SAM plans and remote access

Vicky White

for the SAM team

Lee Lueking, Vicky White, Heidi Schellman, Igor Terekhov, Matt Vranicar, Julie Trumbo, Rich Wellner, Steve White, Sinisa Veseli

The D0 Workshop on Software and Data Analysis

Praha, September 23-25, 1999

Outline

- SAM V1.0
  - *with SAM Manager - a framework package, integrated with d0om, and d0reco*
- Future SAM releases and features
- SAM and Databases - - the design and its effect on portability and remote access
- Using SAM remotely or locally

SAM Versions and Feature

- For the most up-to-date list see sam development web page at http://d0db-dev.fnal.gov/sam

Done In progress To do

Version 1.0

- SAM manager integrated in D0 Framework, with RCP and input options passed on command line
- V0 of Event Catalog and primitive web browser for Raw data entries
- Support for RIP/online data logger
- File Storage Server for RAW, MC and reconstructed data
- Preferred locations to fetch files
- Restrictions on number of parallel file transfers per buffer
- Python scripts for launching user applications
- sam 'project' tools with GUI on web
- User Guide and internal docs
- test multiple i/o pipes and projects with enstore on d0test

Version 1.0

- SAM manager integrated in D0 Framework, with RCP and input options passed on command line
- V0 of Event Catalog and primitive web browser for Raw data entries
- Support for RIP/online data logger
- File Storage Server for RAW, MC and reconstructed data
- Preferred locations to fetch files
- Restrictions on number of parallel file transfers per buffer
- Python scripts for launching user applications
- sam 'project' tools with GUI on web
- User Guide and internal docs
- test multiple i/o pipes and projects with enstore on d0test

SAM/Franework integration

- SAM (from user perspective) is just a few useful commands
  - *all are available on the command line*
  - *a few from a web-GUI (define project etc.)*
  - *some (more later) will be available in V1.0 from within your d0reco or other d0 framework program*

SAM user commands

sam create project definition < *defin. params*>

sam create project snapshot <*project params*>

sam create analysis project <*project params*>

sam verify snapshot <*snap params* >

sam verify project <*project params* >

+

sam translate constraints <*data constraints*>

sam resolve query <*sql params*>

SAM user commands

sam start project <…>

sam start consumer <…>

sam start process <…>

sam get next file <…>

sam release < *file params…*>

sam store <*file and file metadata params…*>

sam declare <*file and file metadata params..*>

sam stop project <…>

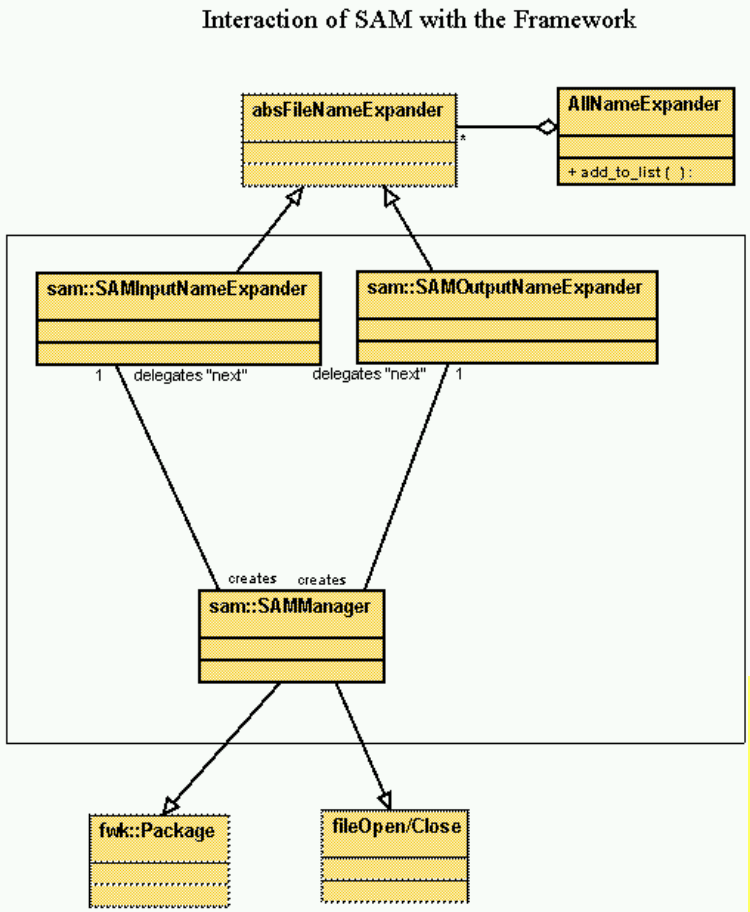
…and others to dump, suspend,resume, etc.

SAM commands available in framework (in V1.0)

- sam start consumer
- sam start process
- sam get next file
- sam release *<file params>*
- sam store *<file and metadata params>*

  – *more in next version ...*

SAMManager and Framework and d0om



SAM interaction through

a) name expanders - used by d0StreamName

b) File Open/Close messages generated by ReadEvent and WriteEvent

sam: in file name will be resolved by a SAM name expander --> SAM Servers to get next file, or get place/name for output file

Note on Name Expanders

- AllNameExpander -- tries all known expanders in turn
- FatmenNameExpander - run I fatmen names
- FileNameExpander - generic environment variables and BSD file name globbing
- ListFileExpander - listfile:file_name with wildcard
- SAM name expander sam:
- will add more e.g. for making output file name from input file(s) name

SAM and Framework

- At file open/close SAM Manager called to
  - *release input file*
  - *keep statistics and file parentage*
  - *write out file meta-data for output file*
  - *initiate sam store of output file*

- SAM Manager at initialization deals with attaching to a project, starting up consumer and process for you… more in the future

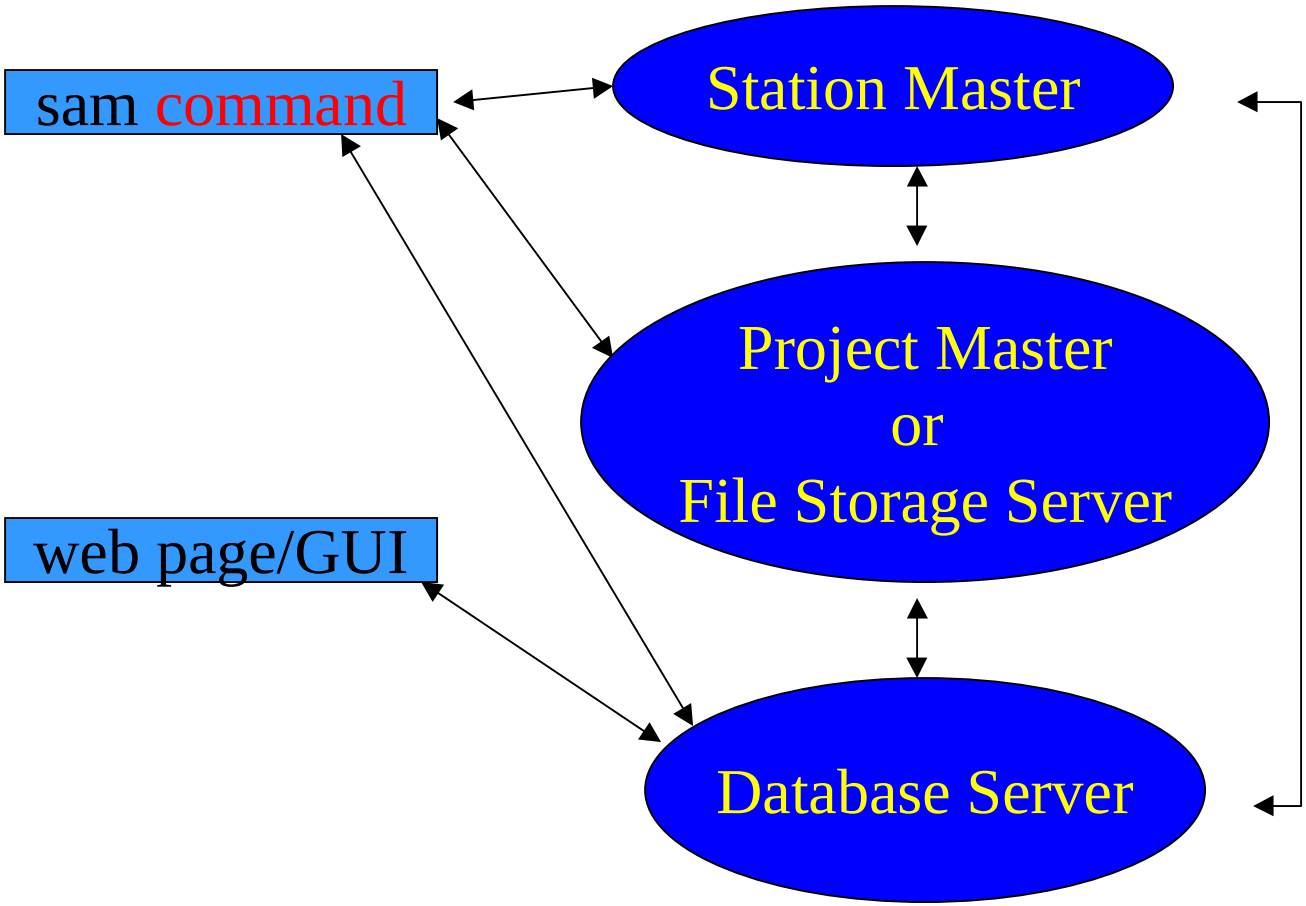
SAM command and Servers

- The sam commands are all implemented as
  - *sam python scripts*
  - *executables called from sam shell script*
  - *C++ SAMManager framework package*
- They will build/run an any machine supported by D0, with D0 release, + installation of standard Fermilab/kits products. (eventually, today linux,irix)
  - *python, orbacus, fnorb*

SAM Servers

- sam user commands talk to SAM Servers
  - *exchange small amounts of information*
- Servers can be anywhere on the network (including locally, or on the same machine)
- Don't be afraid … Servers are everywhere
  - *ftp, mail, telnet, http, nfs, etc. etc.*
- The SAM system is built to run in a fully distributed environment
  - *flexibility for where the parts run*
  - *interchangeable components*

SAM command -> Servers

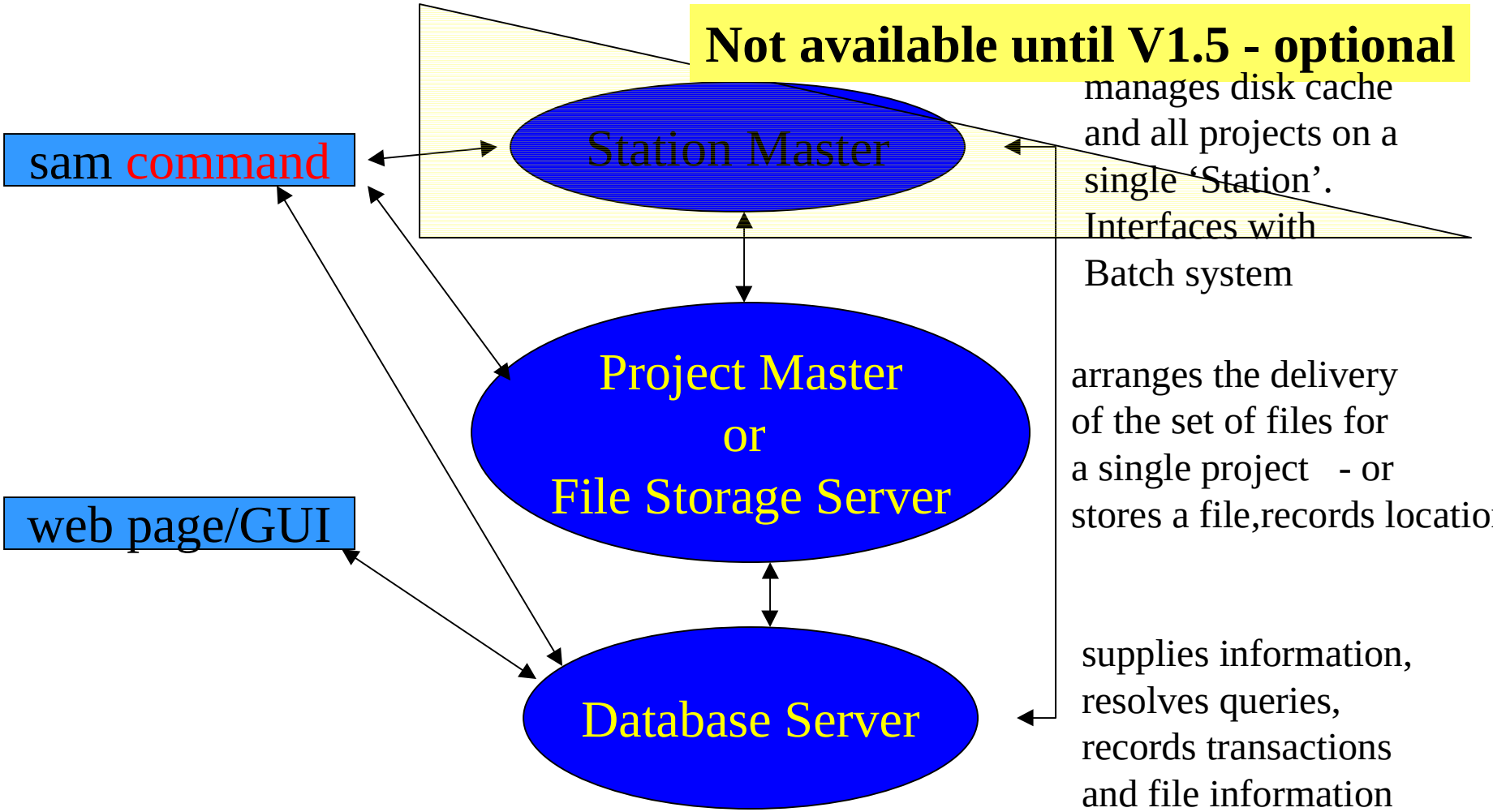


manages disk cache and all projects on a single 'Station'. Interfaces with Batch system

arranges the delivery of the set of files for a single project - or stores a file,records location

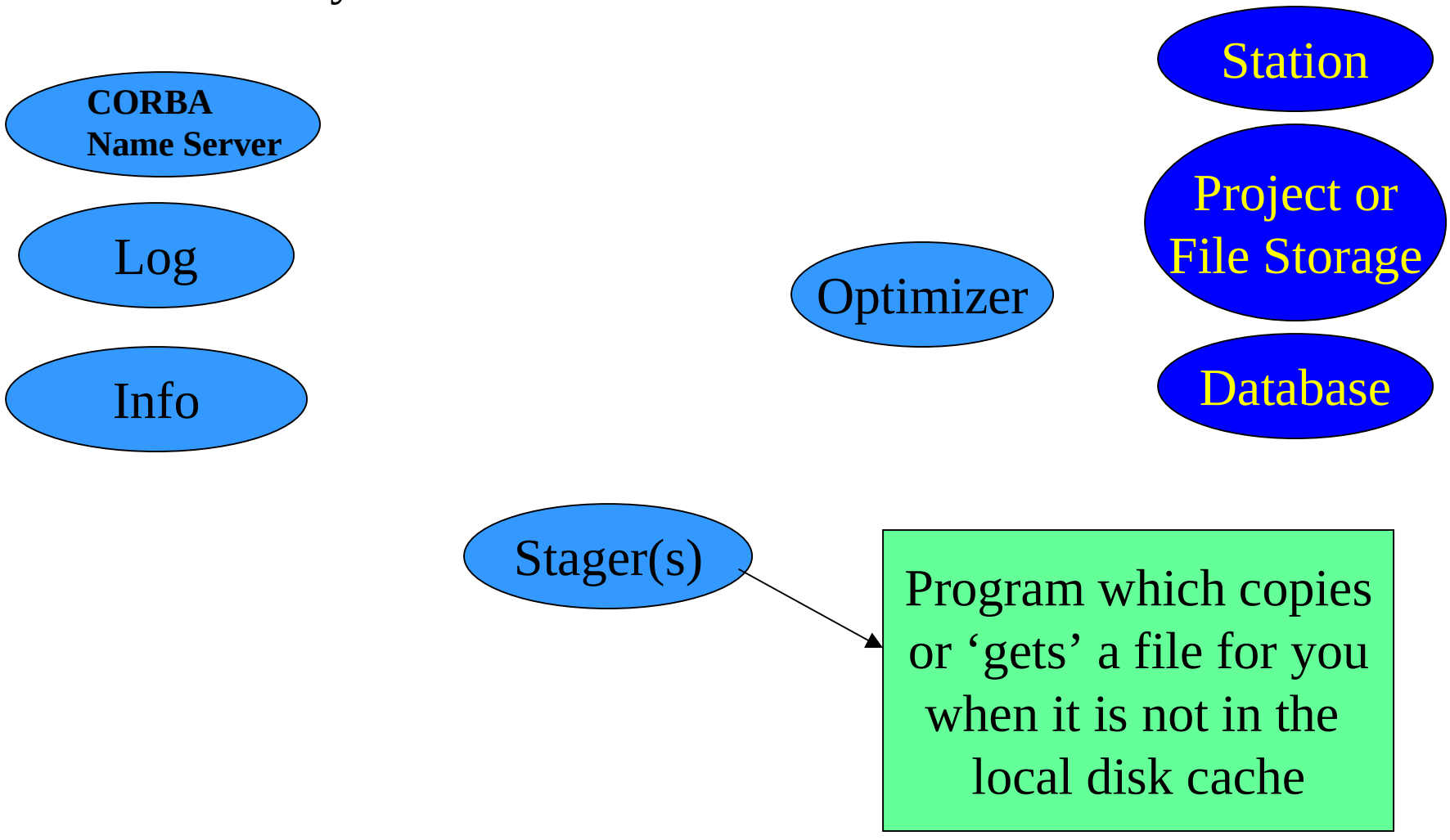
supplies information, resolves queries, records transactions and file information

SAM command -> Servers

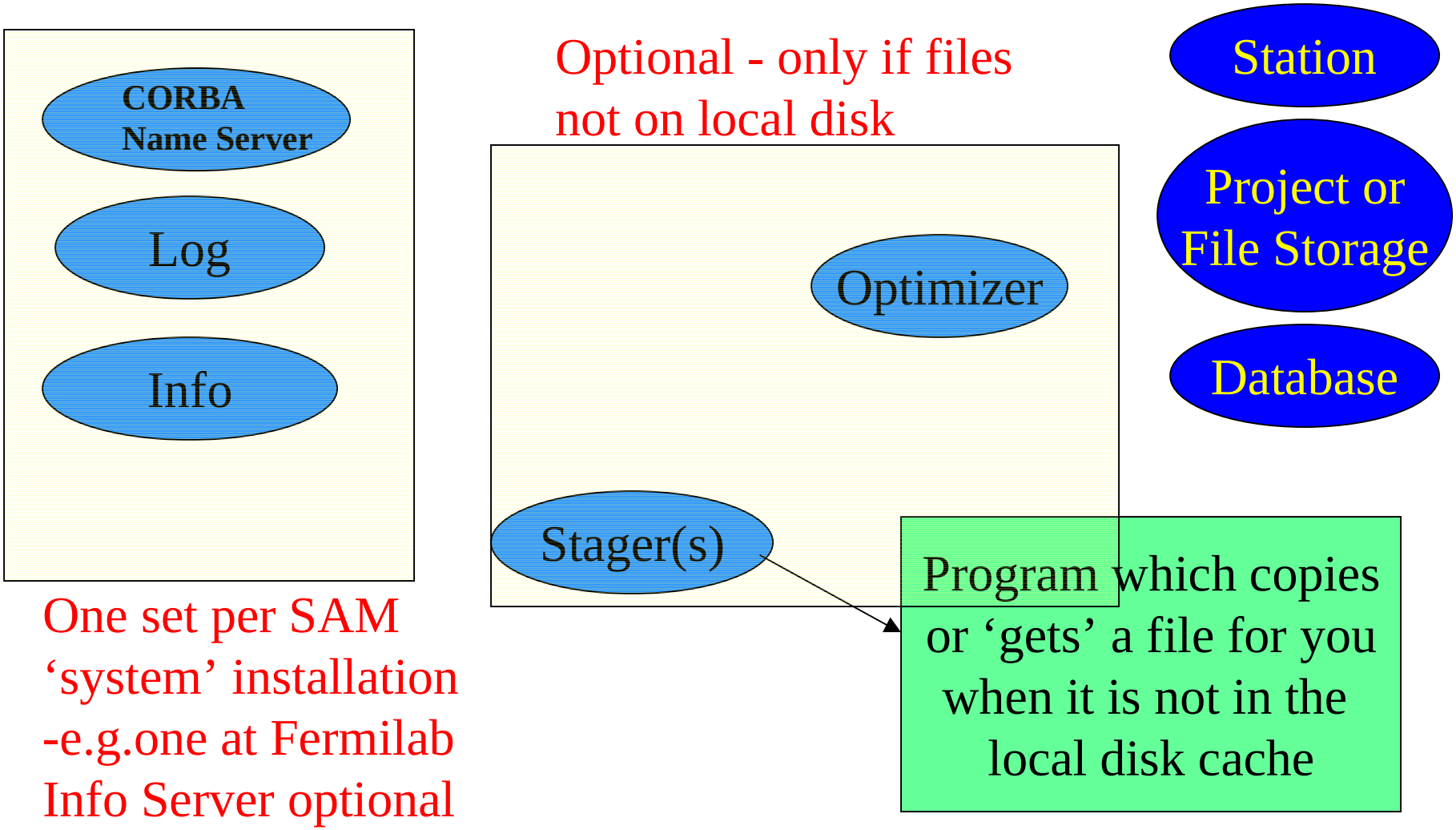


More of the Server story...

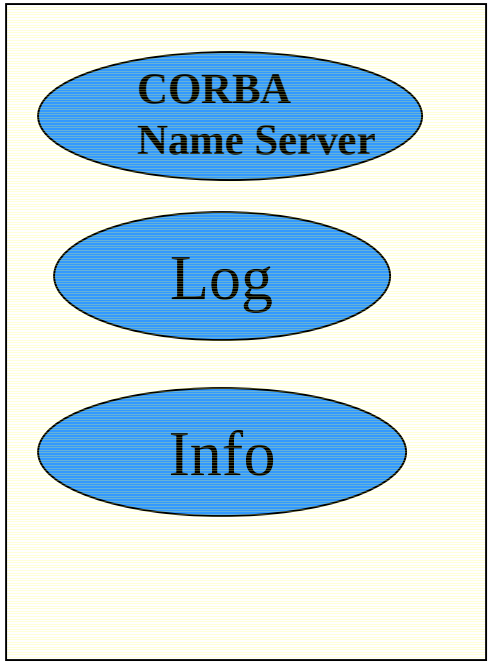
The servers rely on other servers behind the scenes ...



More of the Server story...

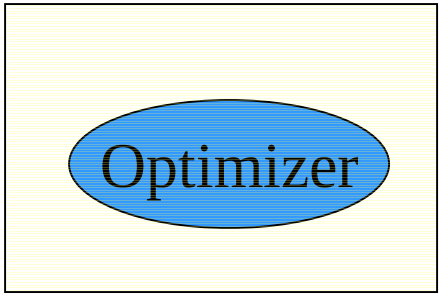


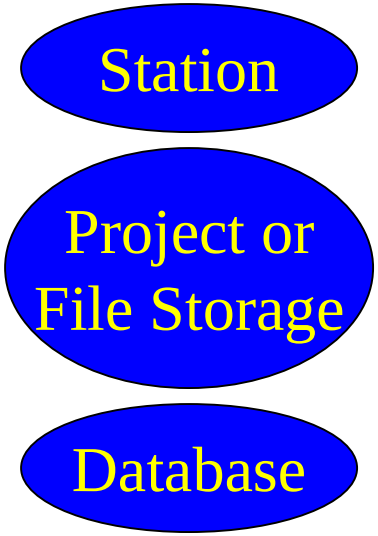
More of the Server story...



Somewhere -on the network

always optional





Stager(s)

If need to stage files - must run on a machine with access to the local disk cache

Program to copy files
i) encp (Enstore)
ii) 'ftp' or rcp
iii) your local way of staging files

V1.0 sam commands - improvements

- Early-bird users caught the worm (ugh!) - had to type commands to start up some of the Servers and the Stagers (if needed)
- Usually want to do a whole bunch of sam commands in sequence - passing info from one to the other … inconvenient, messy
  - *now - many commands inside your program*
  - *now - Python script wrapper with places to put*
    - **your parameters and options**
    - **your executable**

SAM Diagnostics PageSAM Diagnostics Page - Netscape

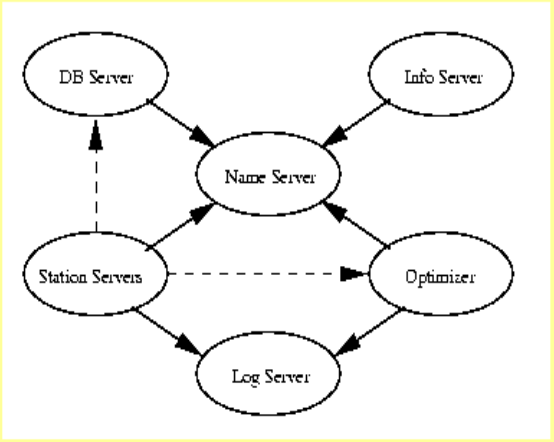
File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Location: http://d0db-dev/sam/diagnostics.html What's Related

SAM Diagnostics

Server Restart Dependencies



Whenever one sam server is restarted, it may be also be necessaty to restart other sam servers. This chart shows the "other restart" requirements for all the sam servers. An arrow going from server x to server y indicates that server x is dependent on server y. A dashed line specifies a dependency relationship which will be eliminated in the future.

How to get Information from Sam Servers

Document: Done

Start 7:30 PM

Version 1.5 - Dec, 1999

- fixes for early users and for online data logger + urgent missing features
- Station Servers with disk cache management
- enhance sam 'project' tools
  - *verify, delta,union &differ*
  - *project restart and continuous projects*
- use of multi-threaded framework to work with d0omCORBA (for calibration)
- enhanced sam test harness (systemwide testing)
- enhanced system monitoring and administrative tools
- start of full system stress tests - 200MB/sec in/out robot
- ….. Continued….

Version 1.5 - Dec, 1999 (cont)

- full MC meta-data creation mechanisms
- simplified luminosity accounting - MC only
- MC import facility and server, with documented process
- Tape injest (Enstore) + sync with SAM database
- start of Batch system integration and Resource Management design for Station Servers

Version 2.0 - March 2000

Enable cosmic ray commissioning

- fixes to V1.5 + urgent features
- Farms/File merge (i/o node integration)
- Station with batch system interface and i/o resource management
- Multi-connection robust Database Server
- Error and robustness features
- Full scale system tests and simulated database size and performance tests
- network interface balancing (with Enstore)
- design of Luminosity Manager/database/processes
- design of PickEvents subsystem and full Event Catalog(s)

Version 3 - April/May 2000

- fixes to V2 + urgent missing features
- implementation of luminosity accounting
- start of Thumbnail data design and access
- other features …. TBD

Version 4 - June/July 2000

Ready for Data Taking (almost)

- features --- TBD

Version 5 - Aug/Sep 2000

PickEvents and Thumbnail data services

- other features --- TBD

Version 6 - Nov/Dec 2000

Support for Remote sites +

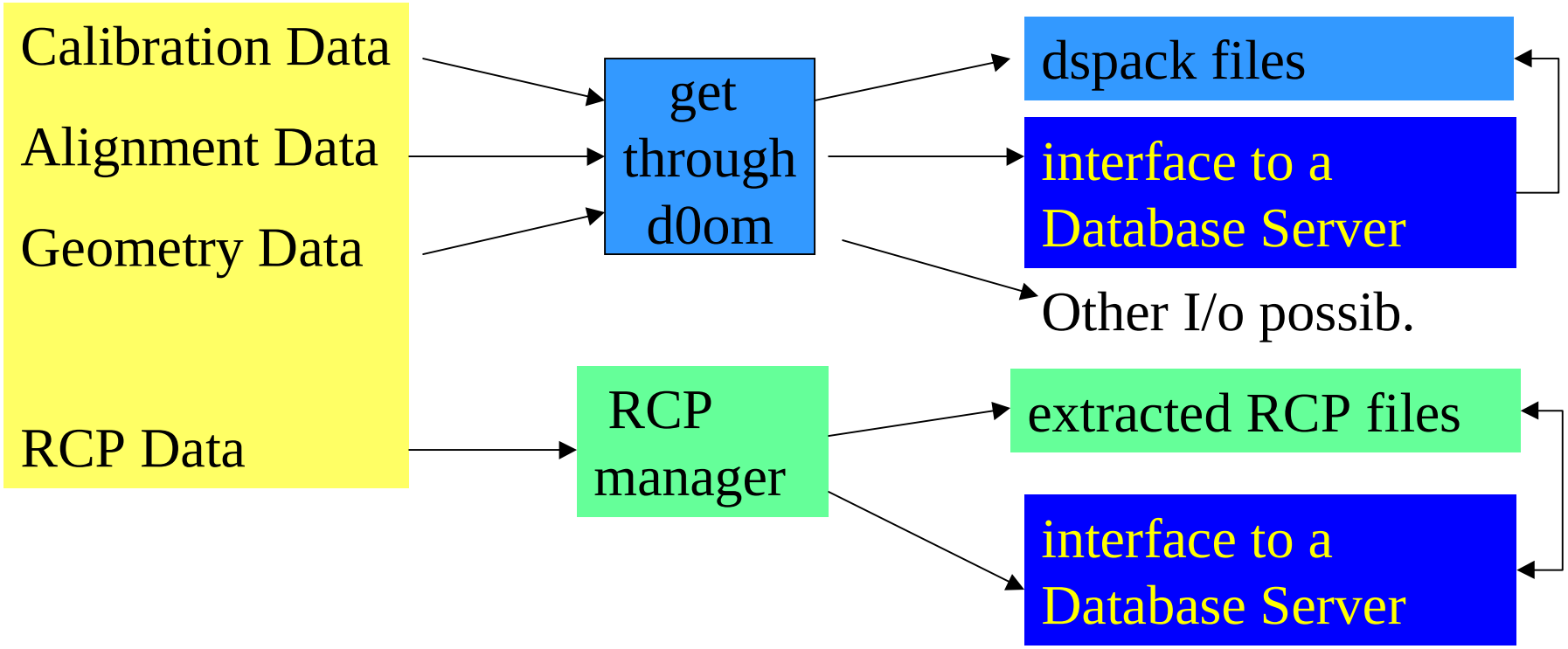
- Other features --- TBD

Remaining Features list

- **Use of Logical Streams in db and project definitions and interface with trigger list**
- **File staging algorithms for sample across logical stream**
- **PickEvent access mode (involves D0 framework i/o packages)**
- **Event catalog for PickEvents support and all data tiers (not just RAW)**
- **PickEvents Server**
- **Luminosity data in database and D0 framework**
- **Export of physics data to remote institutions - server**
- **Export of meta-data to remote institutions + synch of remote meta-data**
- **SAM running at remote institutions, including database extract and synch**
- **Thumbnail data design, file format, and access strategy**
- **Import of Run I metadata and access to Run I data via SAM**
- **Prompt (and on-demand) Reconstruction Pipeline**
- **Summary reports and informational tools for Physics use**
- **Network interfaces balancing, in conjunction with Enstore**
- **ROOT objects and file format? - - implications**
- **Online databases upload and synch of data (with help from Support Databases)**
- **Database monitoring tools (with help from Support Databases)**
- **??? things we forgot**

Analysis outside Fermilab, using SAM

- In addition to your program, which must talk to a SAM Project Server and Database Server somewhere, and may need to have files staged, you will need



D0om and deferred I/O

- D0om has extremely smart (brilliant) pointers for objects stored in a database
  - *may defer fetching data from database until that part of the sub-tree of data is referenced*

Physics Data and Database Data

Physics Data - store and manage locally or fetch across network from Fermilab and cache locally?

- few events
- few files
- large dataset

Database Data - create local database or interact across network with d0 central database? Cache results locally if network down?

- information
- transactions
- substantial data e.g. calibration data

Database knows all!

The central database keeps excellent track of the correlation between “Physics Data” and “Database Data”.

- *e.g. each time period of a particular set of calibration constants forms a ‘tree’ of data - precisely tracked in database*
- *lineage and meta-data for every file is known*

This will make export of a subset of Physics Data and ALL of the related calibration, geometry, RCP, etc. possible

--- we have to worry only about overloading the db machine

Access to data and databases can be configured many ways

- depends where, and which, Servers run
- depends if physics data comes over network or on tape
- depends if you cache all data locally on disk or have to keep fetching from tape locally
- depends if you have a local extracted database or not

Any combination is possible…

Physics Data files - over network

If few events/files

- *Use a workgroup cluster at Fermilab to run a Project to pre-stage files from robot for you/cache them on disk. (we won't let you go to robot directly from outside Fermi)*
- *Local Stager can 'ftp' files to your local disk, where they can be managed in a disk cache by SAM (if you want), running a local Station Server and Project Server*

Physics data files - by tape

use central database to determine files you need and associated calibration, geometry, alignment ‘trees’ and RCPs

- *get physics data exported to you on tape*
- *optionally get other data exported in either ‘database’ or flat file dspack or other format*

a) cache data on local disk

- *declare new file locations on your disk to database (local or central)*
- *run locally - no need for stager*
- *record info in database (local or central)*

Physics data by tape

b) too much data for disk? - - set up a local staging system from tape or mass store

- *write your own command for a Stager to use to fetch a specific file and interface this to your operations/tape mounting/robot*
- *SAM Station Server will handle disk cache for you - release least used files, or files according to group policy*

Our almost-exclusive streaming strategy should help to minimize the number of DST, or other files, you need to get on tape

Database Server - local or remote?

- Any of the database servers can run at your site, connected to the Fermilab central database, provided you install
  - *oracle client software (no licence fee), will be available for linux, windows/nt, solaris, irix, dec-unix*
- A Calibration database server will be able to cache constants in memory locally once fetched from central database - until it is restarted (up to some limit)

Database server ….

- A database server at your site, using a remote database at Fermilab, can store some transactions in case of network down and post them later, but won't be able to query for file lists etc. during down time.
- If you use a remote database server at Fermilab you will be out of luck unless the network is up - but you won't have to worry about running database servers…
  - *(just like web server access)*

Database local or remote?

- In principle the various database servers can interface to any reasonable sql relational database (but its all work!)
- We hope to make a decision in early 2000 on which ‘freeware’ or ‘cheap’ database will be supported for those that want a local database for performance/reliability reasons
- An extract of available information from the central database will be prepared for export to a local database (no event catalog)
- Incremental exports/updates will be needed also

Freeware or cheap database candidates

- Oracle on linux looks good - not free, but cheap, and Fermilab could deal with licences
  - *CDF acting as early adopters*
  - *Migratory databases on a CD probably by end 2000*
- MSQL - not a good choice
- mySQL - might be a possibility
- Microsoft Access using odbc - also possible

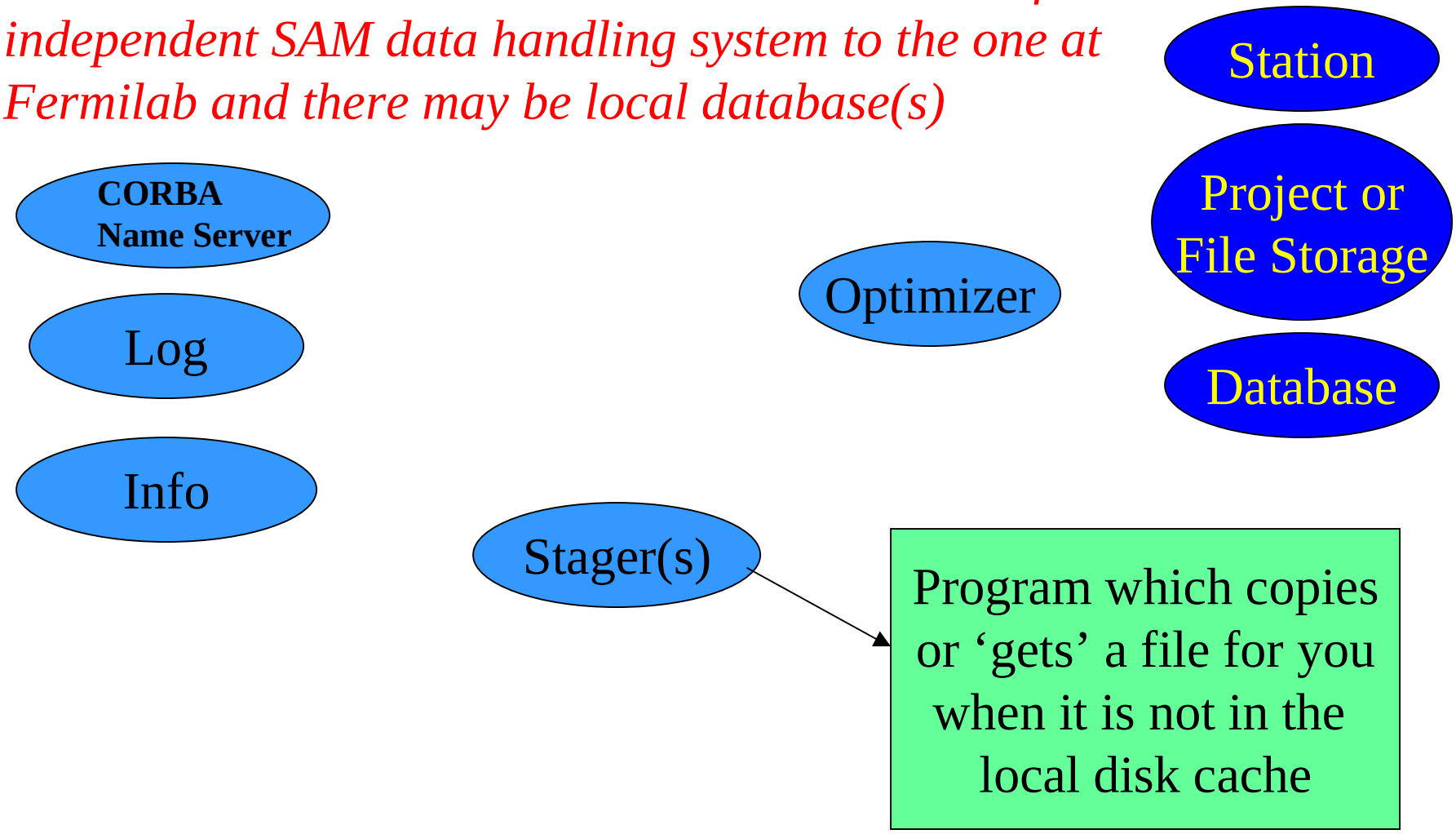
Let's choose just one, if possible!

Making Database Servers work with a non-Oracle database

- May sound like several servers to deal with (SAM, Calibration, RCP, etc.) …but..
  - *All servers are built using same technology and* ***using code generation****, from the database table and C++ class definitions*
  - *this will help ease the job of providing a version of each server interfaced to a non-Oracle database -- if we have to*
  - *note - all the clients of the Database Servers remain totally unchanged*

SAM system outside Fermilab

All servers must run somewhere at the local site if it is to run an independent SAM data handling system to the one at Fermilab and there may be local database(s)



SAM at your place?

- Copy of most of file/event catalog and calibration data
- File and Tape Export facility needed
- Run entire SAM system with all Servers locally
- Interface Stager to your own staging system - via a single command to fetch a file not present in the disk cache
- Re-synchronize with Fermilab central database for transactions and new file locations.
- Incremental updates of databases

- Best if you have Oracle and a Database Administrator (DBA)
- Outside the scope of SAM - Enstore/Operations project (SAM provides file/tape list)
- Code will run (certainly by V6.0)
- need to write this interface to your data center, HPSS?, tape mounting, etc.
- This will be done for V6.0 SAM and perhaps for calibration?
- Support Databases project will help with this

Conclusions

- We are trying hard to ensure that the data access system will provide the access layer for all types of data, for those at Fermilab and outside.
- SAM, d0om, Calibration, etc are all designed to allow for various different i/o mechanisms
- There are many ways to configure the SAM system - with different performance, reliability, and support trade-offs
- Access to central databases directly should not be ruled out even though local extracts or copies will be supported (using a ‘cheap’ database) and might sound attractive.
- We welcome suggestions and want to hear your concerns
- We would welcome help from people outside Fermilab trying to set up a whole system, or work on database data export/synchronization procedures earlier than V6