MicroBooNE Experience with SAM

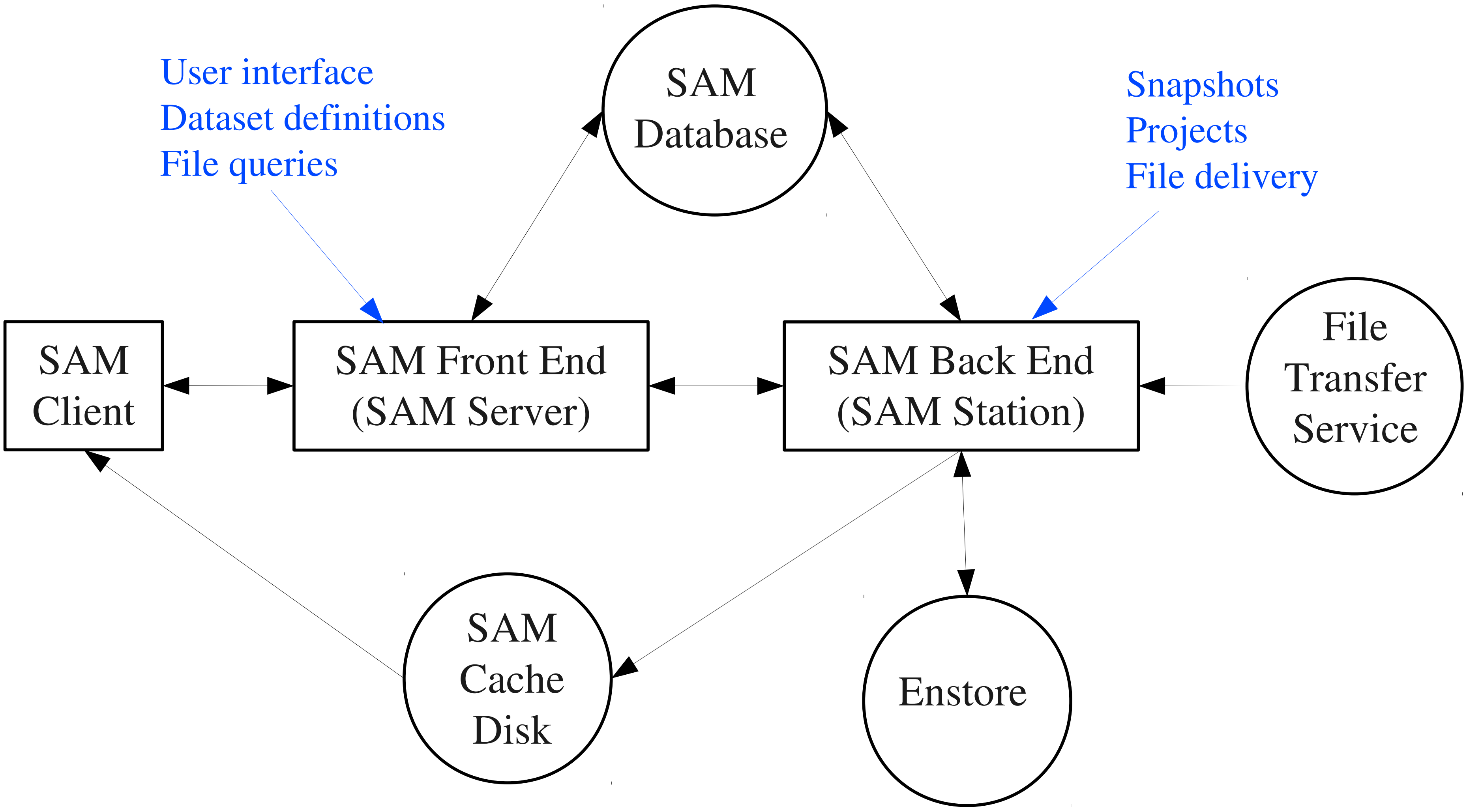
LBNE Software and Computing Meeting
Nov. 14, 2013

H. Greenlee

Elements of Fermilab Data Handling

- SAM microboone resources.
  - Database (file catalogue).
  - Server (http://samweb.fnal.gov:8480/sam/uboone/api/).
  - Station (manages sam projects, file delivery).
  - Cache disk.
- Enstore (tape system).
  - Currently using “t10kc” tapes with a capacity of 5 Tb.
- File transfer service.
  - Handles uploading of files into enstore.

Data Handling Overview



User interface
Dataset definitions
File queries

Snapshots
Projects
File delivery

SAM Concepts

- Metadata – Descriptive information associated with files.
  - Main purpose is to allow files to be queried.
- Dataset definition – A memorized sam database query, based on metadata, that returns a collection of files.
- Dataset snapshot – A collection of files (result of query).
- Project – A collection of files (snapshot) scheduled for delivery to worker nodes.

Sam Use Cases

- Generating sam metadata for output files.
  - Does not require interacting with sam server.
  - Microboone metadata design microboone docdb 2414 (public).
- Declaring files to sam (loading metadata into sam database).
- Adding disk locations for disk-resident files.
- Uploading files to enstore tape.
- Creating dataset definitions (e.g. for specific MC samples).
- Fetching files from sam (running sam projects) for a given dataset definition.

Sam Clients

- All sam clients have in common that they send requests to the samweb http server (http://samweb.fnal.gov:8480/sam/uboone/api).
- Web browser.
- Samweb (setup sam_web_client).
  - Line mode client (samweb -e uboone <subcommand> …).
  - Python client (import samweb_cli).
- Ifdh client tools (setup ifdhc).
  - Line mode client (ifdh <subcommand>).
  - Python client
  - C++ client (class ifdh, not art-specific).
- Art client (setup ifdh_art).
  - Wraps ifdhc c++ sam client as art service (IFDH service), and provides sam-capable instances of file delivery and file transfer services.

Microboone Sam Metadata

- A typical Microboone dataset definition:

```
% samweb -e uboone describe-definition prod_muminus_0.5-5.0GeV_25degf_t0_uboone_fall2012
Definition Name: prod_muminus_0.5-5.0GeV_25degf_t0_uboone_fall2012
  Definition Id: 66
  Creation Date: 2013-06-17T14:34:46
       Username: greenlee
          Group: uboone
     Dimensions: file_type mc and data_tier reconstructed and ub_project.name
'prod_muminus_0.5-5.0GeV_25degf_t0_uboone' and ub_project.stage merge and ub_project.version v1_1
```

  - Note that mc files are queried using five metadata attributes:
    - file_type
    - data_tier
    - ub_project.name
    - ub_project.stage
    - ub_project.version
  - Real data files will use different metadata attributes (e.g. run number).

Microboone Sam Metadata (cont.)

- Microboone also uses sam metadata to preserve the art program configuration (fcl configuraiton) for the complete processing history of all files in sam.
  - FCL-relates metadata parameters:
    - fcl.name.
    - fcl.version.
  - SAM fcl paramaters reference external fcl repository.
  - These parameters are only informational. We do not normally use them in constructing queries.

Sam Output Art Services and Modules

- FileCatalogMetadata service (art).
  - Defines basic metadata.
- RootOutput module (art).
  - Defines basic metadata.
- FileCatalogMetadataExtras service (larsoft/Utilities).
  - Adds additional metadata, including experiment-specific.

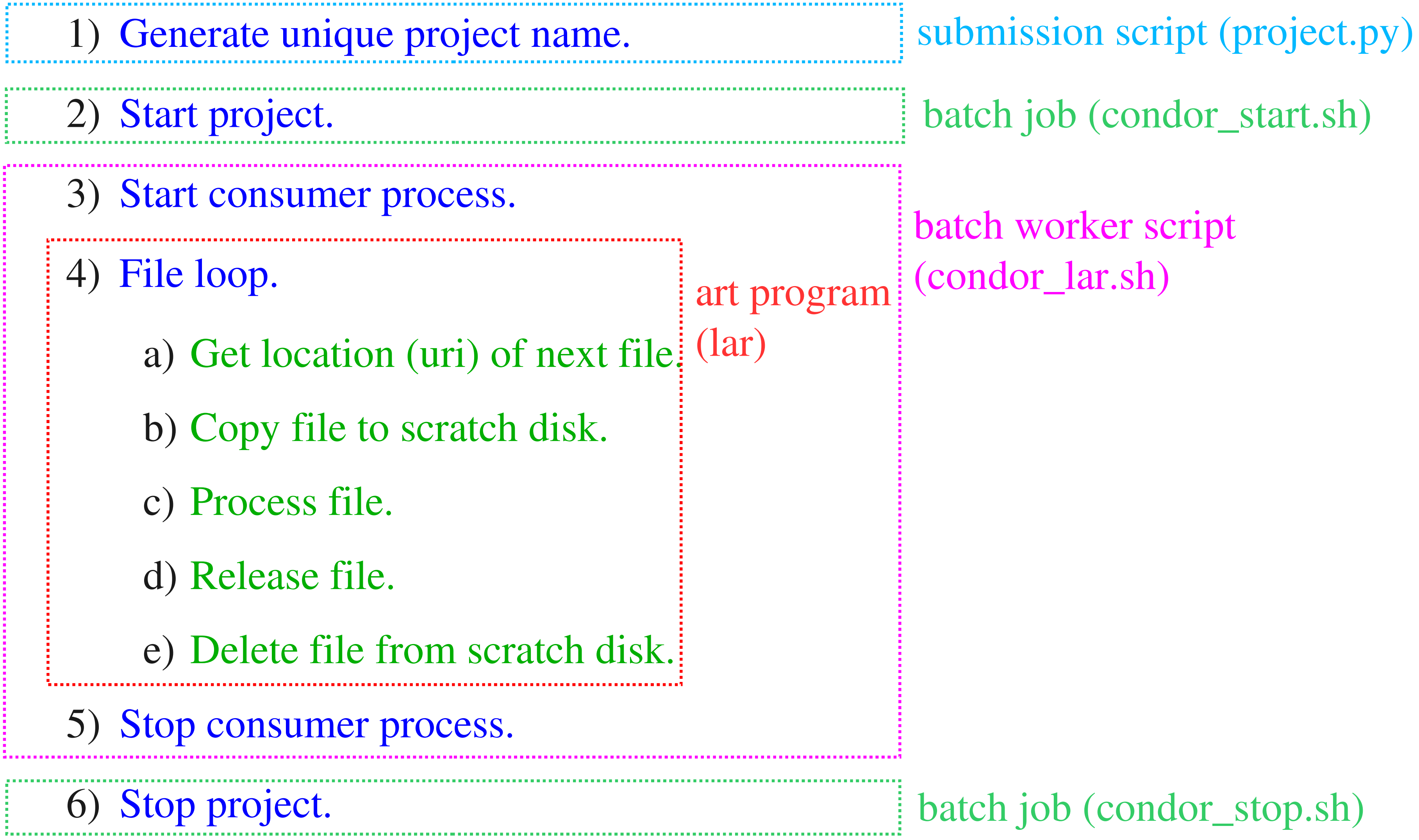
SAM Input Art Services and Modules

- CatalogInterface virtual service (art).
  - TrivialFileDelivery service (art). Supports files and file lists.
  - IFCatalogInterface service (ifdh_art). Supports sam/ifdh.
- FileTransfer virtual service (art).
  - TrivialFileTransfer service (art). Supports files ans file lists.
  - IFFileTransfer service (ifdh_art). Supports sam/ifdh.
- IFDH service (ifdh_art). Full C++ samweb client.
- RootInput module.

Using SAM in Art Programs

- Using SAM in art programs.
  - Users do not normally interact with any sam services or module in their c++ code.
  - Users need to properly configure sam-related services and modules via fcl job configuration.
  - Reading from sam requires some support from enclosing batch worker script and batch submission script.

Sam Input: Project Life Cycle



1) Generate unique project name.

submission script (project.py)

2) Start project.

batch job (condor_start.sh)

3) Start consumer process.

batch worker script (condor_lar.sh)

4) File loop.

art program (lar)

a) Get location (uri) of next file.

b) Copy file to scratch disk.

c) Process file.

d) Release file.

e) Delete file from scratch disk.

5) Stop consumer process.

6) Stop project.

batch job (condor_stop.sh)

Microboone SAM Usage

- Microboone has (so far) run three large MC productions.
  - MCC 1.0 (Dec. 2012).
  - MCC 2.0 (July, 2013).
  - MCC 2.1 (Oct. 2013).
- All files generated in these productions have metadata in sam database, and files are stored on tape.
  - For MCC 1.0, metadata was generated after the fact using scripts.
  - For MCC 2.0 and 2.1, metadata was generated automatically using art and larsoft tools.

Summary

- Over about the past year, Microboone has been involved in an intensive effort to adopt SAM and related tools for data handling.
  - Generating metadata.
  - Storing data on tape.
  - Reading data using sam.
  - Integrating sam into art framework and microboone production system.