao Paolo, SVT
Jon Gangs, Yale, online
Dave Hardtke, LBNL, Calibrations, DB
Mike Heffner, Davis, FTPC
Eric Hjort, Purdue, TPC
Amy Hummel, Creighton, TPC, production
Holm Hummler, MPG, FTPC
Matt Horsley, Yale, RICH
Jennifer Klay, Davis, PID
Matt Lamont, Birmingham, QA
Curtis Lansdell, UT, QA
Brian Lasiuk, Yale, TPC, RICH
Frank Laue, OSU, online
Lilian Martin, Subatch, SSD
Marcelo Munhoz, Sao Paolo/Wayne, online
Aya Ishihara, UT, QA
Adam Kisiel, Warsaw, online, Linux
Frank Laue, OSU, calibration
Hui Long, UCLA, TPC
Vladimir Morozov, LBNL, simulation
Alex Nevski, RICH
Sergei Panitkin, Kent, online
Caroline Peter, Geneva, RICH
Li Qun, LBNL, TPC
Jeff Reid, UW, QA
Fabrice Retiere, calibrations
Christelle Roy, Subatech, SSD
Dan Russ, CMU, trigger, production
Raimond Snellings, LBNL, TPC, QA
Jun Takahashi, Sao Paolo, SVT
Aihong Tang, Kent
Greg Thompson, Wayne, SVT
Fuquian Wang, LBNL, calibrations
Robert Willson, OSU, SVT
Richard Witt, Kent
Gene Van Buren, UCLA, documentation, tools, QA
Eugene Yamamoto, UCLA, calibrations, cosmics
David Zimmerman, LBNL, Grand Challenge

STAR Software Environment

C++:Fortran ~ 2:1 in Offline

- from ~1:4 in 9/98



Migratory Fortran => C++ software environment central to STAR offline design

In Fortran:

- Simulation, reconstruction

In C++:

- All post-reconstruction physics analysis
- Recent simu, reco codes
- Infrastructure
- Online system (+ Java GUIs)

~75 packages

~7 FTEs over 2 years in core offline

~50 regular developers

~70 regular users (140 total)

QA

Major effort in the past year

- Suite of histograms and other QA measures in continuous use and development
- Automated tools managing production and extraction of QA measures from test and production running
- 'QA signoff' integrated with software release procedures
- Automated web-based management and presentation of results
- Acts as a very effective driver for debugging and development of the software, engaging a lot of people

STAR Offline Framework

STAR Offline Framework must support

- 7+ year investment and experience base in legacy Fortran
  - Developed in a migration-friendly environment – StAF – enforcing IDL-based data structures and component interfaces
- OO/C++ offline software environment for new code
- Migration of legacy code: concurrent interoperability of old and new

11/98 adopted C++/OO framework built over ROOT

- Modular components 'Makers' instantiated in a processing chain progressively build (and 'own') event components
- Automated wrapping supports Fortran and IDL based data structures without change
- Same environment supports reconstruction and physics analysis
- In production since RHIC's second 'Mock Data Challenge', Feb-Mar '99 and used for all STAR offline software and physics analysis

STAR Event Model StEvent

C++/OO first introduced into STAR in physics analysis

- Essentially no legacy post-reconstruction analysis code
- Permitted complete break away from Fortran at the DST

StEvent C++/OO event model developed

- Targeted initially at DST; now being extended upstream to reconstruction and downstream to micro DSTs
- Event model seen by application codes is "generic C++" by design – does not express implementation and persistency choices
  - Developed initially (deliberately) as a purely transient model – no dependencies on ROOT or persistency mechanisms
- Implementation later rewritten using ROOT to provide persistency
  - Gives us a *direct* object store – no separation of transient and persistent data structures – *without* ROOT appearing in the interface

Event Store Design / Implementation Requirements

Flexible partitioning of event components to different streams based on access characteristics

Support for both IDL-defined data structures and an OO event model, compatible with Fortran => C++ migration

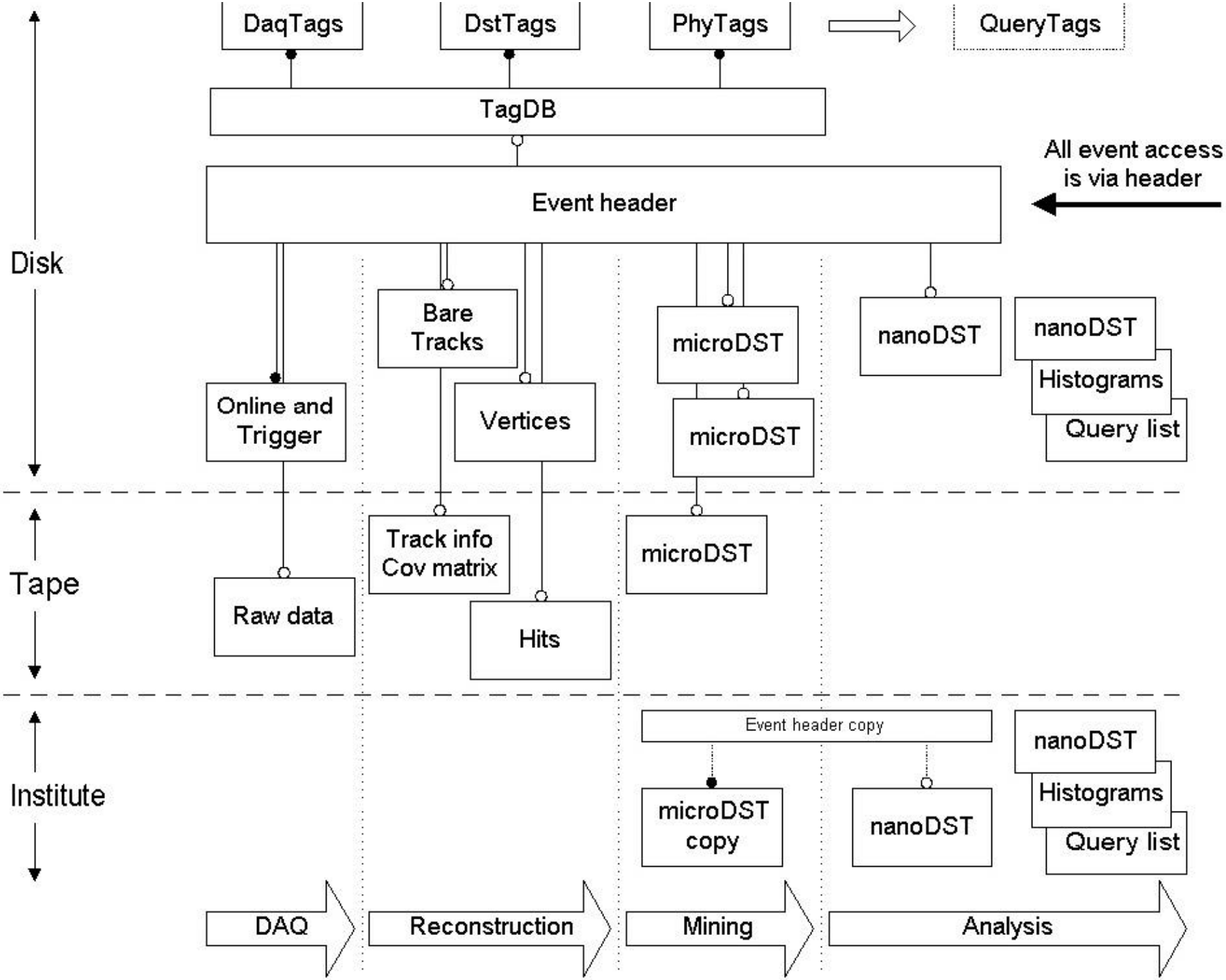
Robust schema evolution (new codes reading old data and vice versa)

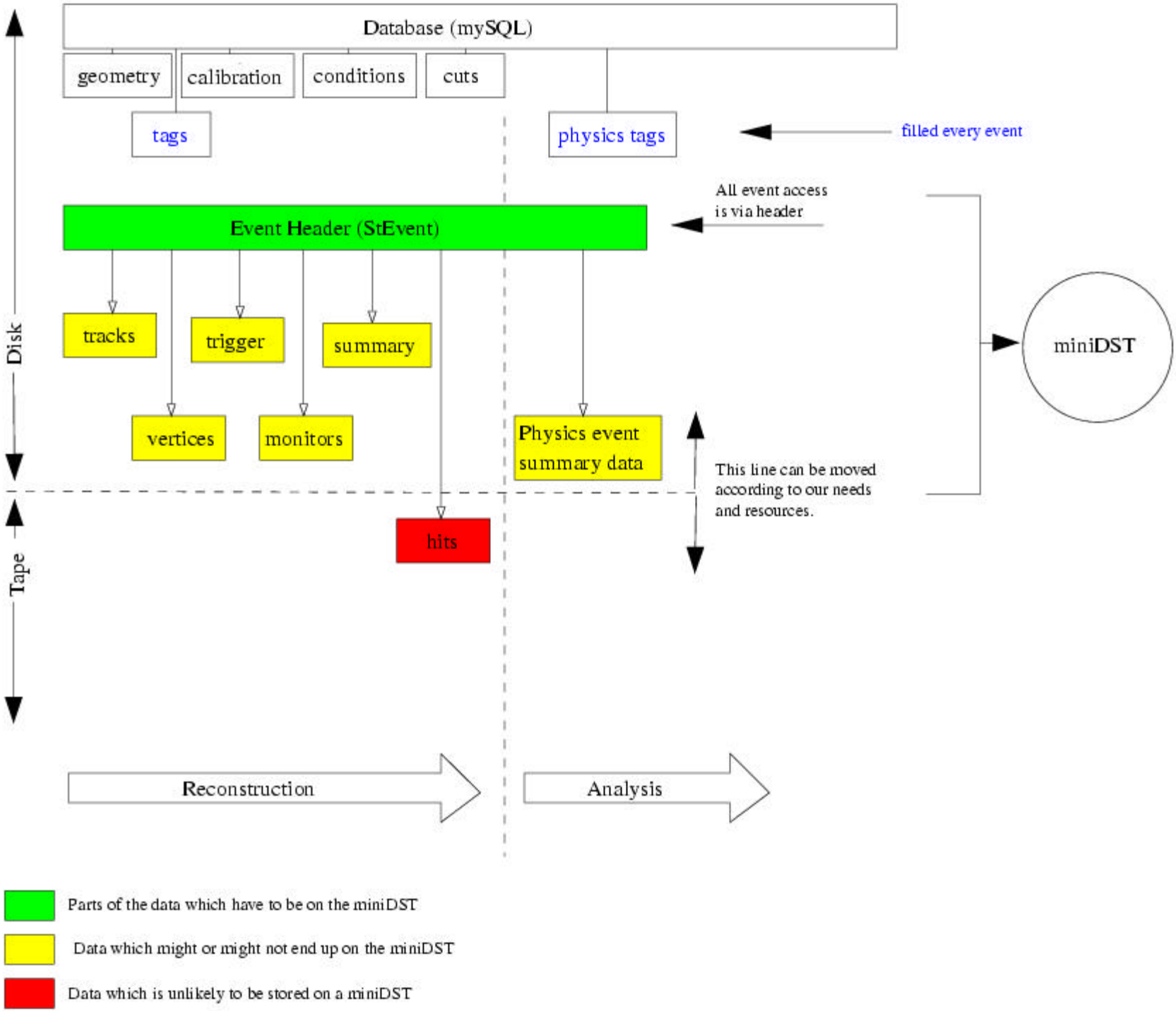
Easy management and navigation; dynamic addition of event components

- Named collections of events, production- and user- defined
- Navigation from from a run/event/component request to the data

No requirement for on-demand access

- Desired event components are specified at start of job, and optimized retrieval of these components is managed for the whole job (Grand Challenge)
- On-demand access not compatible with job-level optimization of event component retrieval


DaqTags
DstTags
PhyTags
QueryTags
TagDB
Event header
All event access is via header
Disk
Bare Tracks
microDST
nanoDST
nanoDST
Histograms
Query list
Online and Trigger
Vertices
microDST
Tape
Track info Cov matrix
microDST
Raw data
Hits
Event header copy
Institute
nanoDST
Histograms
Query list
microDST copy
nanoDST
DAQ
Reconstruction
Mining
Analysis


Database (mySQL)
geometry
calibration
conditions
cuts
tags
physics tags
filled every event
All event access
is via header
Event Header (StEvent)
tracks
trigger
summary
vertices
monitors
Physics event
summary data
hits
miniDST
This line can be moved
according to our needs
and resources.
Disk
Tape
Reconstruction
Analysis
Parts of the data which have to be on the miniDST
Data which might or might not end up on the miniDST
Data which is unlikely to be stored on a miniDST

STAR Event Store Technology Choices

Original (1997 RHIC event store task force) STAR choice: Objectivity

Prototype Objectivity event store and conditions DB deployed Fall '98

- Worked well, BUT growing concerns over Objectivity

Decided to develop and deploy ROOT as DST event store in Mock Data Challenge 2 (Feb-Mar '99) and make a choice

ROOT I/O worked well and selection of ROOT over Objectivity was easy

- Other factors: good ROOT team support; CDF decision to use ROOT I/O

Adoption of ROOT I/O left Objectivity with one event store role remaining to cover: the true 'database' functions

- Navigation to run/collection, event, component, data locality
- Management of dynamic, asynchronous updating of the event store

But Objectivity is overkill for this, so we went shopping…

- with particular attention to Internet-driven tools and open software

and came up with MySQL

Technology Requirements: My version of 1/00 View

Requirement	Obj 97	Obj 00	ROOT 97	ROOT 00
C++ API	OK	OK	OK	OK
Scalability	OK	?	No file mgmt	MySQL
Aggregate I/O	OK	?	OK	OK
HPSS	Planned	OK	No	OK
Integrity, availability	OK	OK	No file mgmt	MySQL
Recovery from lost data	OK	OK	No file mgmt	OK, MySQL
Versions, schema evolve	OK	Your job	Crude	OK
Long term availability	OK?	???	OK?	OK
Access control	OS	Your job	OS	OS, MySQL
Admin tools	OK	Basic	No	MySQL
Recovery of subsets	OK	OK	No file mgmt	OK, MySQL
WAN distribution	OK	Hard	No file mgmt	MySQL
Data locality control	OK	OK	OS	OS, MySQL
Linux	No	~OK	OK	OK

Event Store Characteristics

Flexible partitioning of event components to different streams based on access characteristics

- Data organized as named components resident in different files constituting a ‘file family’
- Successive processing stages add new components

Automatic schema evolution

- New codes reading old data and vice versa

No requirement for on-demand access

- Desired components are specified at start of job
  - permitting optimized retrieval for the whole job
  - using Grand Challenge Architecture
- If additional components found to be needed, event list is output and used as input to new job
- Makes I/O management simpler, fully transparent to user

Features of ROOT-based Event Store

Data organized as named components resident in different files constituting a 'file family'

- Successive processing stages add new components

No separation between transient and persistent data structures

- Transient object interfaces do not express the persistency mechanism, but the object implementations are directly persistent
- Used for our OO/C++ event model StEvent, giving us a direct object store *without* ROOT appearing in the event model interface

Automatic schema evolution implemented

- Extension of existing manual ROOT schema evolution

MySQL as the STAR Database

Relational DB, open software, very fast, widely used on the web

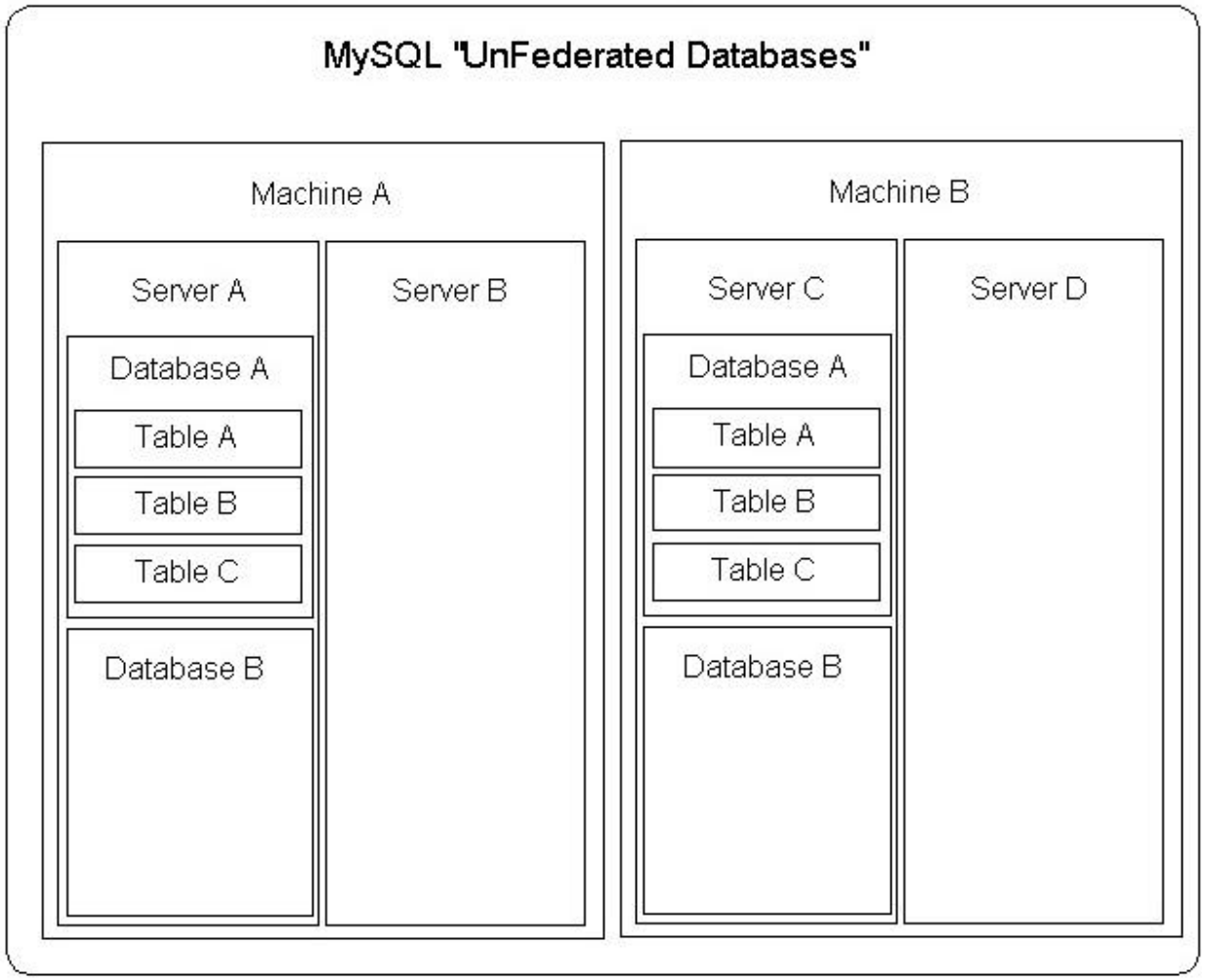
- Not a full featured heavyweight like Oracle
- No transactions, no unwinding based on journalling
- Good balance between feature set and performance for STAR

Development pace is *very* fast with a wide range of tools to use

- Good interfaces to Perl, C/C++, Java
- Easy and powerful web interfacing
- Like a quick protyping tool that is also production capable – for appropriate applications
  - Metadata and compact data

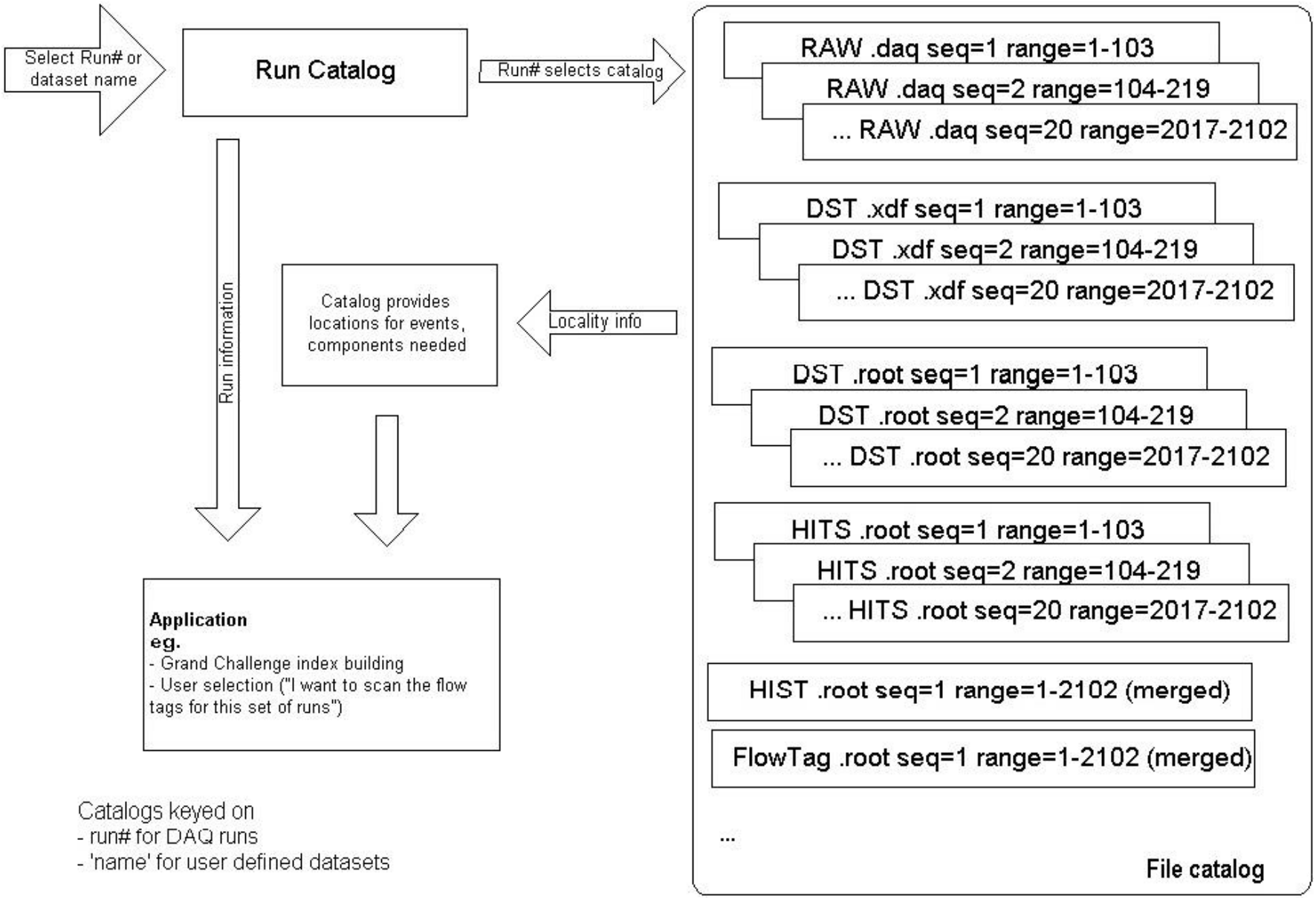
Multiple hosts, servers, databases can be used (concurrently) as needed to address scalability, access and locking characteristics

MySQL Database Organization



Organization of database by machine, server, database, table
- flexibility and adaptibility to load, locking, data volume characteristics to provide scalability, adapting to experience
- organization encapsulated behind DB interface
- can include replication of components and failover

Data lookup via run/file catalogs



Catalogs keyed on
- run# for DAQ runs
- 'name' for user defined datasets

MySQL based DB applications in STAR

File catalogs for simulated and real data

- Catalogues ~22k files, ~10TB of data

Being integrated with Grand Challenge Architecture (GCA)

Production run log used in datataking

Event tag database

- DAQ/Online, Reconstruction, Physics analysis tags
- Good results with preliminary tests of 10M row table, 100bytes/row
  - 140sec for full SQL query, no indexing (70 kHz)

Conditions (constants, geometry, calibrations, configurations) database

Production database

- Job configuration catalog, job logging, QA, I/O file management

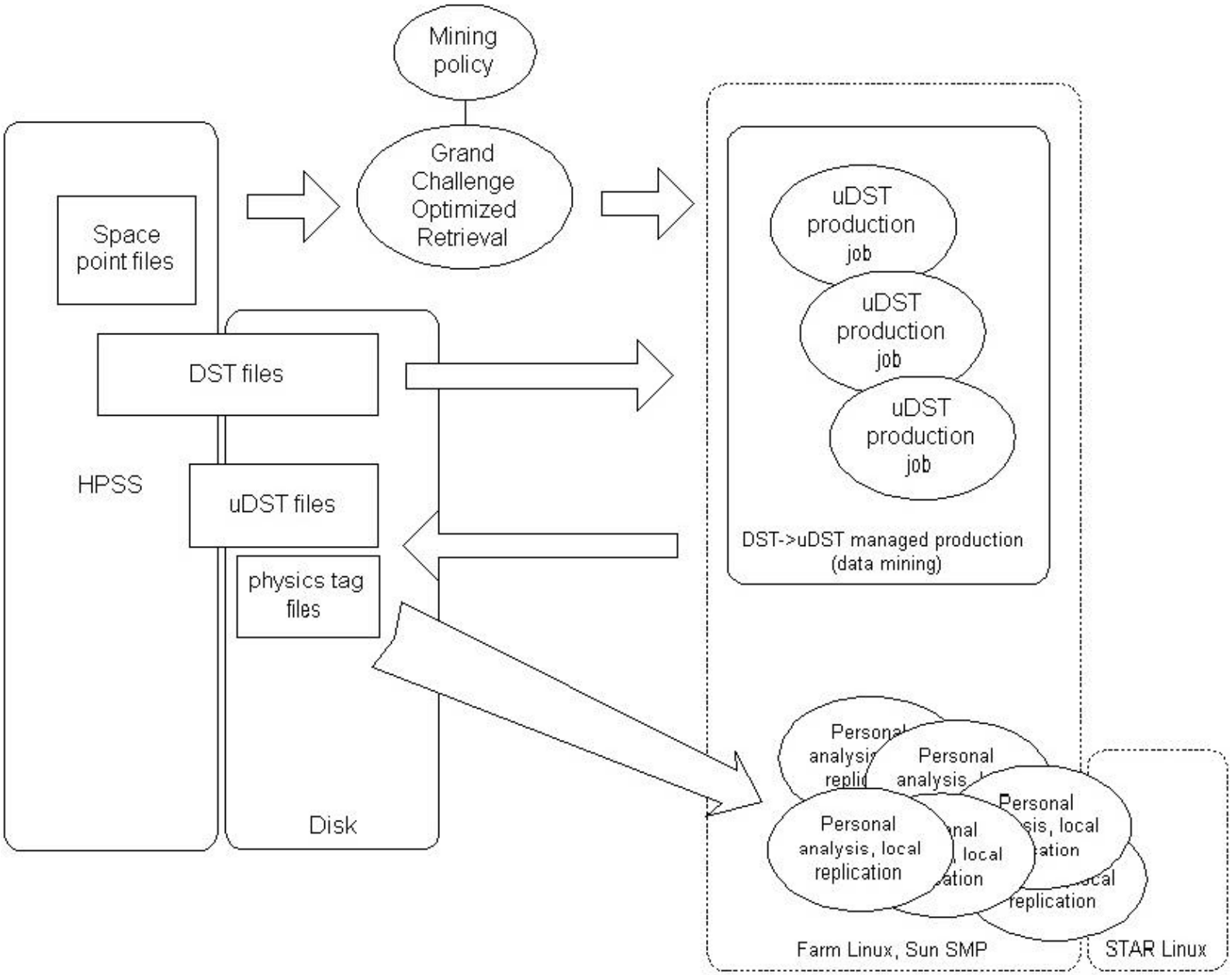
Distributed (LAN or WAN) processing monitoring system

- Monitors STAR analysis facilities at BNL; planned extension to NERSC

Distributed analysis job editing/management system

Web-based browsers

Analysis Operations



HENP Data Management Grand Challenge

What does (will!) the Grand Challenge Architecture (GCA) do for the STAR user?

Optimizes access to HPSS based data store

- Improves data access for individual users
  - Allows event access by query:
    - Present query string to GCA (e.g. NumberLambdas>10)
    - Iterate over events which satisfy query as files are extracted from HPSS
  - Pre-fetches files so that “the next” file is requested from HPSS while you are analyzing the data in your first file
- Coordinates data access among multiple users
  - Coordinates ftp requests so that a tape is staged only once per set of queries which request files on that tape
- General user-level HPSS retrieval tool
  - Can also be used for convenient access to disk-resident data

Grand Challenge queries

Examples of event components:

simu, daq, dst, hits, StrangeTag, FlowTag, StrangeMuDst, …

Mapping from run/event/component to file via the STAR database

GC index assembles tags + component file locations for each event

Tag based query match yields the files requiring retrieval to serve up that event

Event list based queries allow using the GCA for general-purpose coordinated HPSS retrieval

Queries based on physics tag selections:

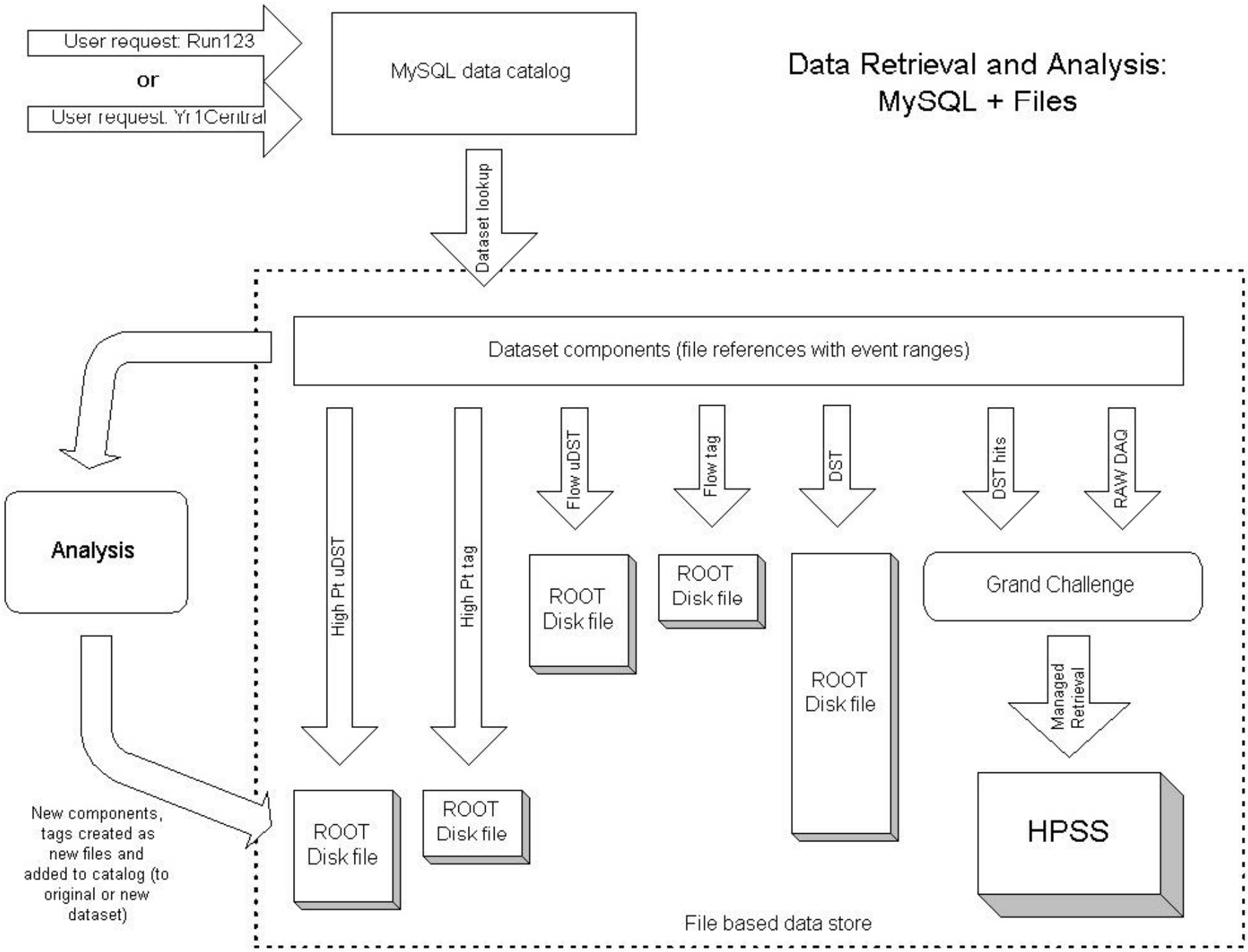
```
SELECT (component1, component2, …)
FROM dataset_name
WHERE
    (predicate_conditions_on_properties)
```

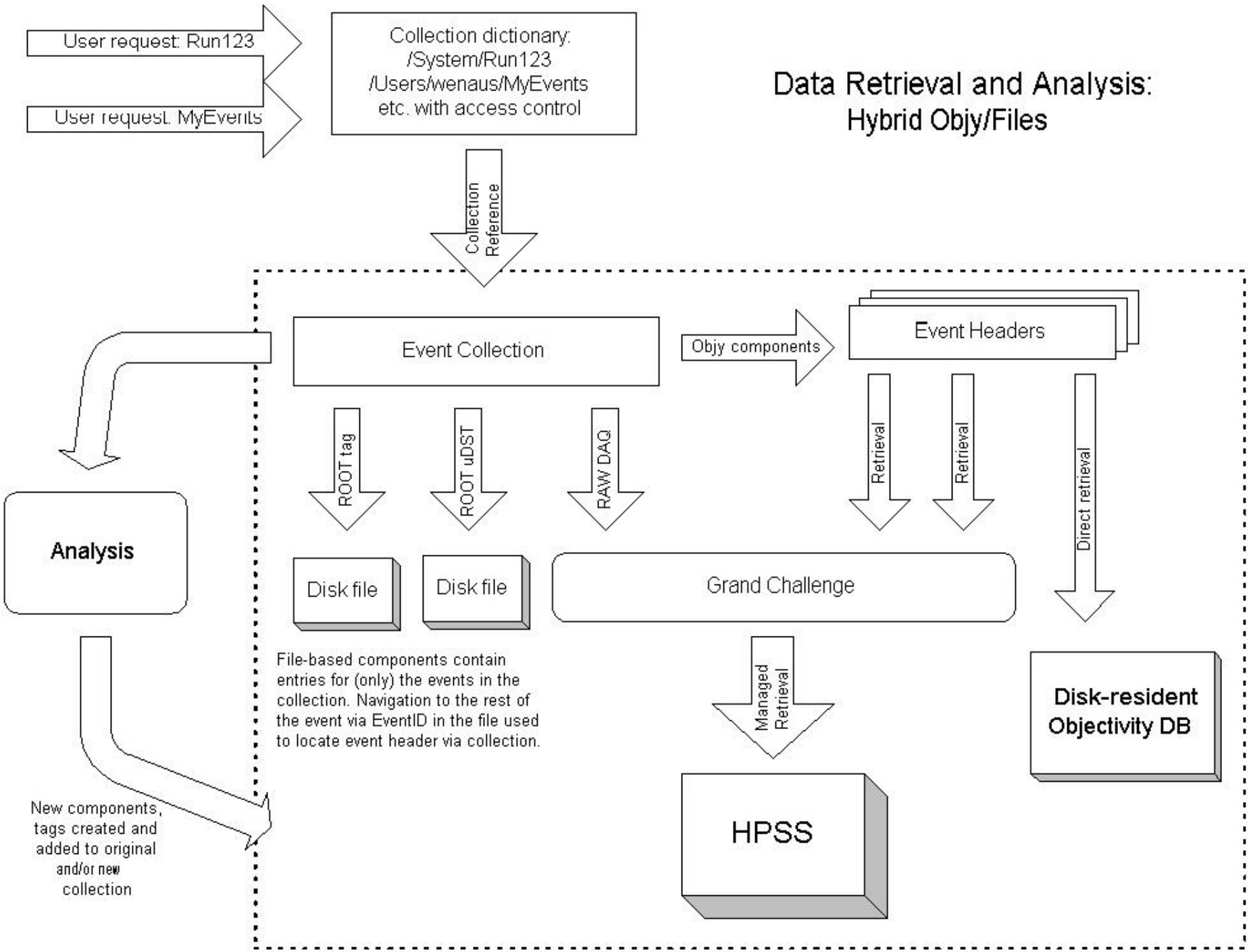
Example:

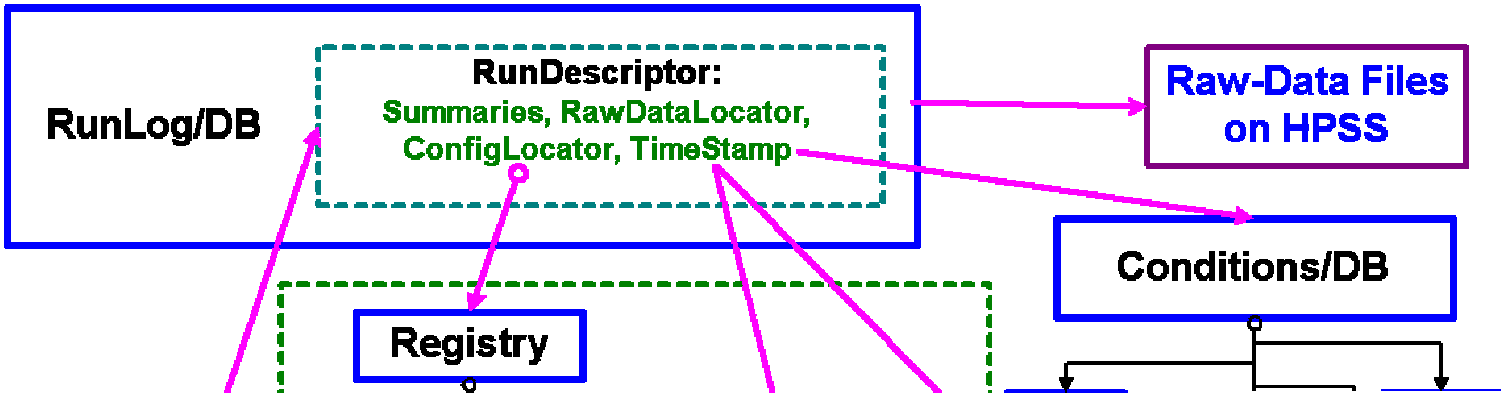
```
SELECT dst, hits
FROM Run00289005
WHERE glb_trk_tot>0 & glb_trk_tot<10
```

Event list based retrieval:

```
SELECT dst, hits
Run 00289005 Event 1
Run 00293002 Event 24
Run 00299001 Event 3
...
```


User request: Run123
or
User request: Yr1Central
MySQL data catalog
Data Retrieval and Analysis:
MySQL + Files
Dataset lookup
Dataset components (file references with event ranges)
Analysis
High Pt uDST
High Pt tag
Flow uDST
Flow tag
DST
DST hits
RAW DAQ
ROOT
Disk file
ROOT
Disk file
ROOT
Disk file
Grand Challenge
Managed
Retrieval
HPSS
ROOT
Disk file
ROOT
Disk file
New components,
tags created as
new files and
added to catalog (to
original or new
dataset)
File based data store


User request: Run123
User request: MyEvents
Collection dictionary:
/System/Run123
/Users/wenaus/MyEvents
etc. with access control
Collection Reference
Data Retrieval and Analysis:
Hybrid Objy/Files
Event Collection
Objy components
Event Headers
ROOT tag
ROOT uDST
RAW DAQ
Retrieval
Retrieval
Direct retrieval
Analysis
Disk file
Disk file
Grand Challenge
File-based components contain entries for (only) the events in the collection. Navigation to the rest of the event via EventID in the file used to locate event header via collection.
Managed Retrieval
Disk-resident Objectivity DB
HPSS
New components, tags created and added to original and/or new collection


RunLog/DB
RunDescriptor:
Summaries, RawDataLocator,
ConfigLocator, TimeStamp
Raw-Data Files
on HPSS
Conditions/DB
Registry

Current Status

Offline software infrastructure and applications are operational in production and 'ready' to receive year 1 physics data

- 'Ready' in quotes: there is much essential work still under way
  - Tuning and ongoing development in reconstruction, physics analysis software, database integration
  - Data mining and analysis operations infrastructure
    - Grand Challenge being deployed now
  - Data management infrastructure

Simulation and reconstruction production operations are now routine

- >10TB simulation and DST data in HPSS
- 6TB simulated data produced in '98-'99; 2TB in Jan-Mar '00
  - 7 event generators; 6 production sites
- ~2TB reconstructed data produced over last 6 months
- Production automation and cataloguing systems based on scripts and MySQL

Current Status (2)

Successful production at year 1 throughput levels in recent 'mini Mock Data Challenge' exercises

Final pre-data Mock Data Challenge just concluded

- Stress tested analysis software and infrastructure, DST and uDST production and analysis, RCF analysis facilities
- Wrap-up meeting yesterday which I missed, so I won't attempt to summarize!