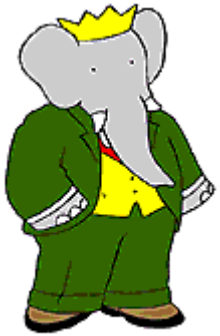
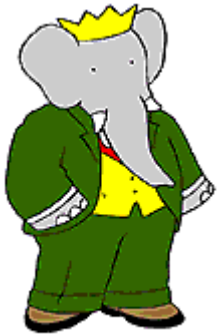


BaBar Bookkeeping

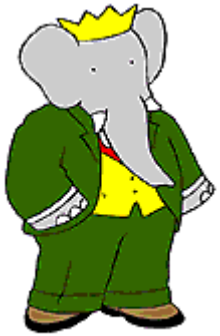
Douglas A. Smith
Stanford Linear Accelerator Center
Tim Adye
Rutherford Appleton Laboratory
Wilhelm Roethel
University of California, Irvine
David Hutchcroft
University of Liverpool

CHEP 2006
Feb 15th, 2006, TIFR, Mumbai


TM & © Nelvana

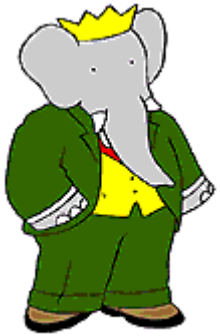
BaBar Computing Model 2 (CM2)

- In 2003 it was decided that BaBar's initial computing model was not going to scale for the whole experiment. An effort to create a new computing model was started, called CM2.
- Resulting in changes to the event store, event structure, and included the need for a new bookkeeping system.
- Initial report on the Bookkeeping: D. Smith, *et al.*, Talk 338, CHEP 2004.
- Other talks at CHEP 2006 about changes due to CM2:
  - Simulation production – D. Smith, Talk 299
  - Condition database – D. Smith, I. Gaponenko, Talk 352


TM & © Nelvana

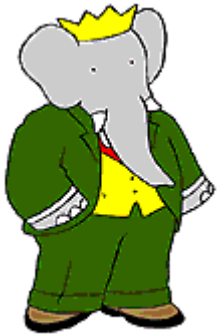
Overview of Bookkeeping

- CM2 event store is in ROOT files.
- Different event components can be placed in separate files.
- Related files of many events called “Collections”, these are the unique elements of the event store.
- Collections are organized in lists, called “Datasets” for use, and there is a *n* to *m* relation between them.
- Also the relation between data run number and collection is kept in a *n* to *m* relation.

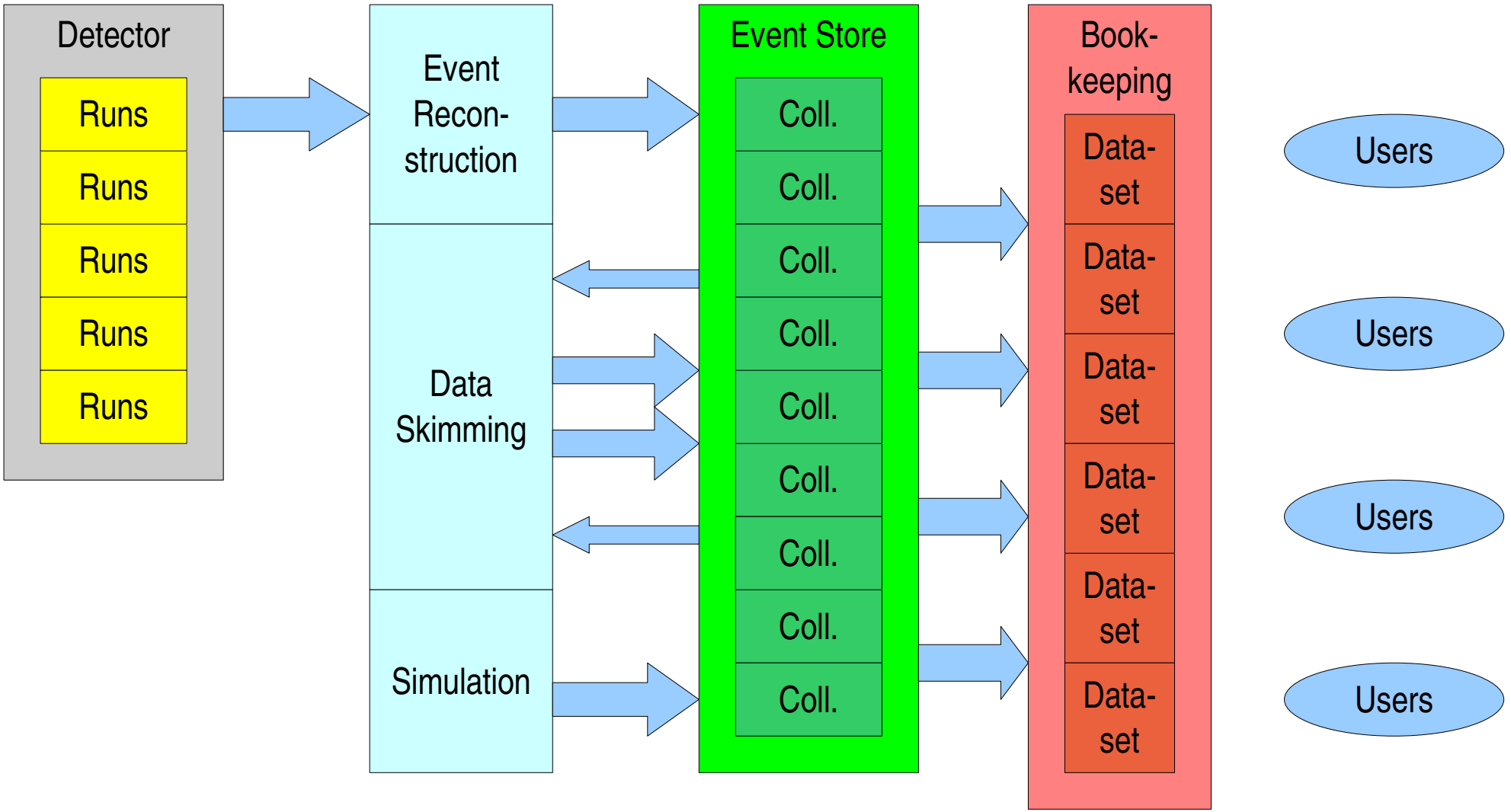

TM & © Nelvana

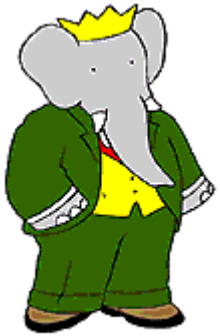
Overview of implementation

- All information for bookkeeping stored in a relational database.
- Use of SQL Abstraction keeps code DB independent, and allows DB schema changes to be independent from code.
- Currently MySQL and Oracle are supported.
- The system was implemented as a software framework in OO-Perl, providing object classes and command line interfaces.



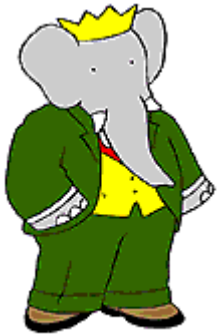
Cartoon of data management




TM & © Nelvana

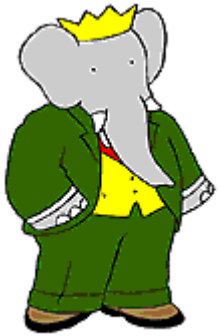
Dataset evolution

- Datasets need to change with the changing state of data.
  - New data becomes available.
  - Changes in quality checks, removes and adds data.
  - Reprocessing, old versions of data shouldn't be used.
- Need to know what happened: Dataset history:
  - The record of the changes to the datasets are kept.
  - At any point the datasets at any state in the past can be selected
- Need stability in the place of change: Dataset tagging:
  - Place a tag on a dataset and give it a name.
  - People can select this state of the dataset at any time.


TM & © Nelvana

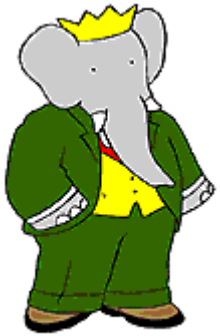
Dataset updates

- Data analysis needs data quickly as it is available, datasets must evolve quickly to support this.
  - Initial datasets were static lists, hard to keep updated.
  - Next classes of datasets were recreated periodically. Updates happened daily, or every other day.
- Changes to collection lists and status are monitored, changes are applied to affected datasets.
  - New collections go into datasets, new datasets are created.
  - Collections not to be used are removed.
  - Changes applied every 10 mins now.



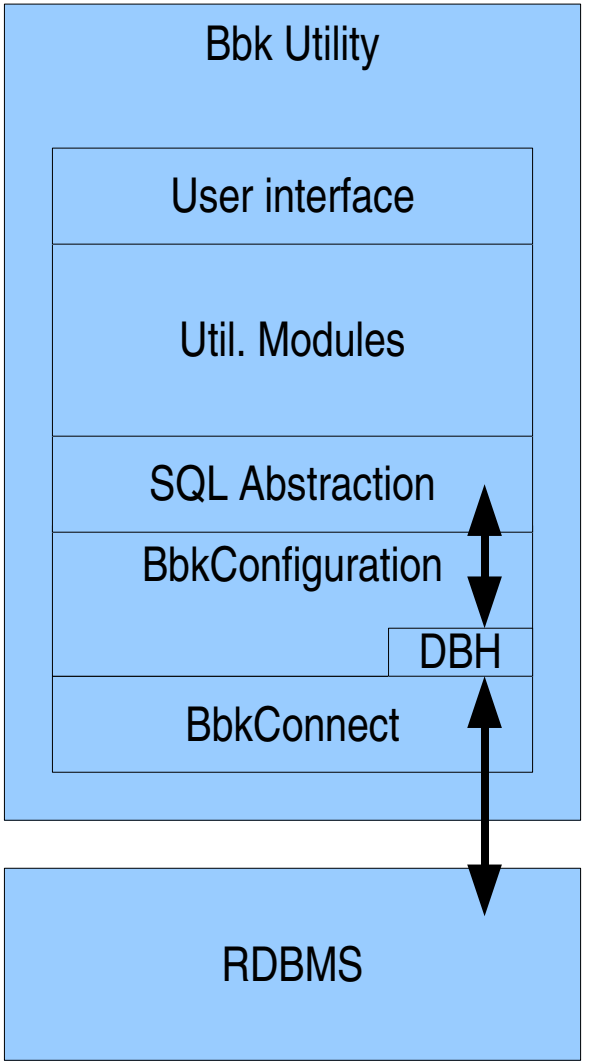
Size of current use

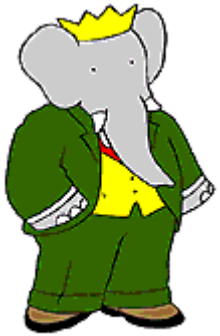
- Event store is 660.4 TB with 1.1M files.
- The Bookkeeping system keeps track of this as 751k collections.
- The collections are organized into 60.7k datasets.
- Size of database is ~2 GB. Can be downloaded as compressed daily snapshot of ~190 MB.
- Selection of information in any dataset done in less than 2 secs.
- Will scale fine to future size of experiment.



Design of a Bookkeeping utility

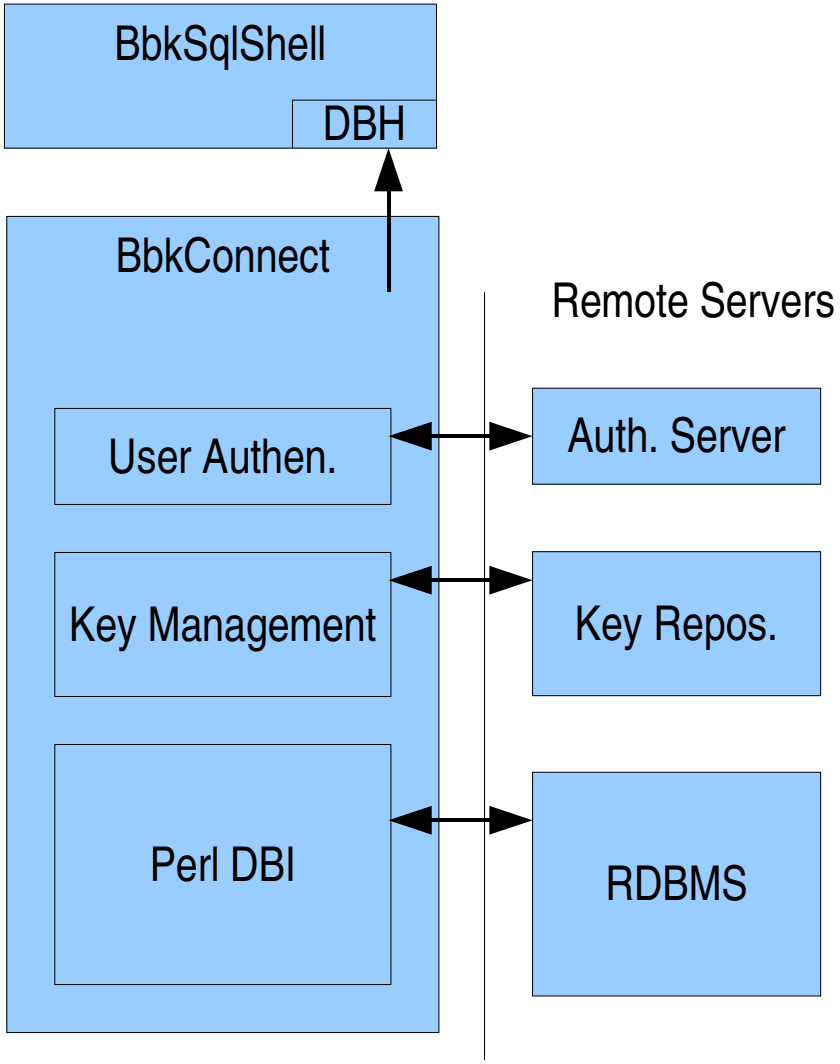
- Utilities built from modular design
  - All information kept in RDB
  - Connection module
  - Configuration module
  - SQL Abstraction module
  - Utility code modules
  - User interface

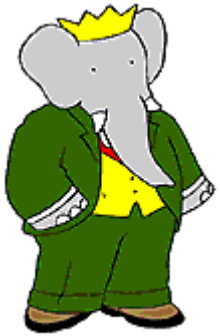




BbkConnect

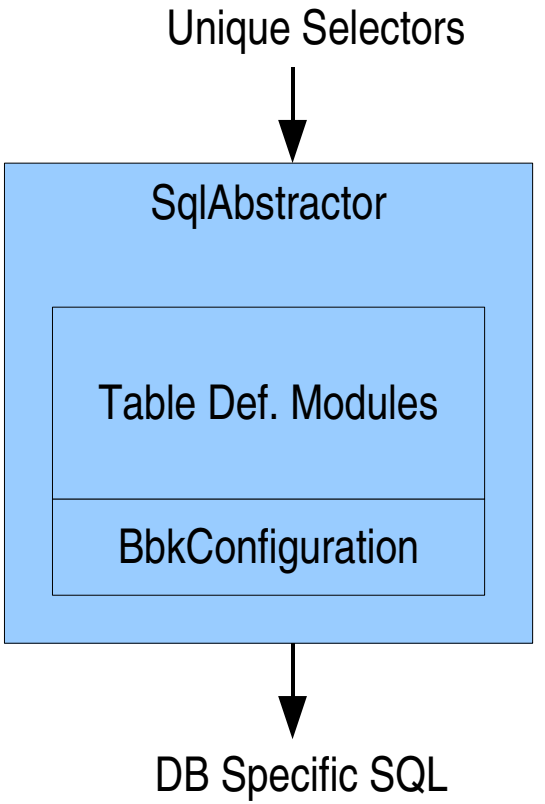
- Database connection manager
  - independent of bookkeeping DB.
  - Distributed user authentication.
  - Central key distribution.
  - Default settings control.
  - Network database connections.
- Defined now for all RDBMS in BaBar not just bookkeeping system.

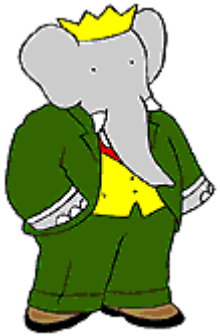




SqlAbstractor

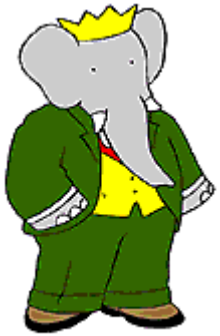
- Abstracts SQL to constraints and selectors
  - All tables and fields given unique labels.
  - Code selects on unique labels, and constrains selects on unique labels.
  - Abstractor class creates SQL for RDB including all needed joins between tables.
- Used for Bookkeeping, Task Management, and Simulation Production DBs.
- Code ready to go to CPAN.





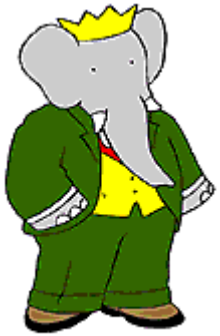
Task Manager

- Skim and analysis data production management:
  - define tasks to apply to datasets.
  - Tasks are divided into jobs, and provides job management with batch system.
  - Output of tasks managed as collections to be placed back into bookkeeping system.
  - Currently used to manage data skimming, many tasks currently defined, managed millions of jobs (approaching 20M), produces hundreds of thousands of collections.



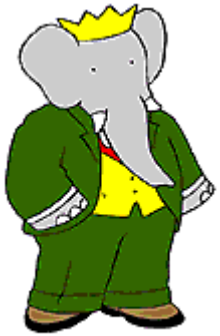
User applications

- BbkDatasetTcl
  - used to create analysis job control file (Tcl) with collections for analysis jobs.
- BbkLumi
  - determines luminosity information for datasets.
- BbkUser
  - Exposes all SQL abstraction selectors to command line.
- BbkExpertTcl
  - Create control tcls bases on selectors from SQL abstraction.



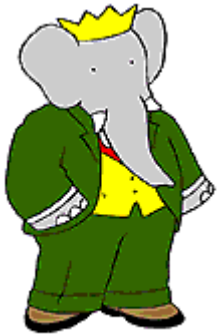
Multiple database design

- Utilities can be used to connect to any number of datasets.
  - For scaling to data size, new reprocessing go into new datasets:
    - First full reprocessing in CM2 with release 14 – bbkr14
    - Next full reprocessing with release 18 – bbkr18
  - For scaling to distributed use, the databases are mirrored:
    - Updates to database done to master in SLAC.
    - Updates mirrored to remote sites.
- Default management system can control use of utility:
  - Analysis with release 18 gets bbkr18.
  - Analysis at remote site gets local mirrored database.


TM & © Nelvana

Data distribution

- System includes data distribution utilities:
  - Distribution controlled by datasets.
  - From large sites with over 100TB to laptops with 10GB.
- Includes data management.
  - Datasets can be removed.
  - If datasets change, collections no longer in dataset removed.
- Records which datasets are local:
  - In local bookkeeping databases, information on which files, collections, and datasets hosted are kept.
  - Local user tools use this, to provide only what exists.



System is working well

- Bookkeeping system has been in use for about 2 years now.
- Manages all meta-data of BaBar event store, and scales well
- Simplifies management of data for most analysis into only a few dataset names.
- For the developer set up a framework with rich set of modules for management of meta-data.
- For the user a small set of easy to use utilities.
- Development continues, to cover more user's needs, and make it still easier for analysis and production management of data.