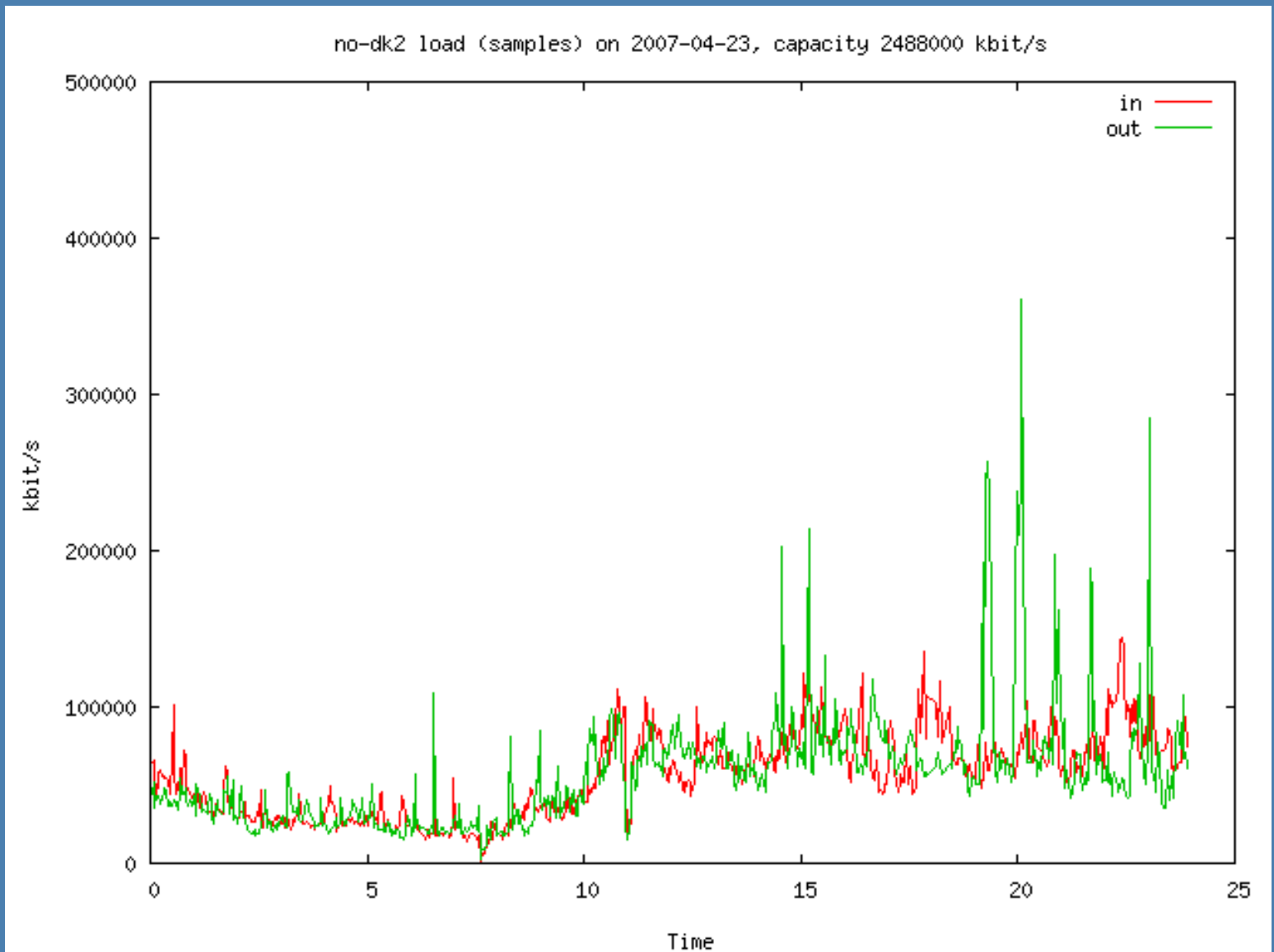


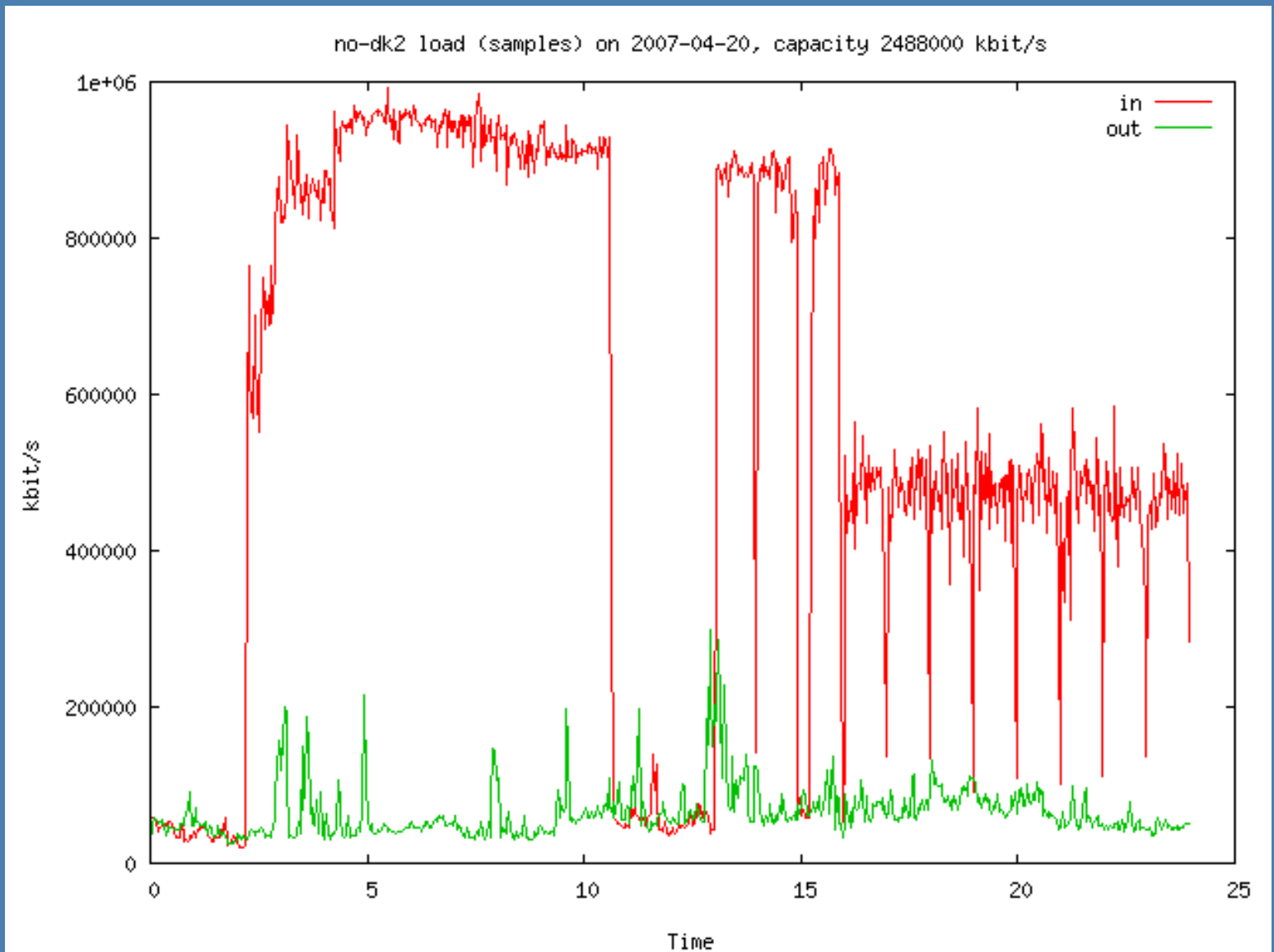


A Unified Wide Area Distributed Storage System

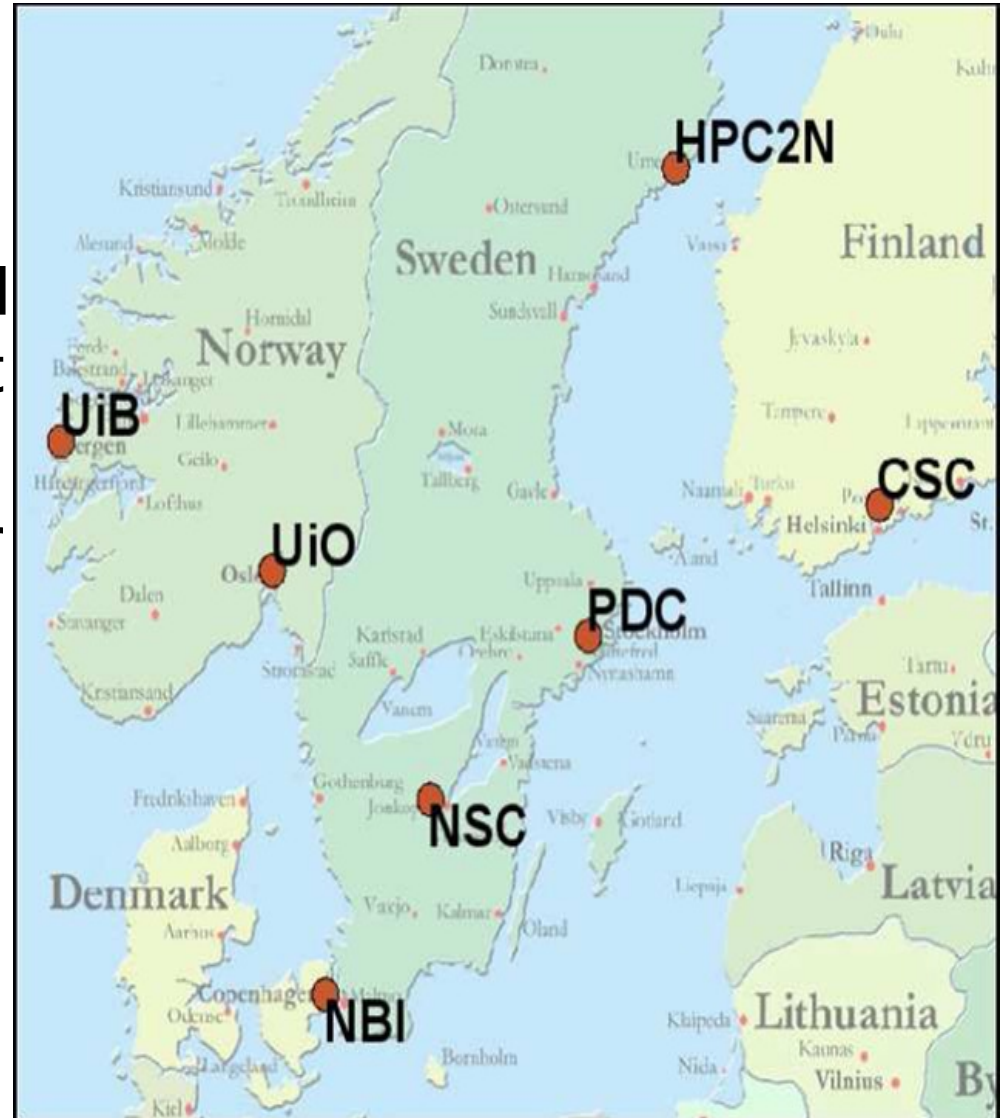
*Josva Kleist
Software Coordinator
Nordic Data Grid Facility*

*2nd Chinese-Nordic Network Workshop
Shanghai, April 25th*





- NDGF Tier-1 sites connected by a dedicated 10GE fibre (end of 07, beginning of 08)
- Storage resources not located at NDGF and not under direct NDGF control
- Required to expose a number of physically distributed sites as a single Tier-1 with a single entry point.

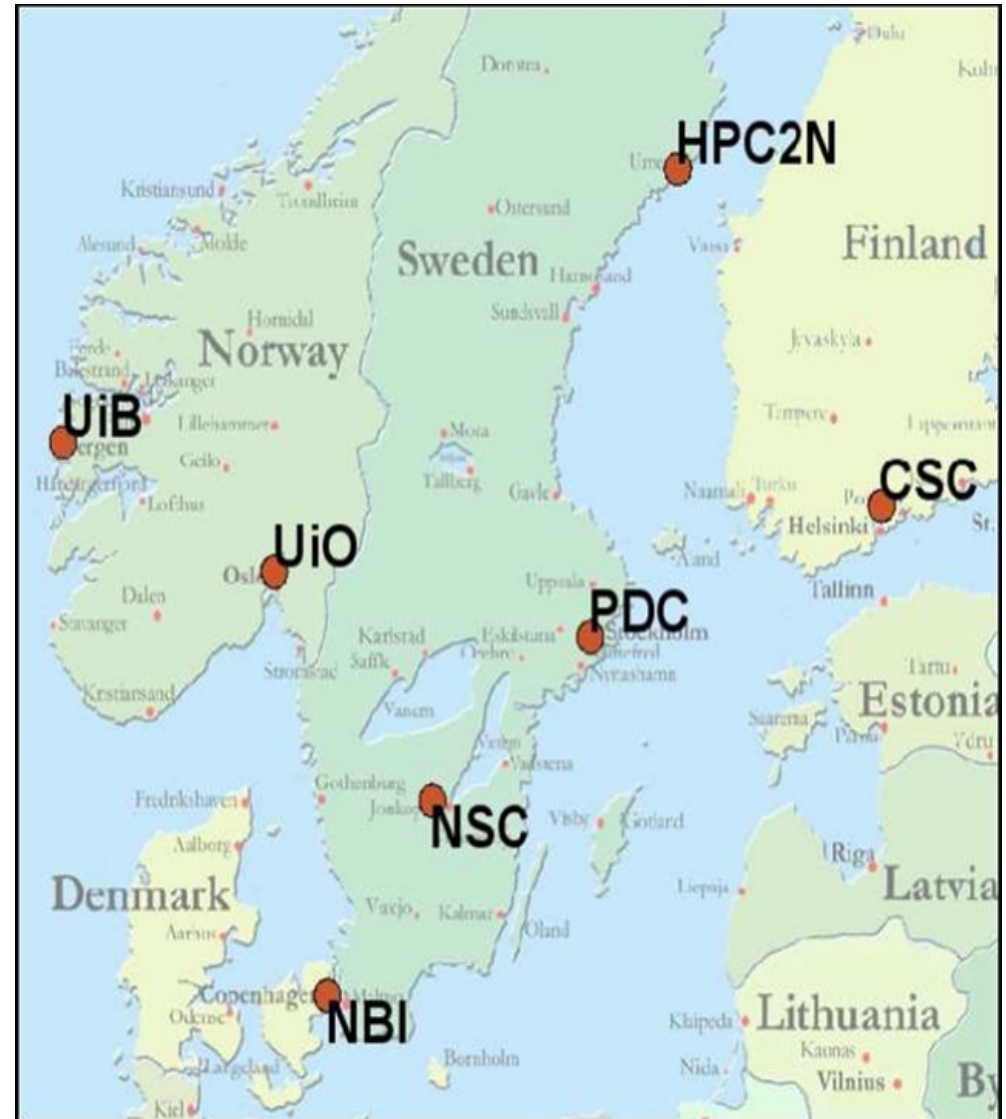


- A system for storing and retrieving huge amounts of data
- distributed among a large number of heterogenous server nodes
- under a single virtual filesystem tree
- supporting a variety of standard access methods



www.dcache.org

- One uniform dCache spanning all sites
- dCache pools operated by site owner



- 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
 - Platform (SL is not widely used in NorduGrid)
 - Upgradability
 - Autonomous operation
- Reliability
 - dCache is fairly resilient against pool failures
 - Head nodes provide central point of failure
 - Network separation in WAN
 - Disconnected operation (at least read-only)
 - Long term hope that dCache becomes less centralised

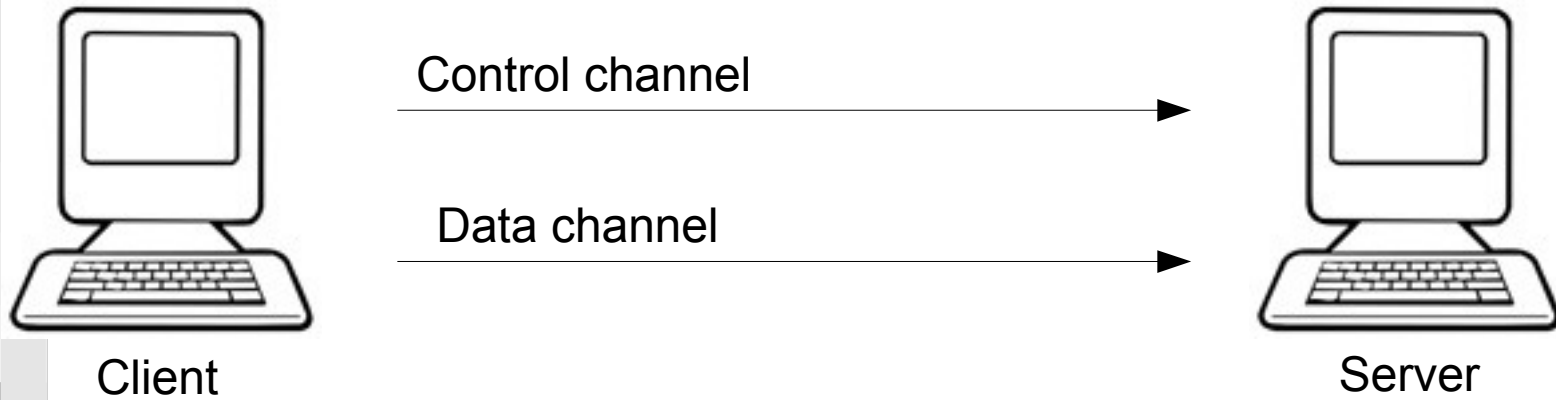
■ Performance

- No network model
 - e.g. SRM door assumes all GridFTP doors are equal (except for current load)
- Proxy operation of GridFTP

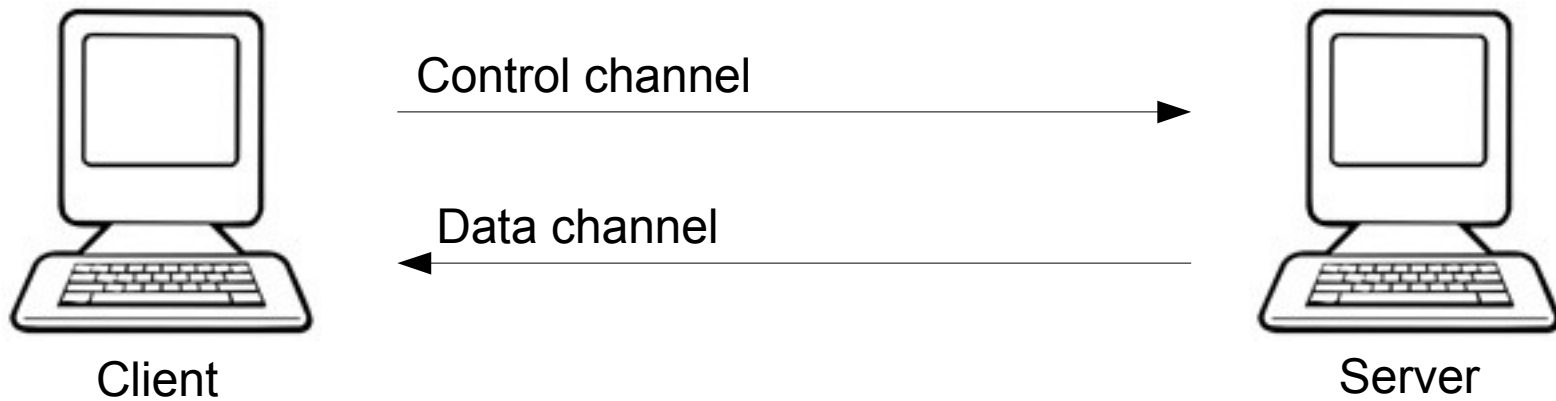
■ Functionality

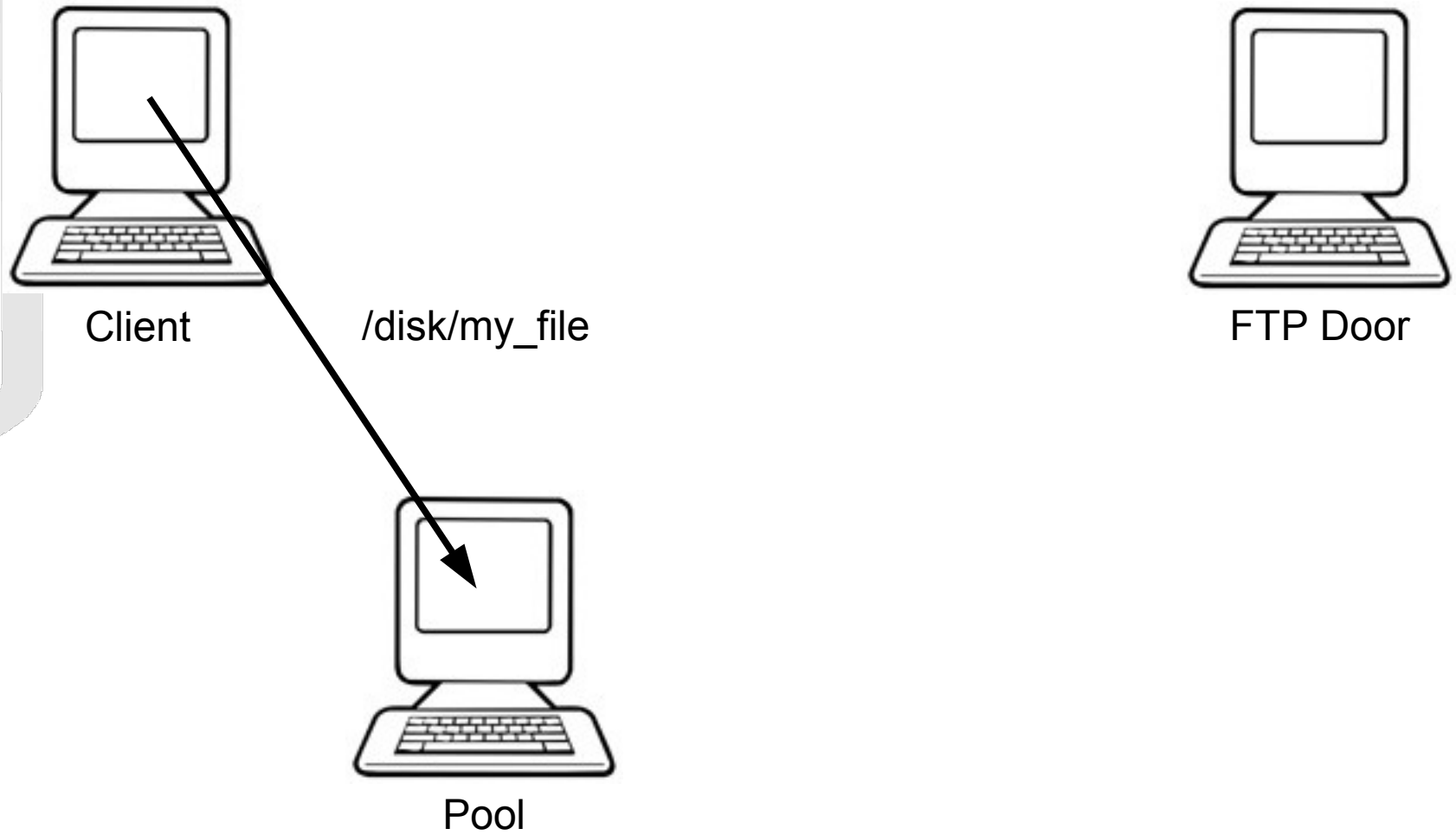
- HSM without PNFS (dCache 1.8)
- Heterogenous access to HSM
 - Stage-in must happen to connected pool
- Tivoli (TSM) integration
- User friendly view of logical name space without PNFS

Passive servers

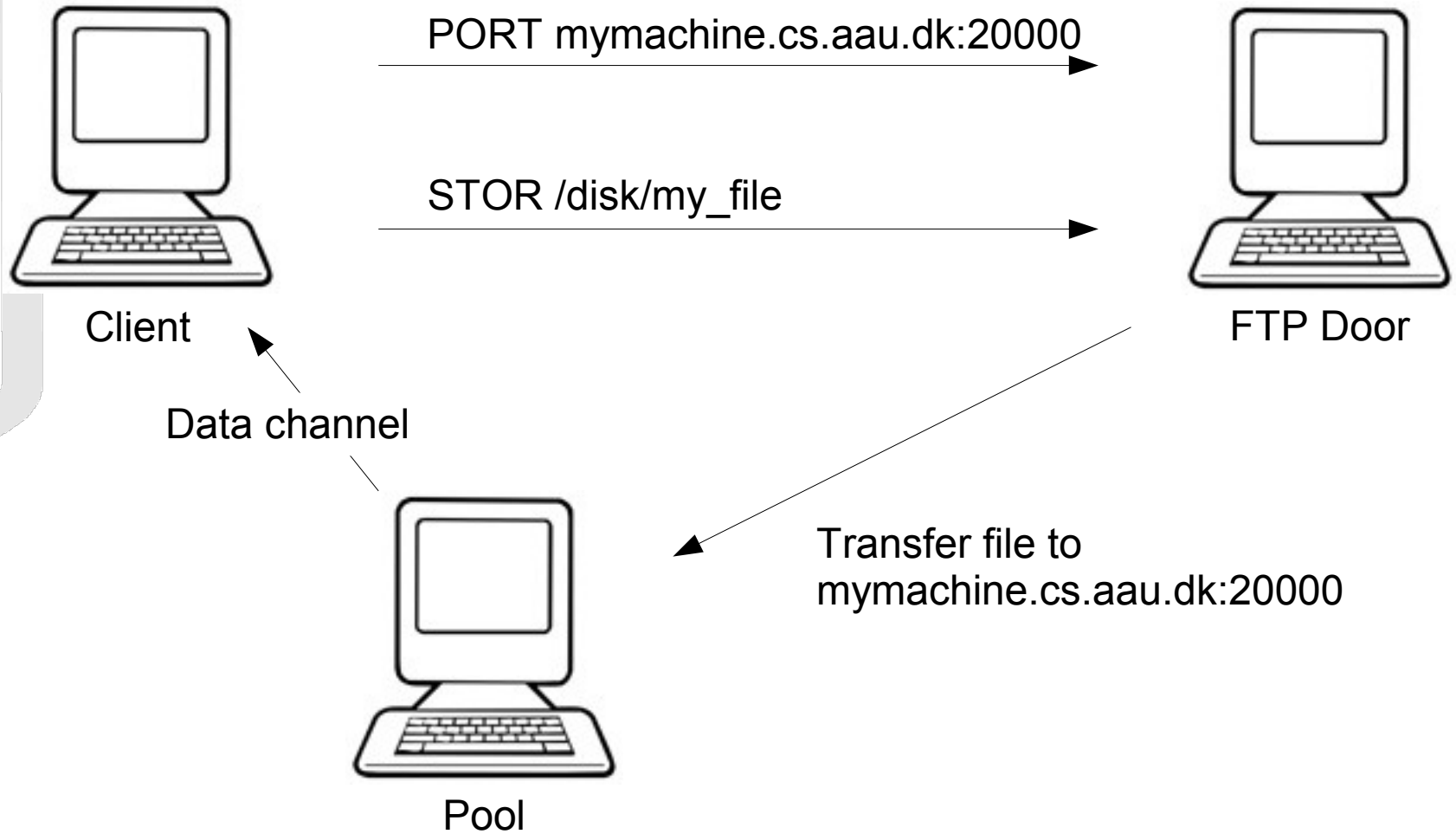


Active servers

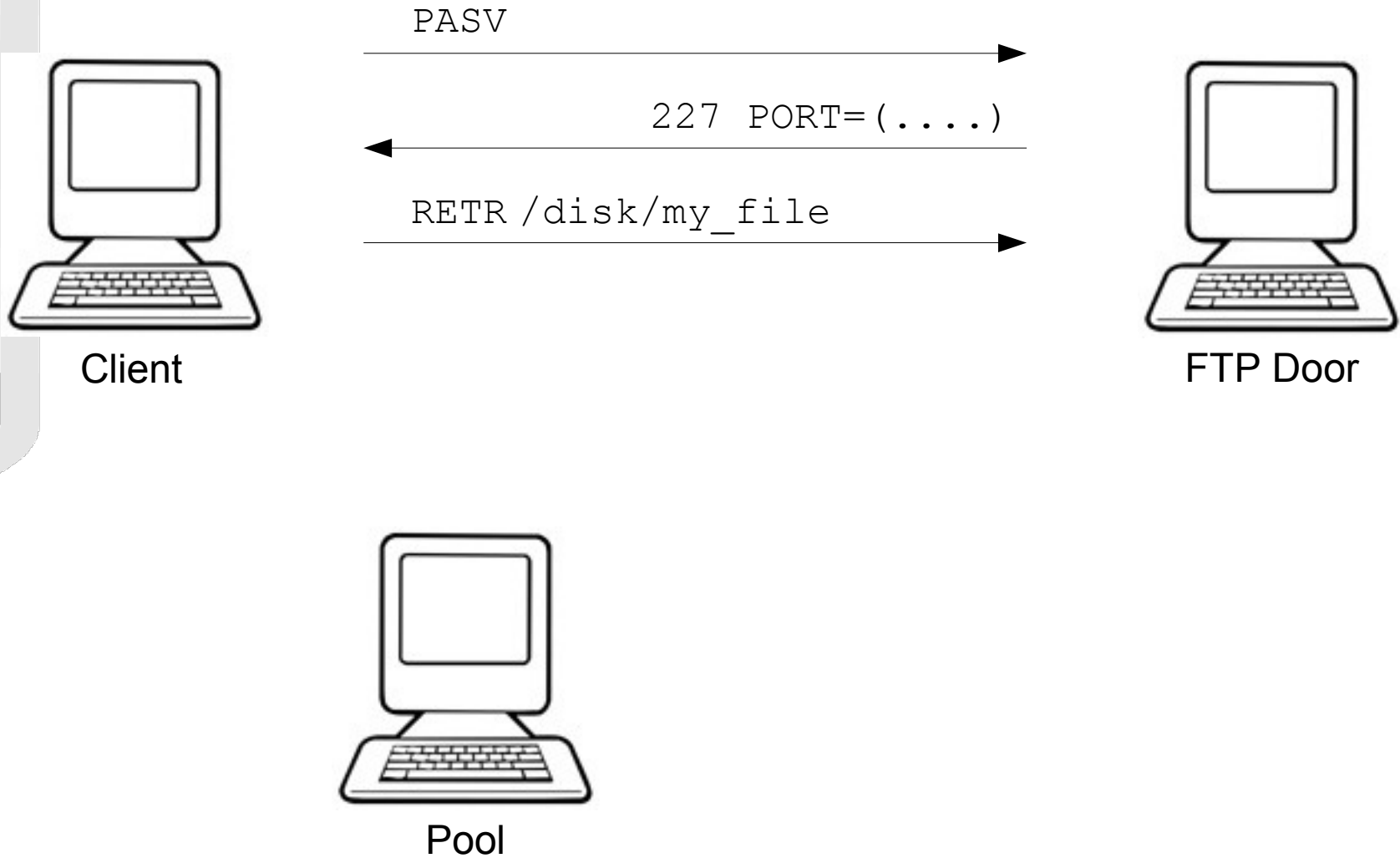




Active transfers in dCache



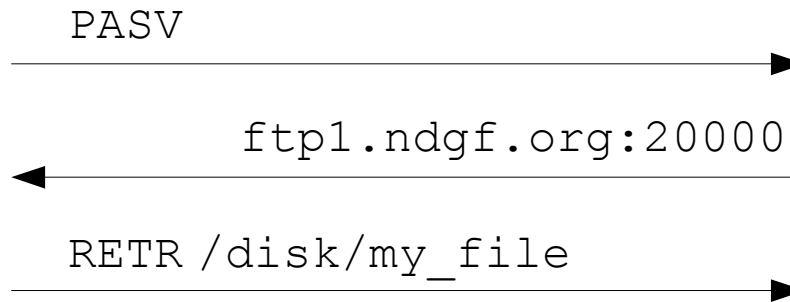
Passive transfers in dCache



Passive transfers in dCache



Client



FTP Door



Pool

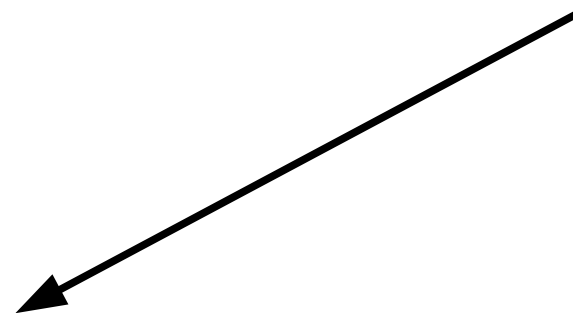
Passive transfers in dCache



Client

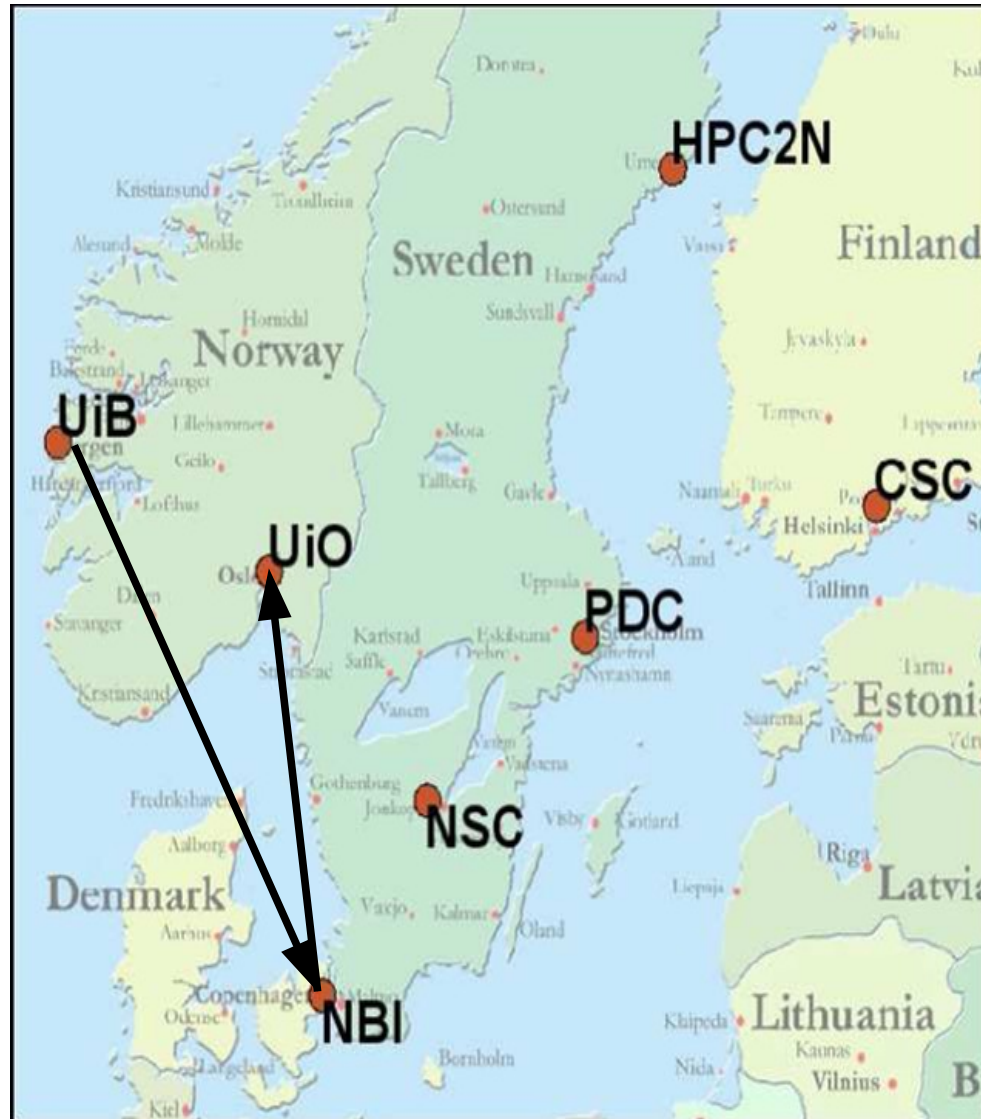


FTP Door



Pool

Passive transfers in dCache



- Enhanced FTP for grid defined in GFD-20.
- Introduced the “Extended block mode”
 - ▣ Parallel transfers
 - ▣ Spanning
- Reliably shutting down multiple data channels is tricky...
- ... mode E contains a known race condition
- ... work around documented in the specification limits the sender to be the active party
- Thus for uploads, we are always faced with the issue of passive transfers!



Client

srmPrepareToGet(data/my_file)

gsiftp://ftp1.ndgf.org/data/my_file



SRM Door



Pool



FTP Door



Client

`srmPrepareToGet(data/my_file)`



SRM Door



Pool



Client

srmPrepareToGet(data/my_file)



SRM Door

Where is the file?

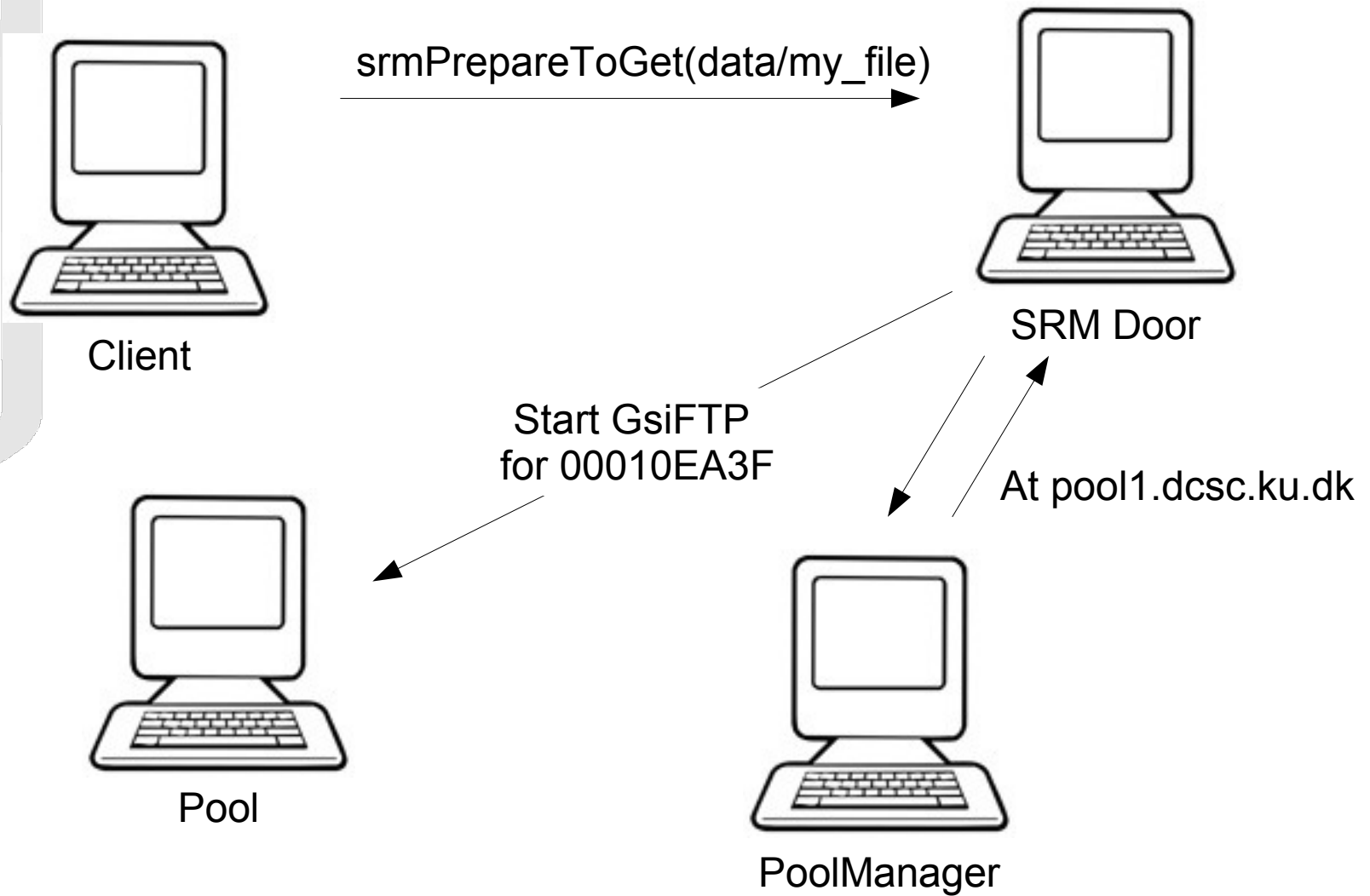
At pool1.dcsc.ku.dk

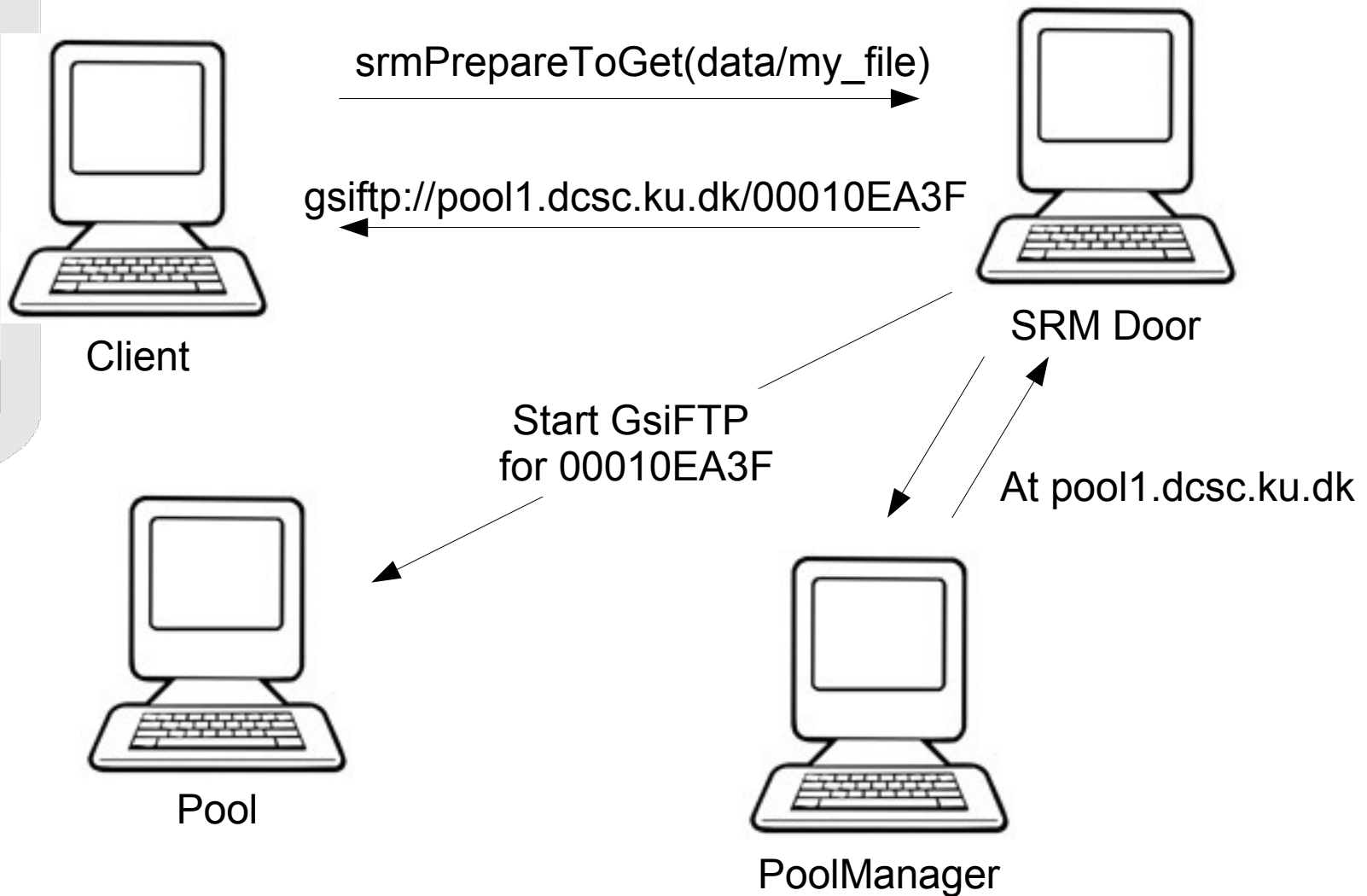


Pool



PoolManager





- Defined in GFD-R-P.047 draft specification
Mandrighenko, Allcock, Perelmutov
- OGF recommendation
- Solves many of the problems of GridFTP 1
 - GETPUT solves PASV/STOR problem
 - MODEX solves race condition in mode E
 - Checksums on blocks or whole files
 - Multiplex transfers on data channels



Client

`PUT pasv;mode=x;file=/disk/my_file`



FTP Door



Pool



Client

PUT pasv;mode=x;file=/disk/my_file



Pool



PoolManager



FTP Door

Where is the file?

At pool1.dcsc.ku.dk



Client

`PUT pasv;mode=x;file=/disk/my_file`



FTP Door

Begin transfer
for 00010EA3F



Pool



Client

`PUT pasv;mode=x;file=/disk/my_file`



FTP Door

Begin transfer
for 00010EA3F



Pool

Listening on
port 20000



Client

PUT pasv;mode=x;file=/disk/my_file

pool1.dcsc.ku.dk:20000



FTP Door

Begin transfer
for 00010EA3F

Listening on
port 20000



Pool

- Solution A
 - ▣ Bypasses FTP door, thus less components involved and reduced risk for failures
 - ▣ If pool is busy, SRM door can tell the client to wait
 - ▣ Changes in many components required

- Solution B: GridFTP2
 - ▣ Draft status. Some clarifications needed.
 - ▣ No signs of progress since June 2005.
 - ▣ No implementations, except
 - dCache head has GETPUT and MODEX
 - Globus patch is under development and Globus people are positive about the patch

Do we need the socket adapter?



Client

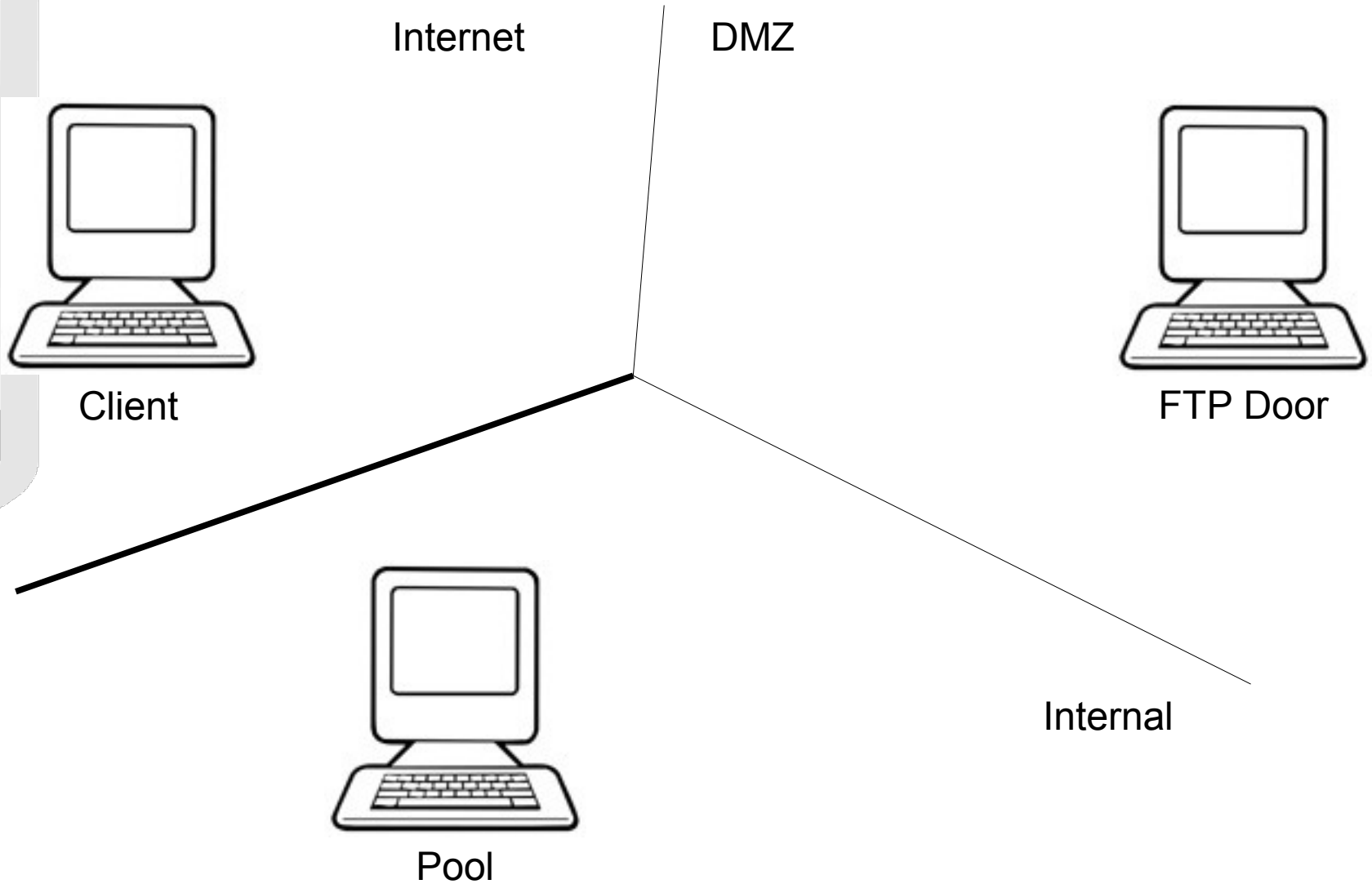


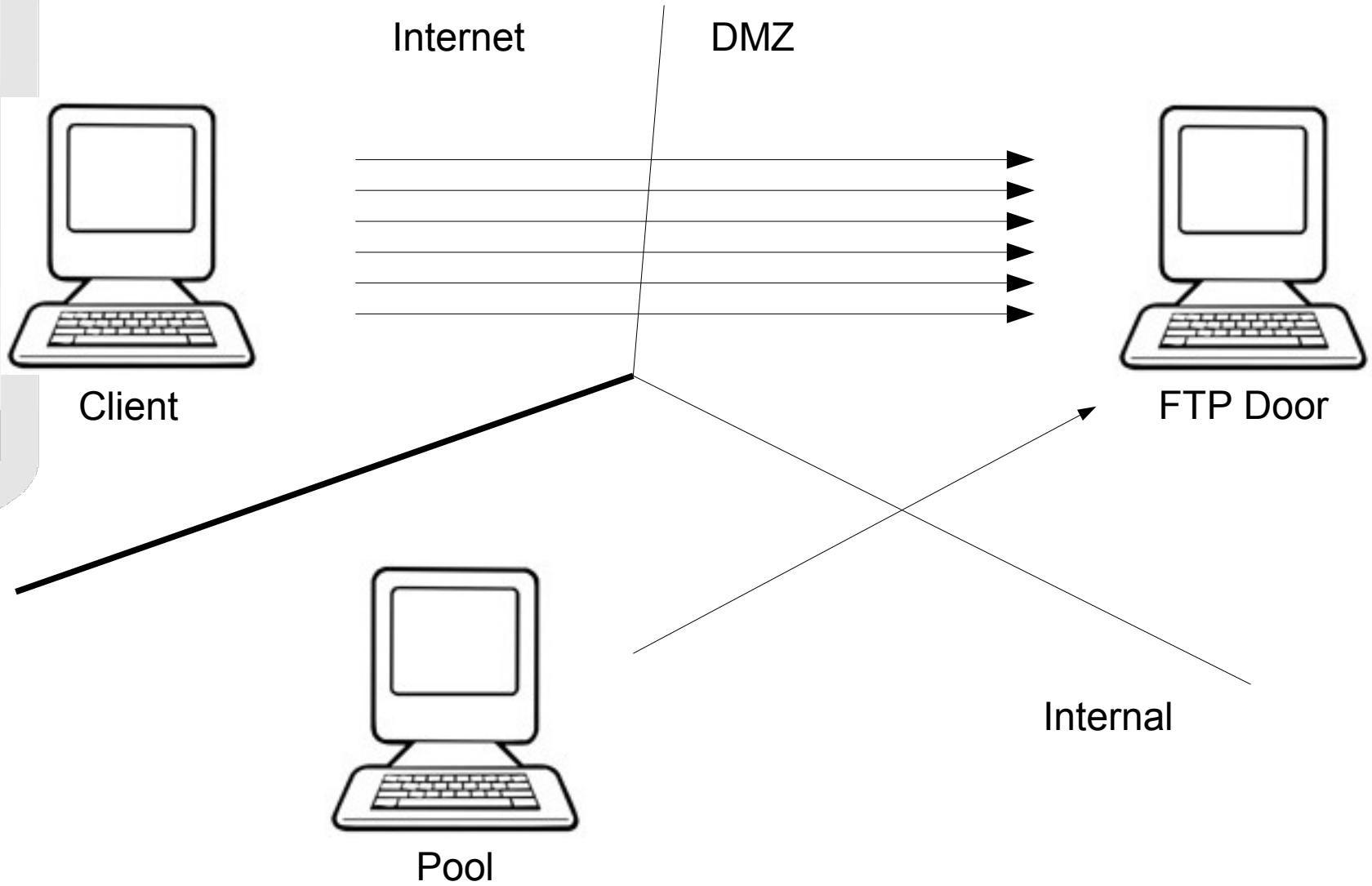
FTP Door

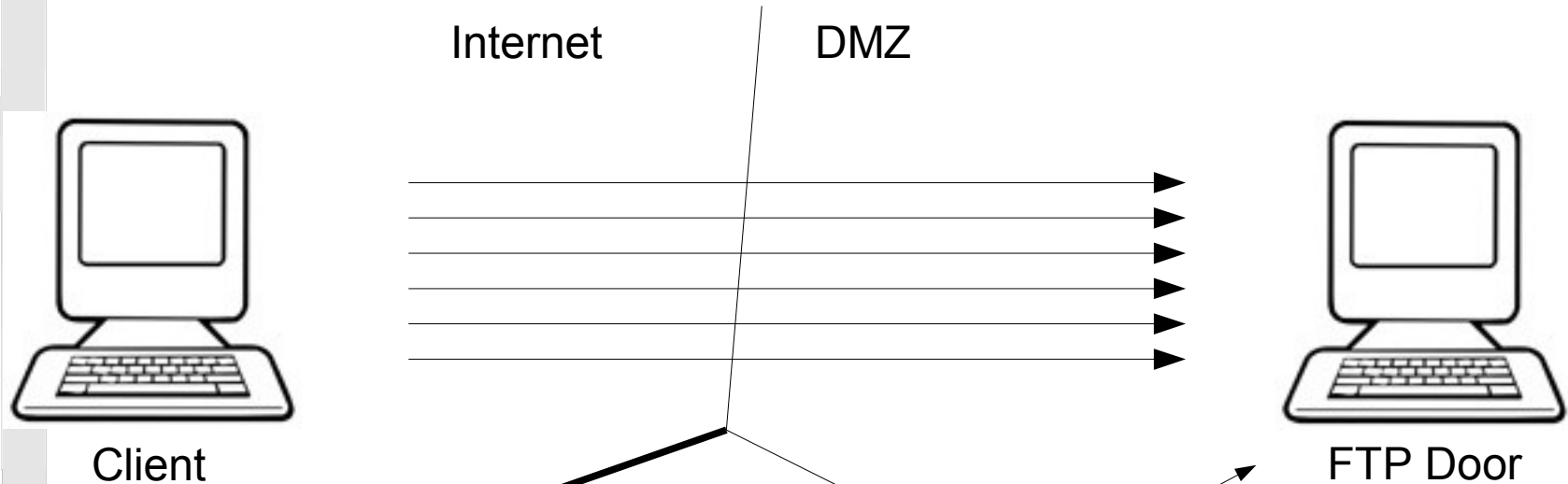


Pool

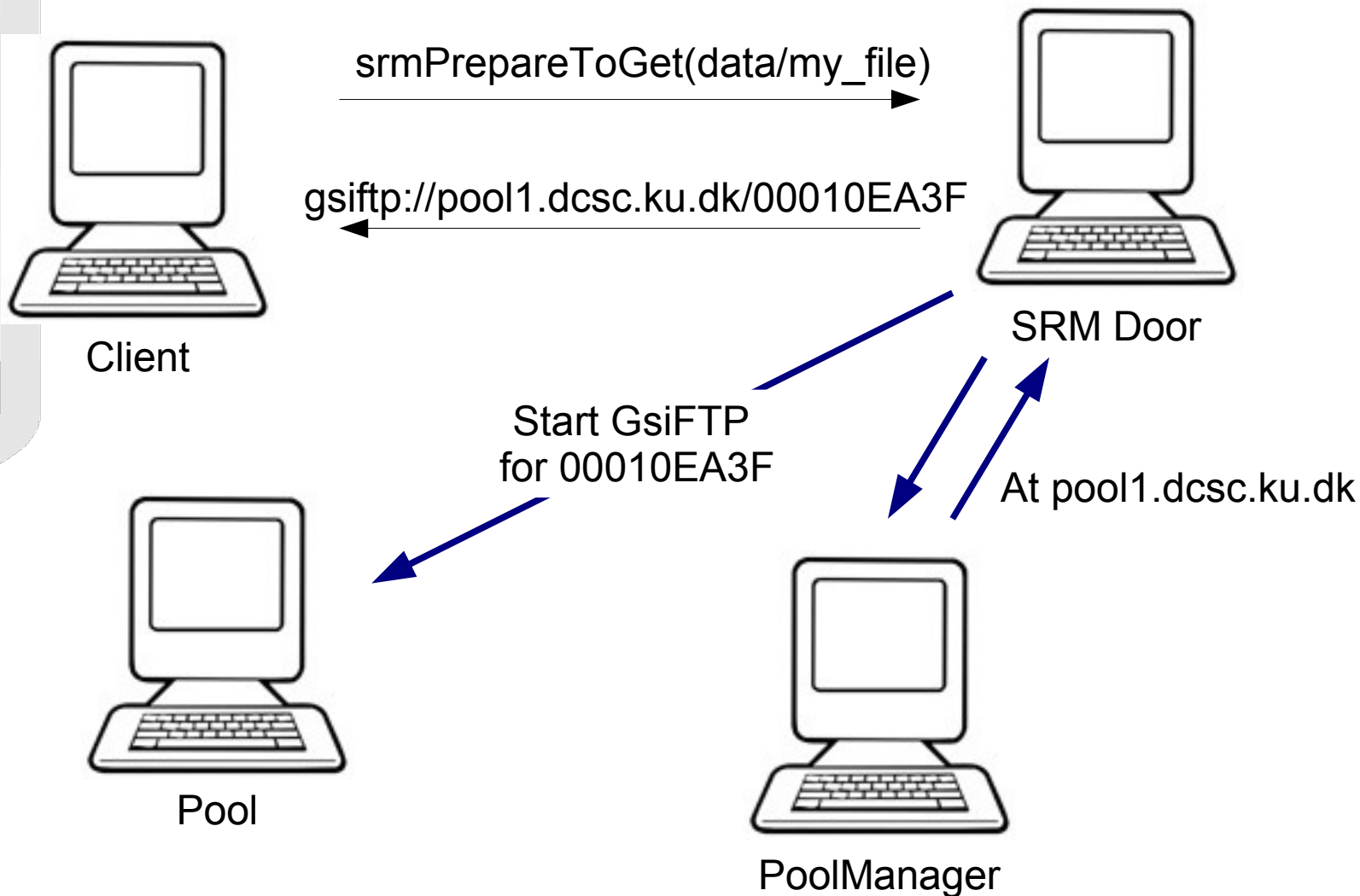
Do we need the socket adapter?







- Demultiplex mode E transfers
- Thread per connection
- Read and write blocks in a loop
- 1MB blocks, 10 connections = 10MB
- 1GB RAM = 100 connections
- Common advice is to run many GridFTP doors



Why we don't have tape yet



Pool



Pool

Why we don't have tape yet



Pool



Pool

Why we don't have tape yet



Pool

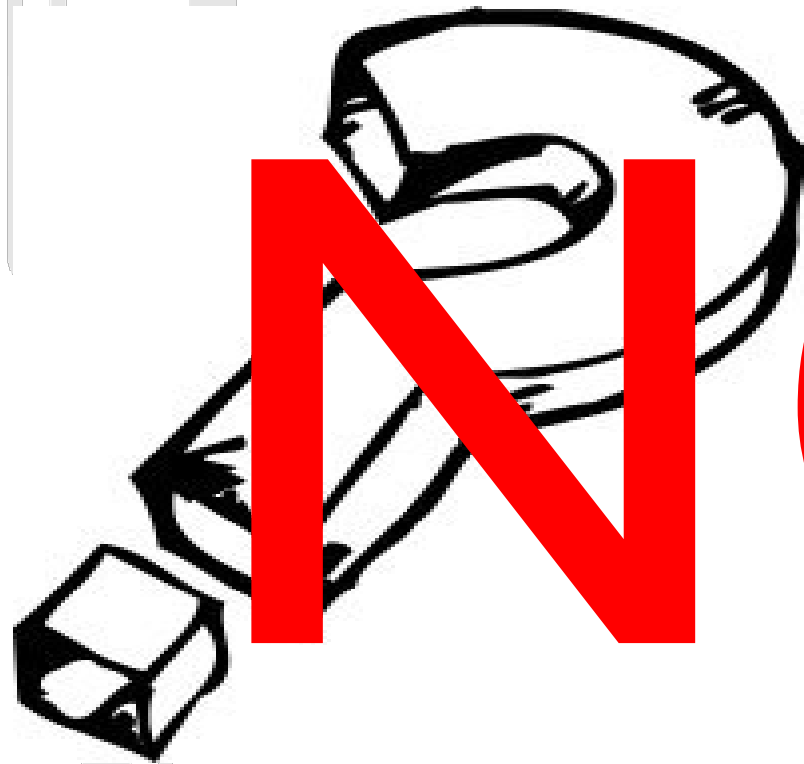


Pool

Are we done then?



Will dCache and
a 10 GE OPN allow us to relax?



Will dCache and
10 GE OPN allow us to relax?

The network pipe is too small



Some computations requires:

- Input files approx 2GB of size
- 15 min of computation
- Output approx 2GB

i.e. each CPU require 128 Gb/h

A 10 GE OPN network will (worst case) give us 3.6 Tb/h

i.e. we can only occupy 280 CPU's

Solution



- Grid middleware need to become *network aware!*
- Schedule jobs based on *where* the data is.
- Schedule jobs based on *when* data is available.

This is a hard scientific problem for computer scientists and network engineers - not physicists

- Grid middleware need to become *network aware!*
- Schedule jobs based on *where* the data is.
- Schedule jobs based on *when* data is available.



- WAN deployment of dCache at the NDGF distributed Tier 1
- Provides unique and interesting problems
- This is not enough – the network pipe can still become too small.
- Current focus is on immediate problems with data flow, HSM, security and administration.