

# dCache Implementation at FZK

Forschungszentrum Karlsruhe GmbH  
Institute for Scientific Computing  
P.O. Box 3640  
D-76021 Karlsruhe, Germany

Dr. Doris Ressmann



[doris.ressmann@iwr.fzk.de](mailto:doris.ressmann@iwr.fzk.de)

<http://www.gridka.de>

## Introduction

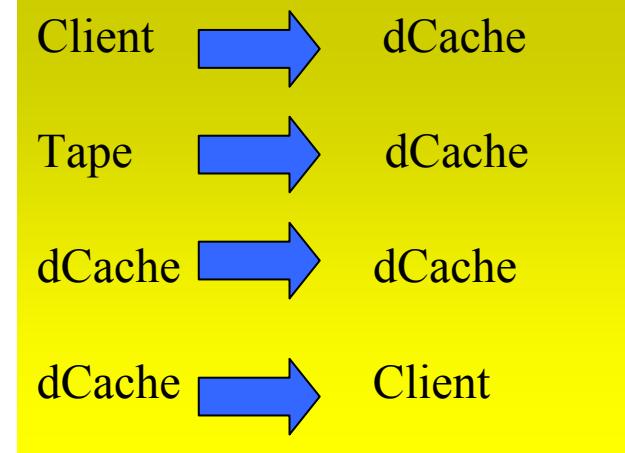
- What is dCache?
- Pool Selection mechanism
- dCache properties
- LCG connection
- Access to dCache – connection to CERN
- Tape Management
- Conclusion

## What is dCache?

- Developed at DESY and FNAL
- Disk pool management with or without tape backend
- Data may be distributed among a huge amount of disk servers.
- Automatic load balancing by cost metric and inter pool transfers.
- Data removed only if space is needed
- Fine grained configuration of pool attraction scheme

## *Pool Selection Mechanism*

- Pool Selection required for:

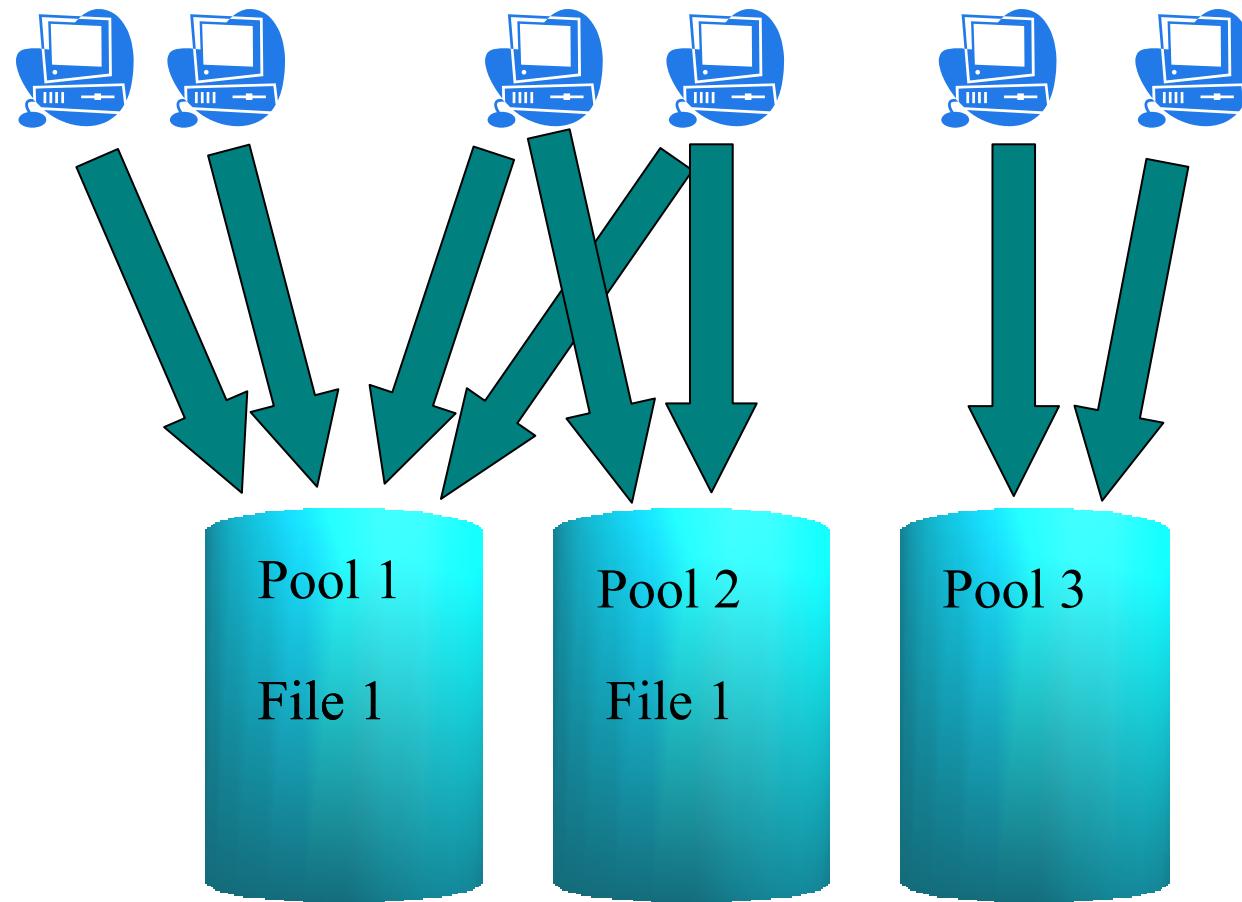


- Pool selection is done in 2 steps
  - Query configuration database :  
→ which pools are allowed for requested operation (intern/extern)
  - Query 'allowed pool' for their vital functions :  
→ find pool with lowest cost for requested operation

## *LCG Storage Element*

- DESY dCap lib incorporates with CERN GFAL library
- SRM version ~ 1.1 supported
- gsiFtp supported

## Multiple access of one file



## Access to dCache

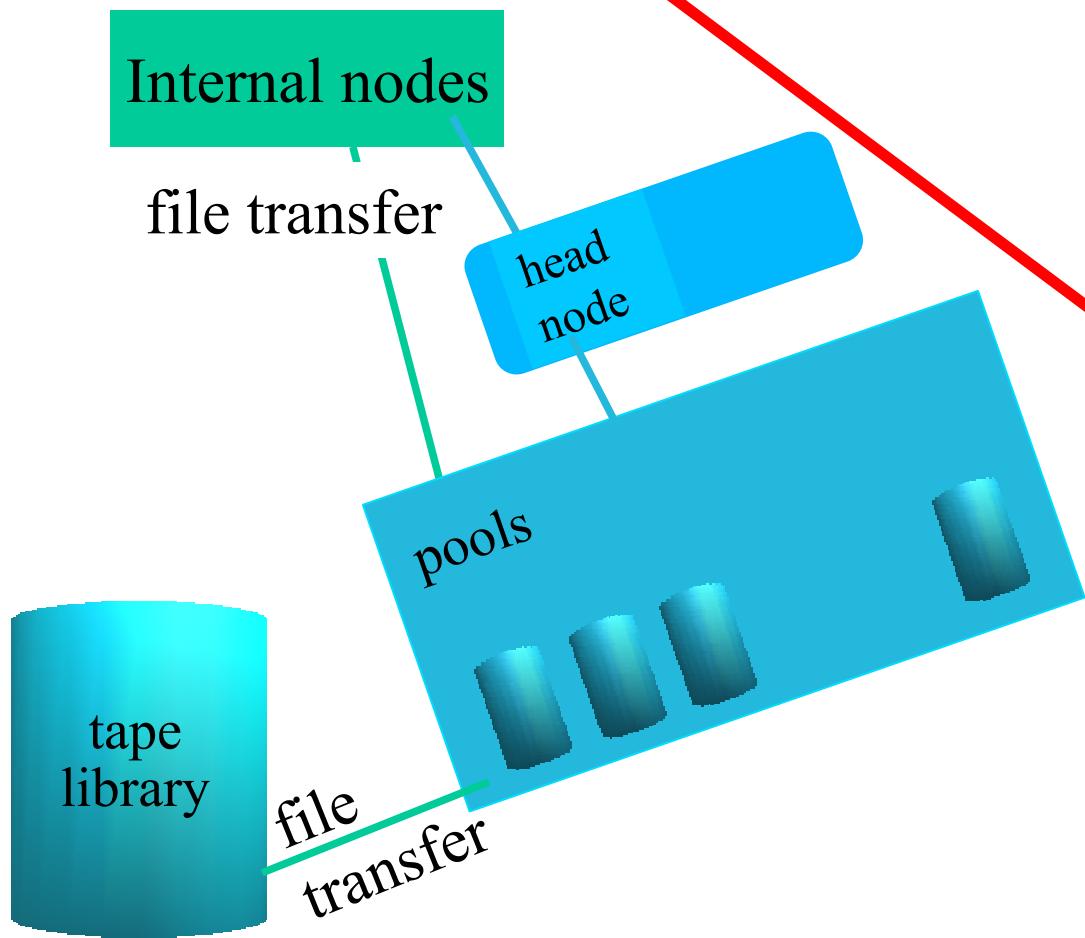
### Intern

- Mountpoint
  - ls
  - mv
  - rm
  - checksum,....
- dCap
  - dccp <source> <destination>
  - dc\_open(...)
  - dc\_read(...)

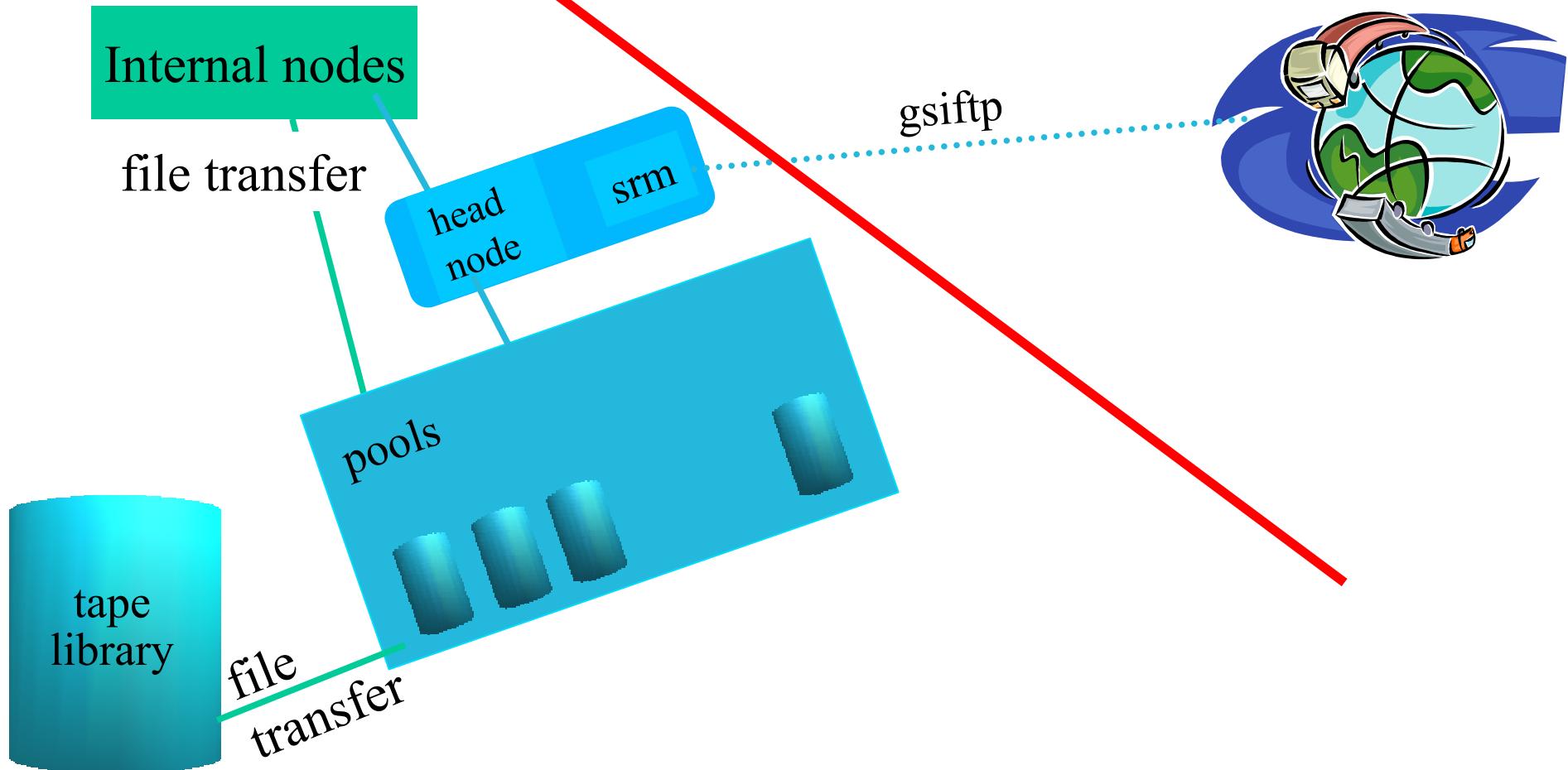
### Extern

- Gridftp
  - Problematic when file needs to be staged first
- SRMCP

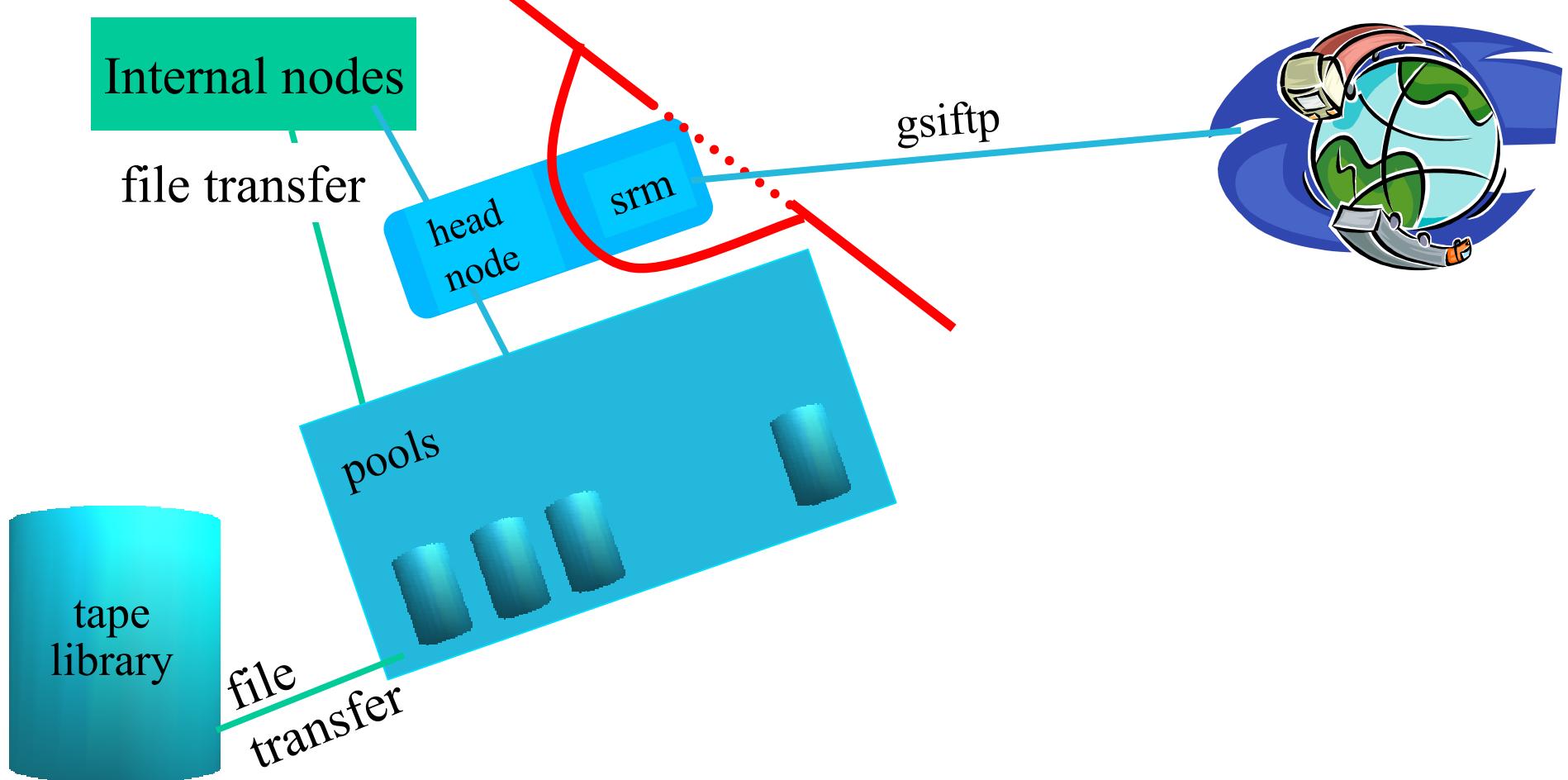
## dCache environment



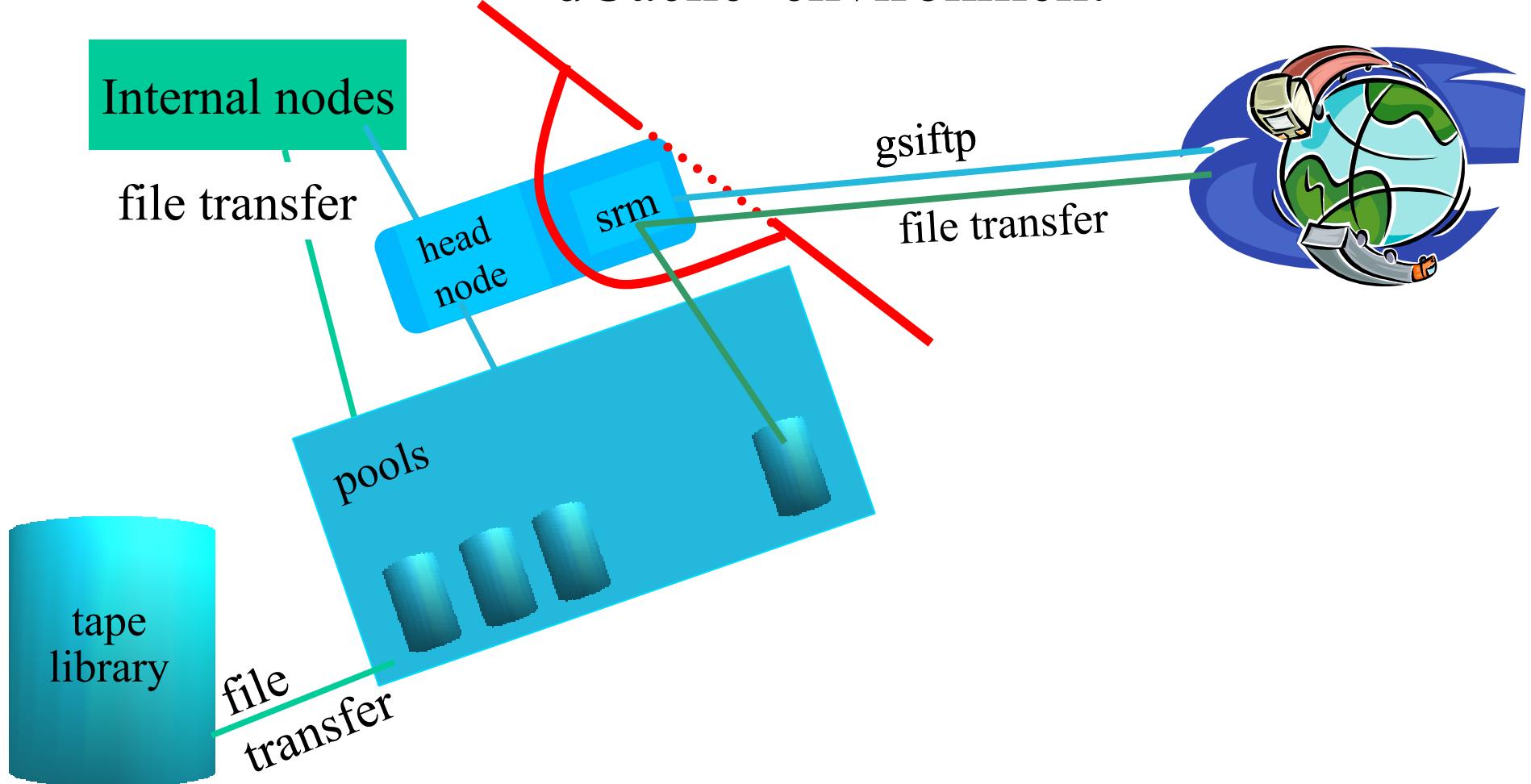
## dCache environment



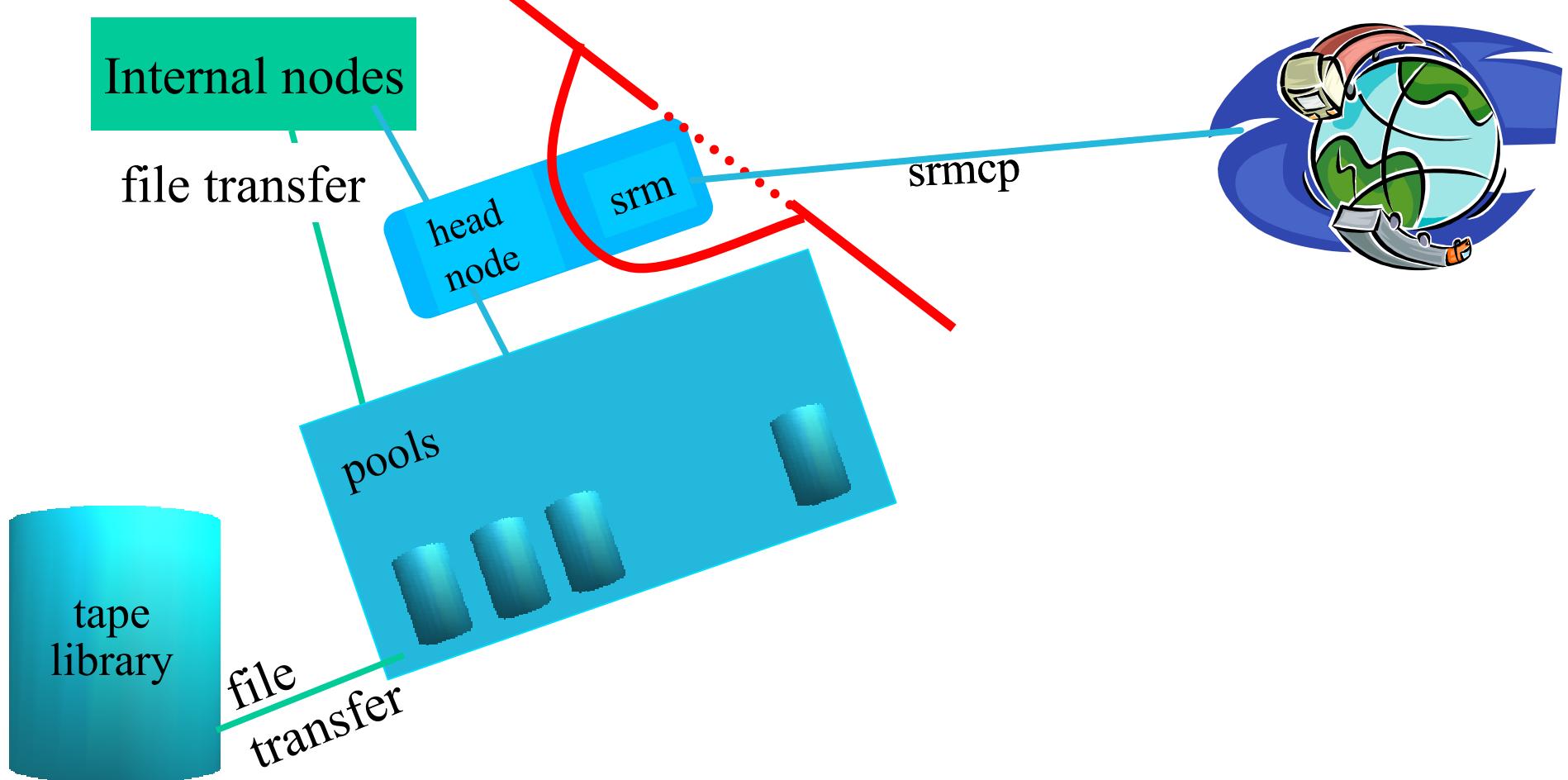
## dCache environment



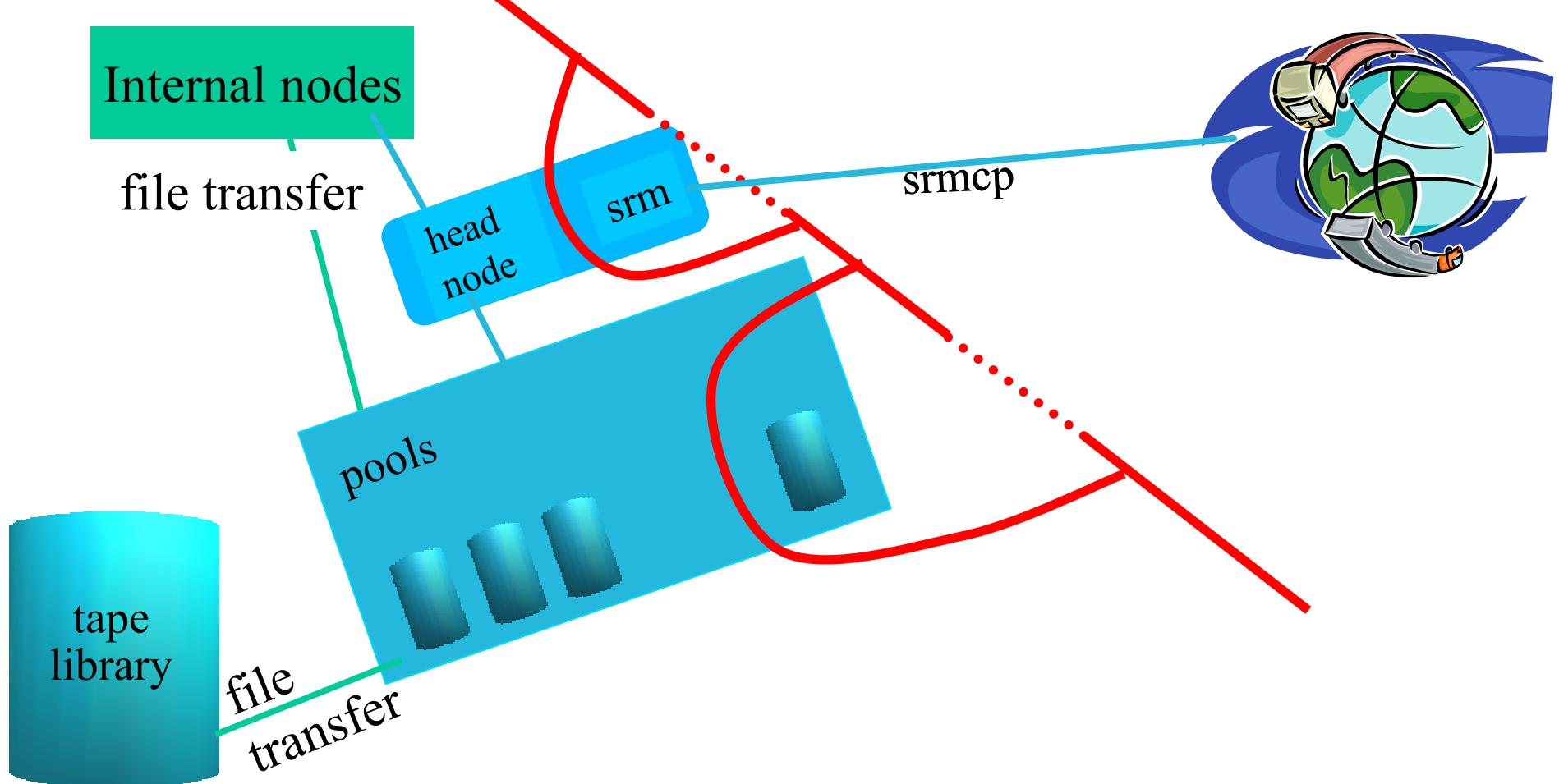
## dCache environment

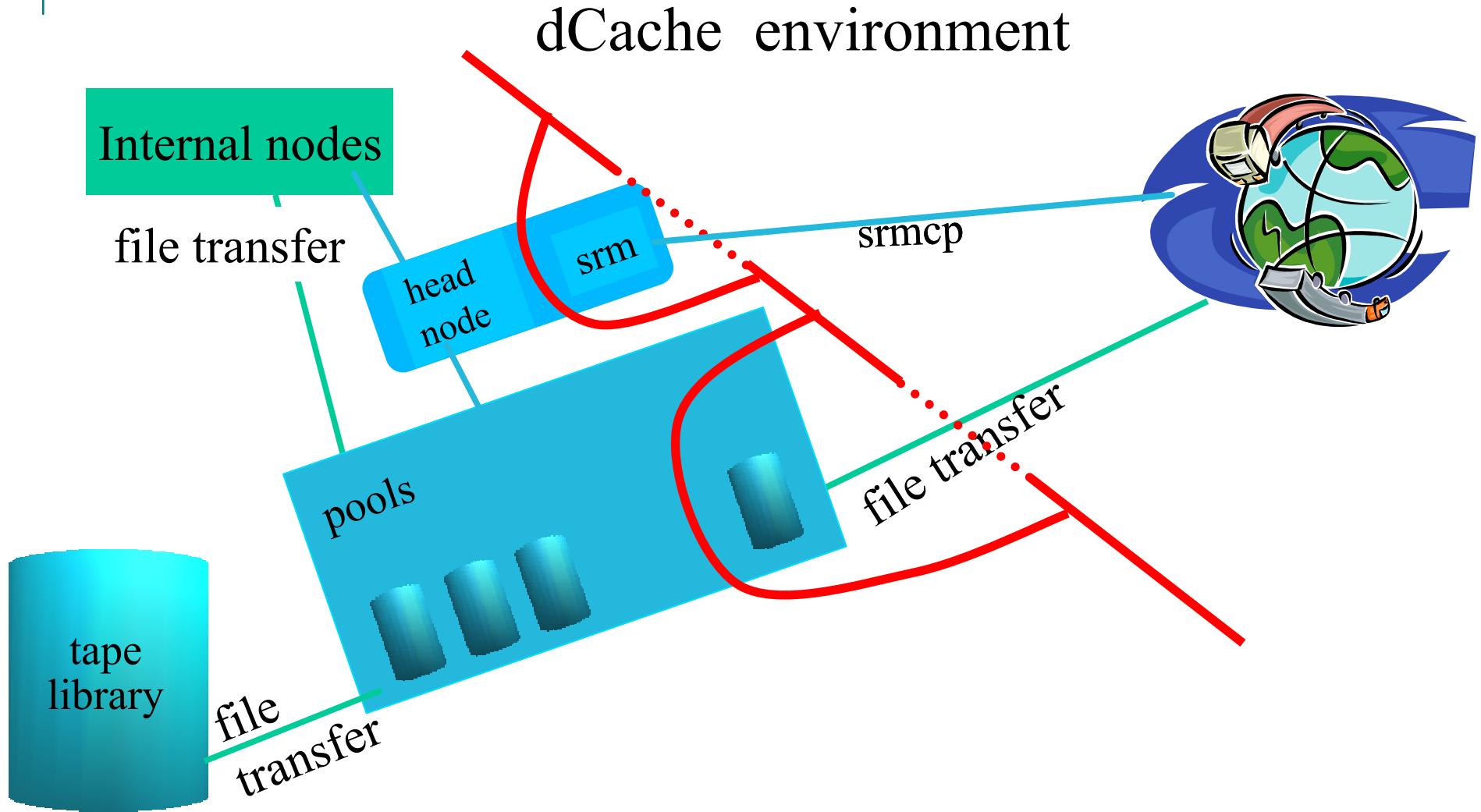


## dCache environment



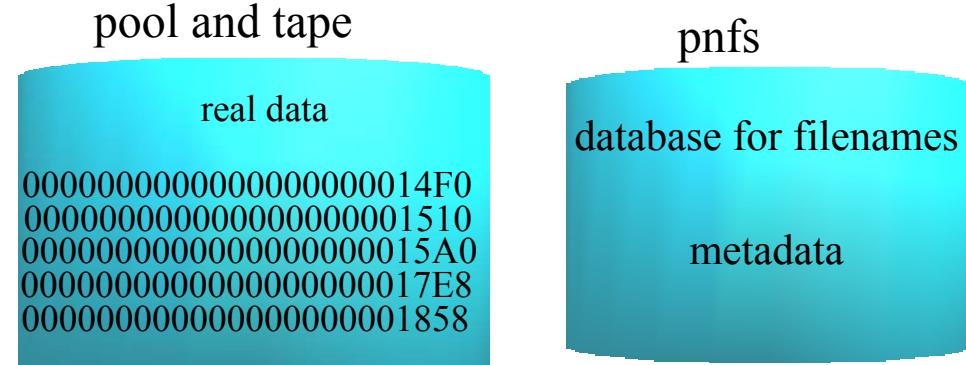
## dCache environment





## PNFS

### Perfectly Normal File System

- gdbm databases
  - Experiment specific databases
  - Independent access
  - Content of metadata:
    - User file name
    - File name within dCache
    - Information about the tape location (storage class...)
    - Pool name where the file is located
- 
- The diagram illustrates the PNFS architecture. It features two rounded rectangles side-by-side. The left rectangle is labeled "pool and tape" at the top and contains the text "real data" followed by five lines of hex values: 00000000000000000000000014F0, 0000000000000000000000001510, 00000000000000000000000015A0, 00000000000000000000000017E8, and 0000000000000000000000001858. The right rectangle is labeled "pnfs" at the top and contains the text "database for filenames" and "metadata" stacked vertically.

## gsiftp

- Only registered dCache user!!!  
grid-proxy-init  
globus-url-copy –dbg \  
file:///tmp/file1 \  
gsiftp://srm1.fzk.de/grid/fzk.de/mounts/pnfs/cms/file1
- dCache gridftp client and server in Java
- copy direct into available pool node
  - pool: data is precious
  - (can't be deleted)
  - flush into tape
  - data is cached (can be deleted from pool)



## srmcp

- Only registered dCache user!!!

grid-proxy-init

```
srmcp --debug=true \
```

```
 srm://srm.web.cern.ch:80//castor/cern.ch/grid/dteam/castorfile \
```

```
 srm://srm1.fzk.de:8443//pnfs/gridka.de/data/ressmann/file2
```

```
srmcp --debug=true \
```

```
 srm://srm1.fzk.de:8443//pnfs/gridka.de/data/ressmann/file2
```

```
file:///tmp/file2
```

## Firewall issues

- Connection to headnode: Ports 8443 and 2811
- Port Range to pool nodes: 20.000 to 50.000

## SRM Disk Version

- FNAL is currently developing a standalone SRM Disk version.
- The client uses a java version of gridftp
- The server uses a standard globus gridftp.
- It is far from production ready and needs:
  - SQL Database
  - jdbc driver
- <http://www-isd.fnal.gov/srm/unix-fs-srm/>

Forschungszentrum Karlsruhe  
in der Helmholtz-Gemeinschaft



Doris Ressmann

GridKa December 2004



## Tape Management

- Tivoli Storage Manager (TSM) library management

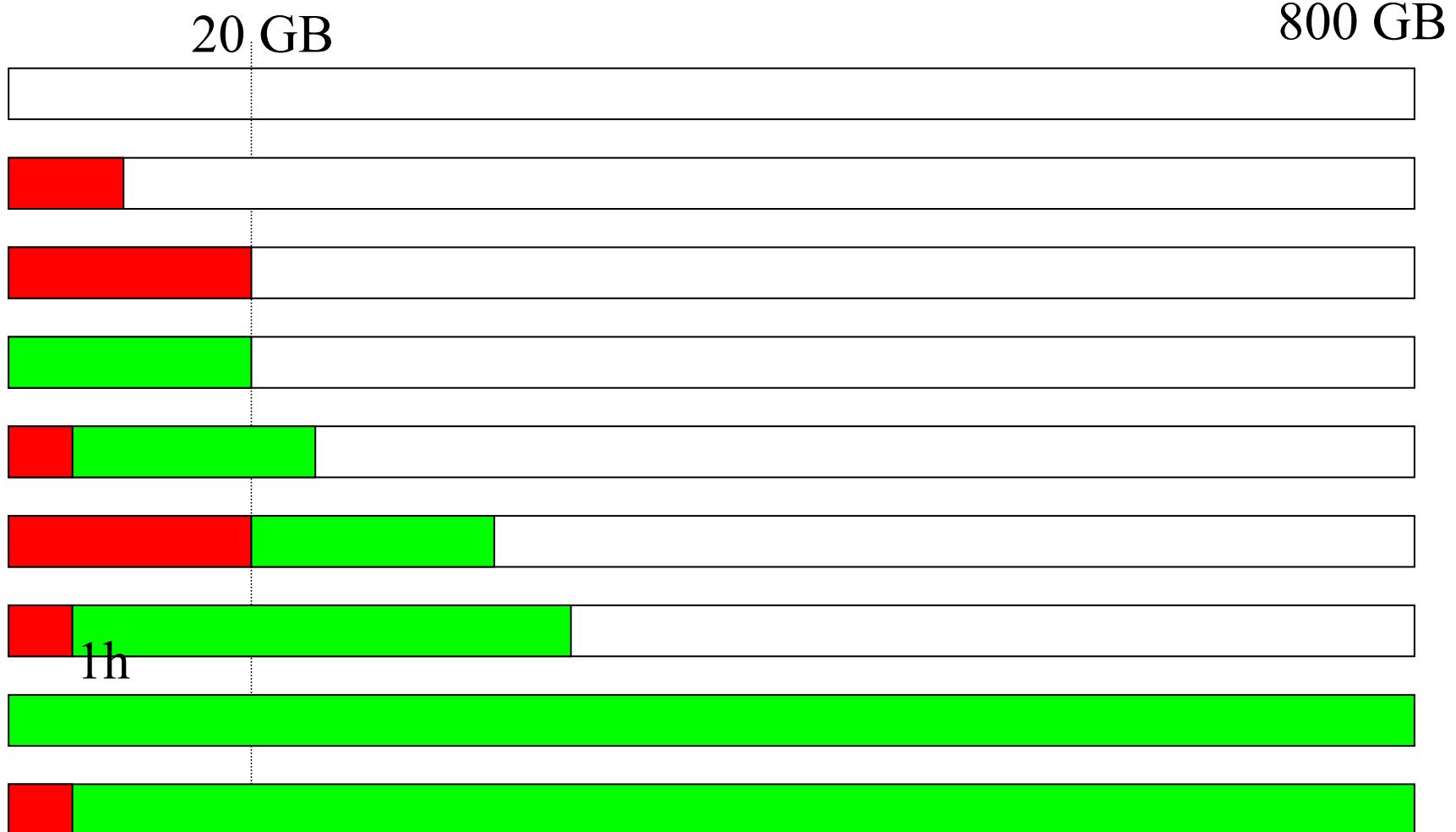


- TSM is not developed for archive
  - Interruption of TSM archive
  - No control what has been archived

## dCache tape access

- Convenient HSM connectivity (done for Enstore, OSM, TSM, bad for HPSS)
- Creates a separate session for every file
- Transparent access
- Allows transparent maintenance at HSM

## dCache pool node



## dCache tape management

- Precious data is separately collected per 'storage class'
- Each 'storage class queue' has individual parameters, steering the tape flush operation.
  - Maximum time, a file is allowed to be 'precious' per 'storage class'.
  - Maximum number of precious bytes per 'storage class',
  - Maximum number of precious files per 'storage class',
- Maximum number of simultaneous 'tape flush' operations can be configured

## Conclusion and Future Work

- Low cost read pools
- Reliable write pools
- Write once never change a dCache file
- Single point of failure
- Working SRM connection between CERN and FZK
- Connection to openlab at CERN
- Adding 15 Pool nodes for the 10 Gbit test from SRM to SRM
- More at [www.dcache.org](http://www.dcache.org)

Thank you for your attention!  
Questions?