dCache 1.9

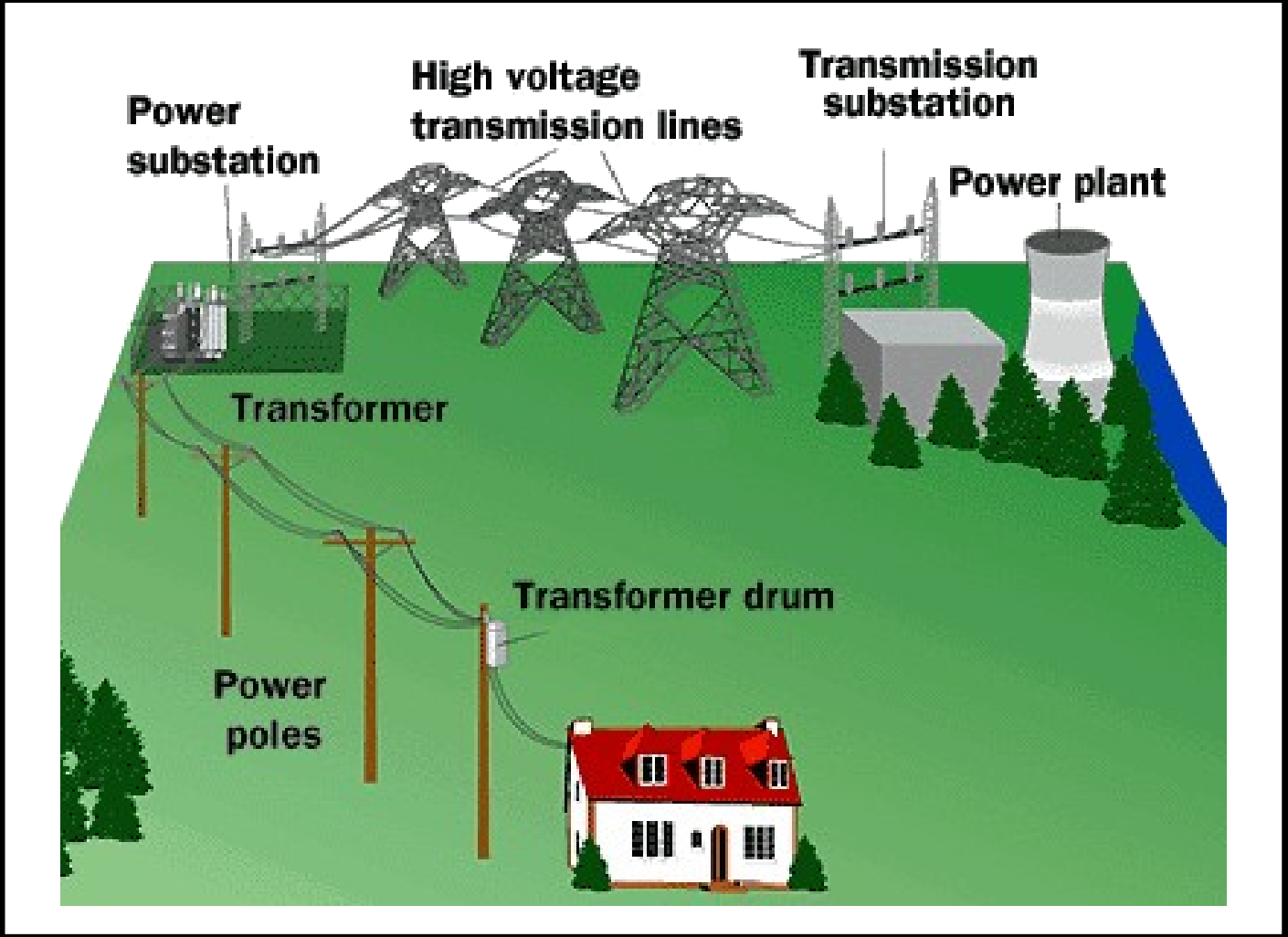
DESY, FNAL, NDGF

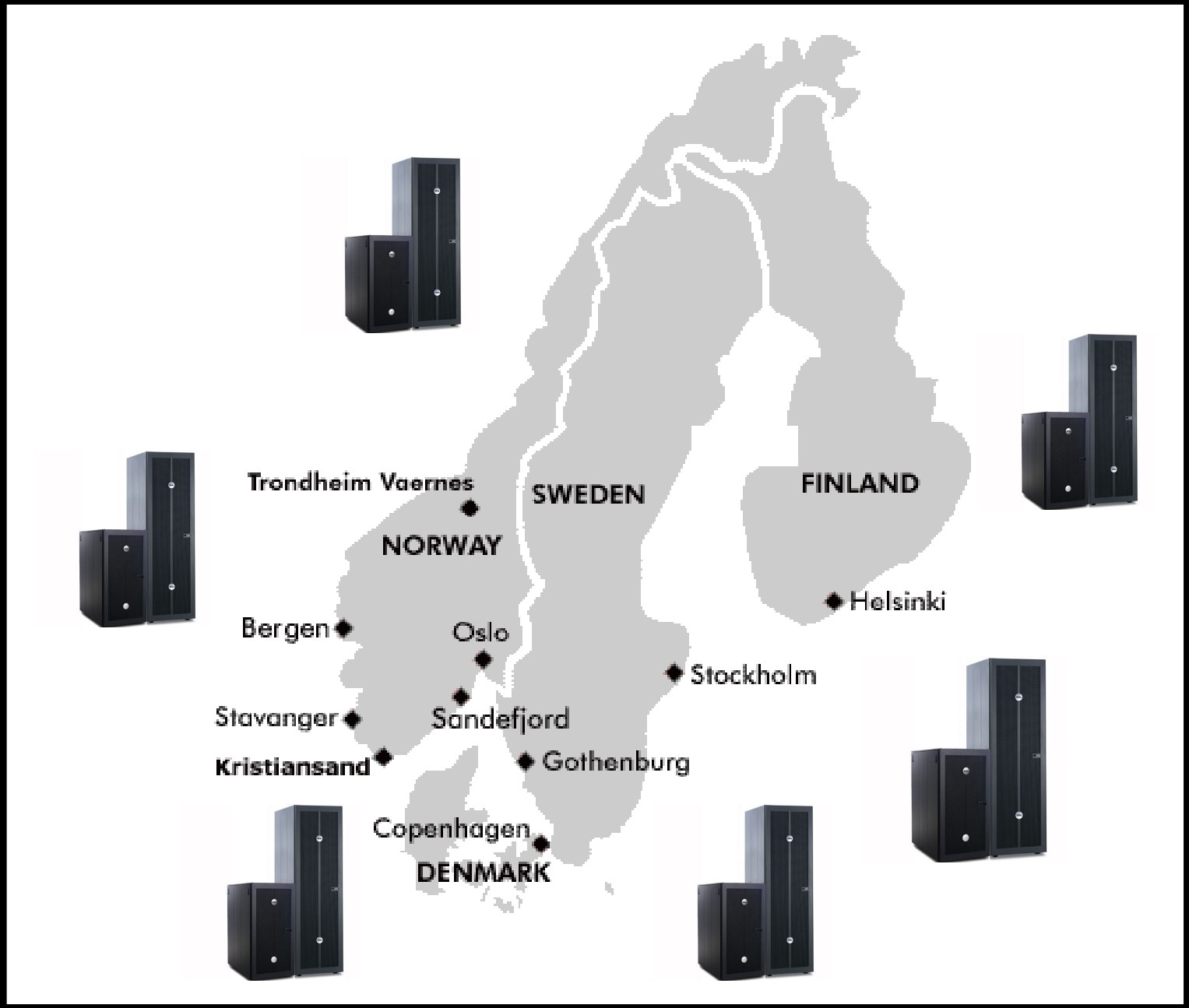
&

NDGF Contributions to dCache

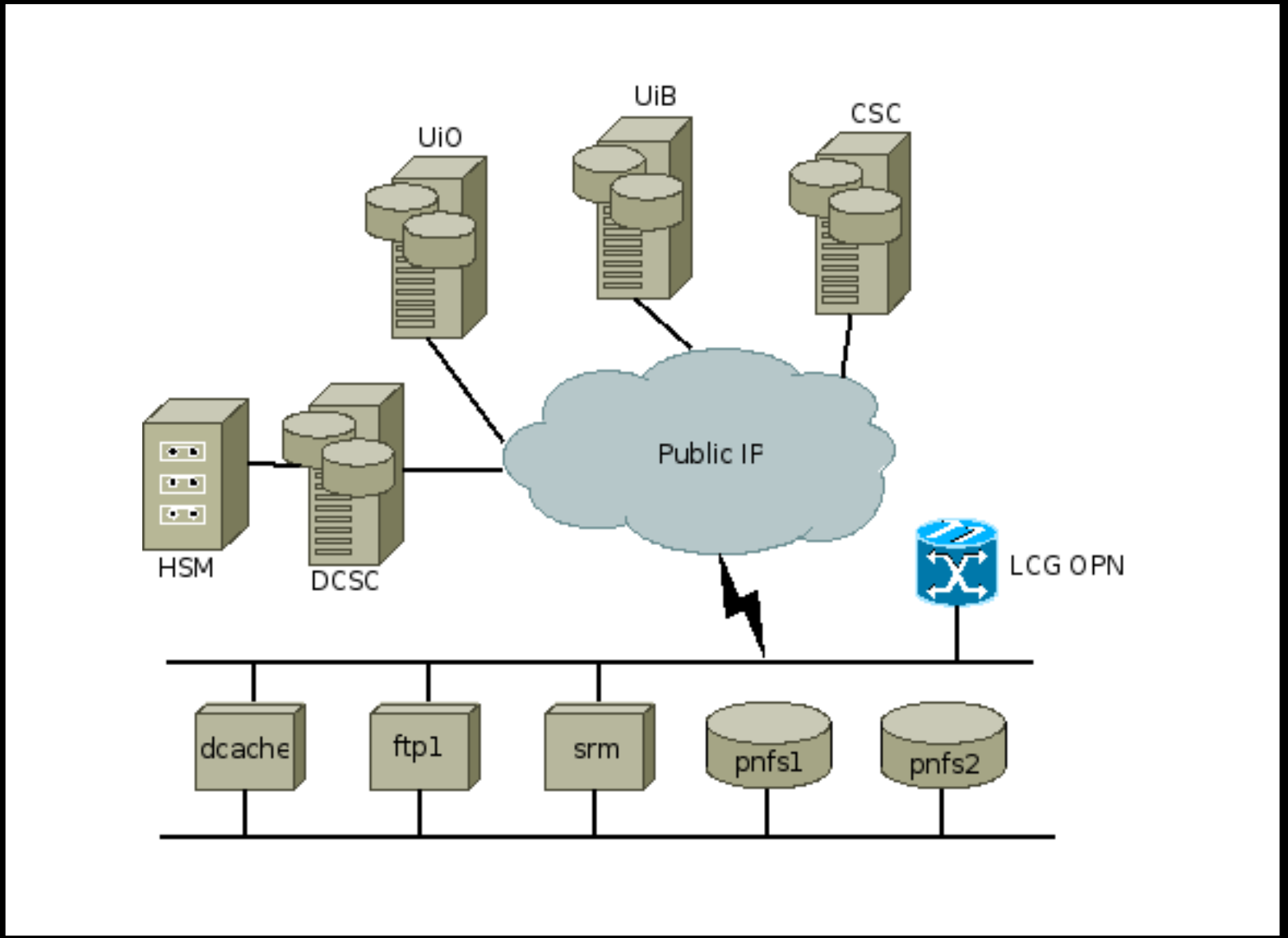
Gerd Behrmann

NDGF
Nordic DataGrid Facility





NDGF
Nordic DataGrid Facility


UiB
CSC
UiO
Public IF
HSM
DCSC
LCG OPN
dcache
ftp1
srm
pnfs1
pnfs2

What you need to host a pool

- Java 5 or 6
- A directory with a lot of storage
- dCache pool code installed

- Free to choose HSM
- Free to choose OS
- Free to choose CPU architecture
- Free to choose storage architecture
- Free to choose storage density
- …

- Limited bandwidth
- High latency
- Frequent network failures
- Spanning many administrative domains

Security

- Many administrative domains
- Local and national rules
- Internal node communication over WAN
- Mounting NFS over WAN is out of the question

Administration

- Site administrators are worried about loosing control
- Mechanisms for delegating control over local ressources

Maintenance

- Upgradability
- Autonomeous operation

Reliability

- dCache is fairly resilient against pool failures
- Head nodes provide single point of failure
- Network saparation in WAN
- Disconnected operation (at least read-only)
- Long term hope that dCache becomes less centralised

Performance

- No network model
- Proxy operation of GridFTP

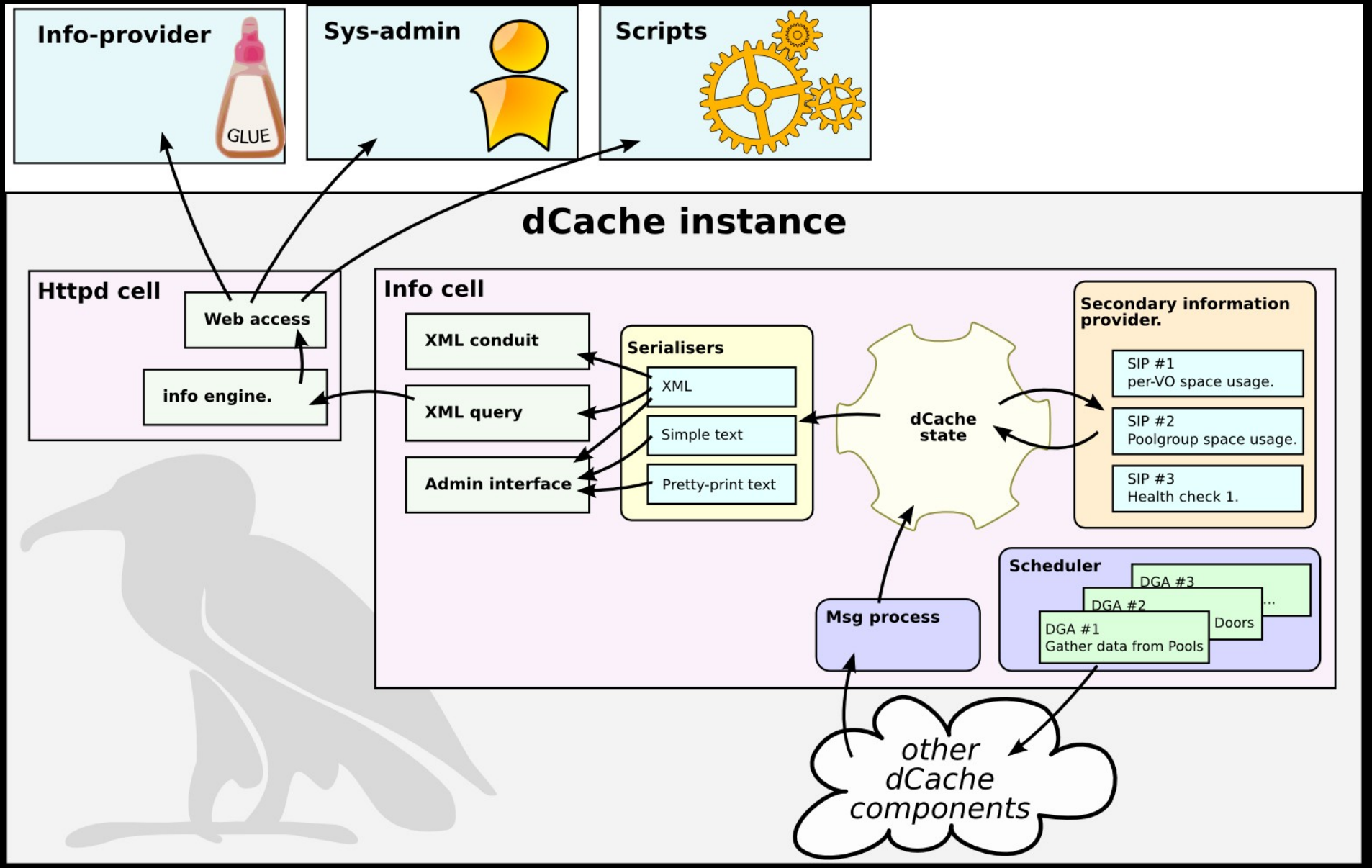
Functionality

- HSM without PNFS (dCache 1.8)
- Heterogenous access to HSM
- Tivoli (TSM) integration

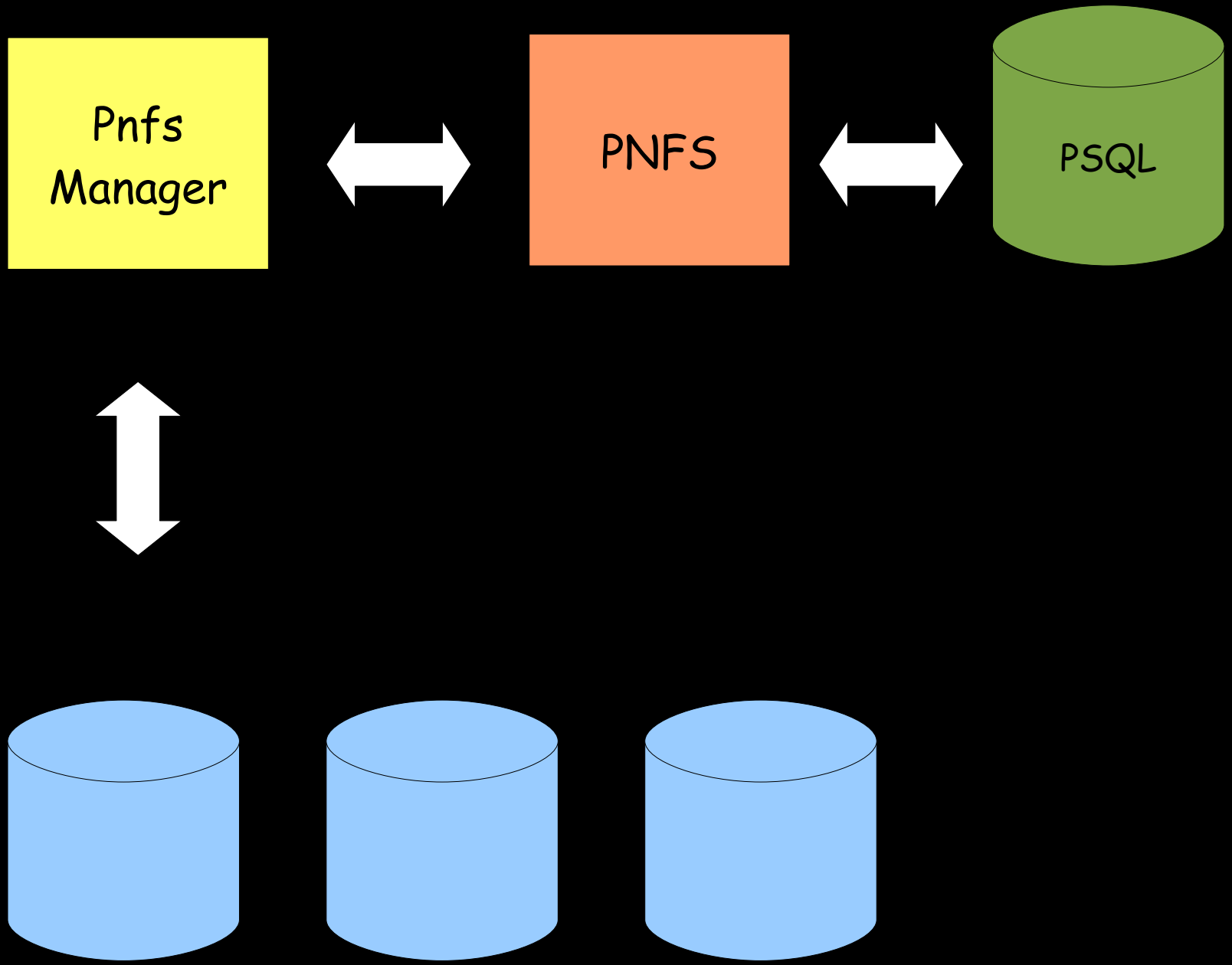
Releases

- 1.9.0
  - Info service, delete registration, logging
- 1.9.1
  - New pool, logging, reservations in info provider
- 1.9.2
  - gPlazma updates, unpin by VO
- 1.9.2+
  - ACL support for files and space reservations

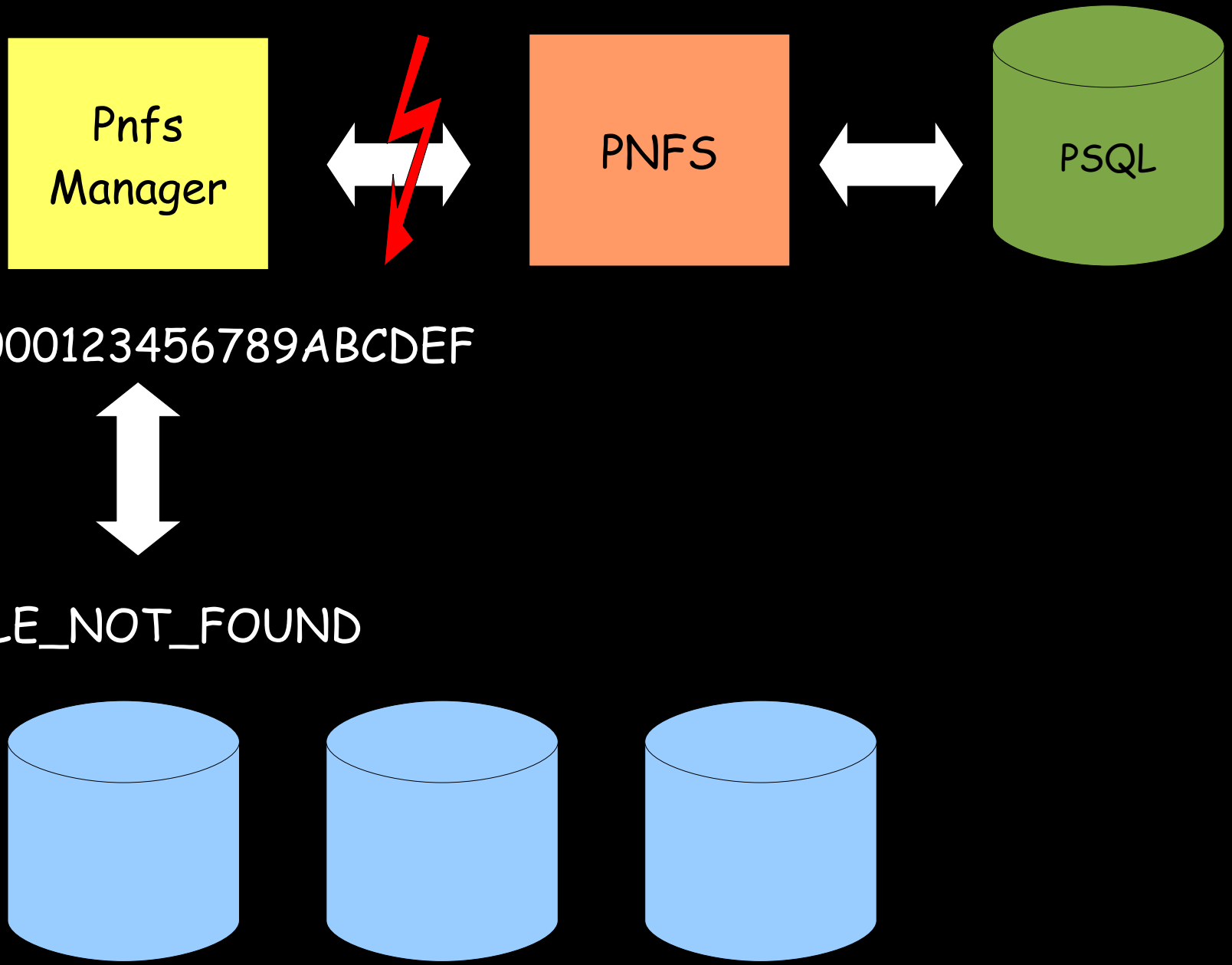
Info Service



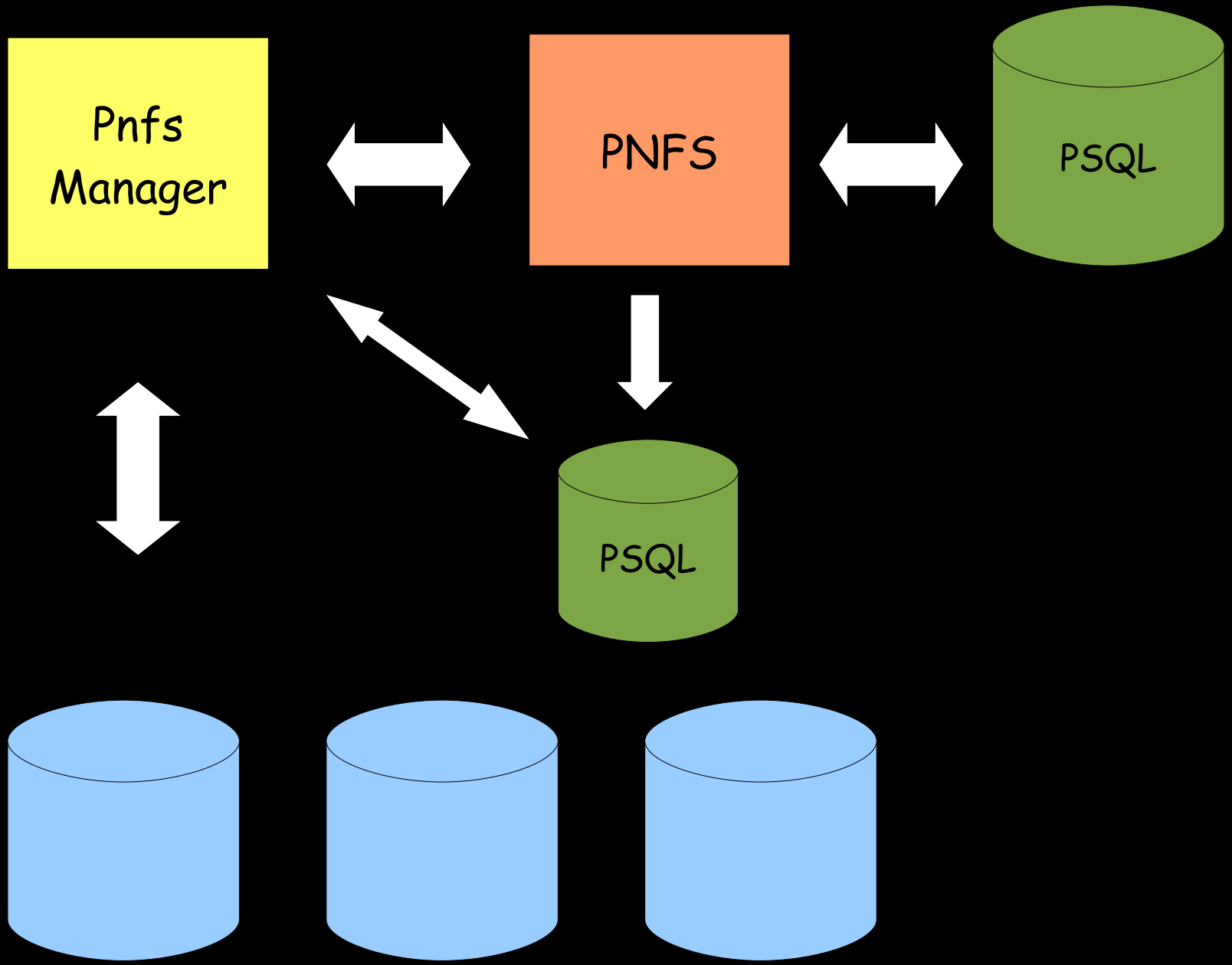
Delete Registration



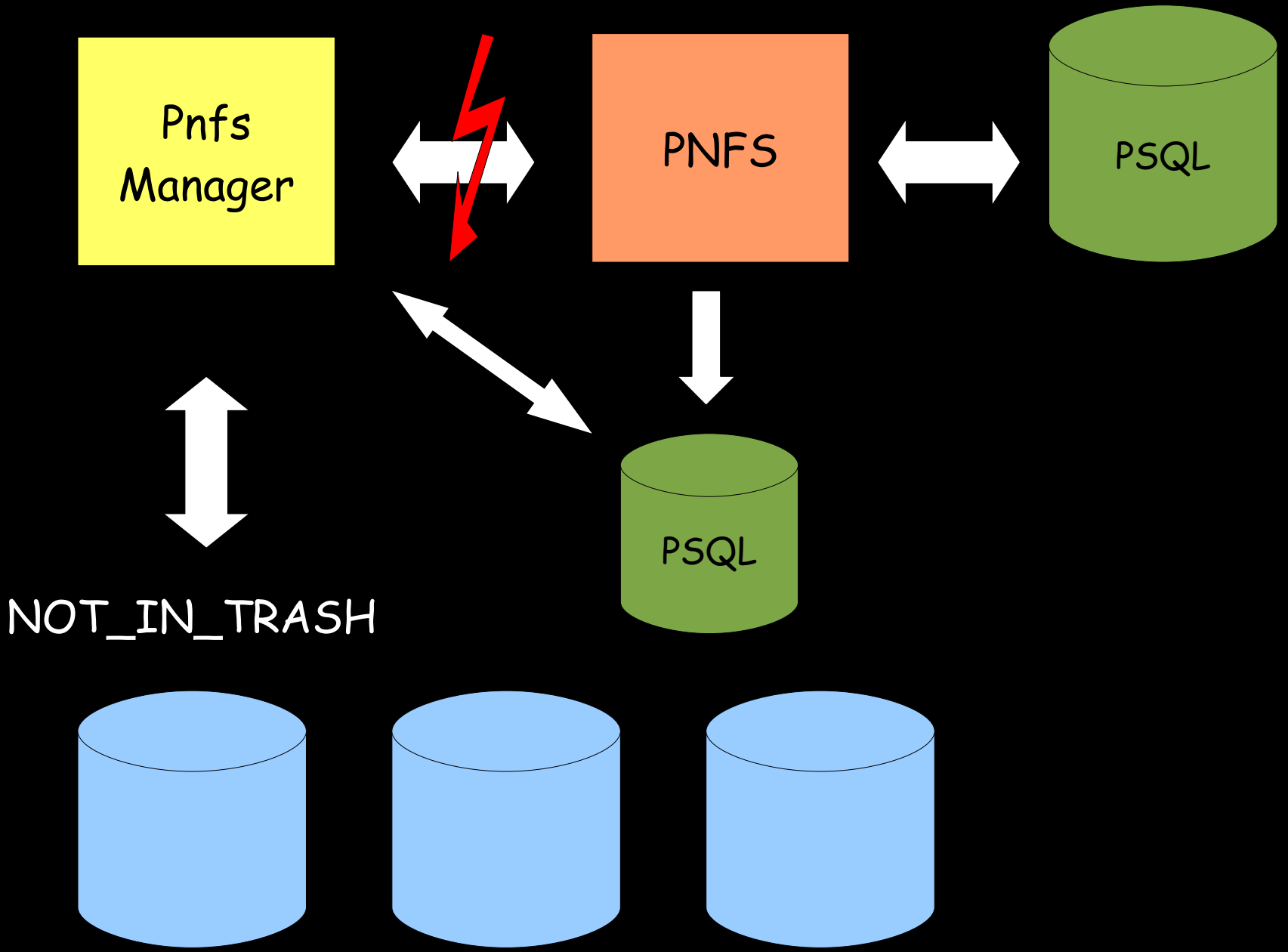
Delete Registration



Delete Registration



Delete Registration

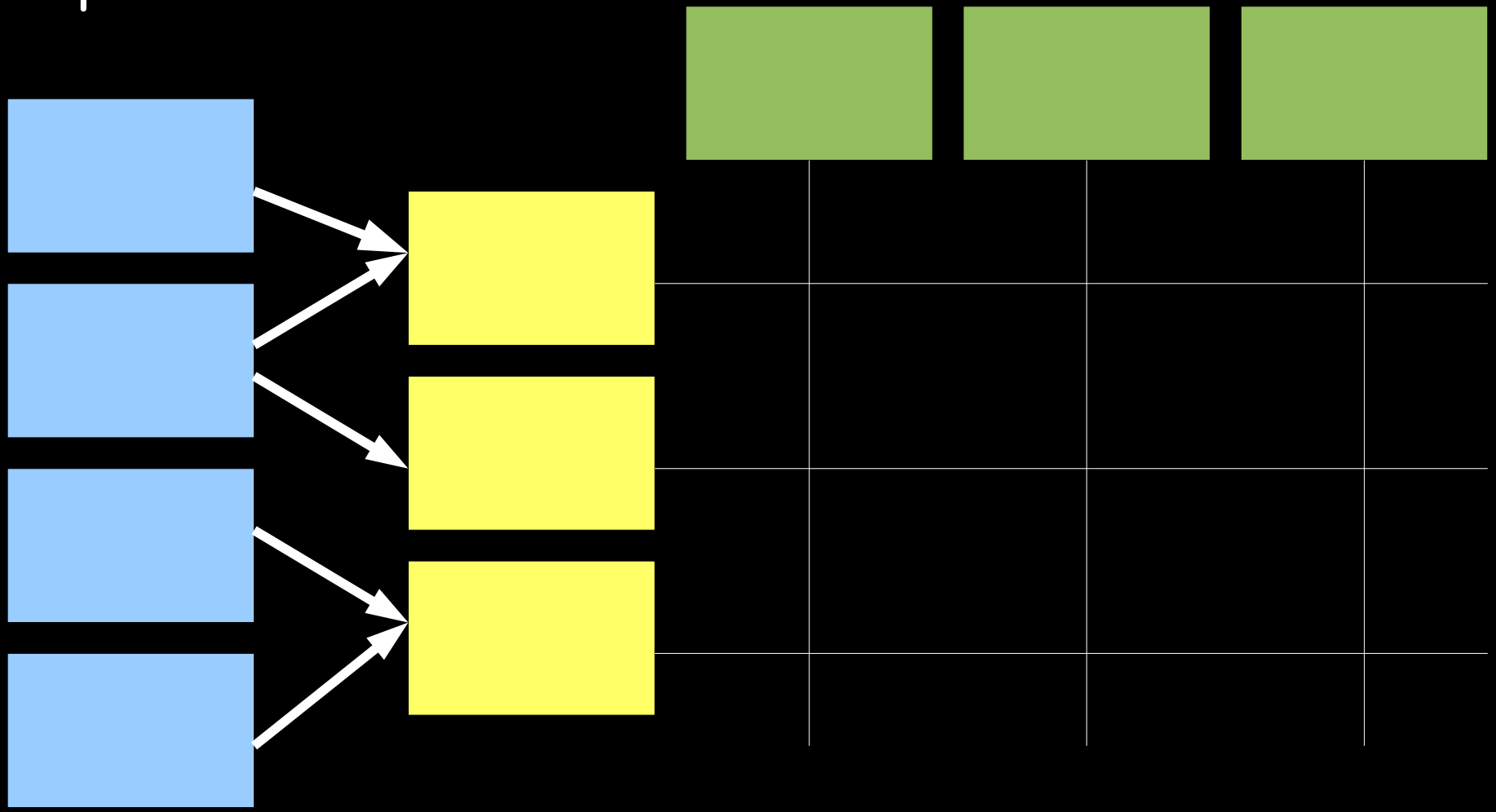


Log4j

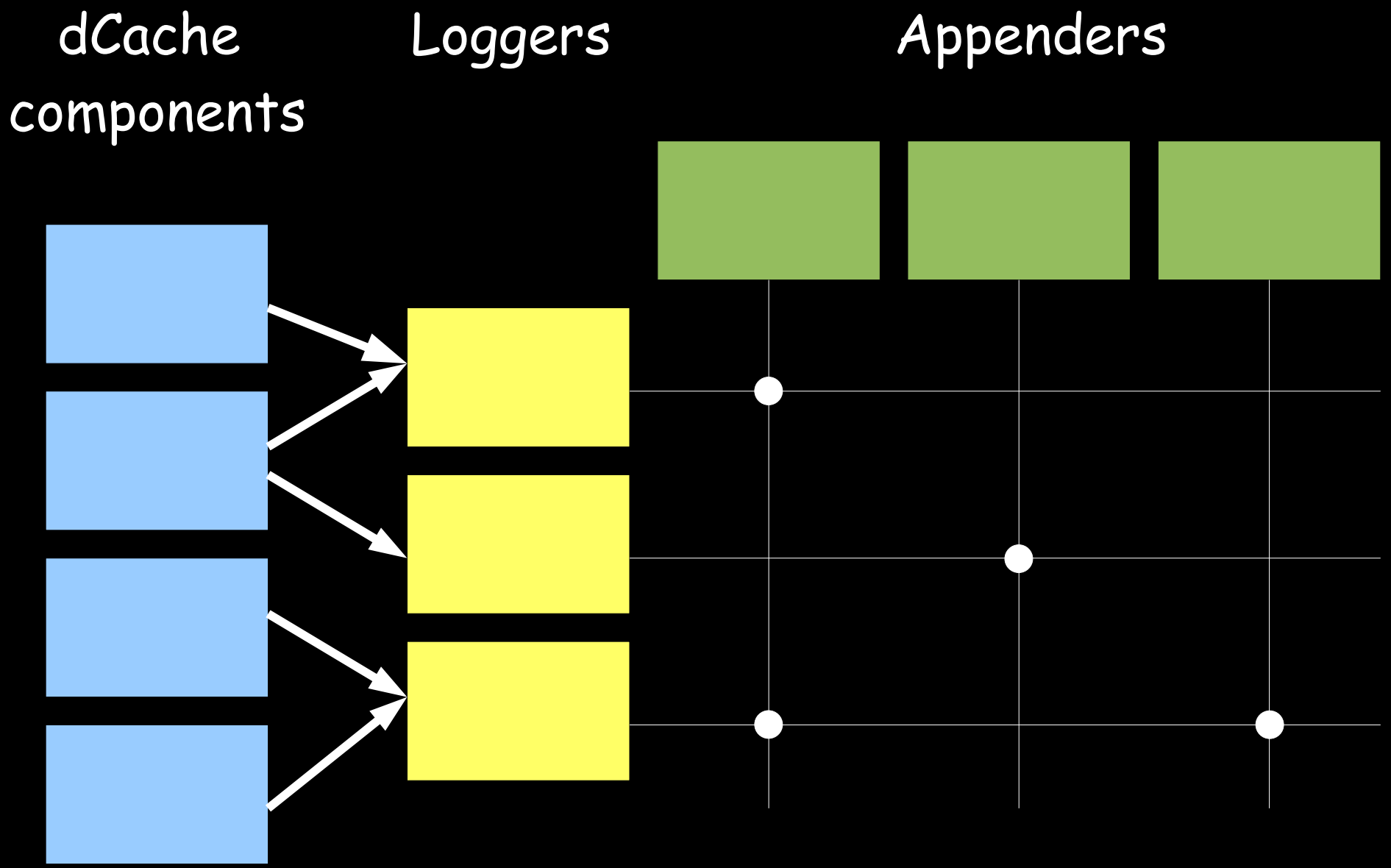
dCache components

Loggers

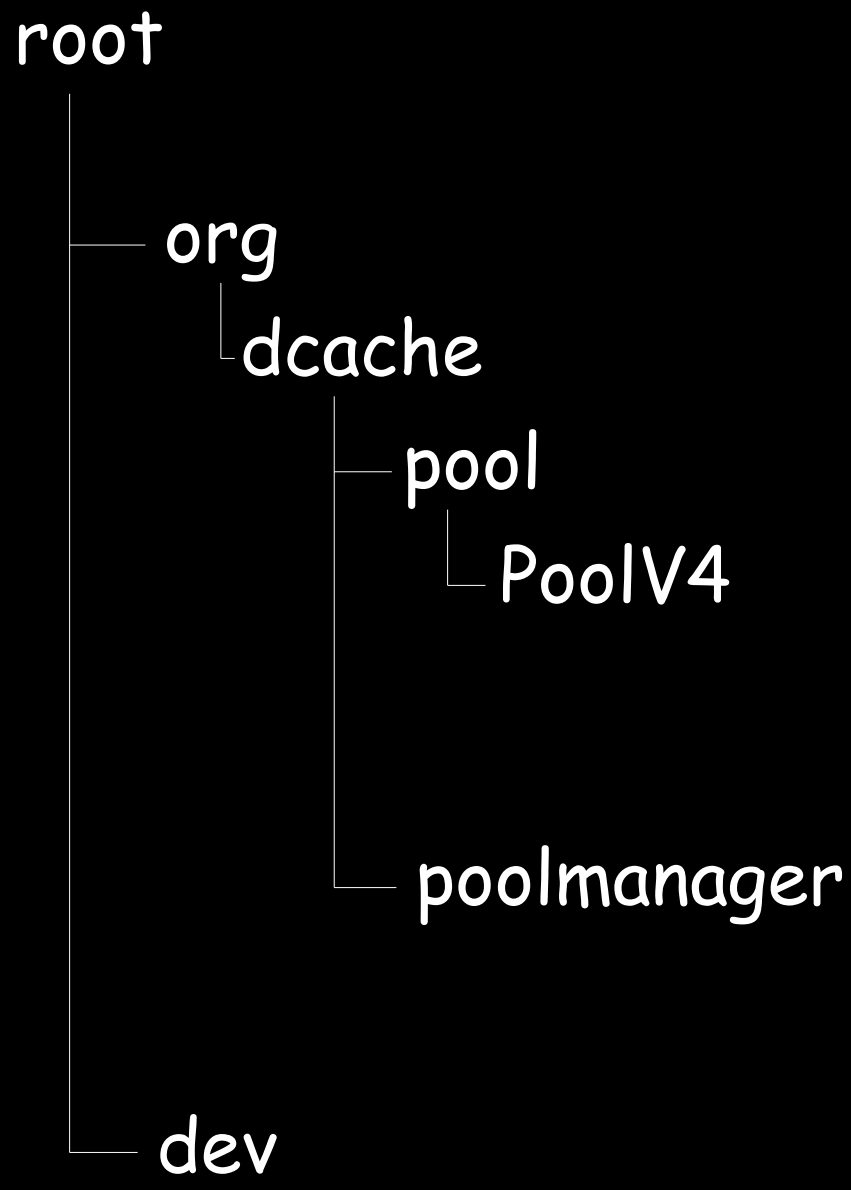
Appenders



Log4j



Log4j Logger Names



New Pool

- Drop in replacement
- Minimal user visible changes
- Much more modular
- Fixes the space calculation issue (we hope)

Migration Module

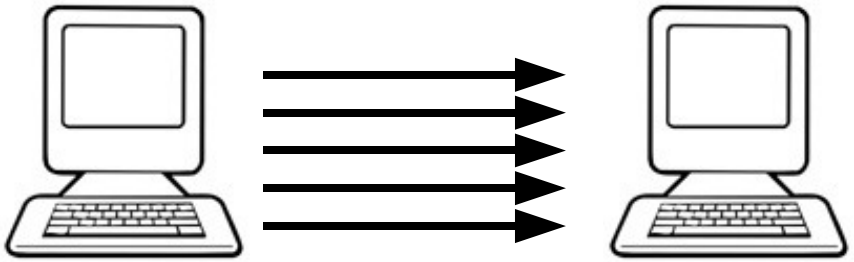
- Embedded in the new pool
- Single-command-copy
- Idempotent
- No persistent state
- Preserves sticky flags
- Can modify both source and target state

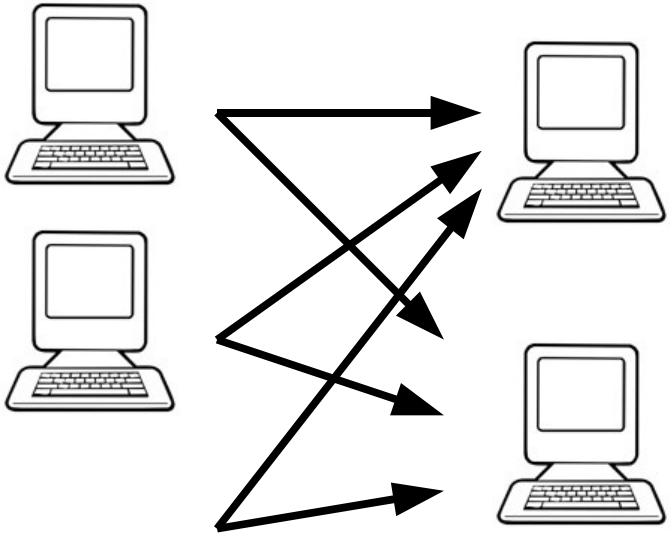
Releases

- 1.9.0
  - Info service, delete registration, logging
- 1.9.1
  - New pool, logging, reservations in info provider
- 1.9.2
  - gPlazma updates, unpin by VO
- 1.9.2+
  - ACL support for files and space reservations

Let's talk GFD.47

GFD.20 aka GridFTP



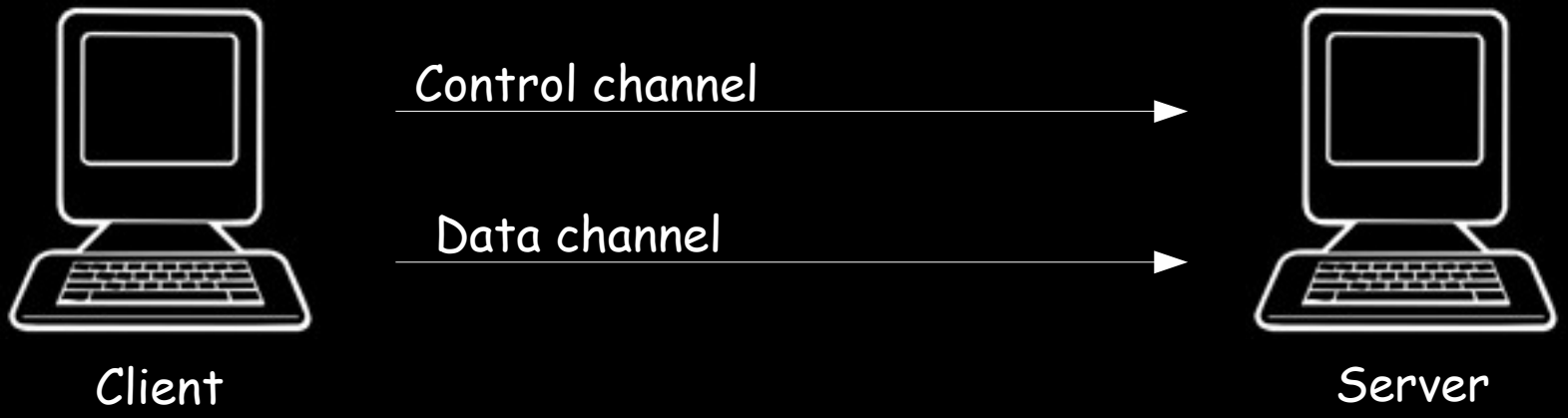


ERET

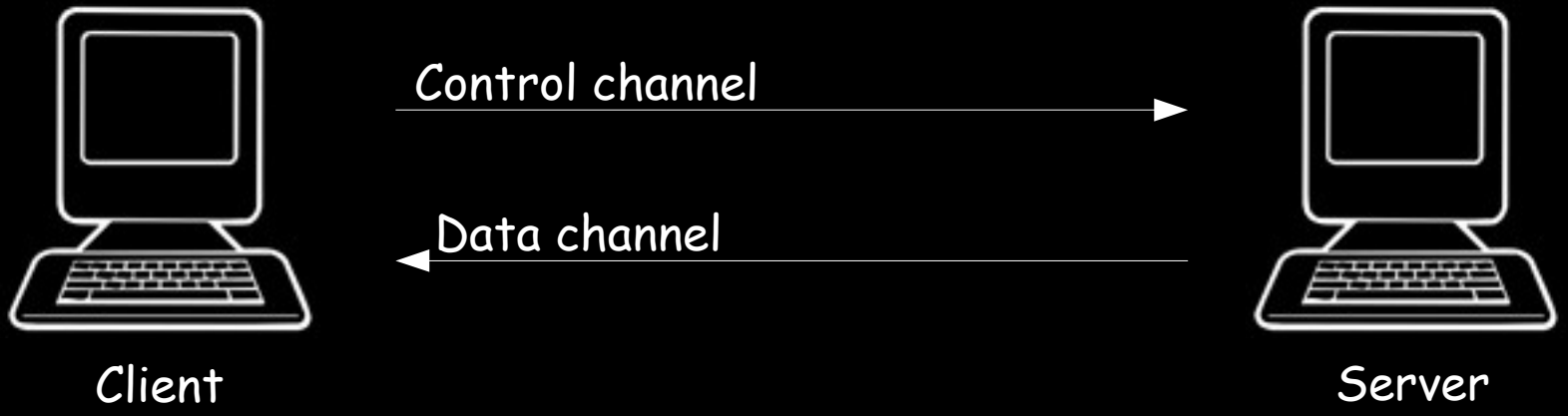
ABUF ESTO

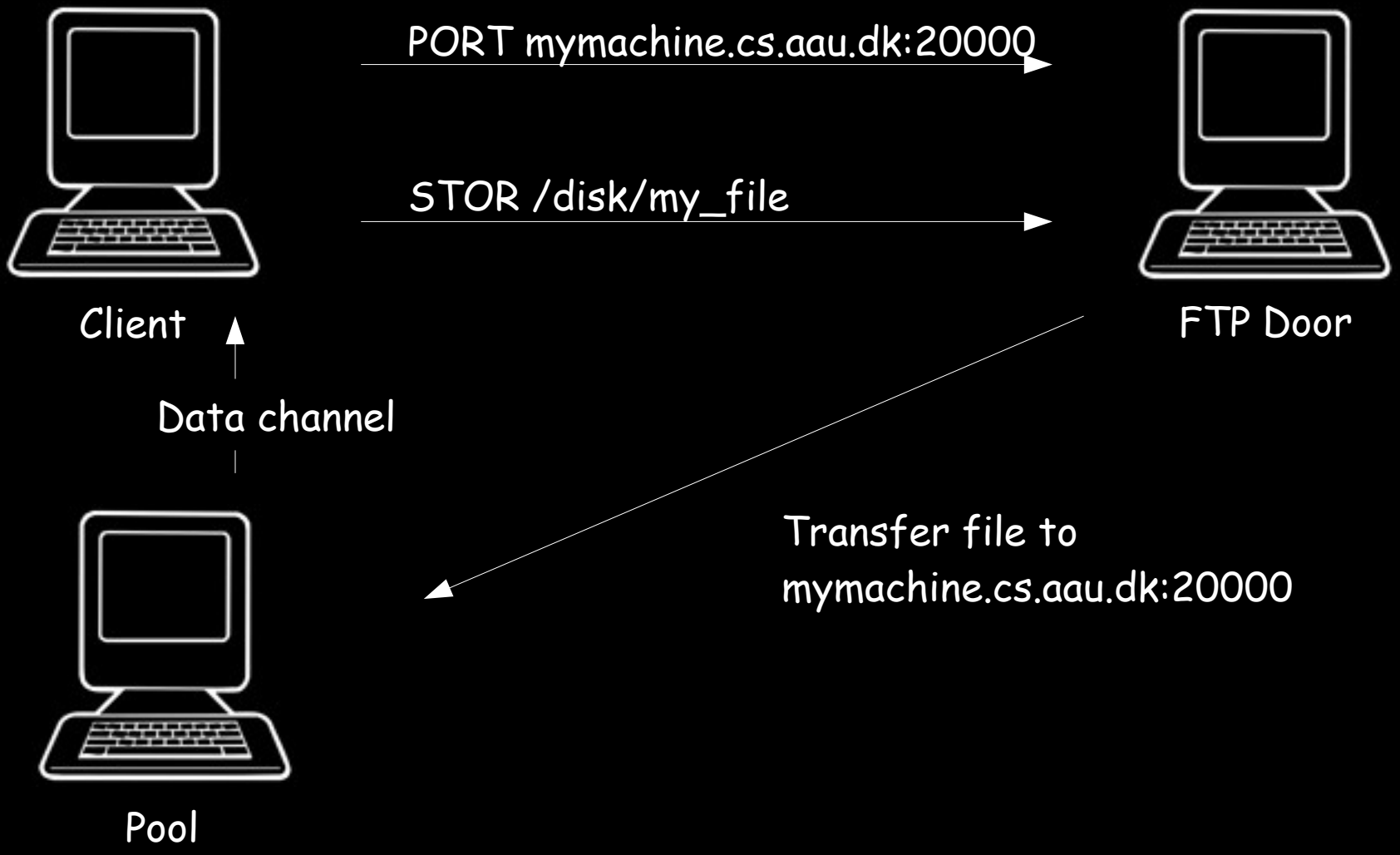
SBUF

Passive servers

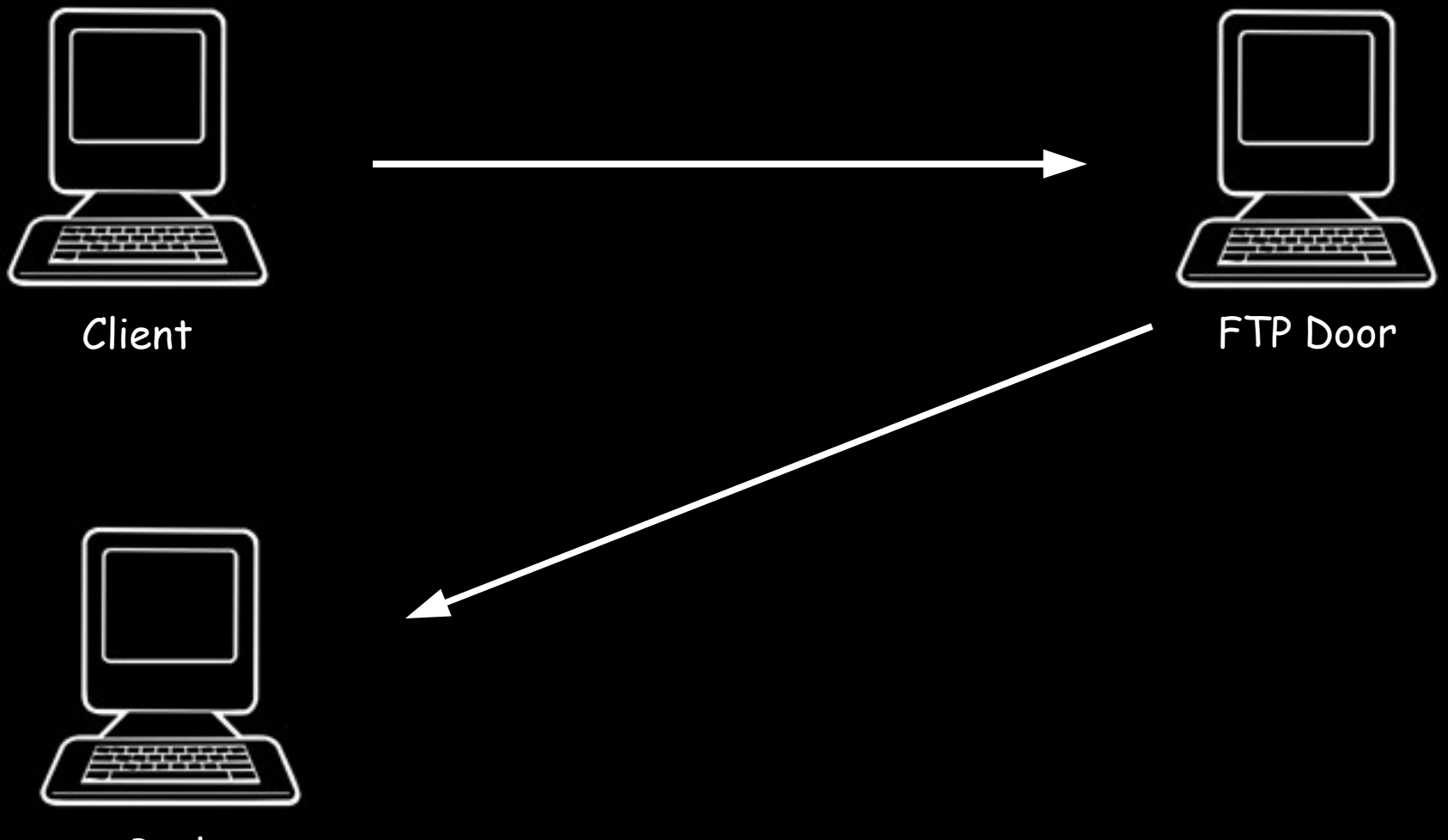


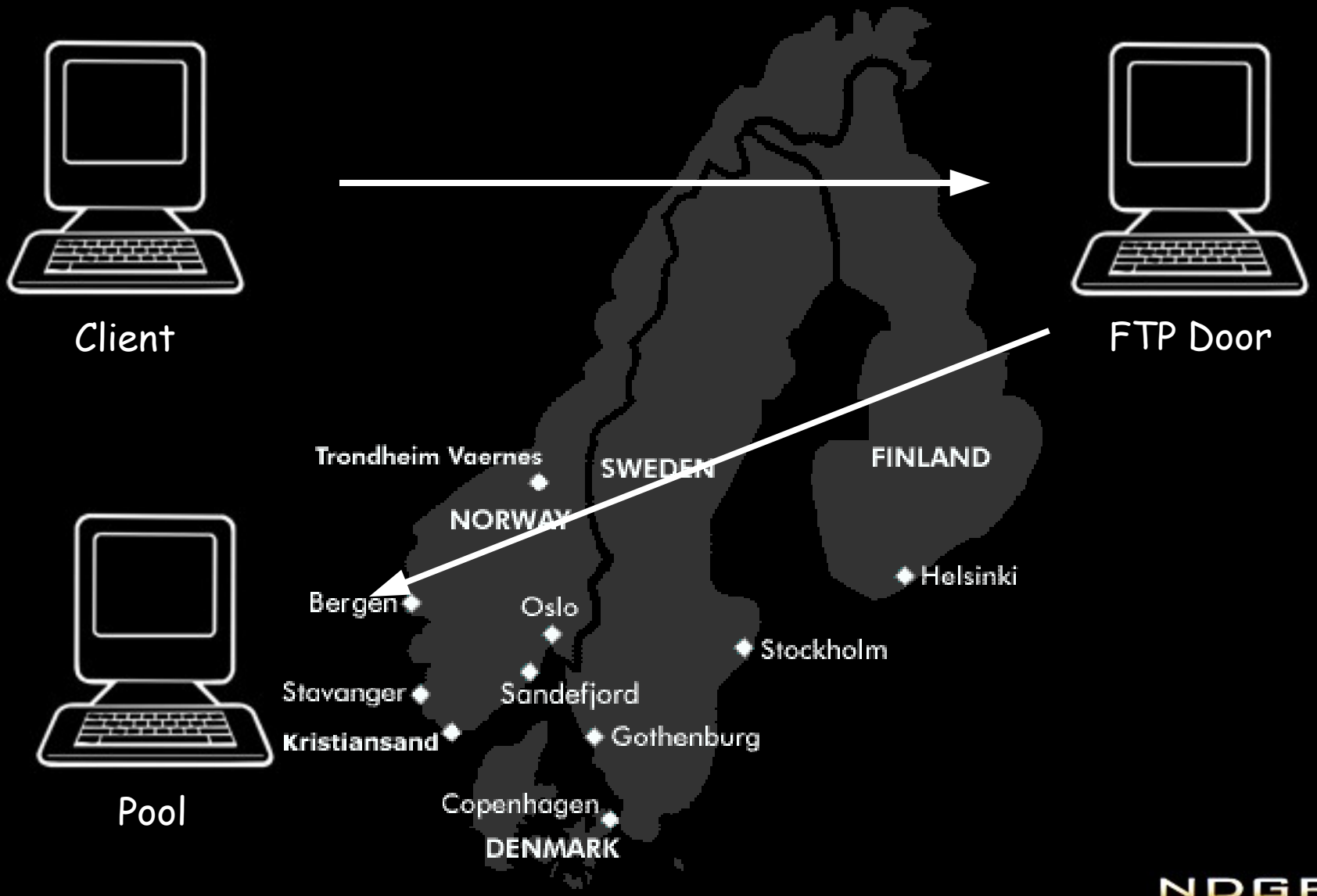
Active servers



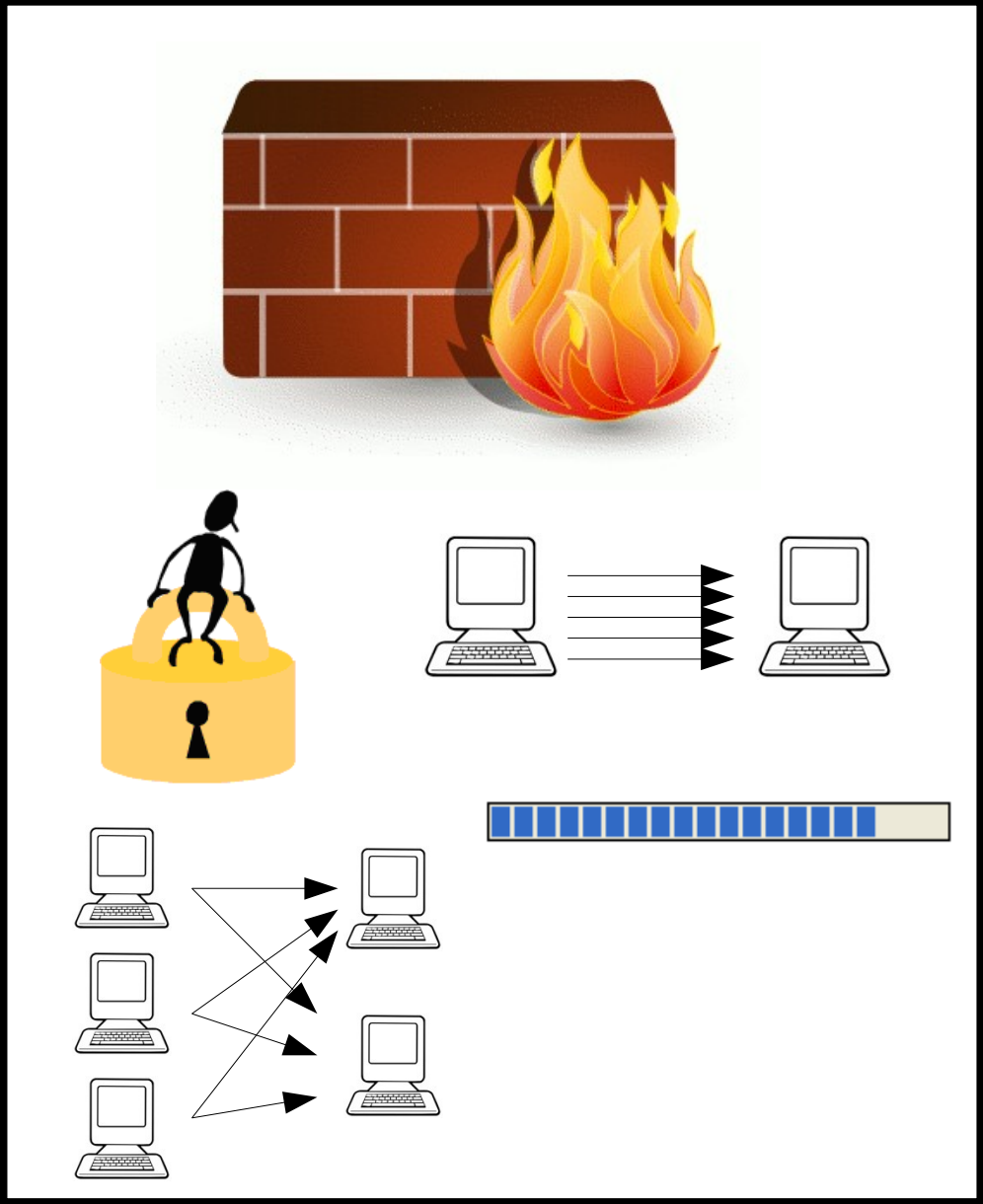

PORT mymachine.cs.aau.dk:20000
STOR /disk/my_file
Client
FTP Door
Data channel
Transfer file to
mymachine.cs.aau.dk:20000
Pool


PASV
ftp1.ndgf.org:20000
RETR /disk/my_file
Client
FTP Door
Pool


Client
FTP Door
Pool



NDGF
Nordic DataGrid Facility



GFD.47 a.k.a. GridFTP v2

- CKSUM
- EOF
- MODEX
- GETPUT
- STREAMING

Active file retrieval in Stream mode:

Client	Server
GET path=/tmp/file.dat;port=34,23,45,12,48,14;mode=s;	
	1xx Data connection established
	2xx Transfer complete

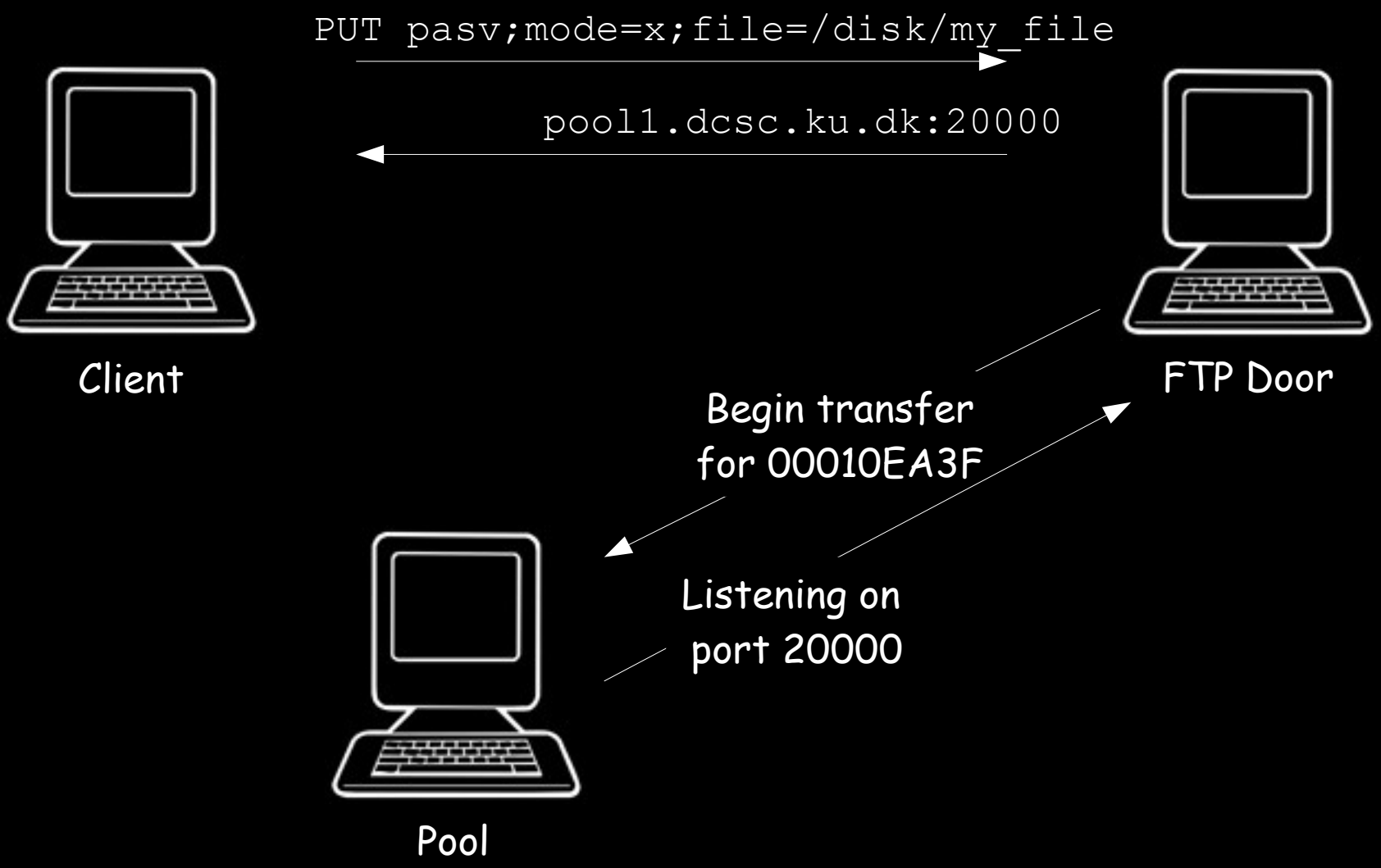
Passive file retrieval in E mode:

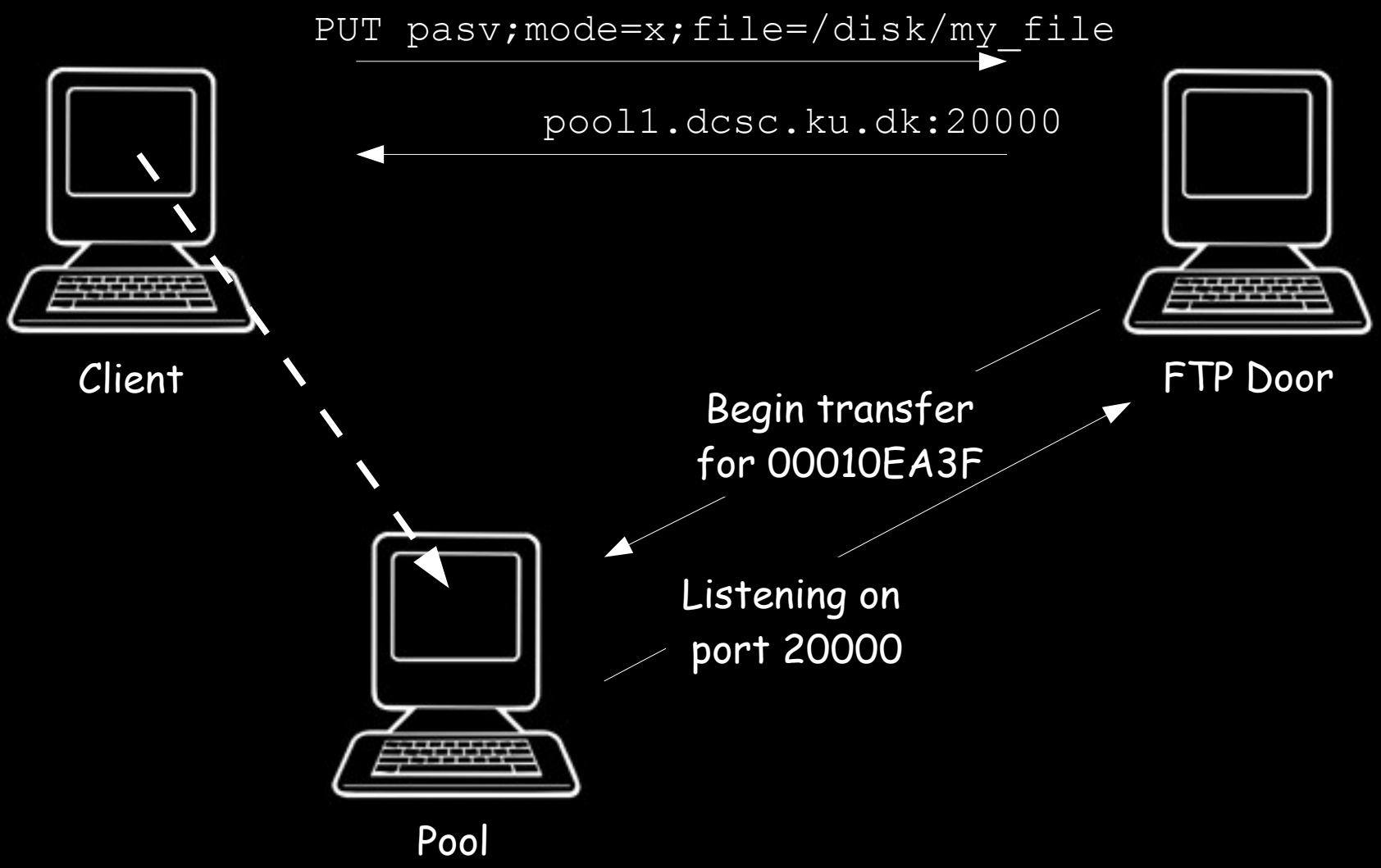
Client	Server
GET path=/tmp/file.dat;pasv;mode=e;	
	1xx wait
	1xx wait
	1xx PORT=134,23,145,2,48,114
	1xx Data connection established
	2xx Transfer complete

Passive file upload in X mode with MD5 signature calculation:

Client	Server
PUT path=/tmp/file.dat;pasv;mode=x;chksum=md5;	
	1xx wait
	1xx wait
	1xx PORT=134,23,145,2,48,114
	1xx Data connection established
	2xx Transfer complete

NDGF
NORDIC DATAGRID FACILITY

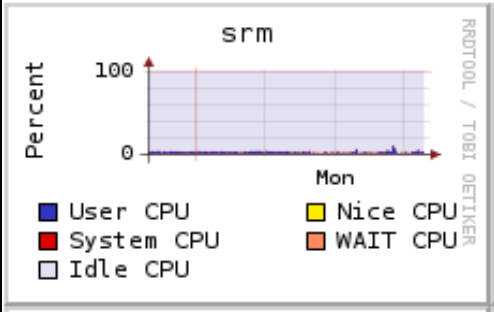

PUT pasv;mode=x;file=/disk/my_file
pool1.dcsc.ku.dk:20000
Client
FTP Door
Begin transfer
for 00010EA3F
Listening on
port 20000
Pool

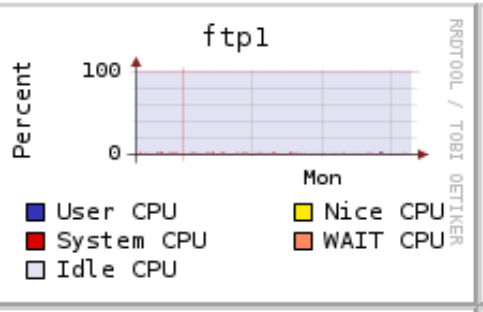

PUT pasv;mode=x;file=/disk/my_file
pool1.dcsc.ku.dk:20000
Client
FTP Door
Begin transfer
for 00010EA3F
Listening on
port 20000
Pool

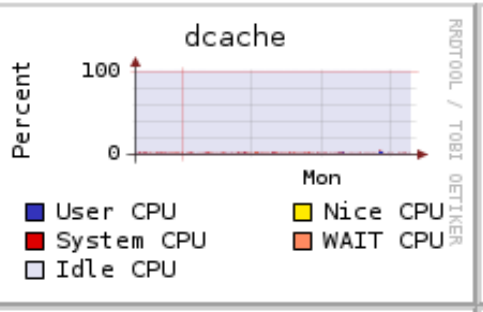
NDGF dashboard - Ganglia - last day

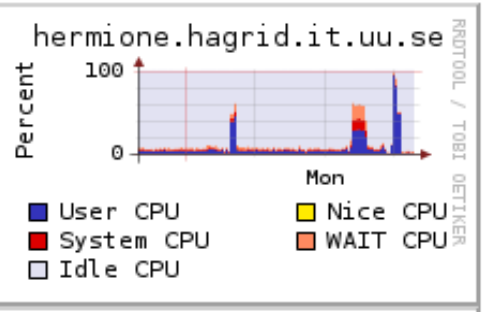
NDGF
Nordic DataGrid Facility

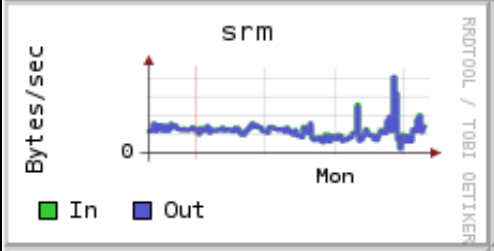
NDGF core services

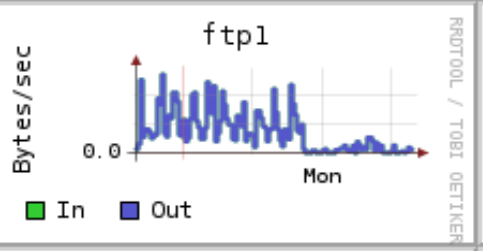


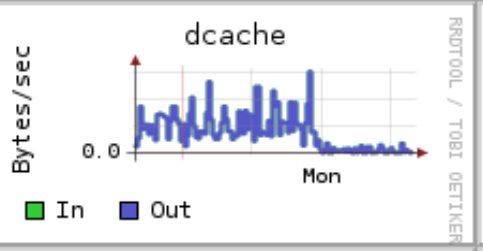


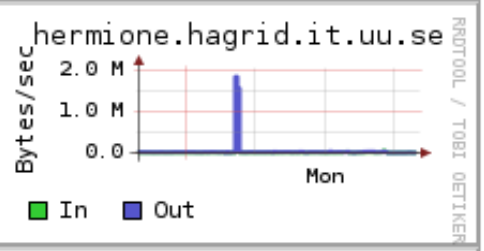


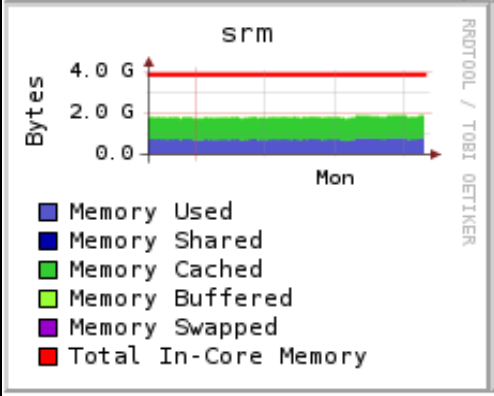


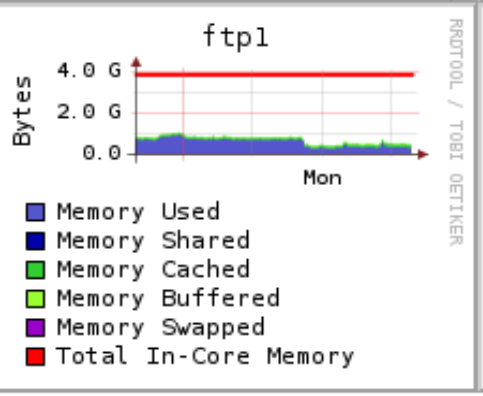


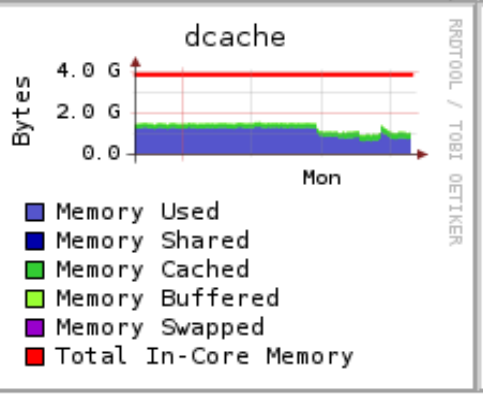


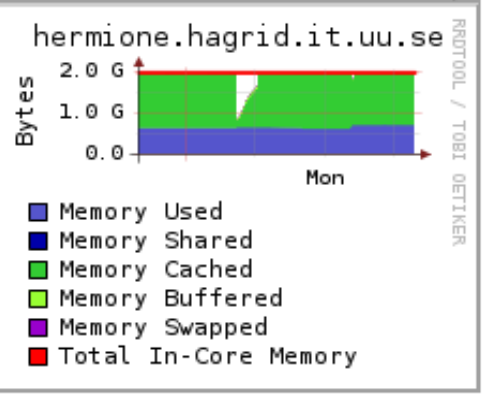












NDGF
Nordic DataGrid Facility

config/dCacheSetup

```
# ---- May pools accept incomming connection for GridFTP transfers?
#    Values: 'true', 'false'
#    Default: 'false' for FTP doors, 'true' for pools
#
#    If set to true, pools are allowed accept incomming connections for
#    for FTP transfers. This only affects passive transfers. Only passive
#    transfers using GFD.47 GETPUT (aka GridFTP 2) can be redirected to
#    the pool. Other passive transfers will be channelled through a
#    proxy component at the FTP door. If set to false, all passive
#    transfers to through a proxy.
#
#    This setting is interpreted by both FTP doors and pools, with
#    different defaults. If set to true at the door, then the setting
#    at the individual pool will be used.
#
# gsiftpAllowPassivePool=false
```

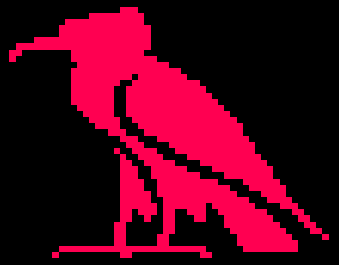
REPEAT AFTER ME

GFD.47
is **NOT** the
Globus GridFTP daemon
release 2.0


NDGF
Nordic DataGrid Facility


the globus toolkit


gLite




Fermilab


DESY


KnowARC

New cache repository in dCache 1.8

or

How to reduce memory consumption and increase startup speed

Gerd Behrmann

NDGF
Nordic DataGrid Facility

Traditional layout of pool directory

```
|-- RepositoryOk
|-- control
|   |-- 000100000000000018EA20
|   |-- 000100000000000018EA48
|   |-- SI-000100000000000018EA20
|   `-- SI-000100000000000018EA48
|-- data
|   |-- 000100000000000018EA20
|   `-- 000100000000000018EA48
`-- setup
```

Berkeley DB based cache repository

```
|-- RepositoryOk
|-- meta
|   |-- 00000000.jdb
|   `-- je.lck
|-- data
|   |-- 0001000000000000018EA20
|   `-- 0001000000000000018EA48
`-- setup
```

```
#  ---- Which meta data repository implementation to use.
#    Values: org.dcache.pool.repository.meta.file.FileMetaDataRepository
#            org.dcache.pool.repository.meta.db.BerkeleyDBMetaDataRepository
#    Default: org.dcache.pool.repository.meta.file.FileMetaDataRepository
#
#   Selects which meta data repository implementation to use. This is
#   essentially a choice between storing meta data in a large number
#   of small files in the control/ directory, or to use the embedded
#   Berkeley database stored in the meta/ directory (both directories
#   placed in the pool directory).
#
# metaDataRepository=org.dcache.pool.repository.meta.file.FileMetaDataRepository
#
```

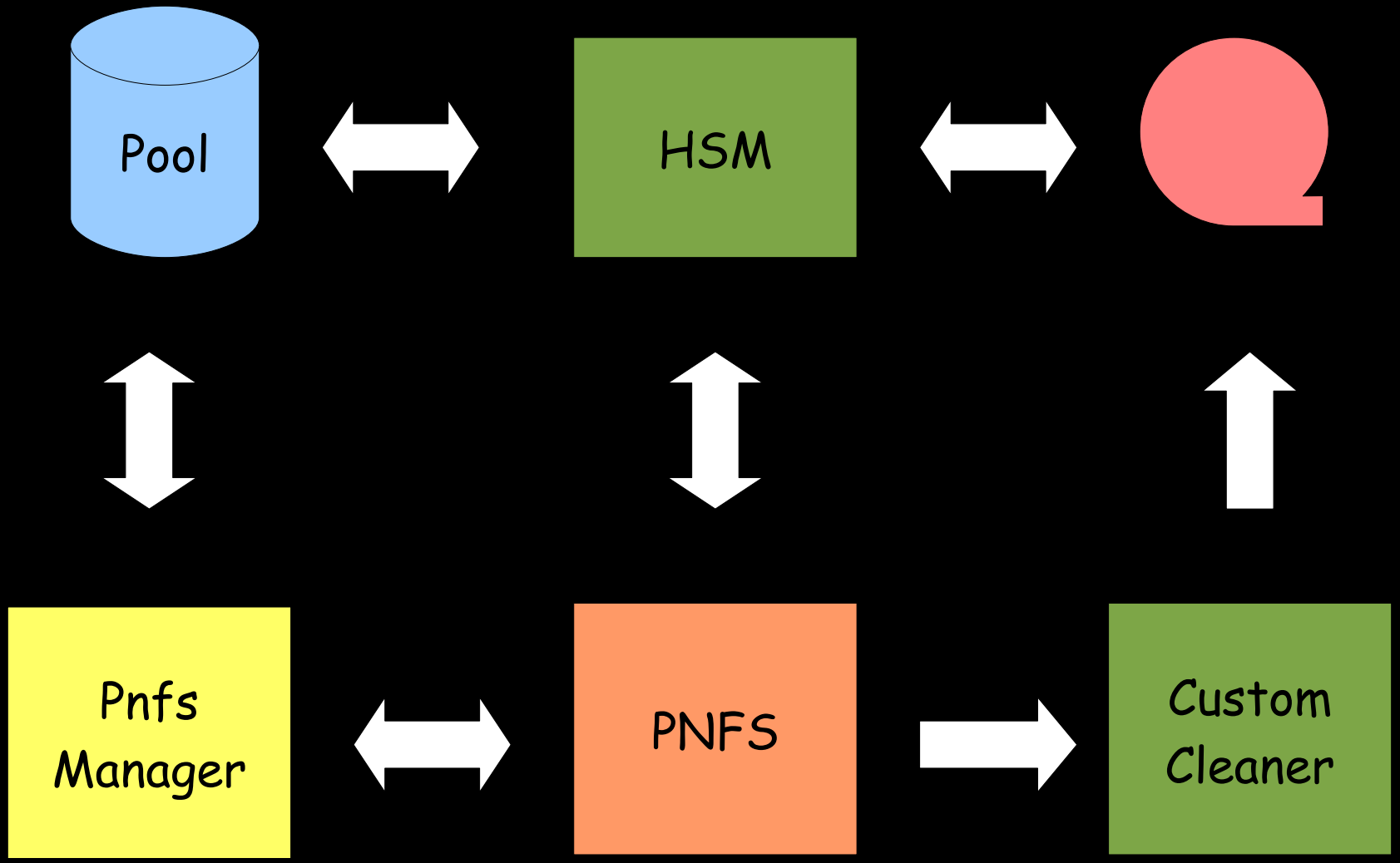
NDGF
NORDUGRID DATAGRID FACILITY

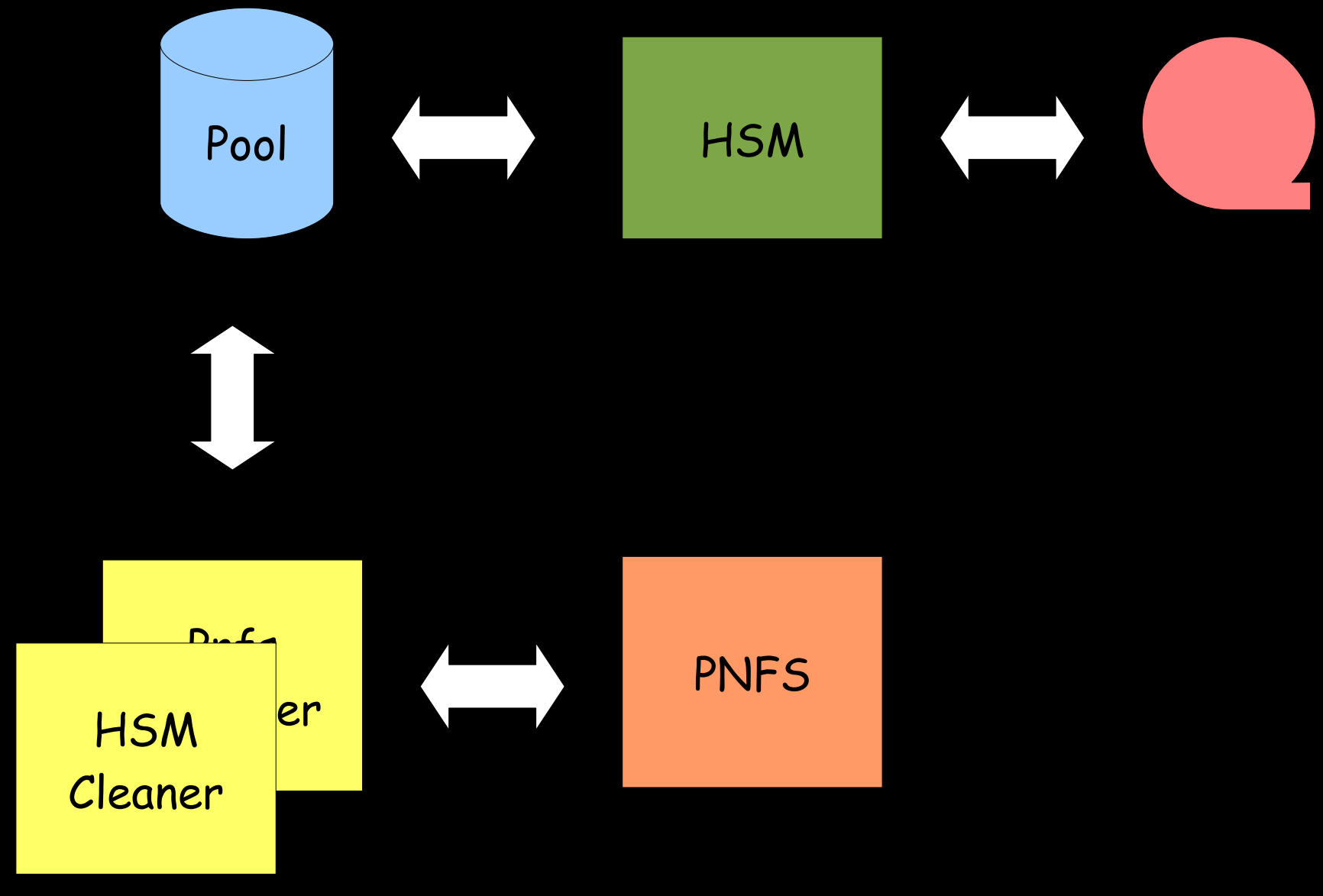
```
#  ---- Which meta data repository to import from.
#    Values: org.dcache.pool.repository.meta.file.FileMetaDataRepository
#            org.dcache.pool.repository.meta.db.BerkeleyDBMetaDataRepository
#    Default:
#
#   Selects which meta data repository to import data from if the
#   information is missing from the main repository. This is useful
#   for converting from one repository implementation to another,
#   without having to fetch all the information from the central PNFS
#   manager.
#
# metaDataRepositoryImport=""
```

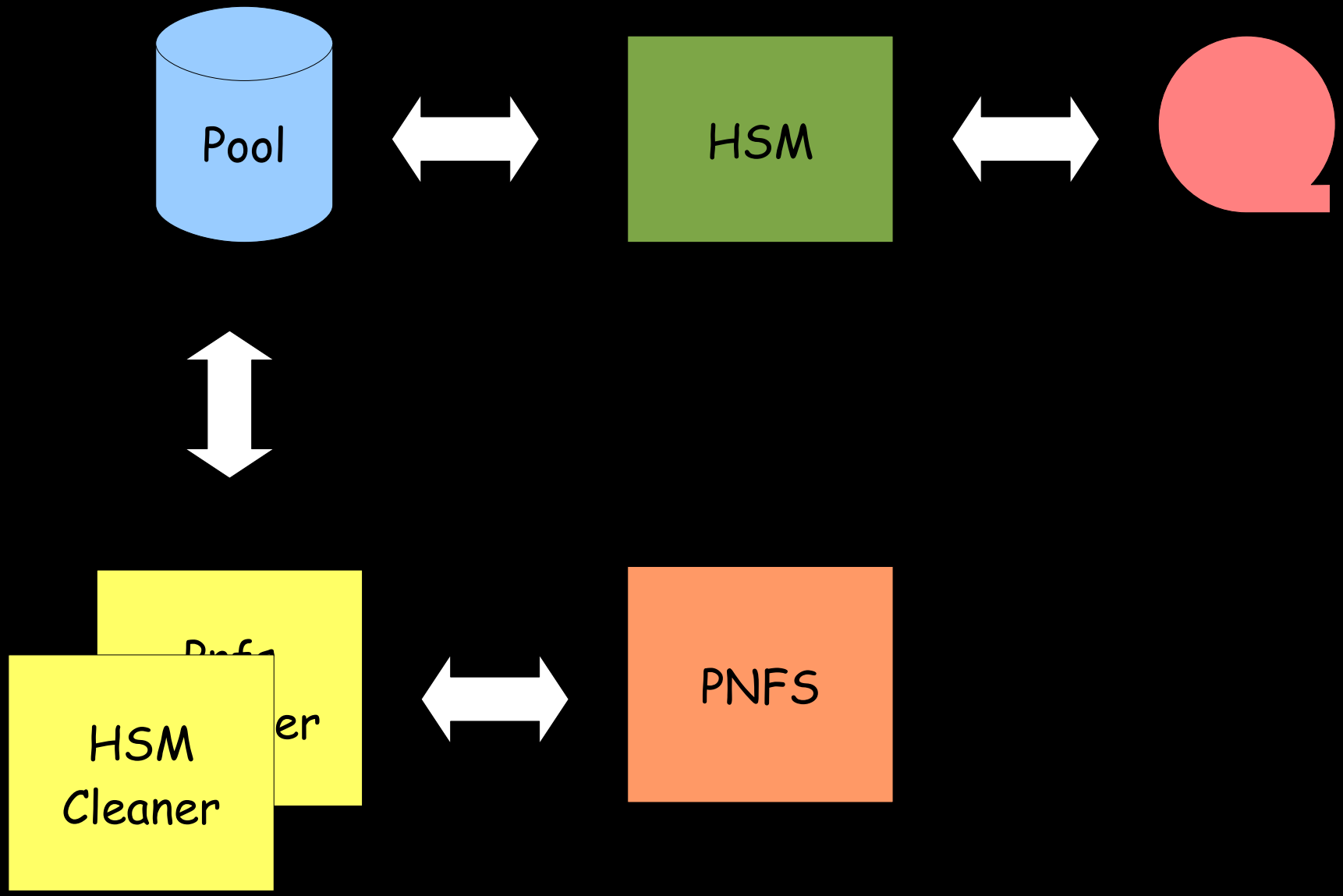
Pseudo algorithm for readEntry(pnfsid)

1. if main repository contains pnfsid
   then return that entry
2. if import repository contains pnfsid
   then return that entry
3. if PNFS Manager can provide the entry
   then return that entry
4. otherwise mark the file as bad

HSM Integration without PNFS


Pool
HSM
Pnfs
Manager
PNFS
Custom
Cleaner


Pool
HSM
HSM
Cleaner
PNFS



osm://osm/?store=myStore&group=STRING&bfid=0001000000000000005A690

Questions?