The POOL File Catalog for LCG-1

Maria Girone, CERN
(on behalf of the POOL Team)

- Requirements from the experiments
- File Catalog functionality
- Some use cases for LCG-1

Component purpose

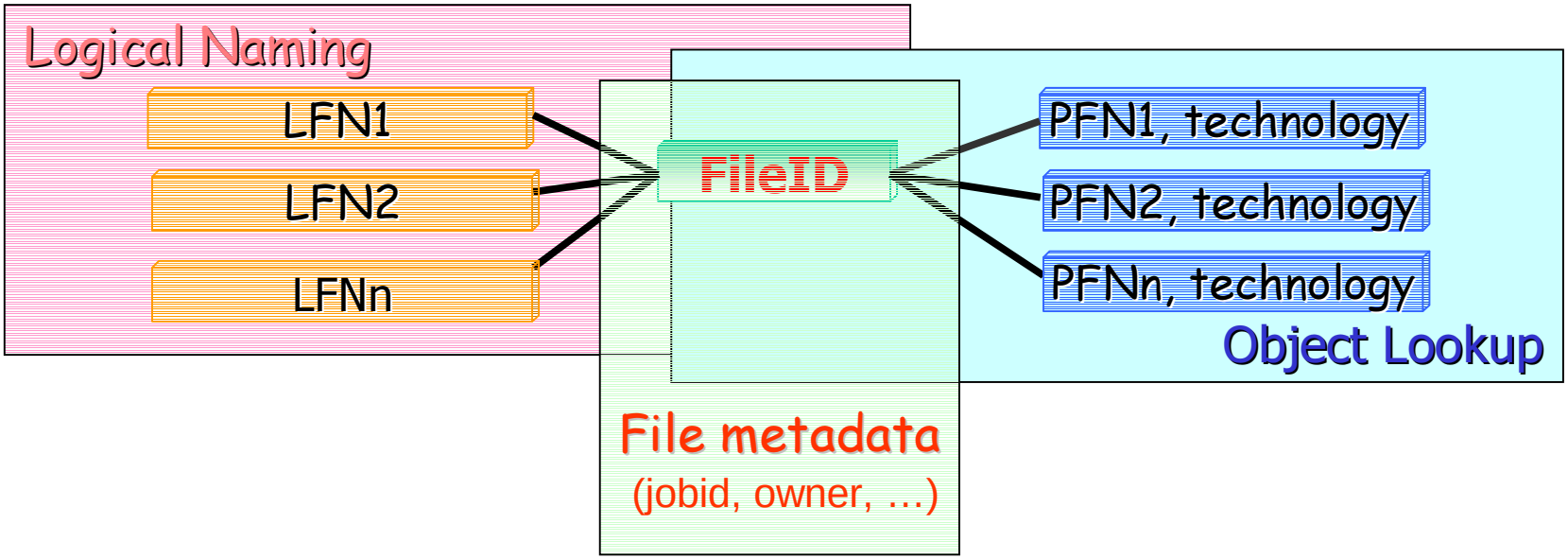
- Maintaining a consistent list of accessible files (physical and logical names) together with their unique identifiers

- GUID implementation for FileID

  - unique and immutable identifier for a file (generated at create time)
  - allows the production of a consistent sets of files with internal references without requiring a central ID allocation service
  - catalog fragments created independently can later be merged without modification to corresponding data files

File Catalog Schema

- FileID-PFN mapping is sufficient for object lookup
- FileID-LFN mapping is supported for user convenience, but not used internally by POOL
- File technology type is stored together with PFN (e.g. root/tree)
- Optionally, File metadata attributes are associated with the FileID, used for fast query based catalog manipulation. Not used internally by POOL



Component Usage

- Abstract interface insulates the user from several concrete implementations
- API clients:
  - C++ API for POOL storage components and experiment framework to register and lookup a file inside the application process
  - Command-line tools for end-users and file administrators (maybe replaced later by python scripts)
  - Python based graphic user interface for the catalog browsing

Concrete implementations

- XML Catalog
  - typically used as local file by a single user/process at a time
  - no need for network
  - supports r/o operations via http
  - tested up to 50K entries
- Native MySQL Catalog
  - handles multiple users and jobs (multi-threaded)
  - tested up to 1M entries
- EDG-RLS Catalog
  - grid aware applications
  - Oracle iAS or Tomcat + Oracle / MySQL backend
  - pre-production service based on Oracle (from IT/DB) , RLSTEST, already in use for POOL V1.0

Requirements from the experiments

- A questionnaire has been circulated recently to the experiments to collect their requirements for the File Catalog component

- POOL will be integrated in experiments' data challenges in
  - Alice: not currently planned
  - Atlas: second quarter of 2004 for next larger production activity DC2 (integration into ATHENA already started)
  - CMS: POOL proposed as baseline for PCP (starting this summer) and later for DC04
  - LHCb: spring 2004

- Expected number of entries in the catalog
  - ATLAS/DC2 – O(100) minimum bias input files/job, O(10^6) total output files
  - CMS/PCP - O(10k) minimum bias input files, O(10^6) total output files
  - LHCb/spring '04 - O(100) input files/job, O(10^5) total output files

Requirements from the experiments

- Required functionality:
  - file registration, lookup on logical and physical filenames
  - delete and clean-up on failure
  - efficient meta data and filenames query support
  - pre-registration mechanism to allocate filenames and final registration after file quality checks
- Bookkeeping is currently handled by experiment (specific) code

File Catalog Functionality

- In POOL V1.0 (some small items to be completed)
- Connection string:
  - XML
    - xmlcatalog_file:/tmp/FileCatalog.xml or file:/tmp/FileCatalog.xml
    - xmlcatalog_http://pc01.cern.ch/file001 (R/O)
  - MySQL
    - mysqlcatalog_mysql://@lxshare070d.cern.ch:3306/testFCdb
  - EDG
    - edgcatalog_http://rlstest.cern.ch:7777/edg-replica-location/services/edg-local-replica-catalog

File Catalog functionality

- Connection and transaction control functions

- Catalog insertion and update functions on logical and physical filenames
- Catalog lookup functions (by filename, FileID or query)
- Clean-up after an unsuccessful job

- Catalog entries iterator (caching will be fully operational in the EDG Catalog in V1.1)
- File Metadata operations (e.g. define or insert file metadata)

- Cross catalog operations (e.g. extract a XML fragment and append it to the MySQL catalog)

File metadata schema

- Uses internally the POOL AttributeList component
- User-defined attribute definition, once per catalog
  - "(jobid,string), (owner,string), (jobtype,string), (run_number,integer)"
- Only catalogs with consistent schema can be cross populated

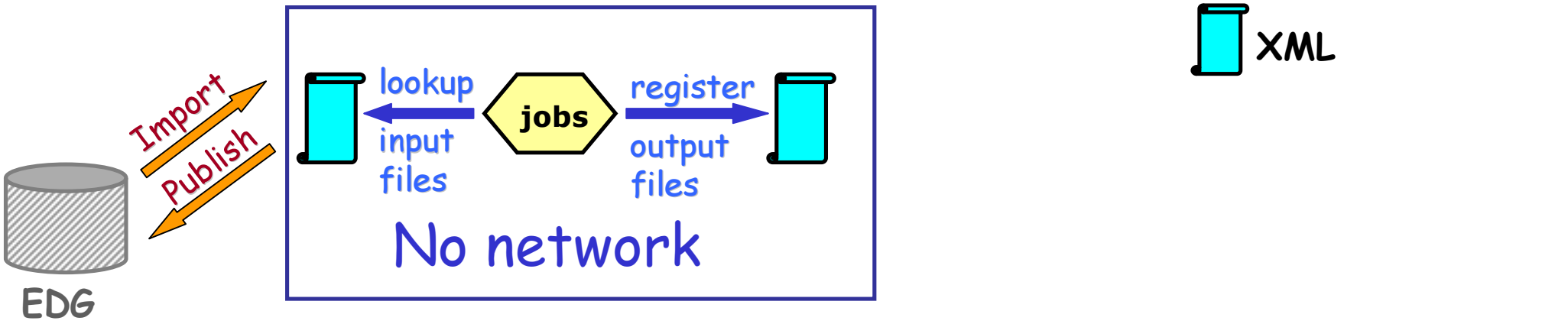
Queries

- Queries are used to look-up or to extract a catalog fragment from the source catalog and attach it to the destination catalog
- Queries are supported at the client side for XML and EDG (EDG is working on support for server side), and at the server side for MySQL
- Some examples
  - "jobid='cmssim' ", "owner like '%wild%' "
  - "pfname like 'path' and owner = 'atlasprod' "
- Evaluation of numerical query expressions not yet fully supported (backend enhancements in progress)

Use of command-line tools

- Definition of a schema for a given experiment catalog
  (FCdefineMetaData)
- Population of a local catalog
  (FCregisterLFN, FCaddMetadata)
- Eventual actions after quality checks
  (FCclearUnsuccessful, FCdeleteEntry, FCrenamePFN, FCaddReplica)
- Publication of the local catalog to the general catalog
  (FCpublish, also by query)
- Retrieval of file information
  (FClistPFN, FClistLFN, FClistMetaData, also by query)

Use case: isolated system



Import: get schema $set schema = `FCdescribeMetaData -u $edgcatalog `

Import: define schema $FCdefineMetaData -u $xmlcatalog -d $schema

Import $FCpublish -d $edgcatalog -u $xmlcatalog -q "pfname like %digi% and owner = lhcbprod"

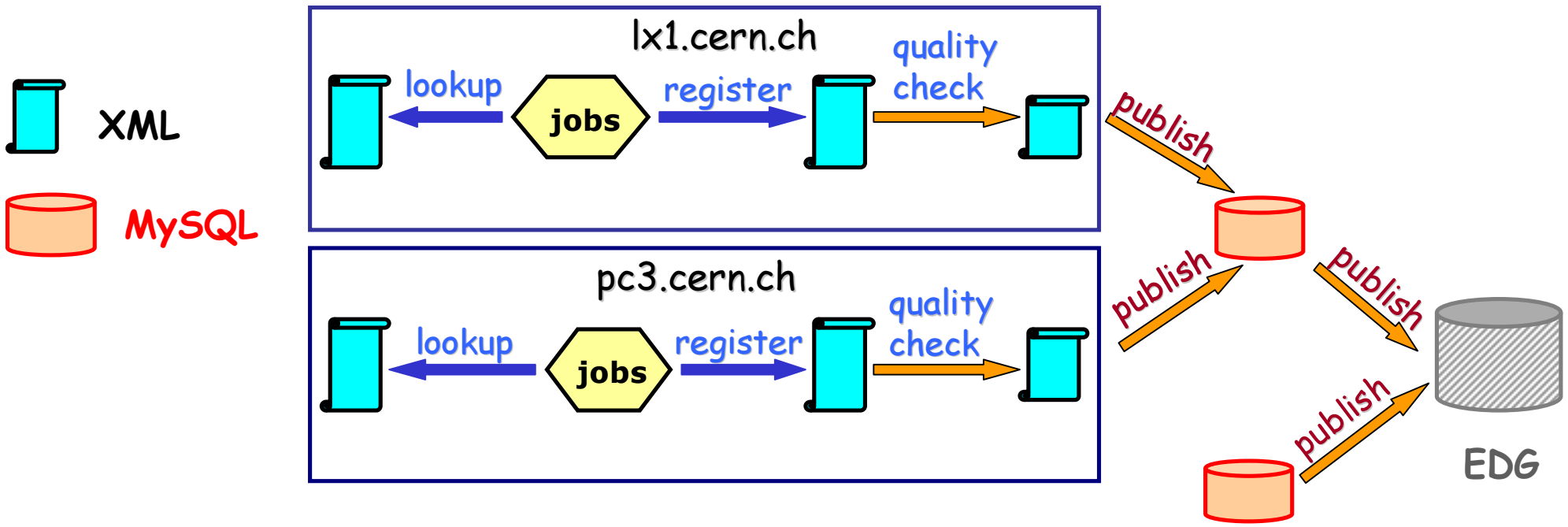
Disconnected from the network, ($setenv POOL_CATALOG $xmlcatalog)

Browse PFN $FClistPFN -l input_logical_filename

Register alias $FCregisterLFN -p out_physical_filename -l out_logical_filename

Insert metadata $FCaddMetaData -p out_physical_filename -m "(owner,'lhcbprod'),(jobtype,'reco')"

Use case: farm production



Actions after quality check
$FCclearUnsuccessful
$FCrenamePFN -p out_physical_filename -n final_physical_filename

Publish (bulk insert in EDG-RLS) $FCpublish -d $edgcatalog -u xmlcatalog_file:lhcbprod01.xml

Summary and perspectives

- In POOL V1.0, the POOL file Catalog satisfies the requirements collected so far from the experiments
  - some remaining implementation constraints are being removed
- A test service, rlstest, is already available since POOL V1.0
- The POOL File Catalog is ready to be used in LCG-1 and needs feedback
- Experiment-specific production services for the EDG Catalog will be provided in conjunction with POOL V1.1 (e.g. RLSATLAS.CERN.CH, …)

Web site: http://pool.cern.ch/

File Catalog performance tests

- Preliminary tests done on POOL V0.5

Pentium III-1.2GHz
free memory-220MB
PFN-200 char; FileID-36 char

- XML: tested up to 50K entries
  - start time:
    - new catalog ~10ms
    - catalog with 20K entries ~6s
  - registerPFN: <0.3ms/entry
- MySQL: tested up to 1M entries
  - up to 300 concurrent clients, commit every 100 entries or less frequent
  - registerPFN: <1.5ms/entry
- EDG-RLS based catalog registerPFN: ~6ms/entry (autocommit)