

LHCb File Transfer framework

N. Brook,
Ph. Charpentier,
A. Tsaregorodtsev



LCG Storage Management Workshop , 6 April 2005, CERN

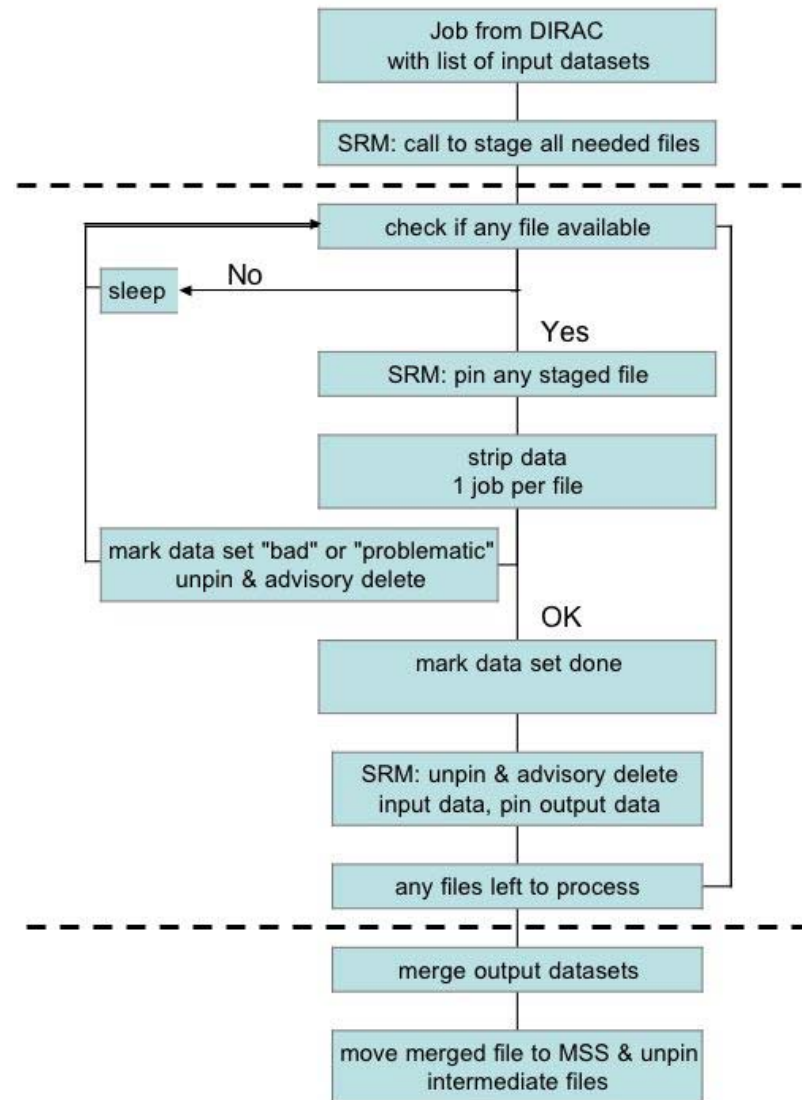
Outline

- ◆ LHCb (advanced) usage of SRM
- ◆ SRM v2 requirements
- ◆ DIRAC Data Management tools
- ◆ File transfer framework
- ◆ Interfacing to FTS

LHCb (advanced) Usage of SRM

Stripping on LCG

- Jobs have several input files (between 40 and 80)
- Jobs sent to site where the data are placed
- Currently 3 sites used CNAF, CERN and PIC based on CASTOR Mass Storage
- Using SRM interface to access MSS



Scale of stripping in Data Challenge

Physics stripping jobs	
Number of events per job	40,000
Number of files	80
Input data size	$80 \times 0.3 = 24 \text{ GB}$
Number of output files	2 (DST + event collection)
Output DST size	600 MB
Event collection size	1.2 MB
<i>Number of events</i>	60M
<i>Number of jobs</i>	1,500
<i>Input data size</i>	36 TB
<i>Output data size</i>	0.9 TB
Trigger stripping jobs	
Number of events per job	360,000
Number of files	400 (files of 900 evts) or 200 (1800 evts)
Input data size	$400 \times 0.18 = 72 \text{ GB}$
Number of output files	1
Output DST size	500 MB
<i>Number of events</i>	90M
<i>Number of jobs</i>	250
<i>Input data size</i>	18 TB
<i>Output data size</i>	125 GB

Usage of SRM

- LHCb CLI tools
 - Stage request
 - File status
 - Advisory delete
- CLI tools built on GFAL library - aim to avoid any SRM version dependencies

SRM (vsn 1.0) Experience with CASTOR

- inability to pin/unpin or mark file for garbage collection - poss. workarounds (redefined SRM “advisory delete” provided)
 - Throttle jobs - manpower intensive (not feasible)
 - New SRM stage request at each file check - use on LCG
 - Technology specific commands - use on LXBATCH for debugging workflow
 - SRM “advisory delete” re-defined
- SRM fails to deal with corrupted/missing files
 - If error returned to SRM all subsequent files are also marked as fail (even if successful!) - needs new CASTOR implementation

SRM (vsn 1.0) Experience with CASTOR

- No control over stage pool - mixing of general user & prod manager
 - Solved - LCG can now check on user and responsibility and assign pool accordingly
- SRM request ID lifetime
 - Implemented last 10 days - noted problem during a re-boot of SRM server which lost lifetime flat file
- Access rights
 - if one server creates files under one user account, it is not readable by the other servers if the mapping is to another user - problem solved

LHCb requirements for SRM

- ◆ Consider SRM v2.1
 - ✦ Ignore the artificial grouping of methods (basic, advanced 1...)
 - ✦ v1 is definitely not enough, v3 not mature
- ◆ Definition:
 - ✦ An SRM endpoint is uniquely defining an SE

SRM namespace

- ◆ An SURL is the concatenation of 3 fields
 - ◆ An SE/SRM endpoint : SRM://mysrmserver.site.xx
 - ◆ A file prefix : e.g. /castor/cern.ch
 - Mind this is a site-dependent information, but due to change...
 - ◆ A filename : e.g.
/lhcb/production/DC04/evtttype1234/DST/01234_2134.dst
 - Need for a convention “a la Castor” : VO/username
- ◆ When replicating a file, no way to know the actual prefix (site dependent)
- ◆ Hence we request the possibility to use relative paths
 - ◆ SRM://mysrmserver.site.xx//lhcb/production/DC04/...

High priority methods

- ◆ File types
 - ✦ *Volatile and permanent* mandatory
- ◆ Space management (required for stripping jobs)
 - ✦ space reservation, extension, deletion
- ◆ Directory management
 - ✦ All methods but possibly *mv* (mind *ls*)
- ◆ Data “transfer” methods
 - ✦ All important, pinning is a must (i.e. lifetime)
 - ✦ srmCopy: cf discussion with fts

High priority methods (cont'd)

◆ Protocols

- ✦ srmPrepareForGet should return a list of possible tURLs
- ✦ User/IO system to select which one to use
 - e.g. at ROOT level (should accept SURL as PFN)

◆ File access control

- ✦ Based on user and role
- ✦ Propose the use of user directories
 - e.g. lhcb/user/a/atsareg
 - How to define a persistent “name”?
 - How to create the directory (no write access to the top directory lhcb/user....)

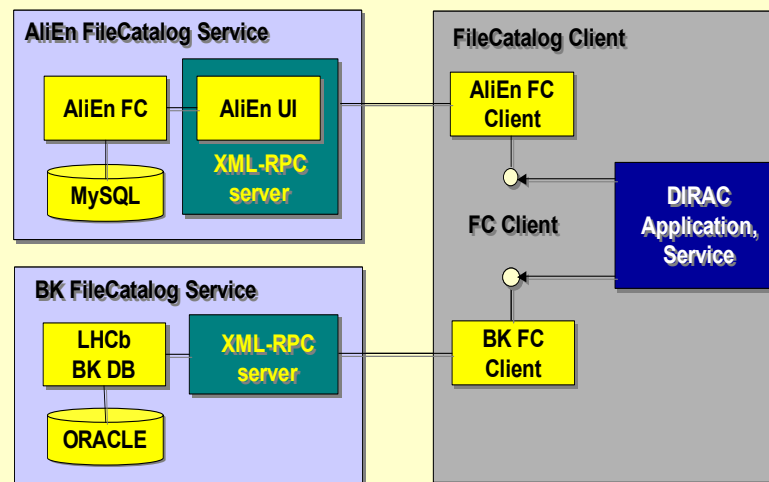
◆ Authentication/authorisation

- ✦ Should allow access from the Grid, direct or local

DIRAC Data Management tools

File Catalogs

- ◆ DIRAC incorporated 2 different File Catalogs
 - ✦ Replica tables in the LHCb Bookkeeping Database
 - ✦ File Catalog borrowed from the AliEn project



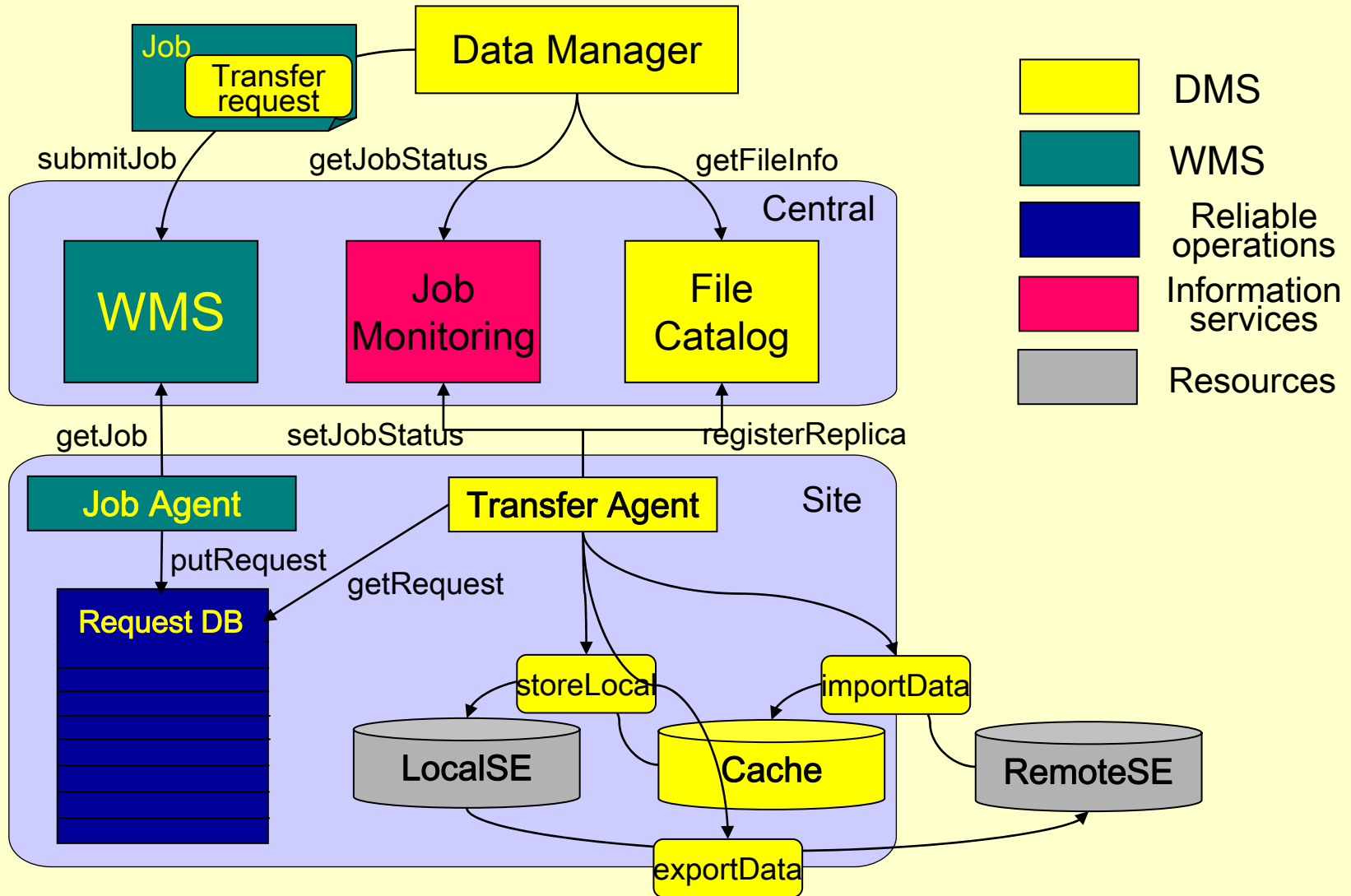
- ◆ Both catalogs have identical client API's
 - ✦ Can be used interchangeably
 - ✦ This was done for redundancy and for gaining experience
- ◆ Other catalogs will be interfaced in the same way
 - ✦ LFC – work in progress
 - ✦ AliEn upgraded
 - ✦ FiReMan

Data management tools

- ◆ DIRAC Storage Element is a combination of a standard server and a description of its access in the Configuration Service
 - ✦ Similar to “Classic SE”
 - ✦ Pluggable transport modules: `gridftp`, `bbftp`, `sftp`, `ftp`, `http`, ...
- ◆ SRM can be incorporated into the DIRAC framework with similar interface
- ◆ DIRAC ReplicaManager interface (API and CLI)
 - ✦ `get()`, `put()`, `replicate()`, `register()`, etc

File Transfer framework

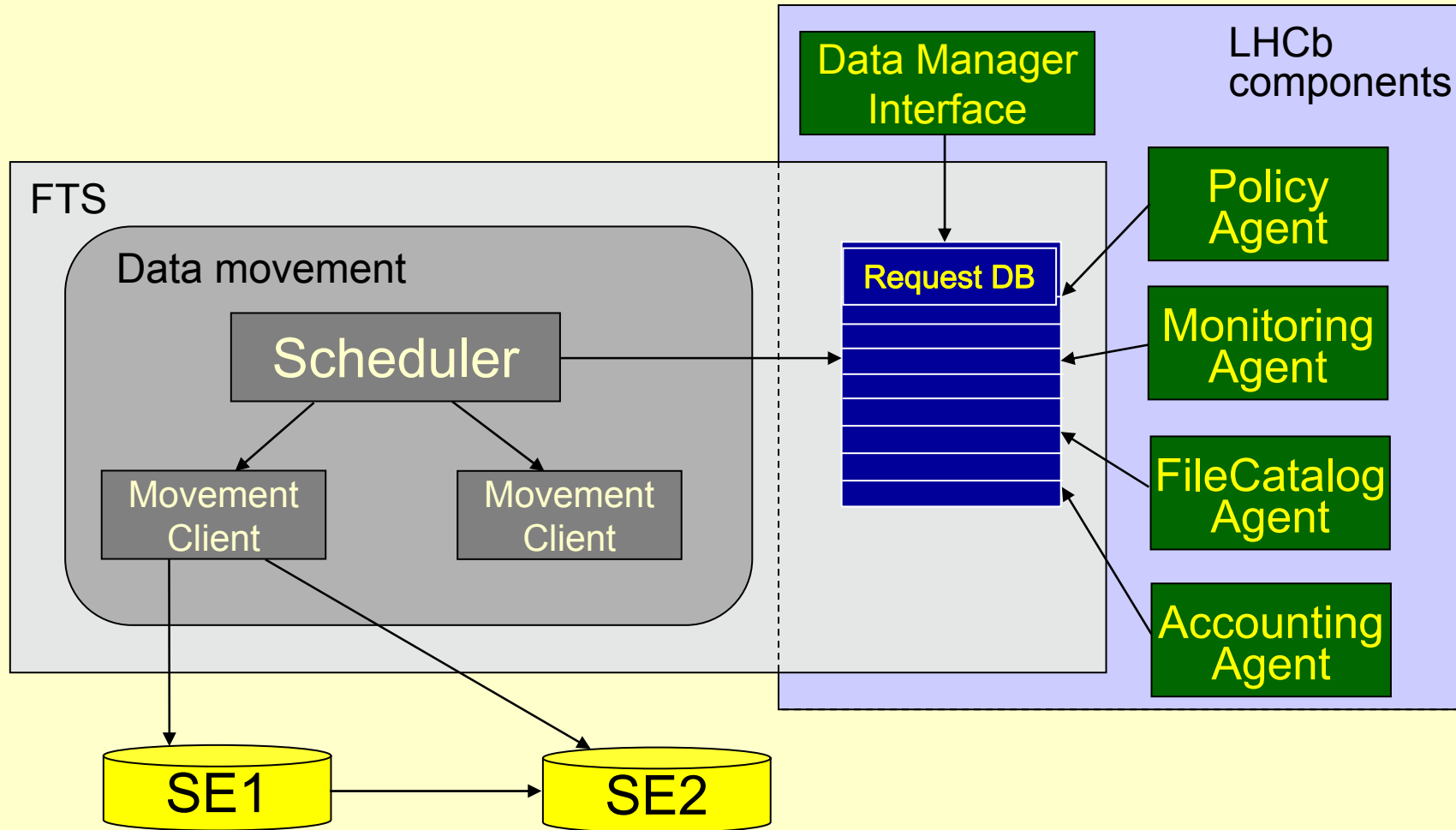
File Transfer framework



File Transfer framework

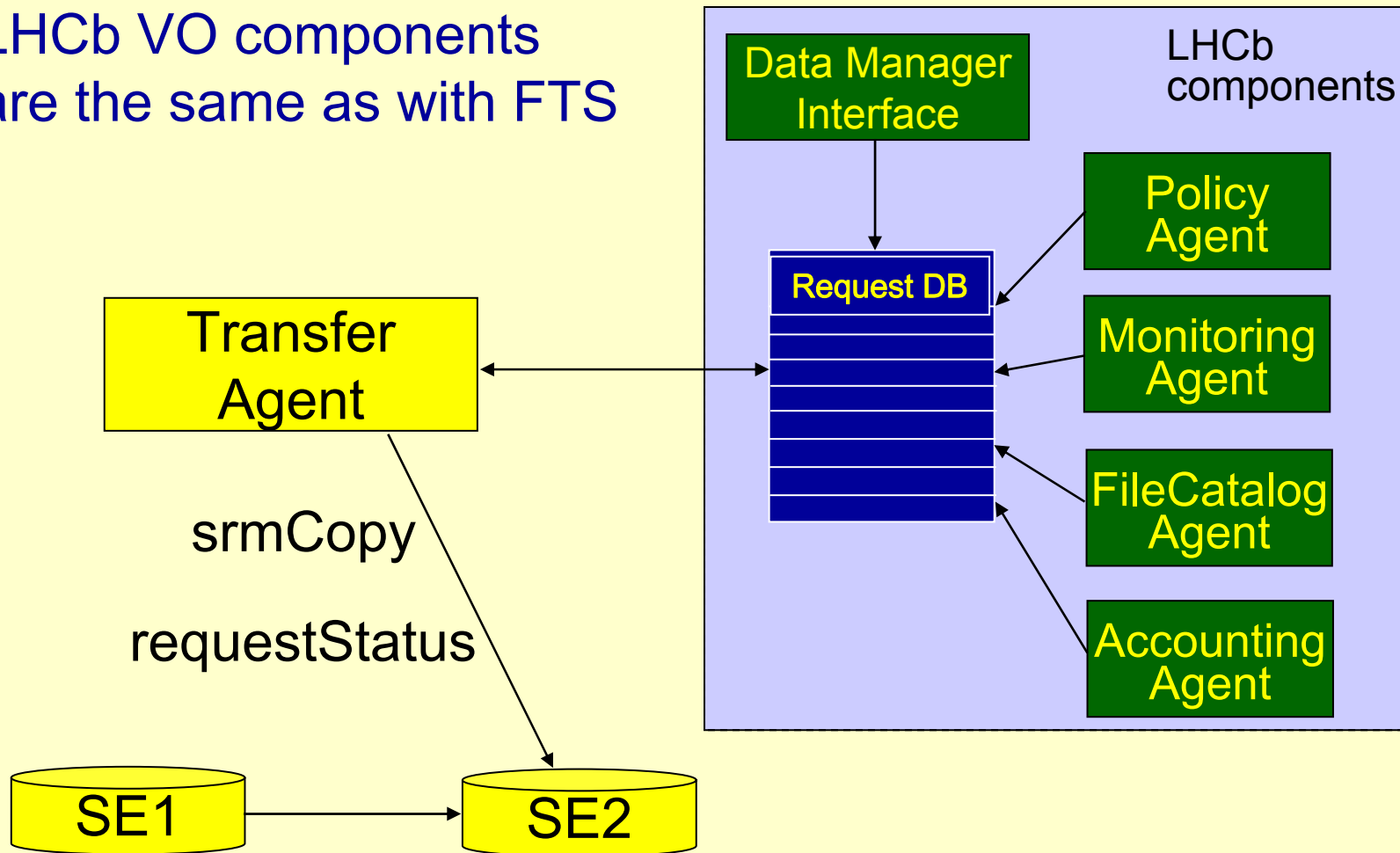
- ◆ Reuses the WMS infrastructure
 - ✦ To deliver transfer requests
 - ✦ To monitor the request execution progress
- ◆ Reliable File Transfer
 - ✦ Transfers are mediated by on-site Transfer Agents
 - ✦ On-site Request DB shared with other “reliable operations”:
 - Bookkeeping registration
 - Application status updates
 - Job parameters/accounting registration
- ◆ This framework is OK for small transfers
 - ✦ Job outputs
- ◆ Need for efficient bulk transfer operations

File Transfer with FTS



File Transfer with SRM-copy

LHCb VO components are the same as with FTS



Service Challenge

- ◆ We would like to get access to the FTS system as early as possible
 - ✦ May ?
 - ✦ Our own evaluation of stability
- ◆ Start with one central Request Store instance
 - ✦ Add more instances as necessary
- ◆ Do bulk transfers for Stripping data distribution to Tier1 centers
 - ✦ September
 - ✦ T0-T1, T1-T1 transfers
 - ✦ Not part of LHCb Data Challenge