



dCache.ORG

dCache.ORG

# *dCache, managed grid storage*

Patrick Fuhrmann

for the dCache Team



support and funding by





# What is dCache.ORG

dCache.ORG

dCache.ORG

## *Head of dCache.ORG*

Patrick Fuhrmann

## *Core Team (Desy, Fermi, NDGF)*

Andrew Baranovski

Gerd Behrmann

Bjoern Boettscher

Ted Hesselroth

Alex Kulyavtsev

Iryna Koslova

Dmitri Litvintsev

David Melkumyan

Dirk Pleiter

Martin Radicke

Owen Synge

Neha Sharma

Vladimir Podstavkov



## *Head of Development FNAL :*

Timur Perelmutov

## *Head of Development DESY :*

Tigran Mkrtchyan

## *External*

### *Development*

Jonathan Schaeffer, IN2P3

### *Support and Help*

Abhishek Singh Rana, SDSC

Greig Cowan, gridPP

Stijn De Weirdt (Quattor)

Maarten Lithmaath, CERN

Flavia Donno, CERN



*Plan for today*

*The LHC Tier model and the Grid.*

*Data Grid and Storage Elements*

*What is the dCache SE*

*The Storage Resource Manager*

*NFS 4.1*



# *The LHC Tier model and the Grid.*



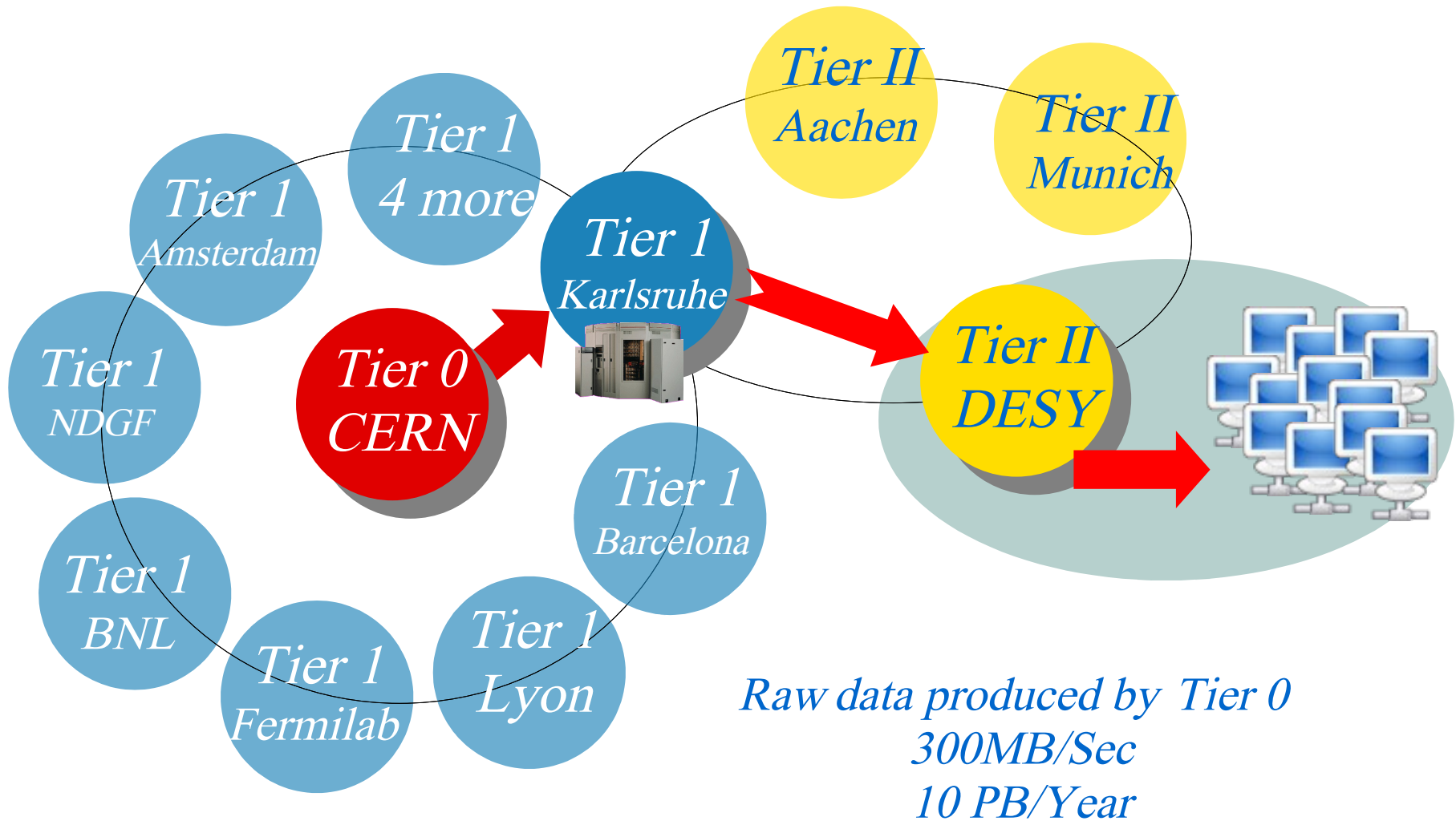


# LHC (Data Management) Tier Structure

*Significantly oversimplified*

dCache.ORG

dCache.ORG

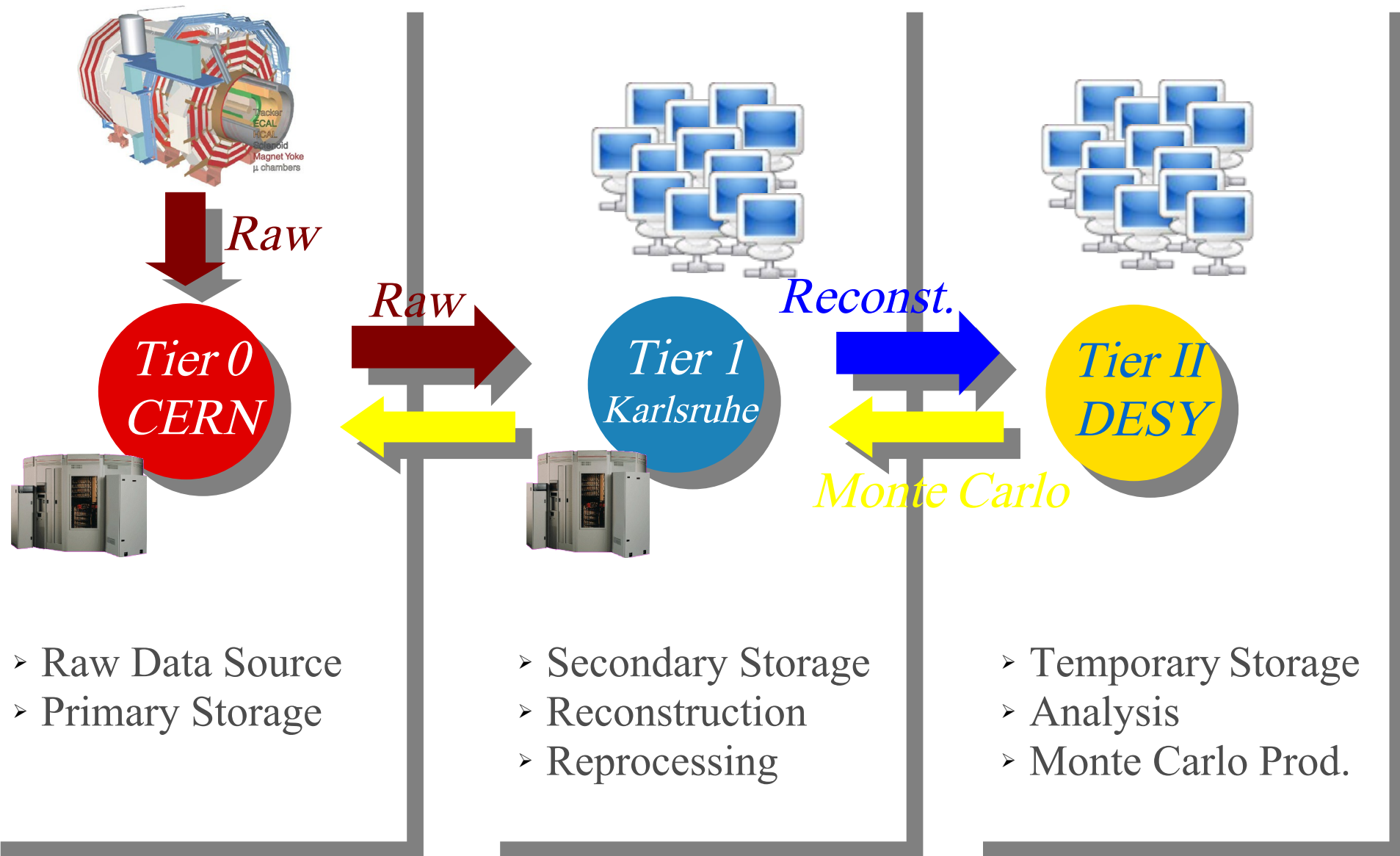




# Storage and data processing

dCache.ORG

dCache.ORG



- › Raw Data Source
- › Primary Storage

- › Secondary Storage
- › Reconstruction
- › Reprocessing

- › Temporary Storage
- › Analysis
- › Monte Carlo Prod.



# *The Storage Element, workhorse of the LHC Data Grid*



# What is an LHC Storage Element ?

- ★ *Stores data on different media (Disk, Tape ?)*
- ★ *Streams data to/from remote SE's*
- ★ *Posix like access from local worker-nodes*
- ★ *Manages storage*
  - ★ *Reserve Space for incoming data*
  - ★ *Create virtual data containers with predefined storage attributes (Access Latency, Retention Policy)*
  - ★ *Steer data location by dynamic (ip number, protocol) or static attributes (space tokens)*

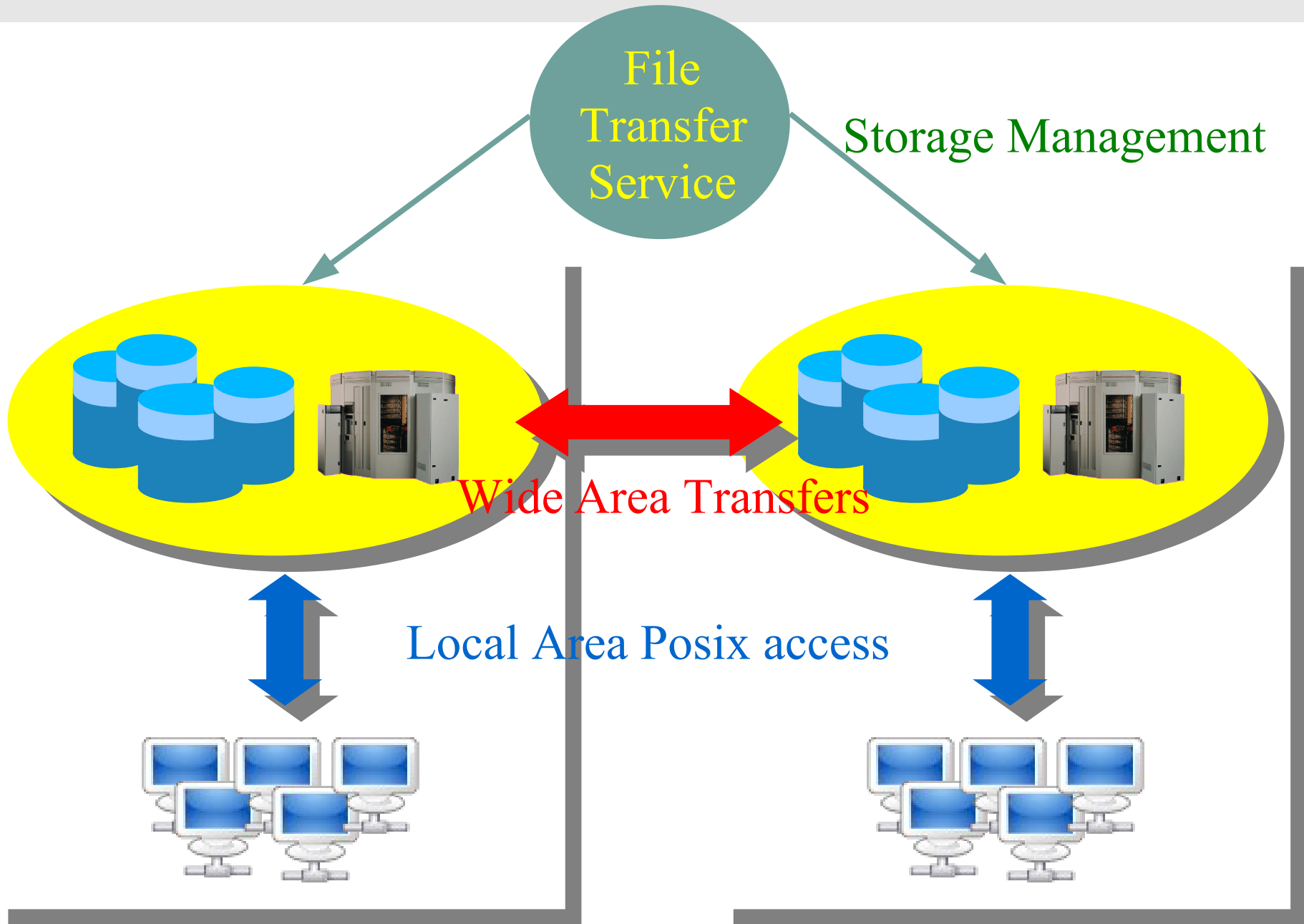




# Storage Element interactions.

dCache.ORG

dCache.ORG





*Intentionally **not** mentioned here*

- *Information Provider Protocols*
- *File Catalogs*



## *What do we need a grid storage element for ?*

*We need to serve large amounts of data locally*

- *Access from local Compute Element*
- *Huge amount of simultaneously open files.*
- *Posix like access (What does that mean ?)*

*We need to exchange large amounts of data with remote sites*

- *Streaming protocols.*
- *Optimized for low latency (wide area) links.*
- *Possibly controlling 'link reservation'.*



## *What do we need a grid storage element for ? (cont.)*

### *We need to allow storage control*

- *Space reservation to guarantee maximum streaming.*
- *Define space properties (TAPE, ONLINE, ...)*
- *Transport protocol negotiation.*

### *We need to publish SE specific information*

- *Clients need to select 'best' SE or CE for a job.*
- *Availability*
- *Available Space (max, used, free ...)*
- *Supported Spaces (Tape, disk ...)*
- *Which VO owns which space ?*



# The Storage Element access Protocol Zoo

dCache.ORG

dCache.ORG

*Might be a standard  
(OGF)*



SRM Storage Resource Management  
Space/Protocol Management

Wide Area Transport Protocol  
In use : `gsiFtp`  
Discussed : `http(s)`

Local Access Protocol  
`(gsi)dCap` or `rfio` and `xRoot`

*This is not at all a standard*





*What makes this a Storage Grid, instead of distributed storage ?*

*! Commonly accepted standards !*



# *dCache in a Nutshell*



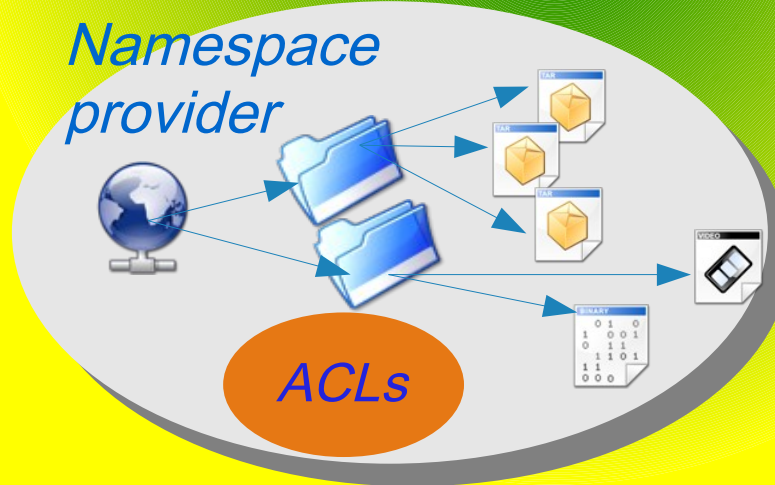
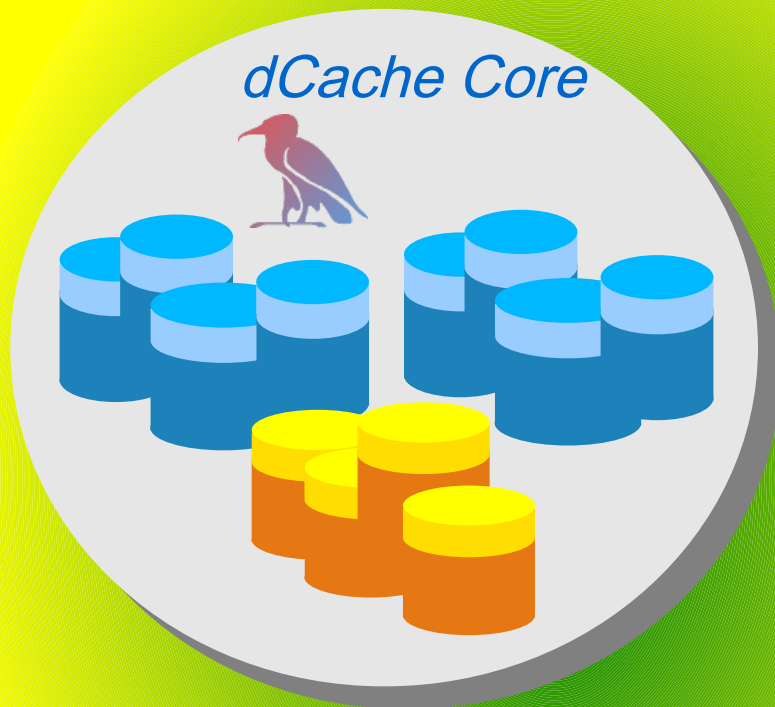
# dCache in a Nutshell

## Black Box View

dCache.ORG

**Tape Storage**

OSM, Enstore  
Tsm, Hpss, DMF

**Protocol Engines**

Information Protocol(s)

Storage Management Protocol(s)  
SRM 1.1 2.2

Data & Namespace Protocols

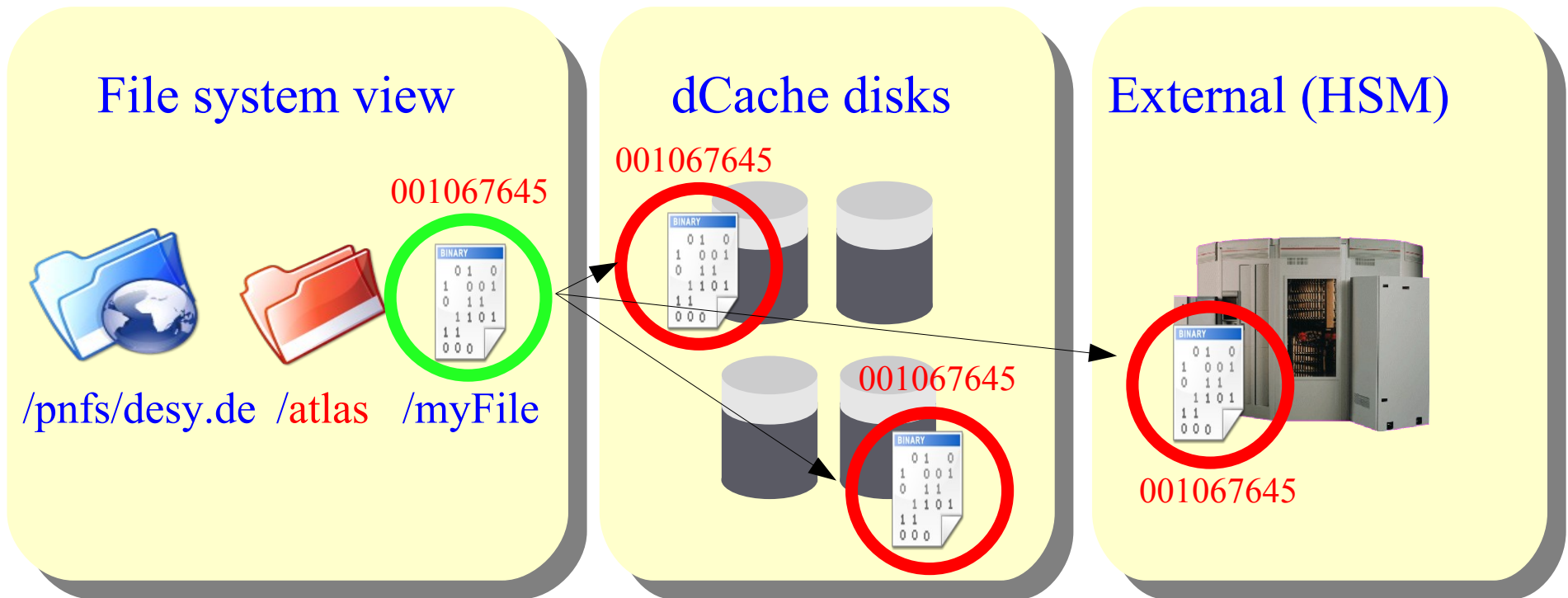
(NFS 4.1) dCap  
ftp (V2) gsiFtp  
xRoot  
(http)

Namespace ONLY  
NFS 2 / 3



# dCache in a Nutshell

- Strict name space and data storage separation, allowing
  - consistent name space operations (mv, rm, mkdir e.t.c)
  - consistent access control per directory resp. file
  - managing multiple internal and external copies of the same file
  - convenient name space management by nfs (or http)





# *dCache in a Nutshell*

- **Overload and meltdown protection**
  - Request Scheduler.
  - Primary Storage pool selection by protocol, IP, directory, IO direction
  - Secondary selection by system load and available space considerations.
  - Separate I/O queues per protocol (load balancing)
- **Supported protocols :**
  - (gsi)ftp
  - (gsi)dCap
  - xRoot
  - SRM
  - nfs2/3 (name space only)



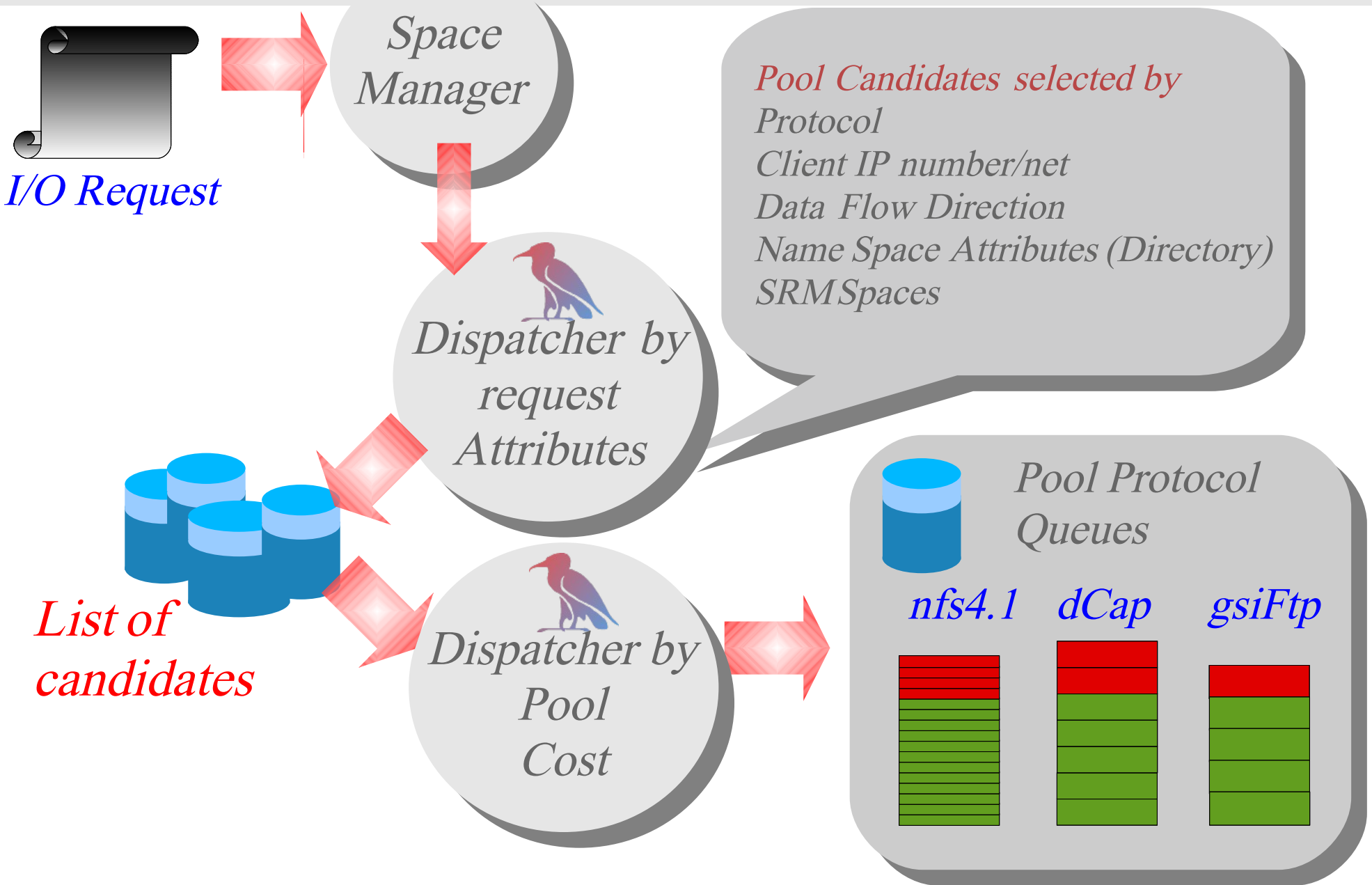


# dCache in a Nutshell

## Scheduler and I/O queues

dCache.ORG

dCache.ORG





# *In a Nutshell*

## → dCache partitioning for very large installations

- Different tuning parameter for different parts of dCache

## → File hopping on

- automated hot spot detection
- configuration (read only, write only, stage only pools)
- on arrival (configurable)
- outside / inside firewalls

## → Resilient Management

- at least  $n$  but never more than  $m$  copies of a file

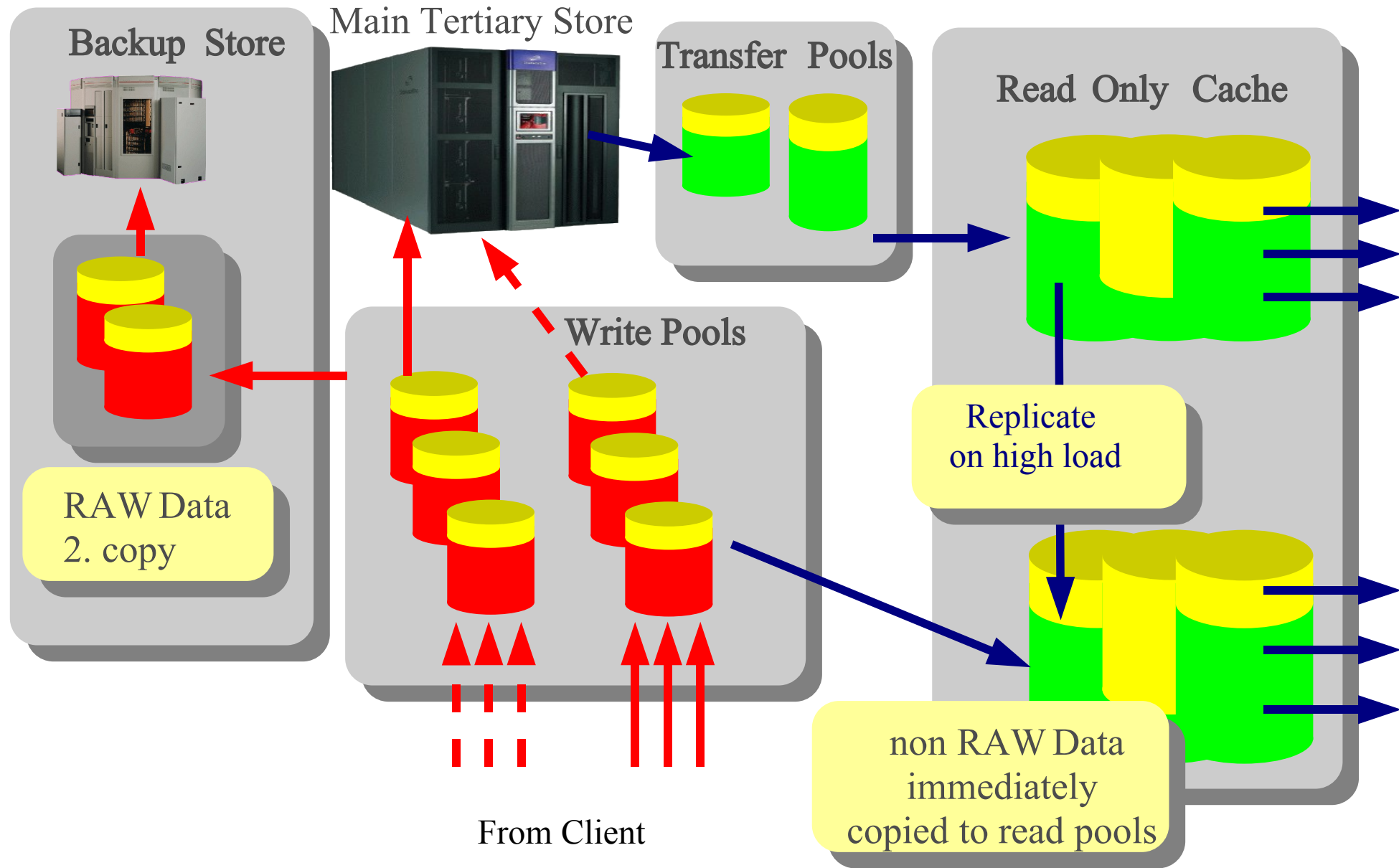


# In a Nutshell

# File Hopping

dCache.ORG

dCache.ORG





# *In the Nutshell*

## → HSM Support

- TSM, HPSS, DMF, Enstore, Osm
- Automated migration to tertiary store and restore from there.
- Central Tertiary Management adaptor in preparation.
- Support of multiple, non overlapping HSM systems (NDGF approach)

## → Misc

- Graphical User Interface
- Command line interface
- Jpython interface
- SRM watch
- NEW : Monitoring Plots



# *dCache and the LHC storage management*

*dCache is in use at 8 Tier I centers*

- *fzk(Karlsruhe, GR)*
- *in2p3 (Lyon, FR)*
- *BNL(New York.US)*
- *FERMILab (Chicago, US)*
- *SARA(Amsterdam. NL)*
- *PIC (Spain)*
- *Triumf(Canada)*
- *NDGF (NordGrid)*

*and at about 60 Tier II's*

*dCache is part of VDT(OSG)*

*We are expecting > 20 PB per site > 2011*

***dCache will hold the largest share of the LHC data.***





*And again*

*! Going for standards !*

*Two examples : SRM and NFS 4.1*



## *The SRM Interface*



## The *Storage Resource Manager Protocol*

### *Key Ideas*

*File Transfer Protocol Negotiation*

*Storage Attributes*

*Space Reservation*

*Space Tokens*



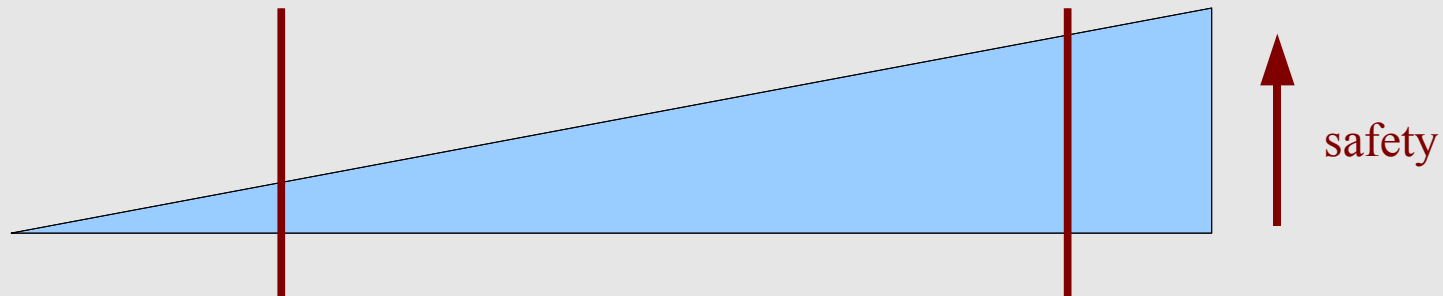


# What are Storage Attributes ?

SRM2.2 introduces two storage attributes

## Retention Policy

*How safe is my file within the system ?  
or What is the likelihood of a file loss ?*

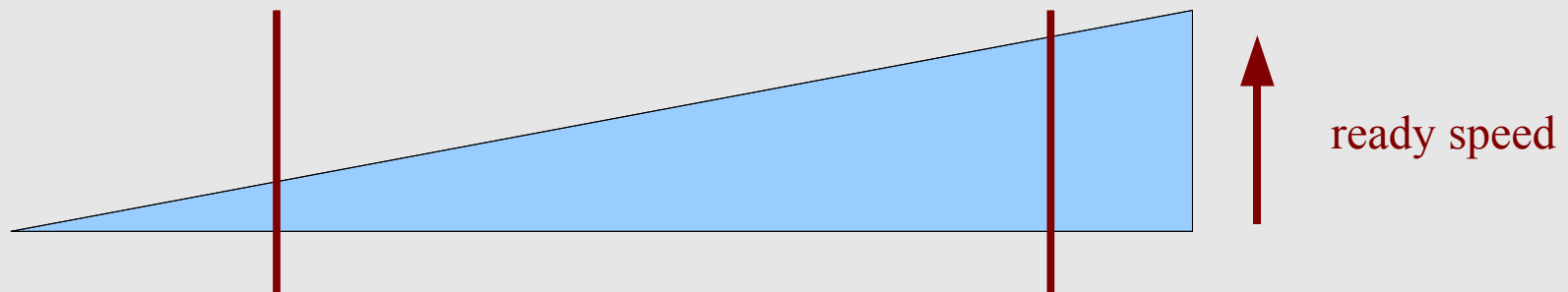


unsafe = **REPLICA** (e.g. JBOD)

Very safe = **CUSTODIAL** (e.g. Tape)

## Access Latency

*How long does it take to make a  
file ready to transfer ?*



fast = **ONLINE** (e.g. Disk)

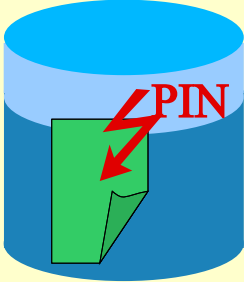
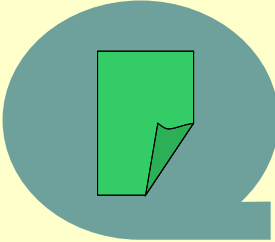
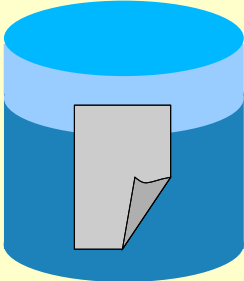
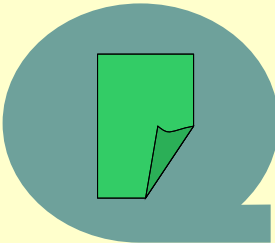
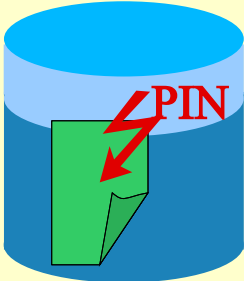
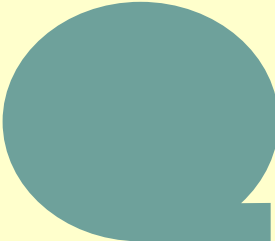
slow = **NEARLINE** (e.g. Tape)



# What are Storage Attributes ?

dCache.ORG

dCache.ORG

<i>Developers</i>	<i>Users</i>	<i>System</i>	
		DISK	TAPE
Custodial/Online	T1D1		
Custodial/Nearline	T1D0		
Replica/Online	T0D1		



# What are Space Tokens?

*Space tokens serve two major purposes :*

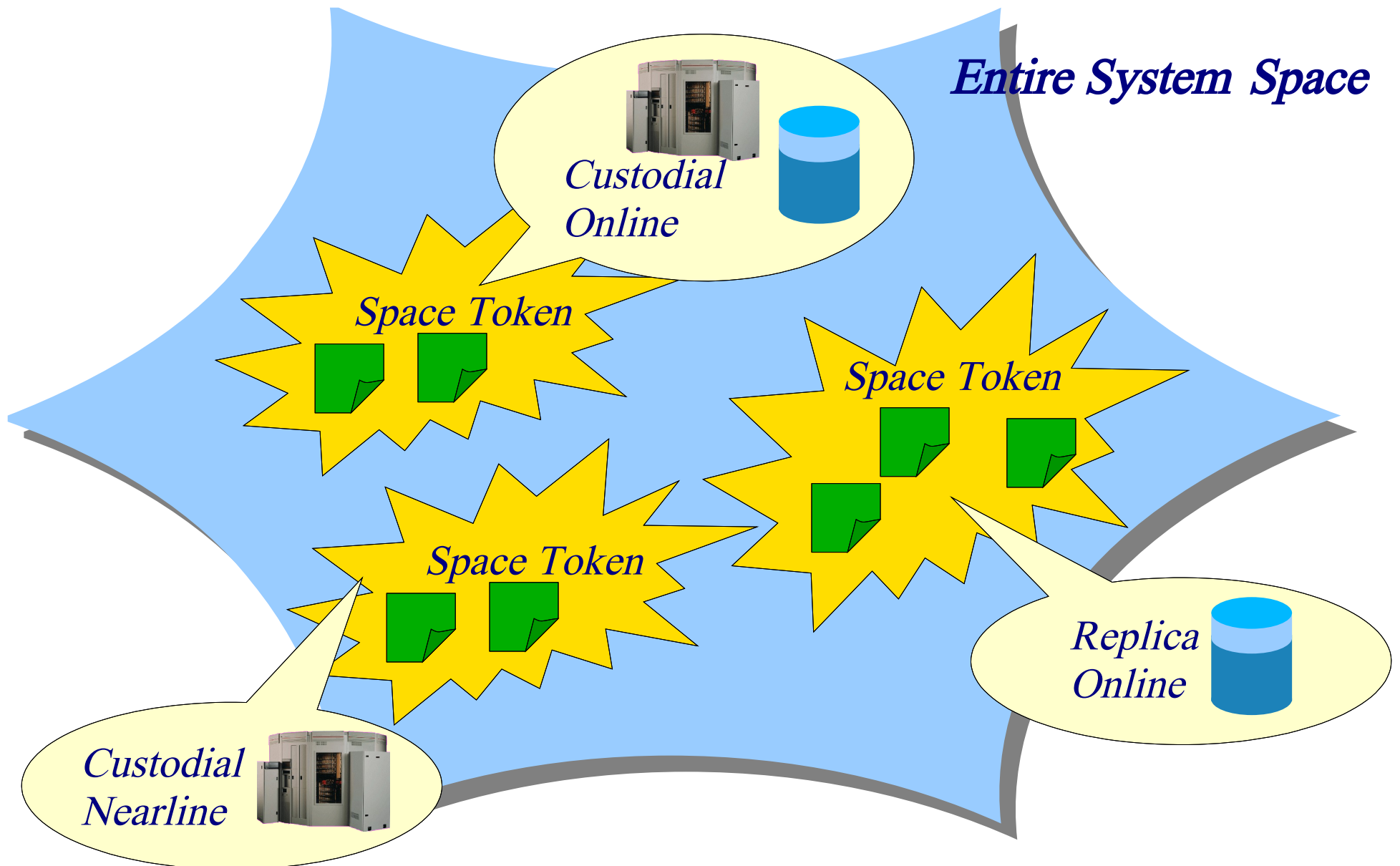
- *A space token is a handle to **reserved space** (disk/tape), which has been allocated dynamically or statically in advance.*
- *Space Tokens have **storage attributes** attached, e.g. **Retention Policy** and **Access Latency**.*



# What are Space Tokens? (cont.)

dCache.ORG

dCache.ORG





## *The NFS 4.1 Interface*





center for  
information  
technology  
integration

*University of Michigan*

*“We are developing an implementation of NFSv4 and NFSv4.1 for Linux.”*

## *Introduction of RFC 3530*

The Network File System (NFS) version 4 is a distributed filesystem protocol which owes heritage to NFS protocol version 2, RFC 1094, and version 3, RFC 1813. Unlike earlier versions, the NFS version 4 protocol supports traditional file access while integrating support for **file locking** and the **mount protocol**. In addition, support for **strong security** (and its negotiation), **compound operations**, **client caching**, and **internationalization** have been added. Of course, attention has been applied to making NFS version 4 operate well in an Internet environment.



## *The NFS 4.1 Protocol*

### *Key Ideas (for use)*

- POSIX **Clients** are coming **for free** (provided by all major OS vendors).
- NFS 4.1 is aware of **distributed data**.
- Will make dCache attractive to other (non-hep) communities.
- LCG could consider to drop the LAN protocol zoo (dcap,rfio,xroot)



## Why is NFS 4.1 : technical perspective

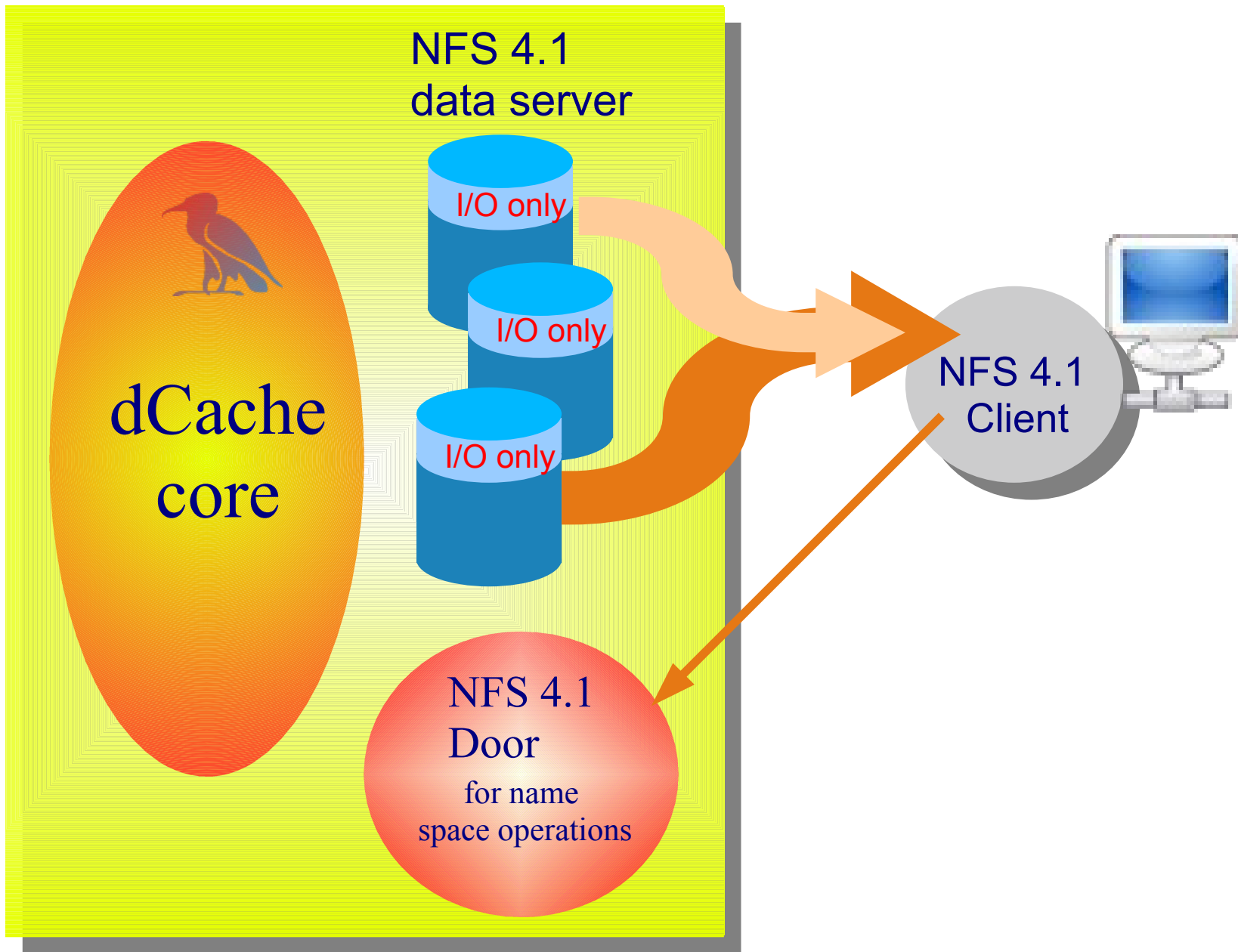
- NFS 4.1 is aware of **distributed data**
- **Faster** (optimized) e.g.:
  - Compound RPC calls
  - e.g. : 'Stat' produces 3 RPC calls in v3 but only one in v4
- GSS authentication
  - Built-in **mandatory security** on file system level
- ACL's
- dCache can **keep track on client operations**
  - OPEN / CLOSE semantic (so system can keep track on open files)
  - 'DEAD' client discovery (by client to server pings)
- smart client caching.



# NFS 4.1 in dCache (This is how we do it)

dCache.ORG

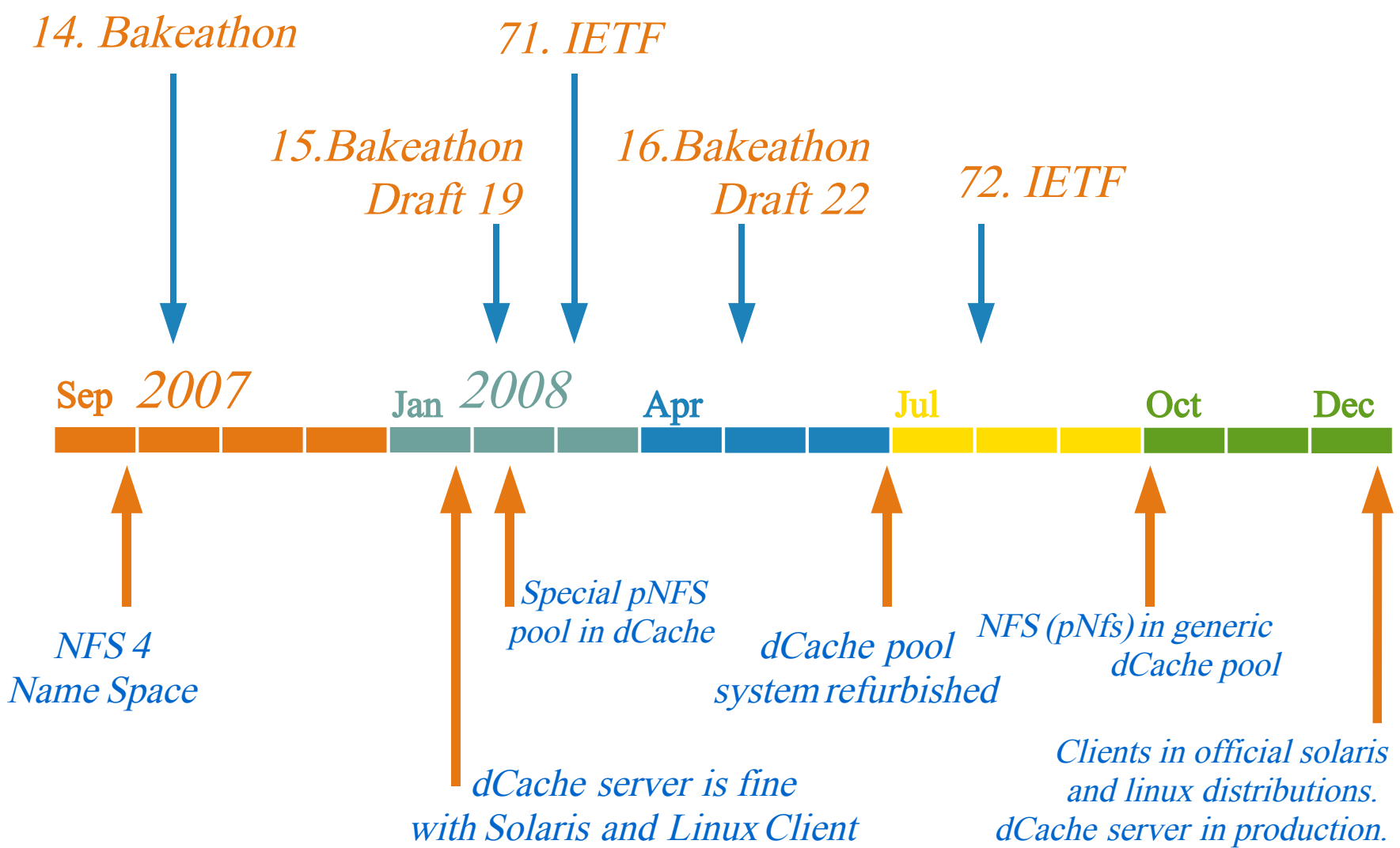
dCache.ORG





# NFS 4.1 in dCache : time-line

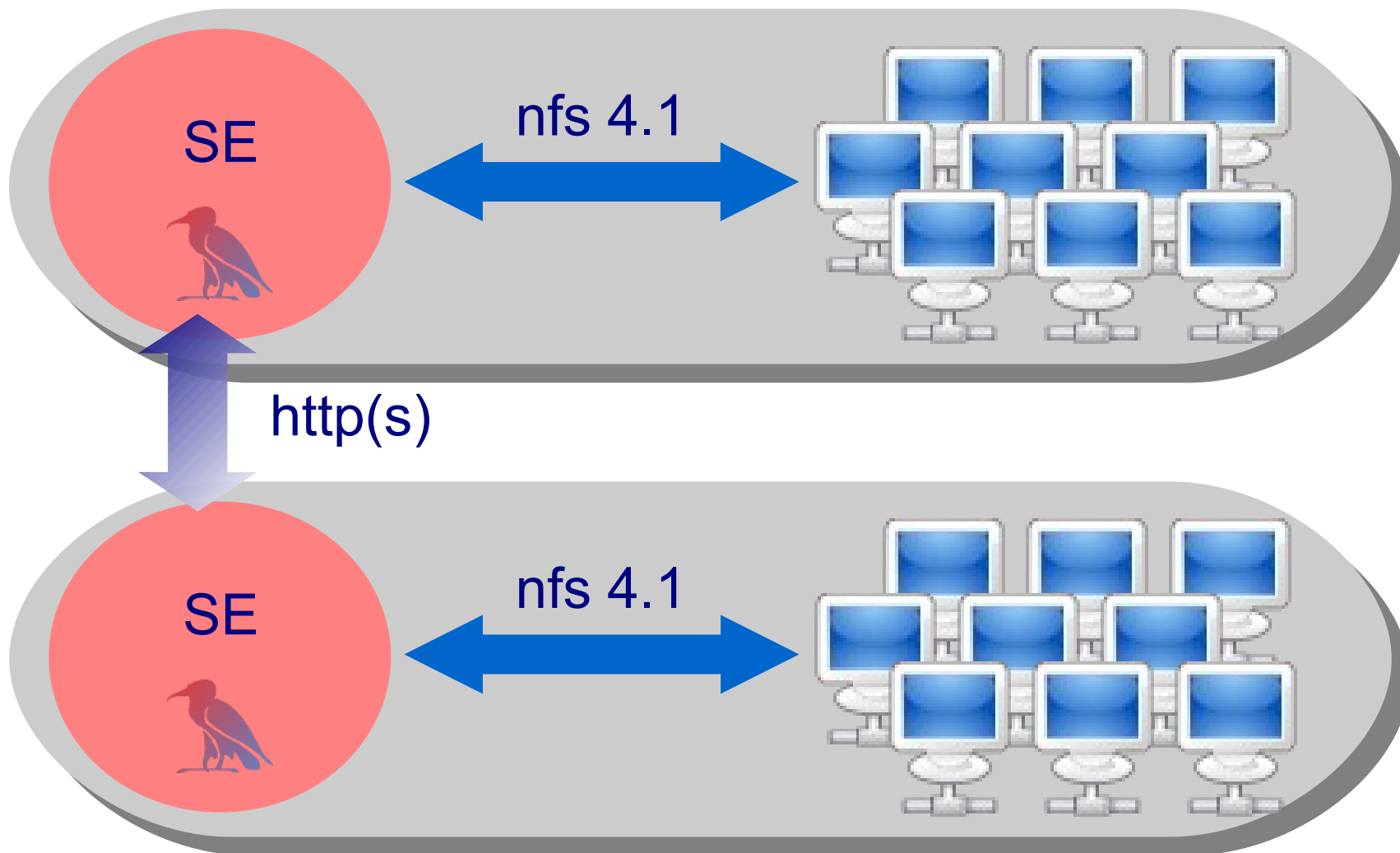
dCache.ORG  
dCache.ORG





Goal : Industry standards in HEP ?

*SRM, can we make this a real standard ?*





## Summary

- *dCache is a highly scalable Storage Element*
- *providing LHC Grid interfaces.*
- *dCache is in production at the majority of the large LCG sites and holds the majority of LHC data.*
- *dCache is pushing for standards allowing to provide community/science independent storage systems.*
- *Do we need remote storage resource management and is the SRM the right solution ?*
- *Should we go for NFS 4.1.*



## *Further reading*

*[www.dCache.ORG](http://www.dCache.ORG)*

*NFS 4.1 : [www.citi.umich.edu/projects/nfsv4/](http://www.citi.umich.edu/projects/nfsv4/)*

*SRM: <http://sdm.lbl.gov/srm-wg/>*





Quotes are stolen from CITI wiki:

And what is *NFS 4.1* ?

- ! “NFSv4.1 extends NFSv4 with two major components: *sessions and pNFS*”  
*Parallel : is exactly what we need !!!*

*IETF Road Map*

- ! “Draft 19 is expected to follow the Austin Bakeathon and be issued as an RFC following the 71st *IETF Meeting in Philadelphia (March 2008)*. This will freeze the specification of sessions, generic pNFS protocol issues, and pNFS file layout”  
*March : exactly when we need it !!!*

Who are the *nfs4, (pNFS) partners* ?

- ! *All known storage big shots, gdfs(IBM), Sun, EMC, Panasas, netApp, Lustre (Sun), dCache*  
*exactly what our clients need !!!*

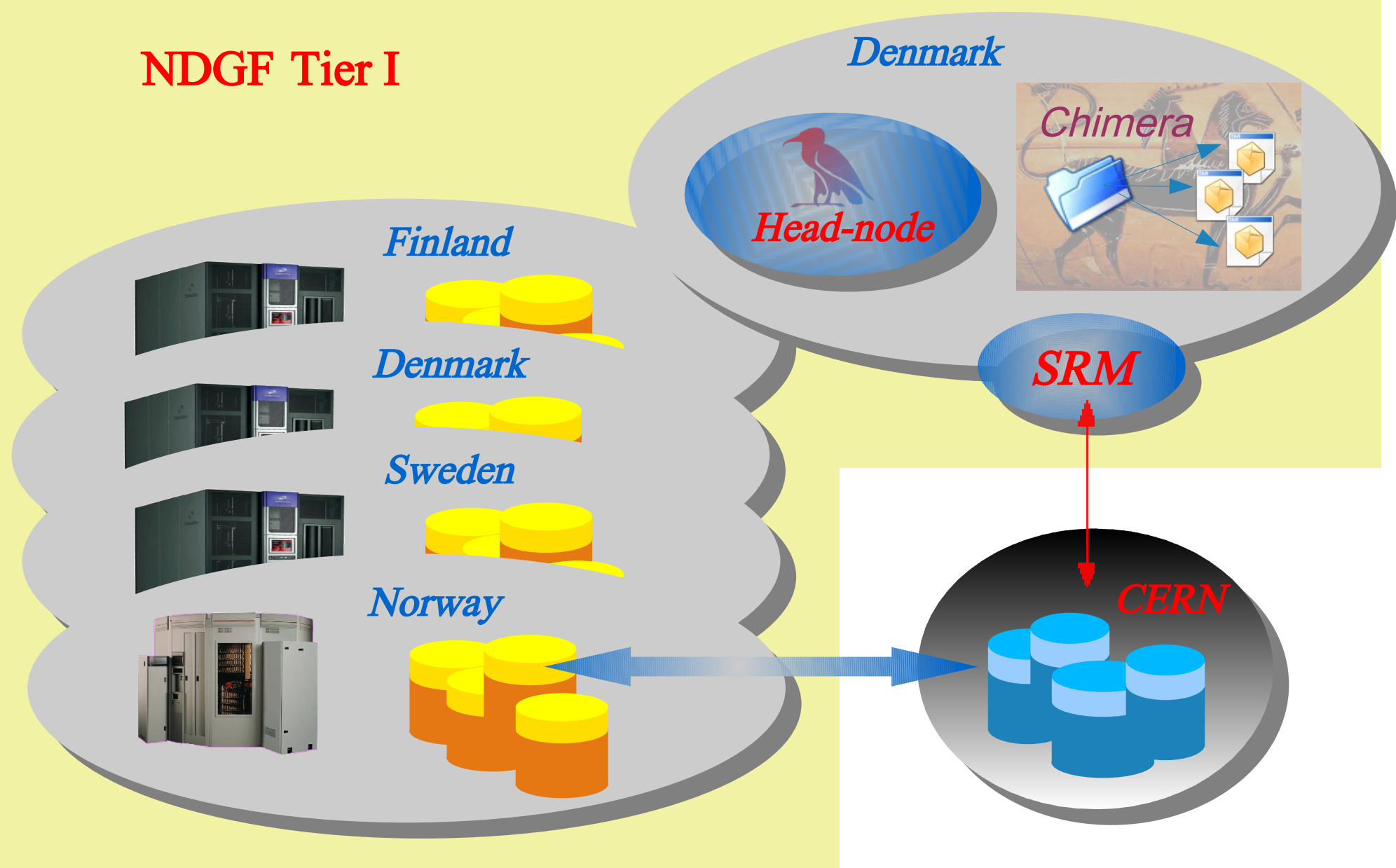


# The NDGF Challenge : gsiFtp Protocol Version II

dCache.ORG

dCache.ORG

## NDGF Tier I



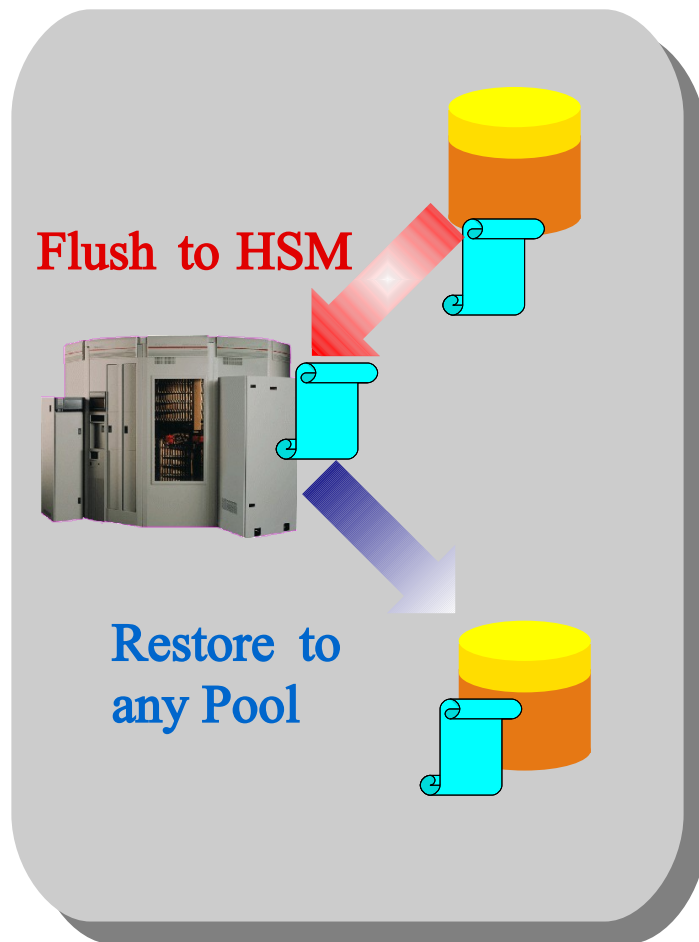


# The NDGF Challenge : Multi Site HSM support

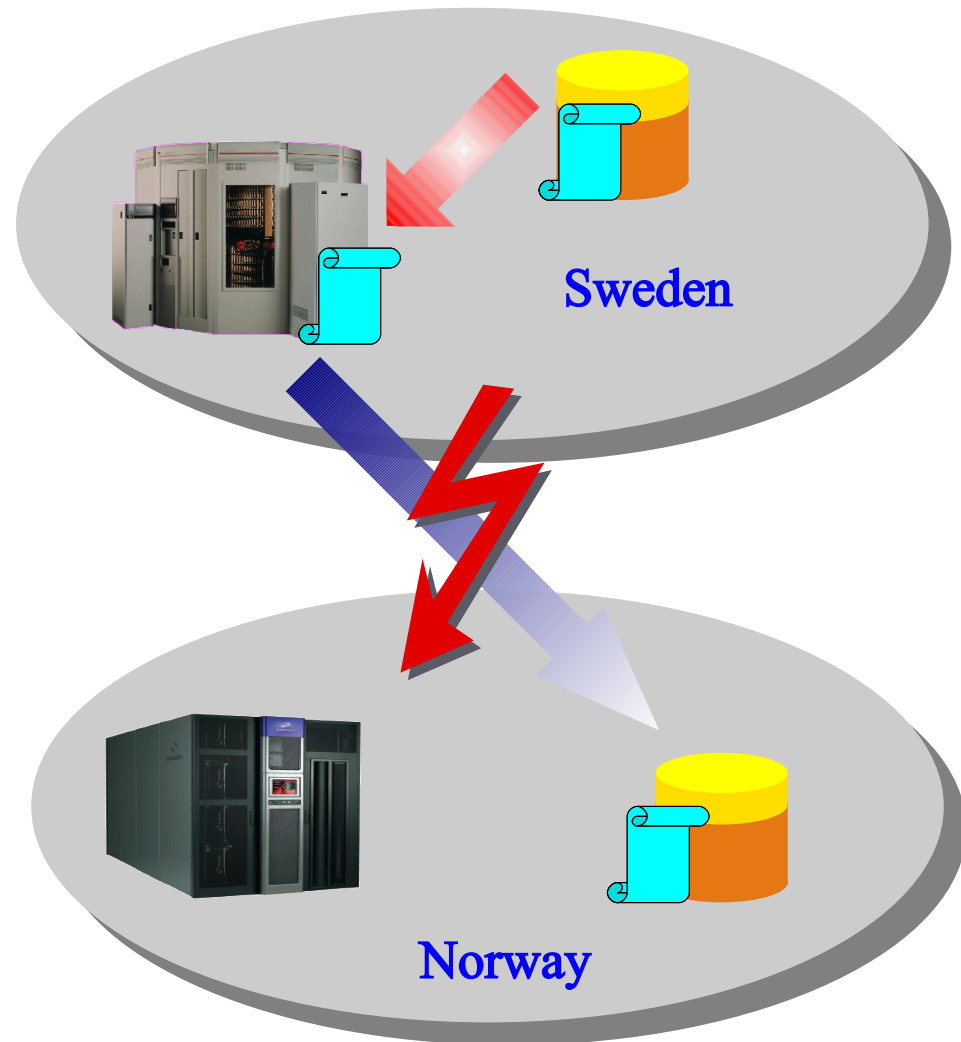
dCache.ORG

dCache.ORG

## Single Site approach



## Multi Site approach

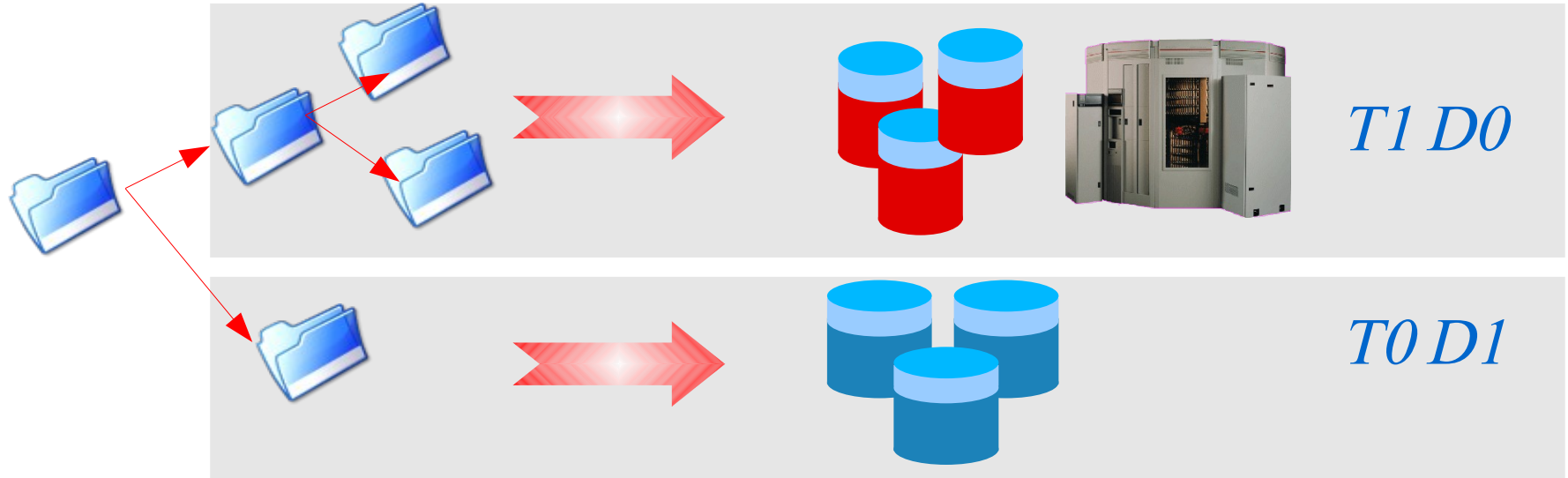


*Not all pools can access all HSM systems*



# SRM2.2 ( The space token )

As it used to be ( <= 1.7 )



As it will be with 1.8

