DIRAC Data Management System

27th May 2014

Christophe Haen
on behalf of the LHCb Offline team

Plan

New DataManager Class

ReplicaManager deprecated

- Historical single point of entry for FileCatalog (FC) and StorageElement (SE)
- Huge class
- Mostly forward calls

Example

```
addCatalogFile(..)
getCatalogIsDirectory(..)
putStorageFile(..)
getStorageFileExists(..)
```

New DataManager Class

New DataManager class (DM)

- Based on the ReplicaManager
- no new functionality
- Introduced to cleanup the code

Know what you are doing !

- Catalog operation only ? Use the FC !
- Physical operation only ? Use the SE !
- DataManager for operations involving both

Minor changes

SingleFile/SingleDir parameters deprecated

- Deprecated. All methods follow the Successful/Failed convention
- Compatibility layer so far, but please change your code

'Catalog' parameter deprecated

- Not given in the DM methods anymore, but in the constructor

Changes in the SE

Minor changes

- add getLfnForPfn (from RM)
- getPfnForLfn and getPfnForProtocol compliant with the Successful/Failed return type convention

Major change : PFN constructed

- Possibility to not use FC PFN any longer
- Controled by UseCatalogPFN in the CS (global now, should be per SE)

New XROOTStorage plugin

New XROOTStorage plugin

- Avoids going through SRM
- Uses pyXRootd $\Rightarrow$ no system calls anymore
- Already coded but...

But...

- Not well tested (please help ! :-))
- Cannot fully replace SRM (staging status)
- Problems foreseen with replication between 2 SE using different protocols (need a protocol negociation)

Down with the PFN

Original idea

- Proposed in RFC #17
- SE methods calls with LFN or PFN $\Rightarrow$ confusion
- No real need for PFN but at the lowest level, i.e. in the SE
- idea : PFN replaced with (LFN, SE)

Problems

- I am lost ! ! getPFNBase, getPFNForLfn, getPfnForProtocol, getPfnPath, getAccessUrl, getTransportUrl ... $\Rightarrow$ proposal to simplify all this
- Usage of Catalog PFN (Are the VO needing this feature here ? Give yourself up !)

And what if...

Yesterday (v6r10 and before)

```
rm = ReplicaManager()
replicas = rm.getReplicas('myLfn', singleFile = True)['Value']
for se, pfn in replicas['myLfn'].items():
  rm.getStorageFileMetadata(pfn, se)
```

Today (v6r11)

```
dm = DataManager()
replicas = dm.getReplicas('myLfn')['Value']['Successful']['myLfn']
for se in replicas['myLfn']:
  StorageElement(se).getFileMetadata('myLfn')
```

And what if...

Tomorrow ?

```
myFile = File('myLfn') # DO NOT CALL THE OBJECT 'FILE'!!
lfn = myFile.LFN
for replica in myFile:
  se = replica.se
  pfn = replica.getUrl('myProtocol') # stored or constructed

# Need bulk operations ?
ds = Dataset ( [File1, File2, File3] )
ds.getUrl('mySE', 'myProtocol')
```

Pro/Cons

- Pro : more object oriented, fits better in DIRAC
- Pro : abstract the complexity
- Pro : *user-friendly*
- Pro : data-driven
- Cons : lot of work to reach that

Asynchronous removal of user files

UseCase

- Allow users to remove files asynchronously in scripts
- Would have been very useful in the past ! (user with 6M files...)

How ?

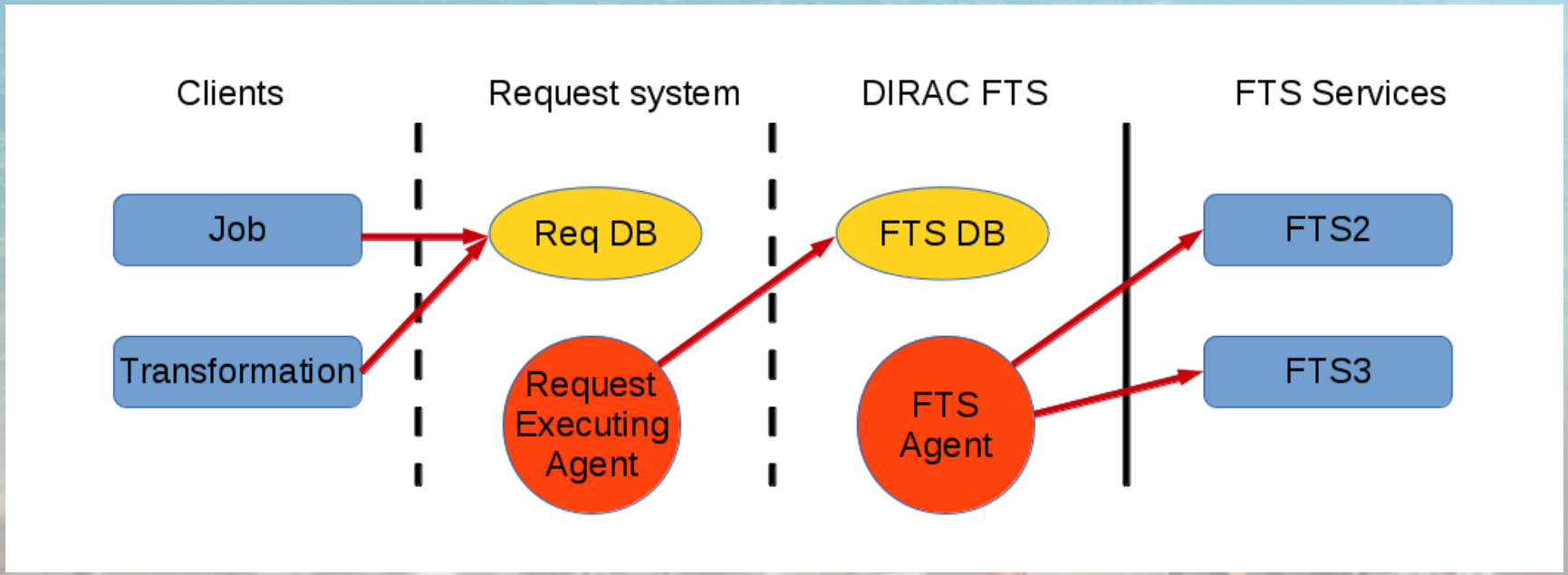
- Using the RMS (see Philippe's talk)
- *'Trash'* status in the DFC + Agent (S.Poss, A.Tsaregorodtsev, RFC #10)

Asynchronous removal of all files

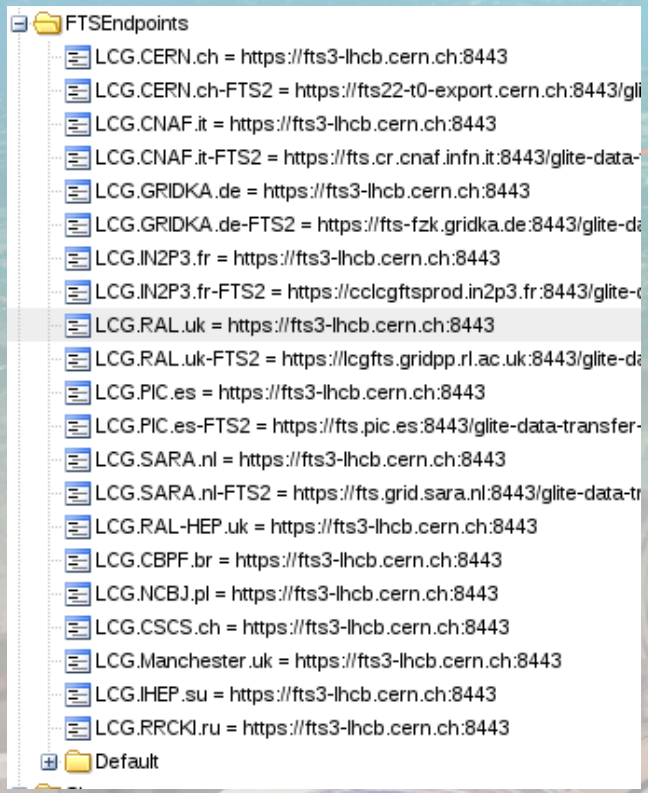
Problem : TransformationSystem

- Files need to be removed from the TS
- We can use the TS as a catalog
- As of now, it circumvents the state machine (see Federico's talk)

FTS workflow



Current use of FTS3



Like FTS2

- 1 FTS server per endpoint
- Job submission grouped by src & dest
- Submit & monitor : system call to glite-transfer-*

FTS3 features

To be implemented in DIRAC

Blacklist, server failover, Snapshot, Advisory, Multiple-source transfers, srm bringonline, REST python interface, free file grouping, bulk deletion

To be discussed

Retry, monitor message bus, replica failover

Will not be implemented

Multi-hop

Detailed feature description

See any presentation of Michail Salichos (e.g.
https ://indico.cern.ch/event/278289/session/7/contribution/62/material/slides/0.pdf)

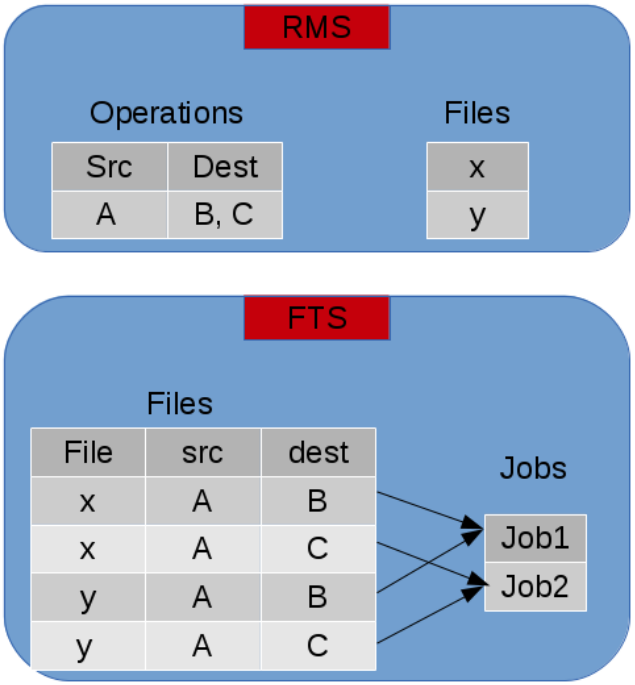
Implementation

Implementation

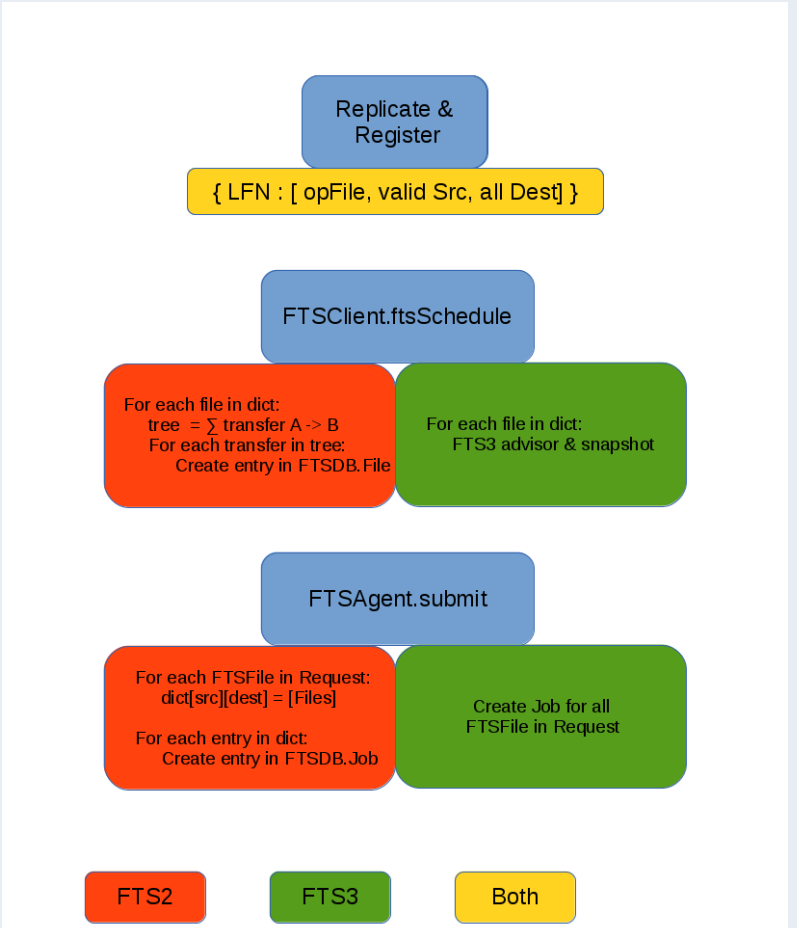
- Driving constraint : compatibility with FTS2 (really ?)
- Option 1 : start a parallel system
- Option 2 : modify the current system with a few *if* statements

Implementation

Current FTS2



FTS3 integration



DFC

DFC

- You all know and use the DFC
- LHCb has still to migrate from the LFC to the DFC

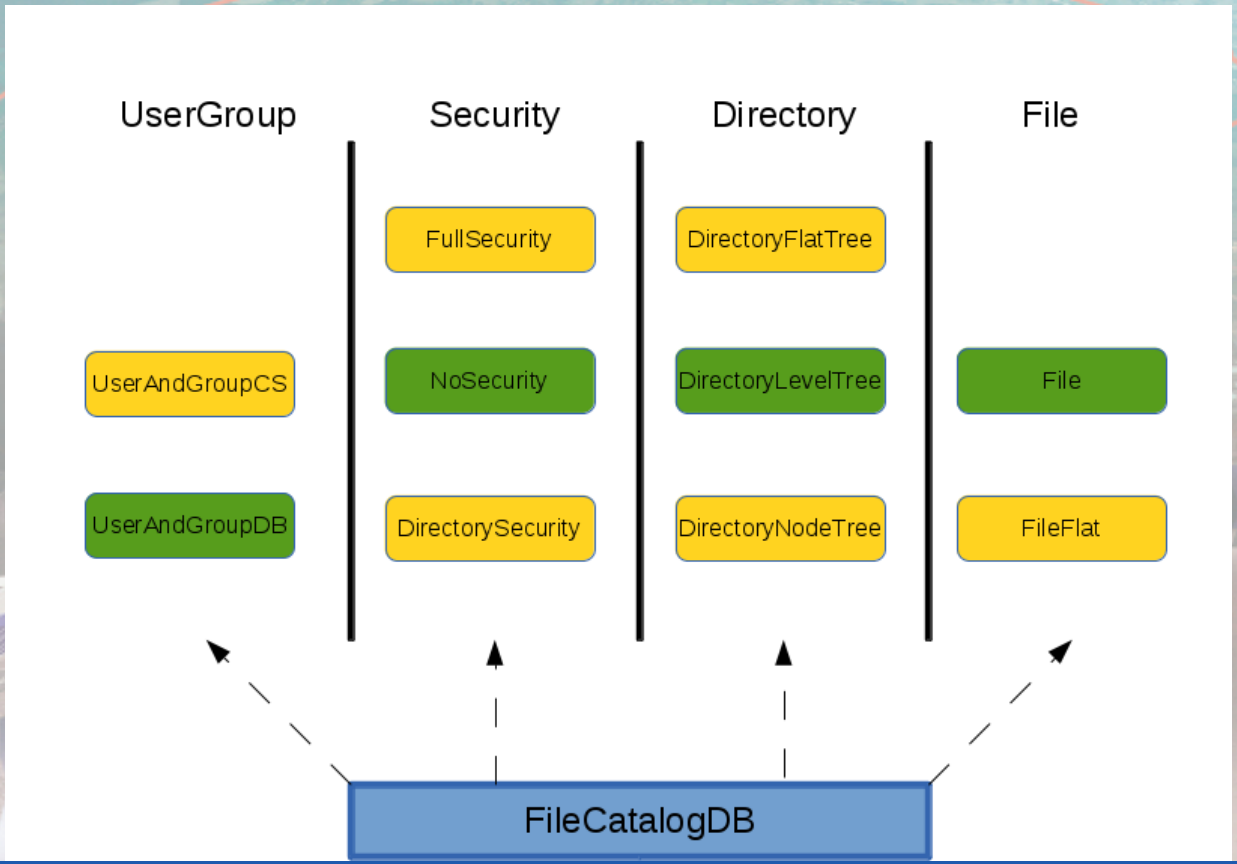
LHCb usage

- Replica catalog only
- No ancestors/descendants
- No metadata

LHCb problem

LHCb cannot migrate

- Max number of sub-directory per directory reached
- Max LFN depth reached



LHCb Requirements

LHCb Requirements

- Scalability target : operation time < 100ms, min 50 operations per second, multiple service per db, 100M Files, 200M replicas and 20M directories
- Consistent : Resistant to network/power cut, interrupt, etc.
- Move and transfer ownership of files (e.g. when a user leave)
- Proper permission management (to be discussed)
- Accounting of space usage
- Advanced search features (e.g. regex)

Proposal

Easy improvements

- InnoDB as engine
- Missing PrimaryKeys
- ForeignKey within managers

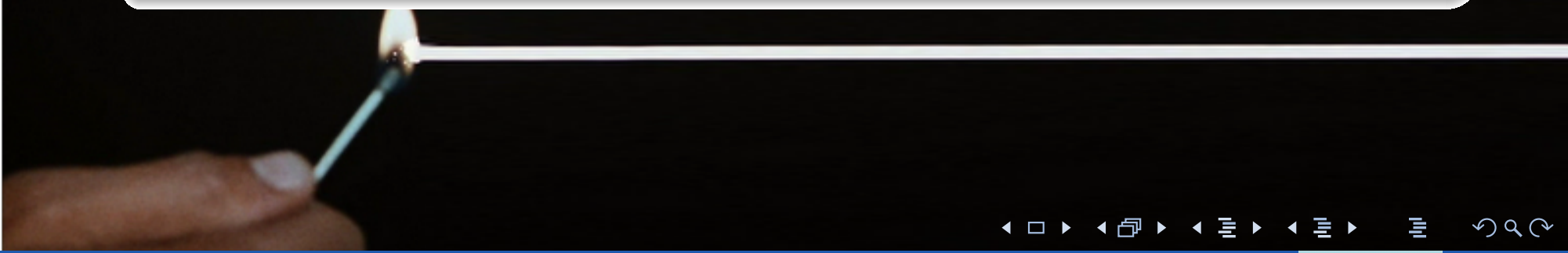
Less easy improvements

- Define sets of managers $\Rightarrow$ FK accros managers
- Directory manager based on closure table
- Storage Usage based on trigger
- Low level procedures, but logic in the code
- New clean db schema

Schedule : tight !

Let the game begin

- In production for Run 2 (early 2015)
- 15th June : requirement list completed
- 1st October : development done
- 1st November : testing done
- 15th November : migration procedure ready
- 1st December : Migration done



HELP !

We need your help !

- LHCb is not the only user
- You know parts of the DFC better than we do

Migration to the new schema

- Would you migrate ?
- Would you want the manager to be compatible with old data ?
- Would you accept to move the data ?

Conclusion

Conclusion

- Quite some changes were done (DataManager, Constructed PFN, ...)
- Many great improvements possibles (FTS3, Object oriented code, DFC...)
- But it won't happen without your help (ILC, CTA, BES III, FG-DIRAC, Auger, Belle II, DESY)