dCache: sneaking up on NFS4.1

Tigran Mkrtchyan
Björn Böttcher

Patrick Fuhrmann

for the dCache Team

support and funding by

DESY

NDGF
Nordic DataGrid Facility

HEP CG
D-GRID

Open Science Grid

VDT
Virtual Data Toolkit

eGee
Enabling Grids for E-sciencE

What is dCache.ORG

Head of dCache.ORG

Patrick Fuhrmann

Core Team (Desy and Fermi)

Andrew Baranovski
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

Gerd Behrmann, NDGF
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 SE.

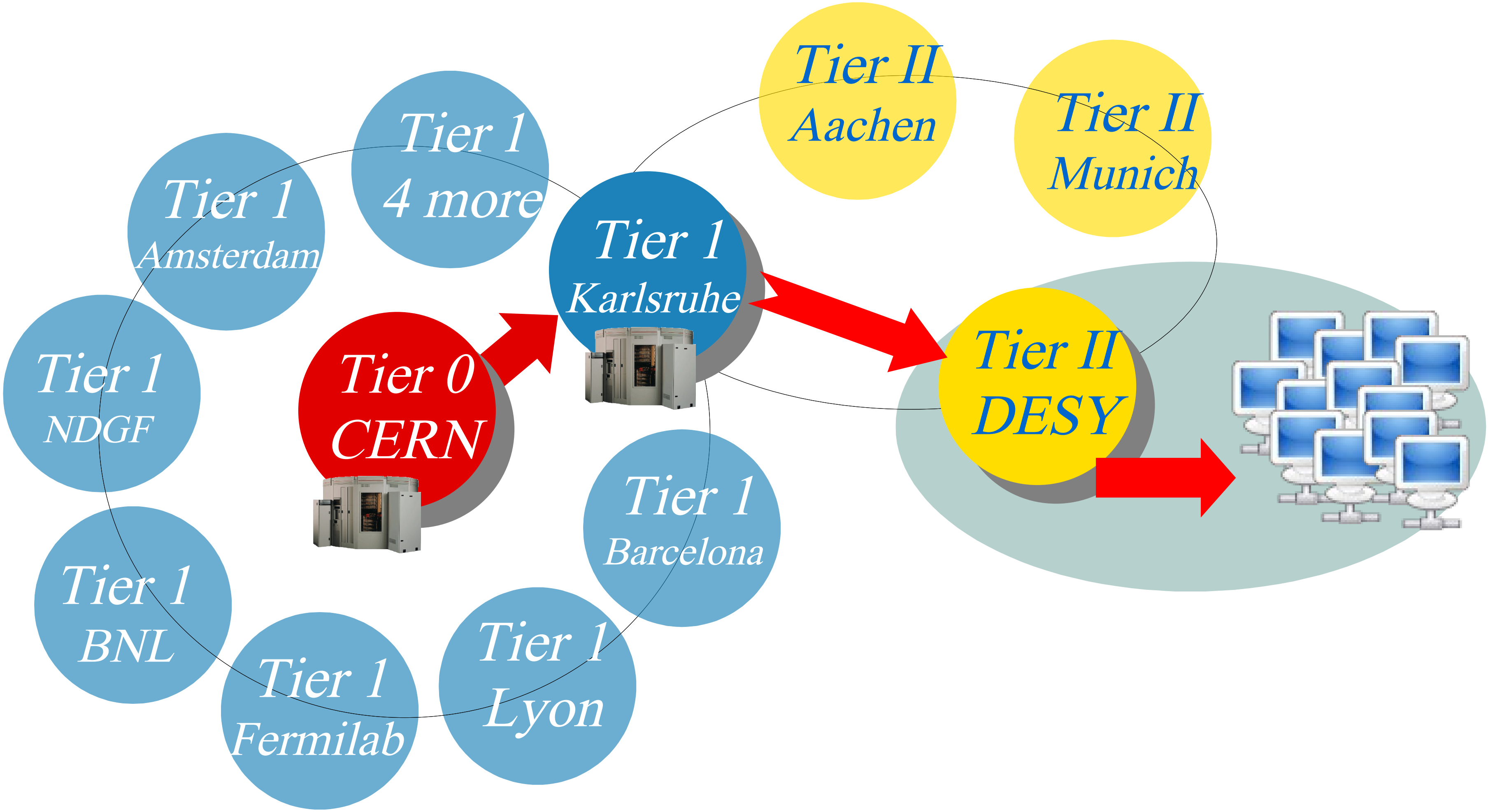
What is a dCache SE ?

WhyNFS 4.1 ?

The LHC Tier model and the SE.

LHC (Data Management) Tier Structure

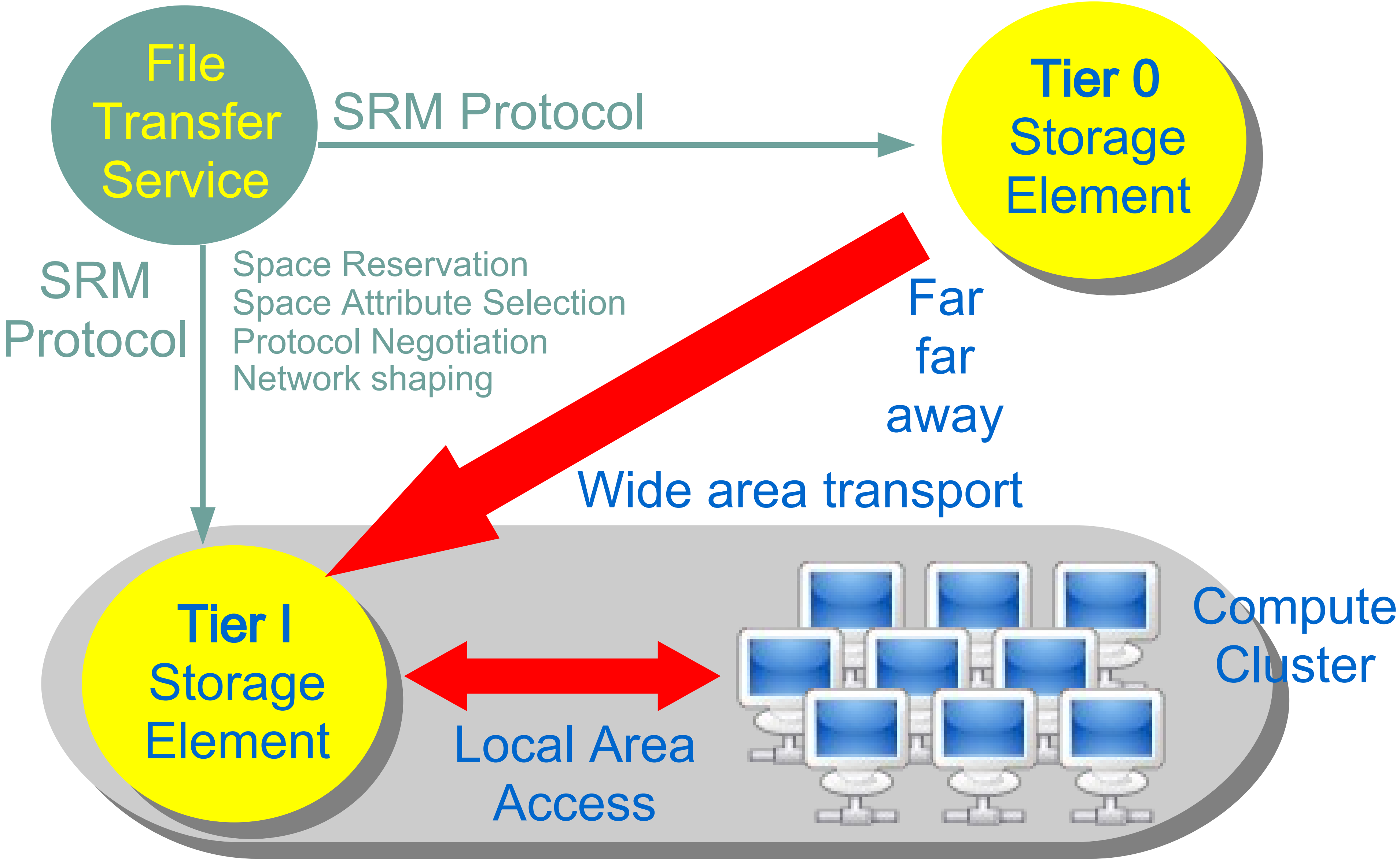
Significantly oversimplified



The Storage Element,
the Storage Management Workhorse

- Streaming data IMPORT and EXPORT
- Posix like access from local worker-nodes
- Managing storage

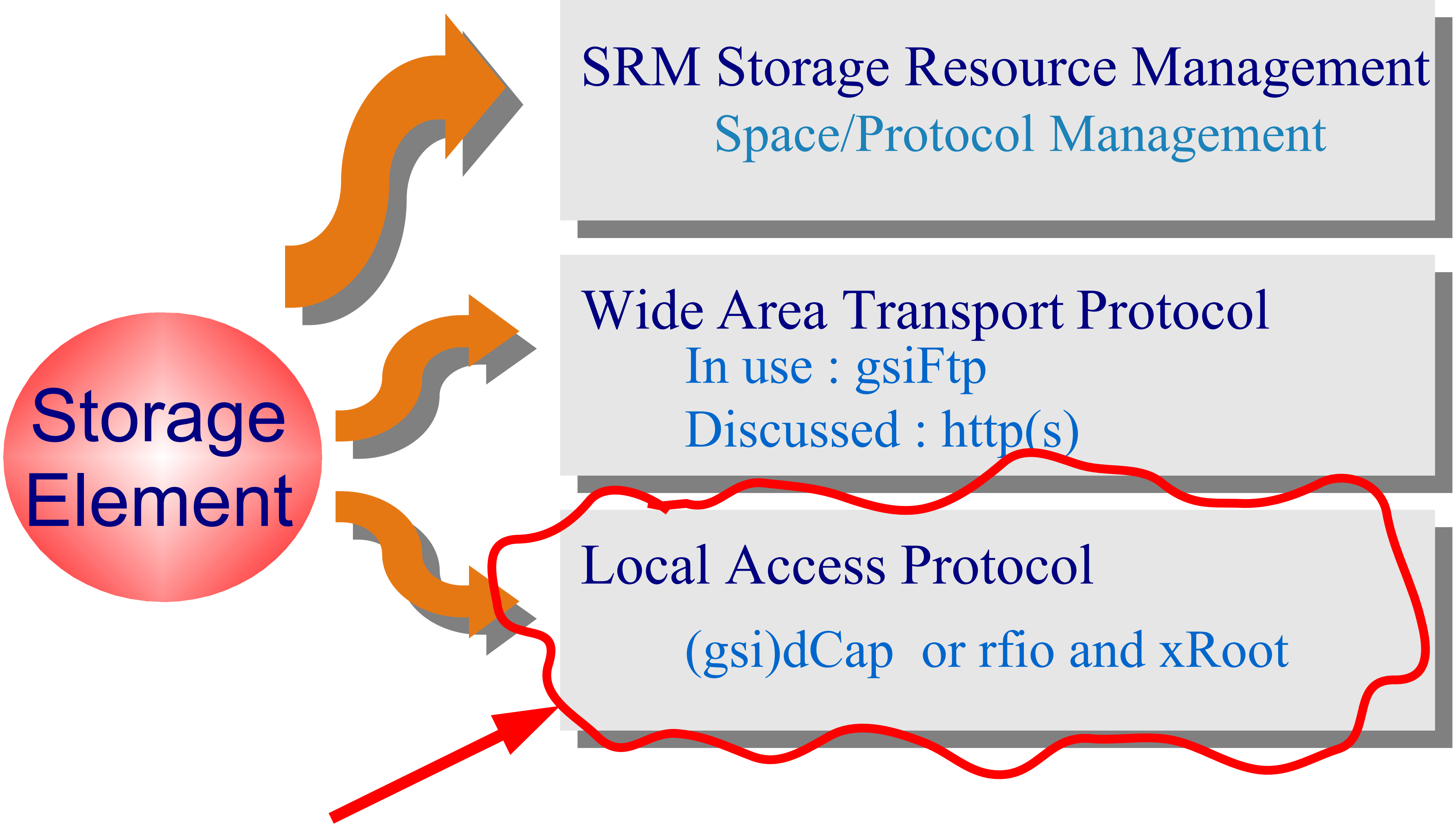
The Idea of a Grid Storage Element



Intentionally not mentioned here

- *Information Provider Protocols*
- *File Catalogs*

The Idea of a (LCG) Grid Storage Element



Storage Element

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 do we need a grid storage element for ?

We needto 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 needto 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 needto allow storage control

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

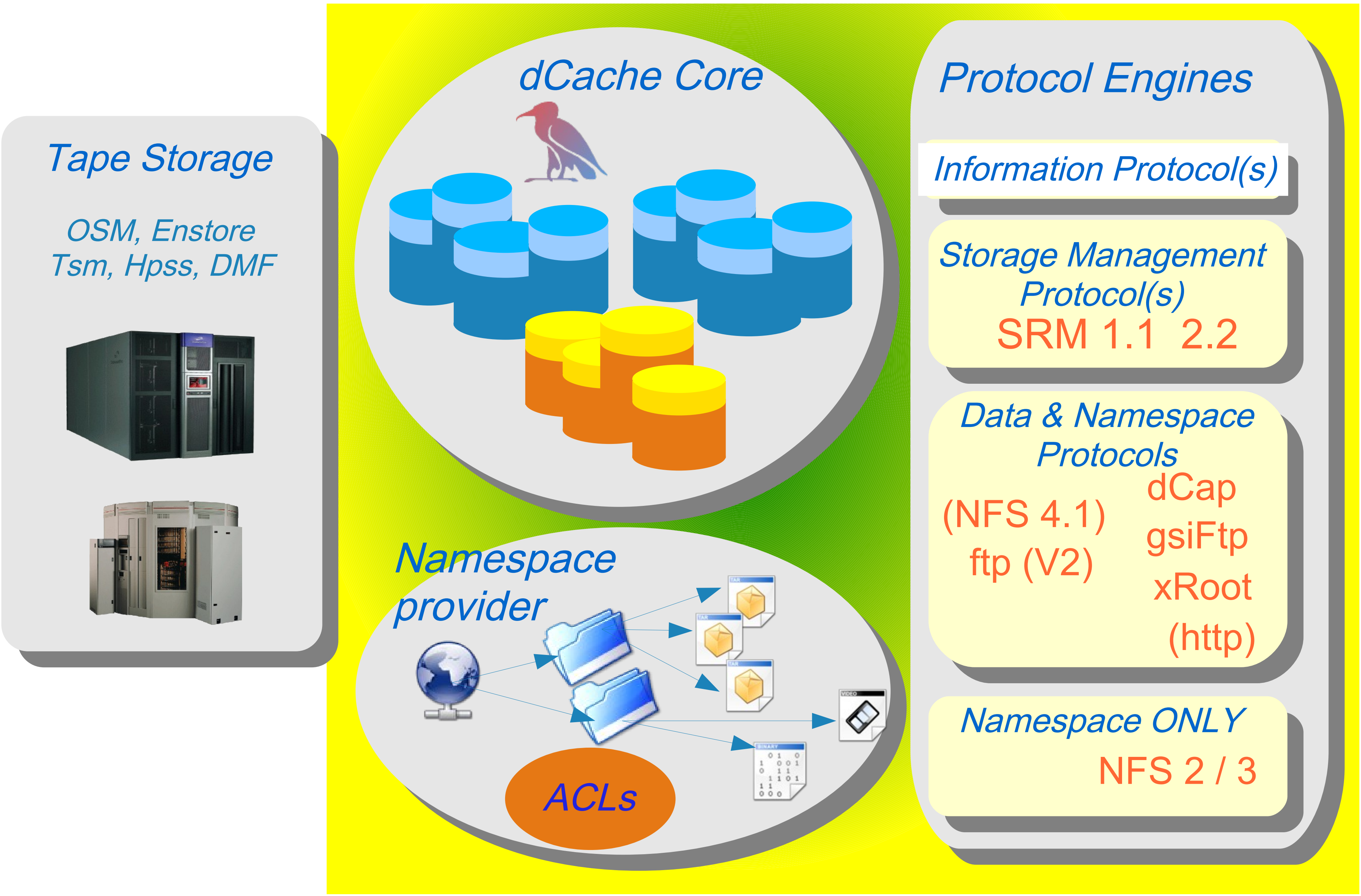
We needto publish SE specific information

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

dCache in a Nutshell

dCache in a Nutshell

Black (yellow) Box View



Tape Storage

OSM, Enstore
Tsm, Hpss, DMF

dCache Core

Namespace provider

ACLs

Protocol Engines

Information Protocol(s)

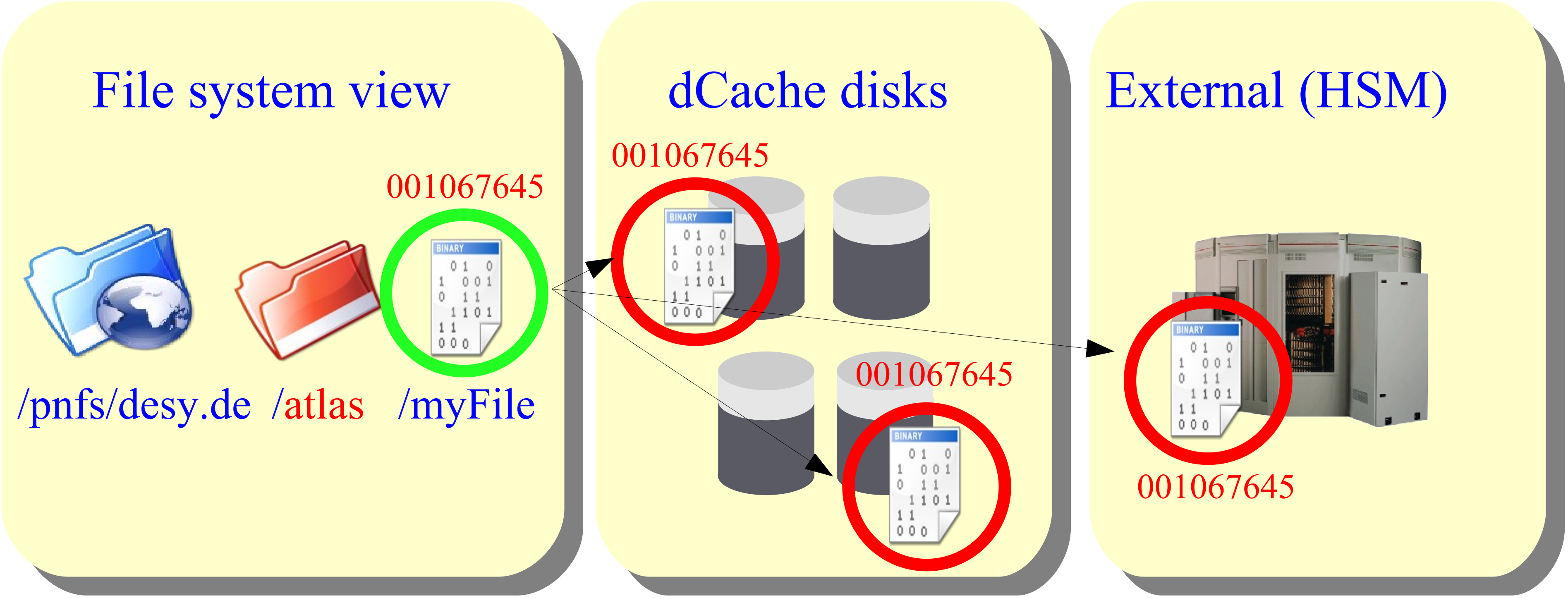
Storage Management Protocol(s)
SRM 1.1 2.2

Data & Namespace Protocols
(NFS 4.1)
ftp (V2)
dCap
gsiFtp
xRoot
(http)

Namespace ONLY
NFS 2 / 3

dCache in a Nutshell

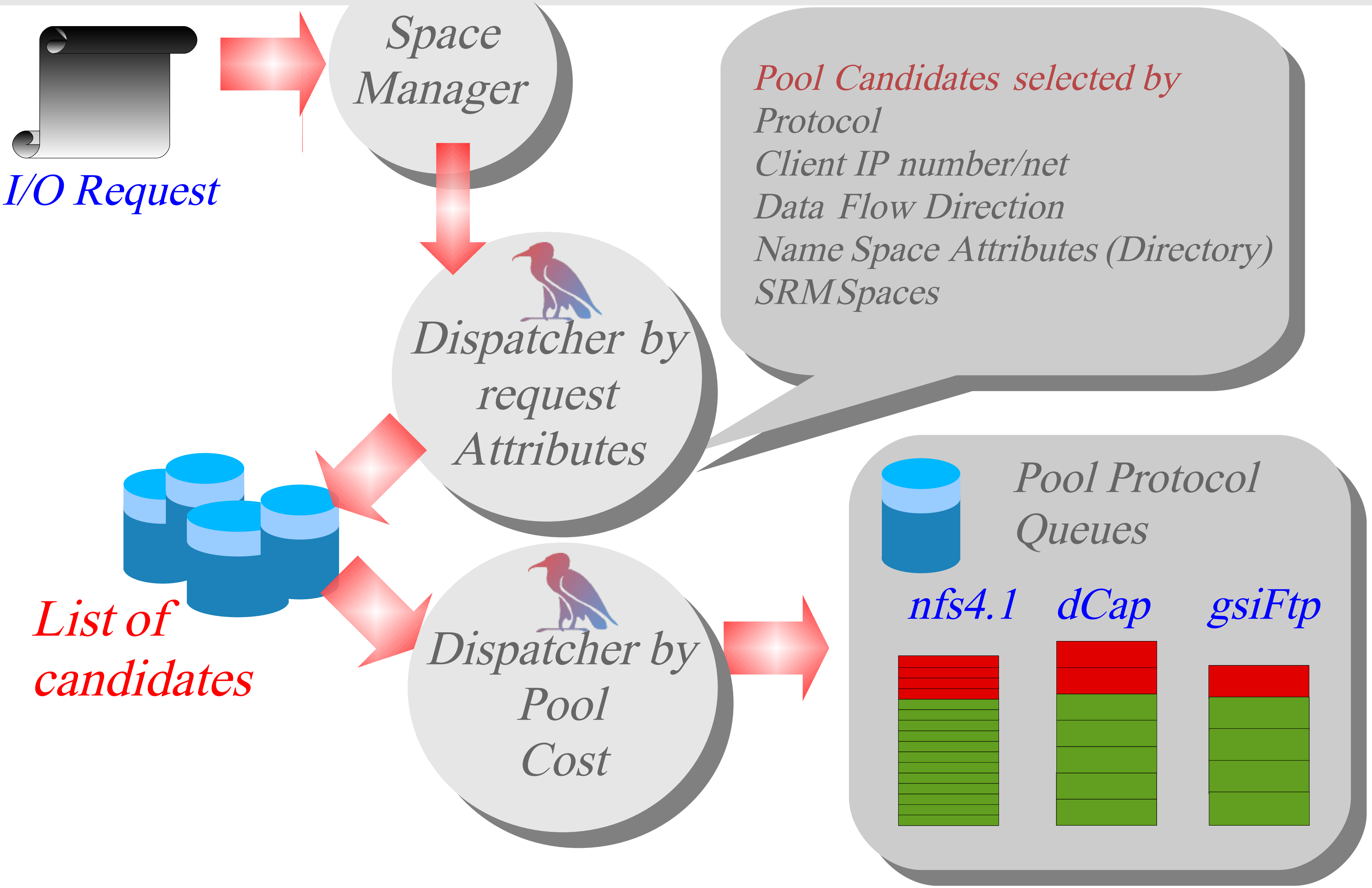
- **Strict name space and data storage separation, allowing**
  - consistent name space operations (mv, rm, mkdir e.t.c)
  - consistent access control per directory resp. file
  - managing multiple internal and external copies of the same file
  - convenient name space management by nfs (or http)



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)

Scheduler and I/O queues



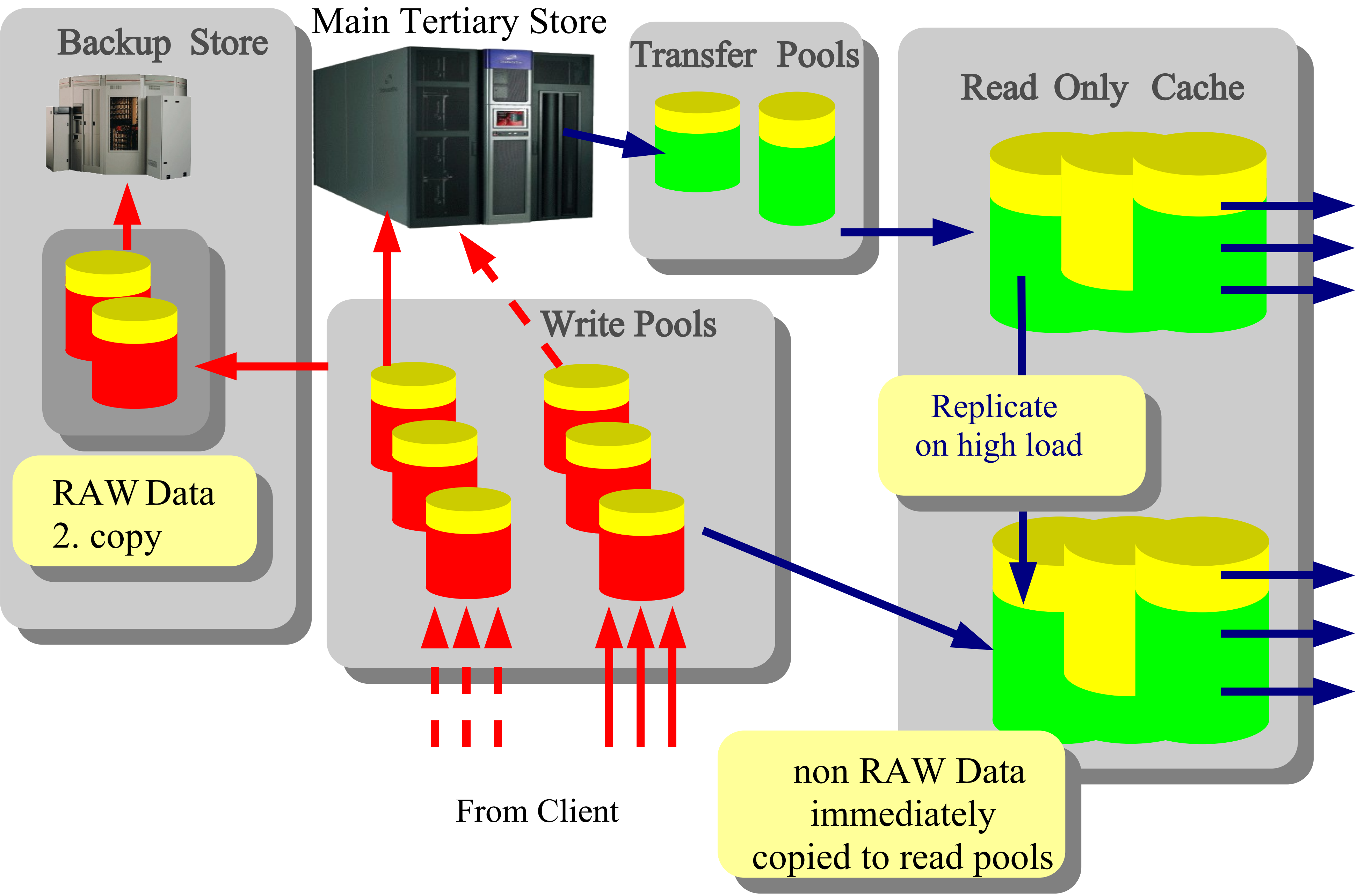
Pool Candidates selected by
Protocol
Client IP number/net
Data Flow Direction
Name Space Attributes (Directory)
SRM Spaces

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



RAW Data
2. copy

Replicate
on high load

non RAW Data
immediately
copied to read pools

In the Nutshell

- **HSM Support**
  - TSM, HPSS, DMF, Enstore, Osm
  - Automated migration and restore
  - Working on Central Flush facility
  - 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 (NorduGrid)

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.

Is this useful for non LCG applications ?

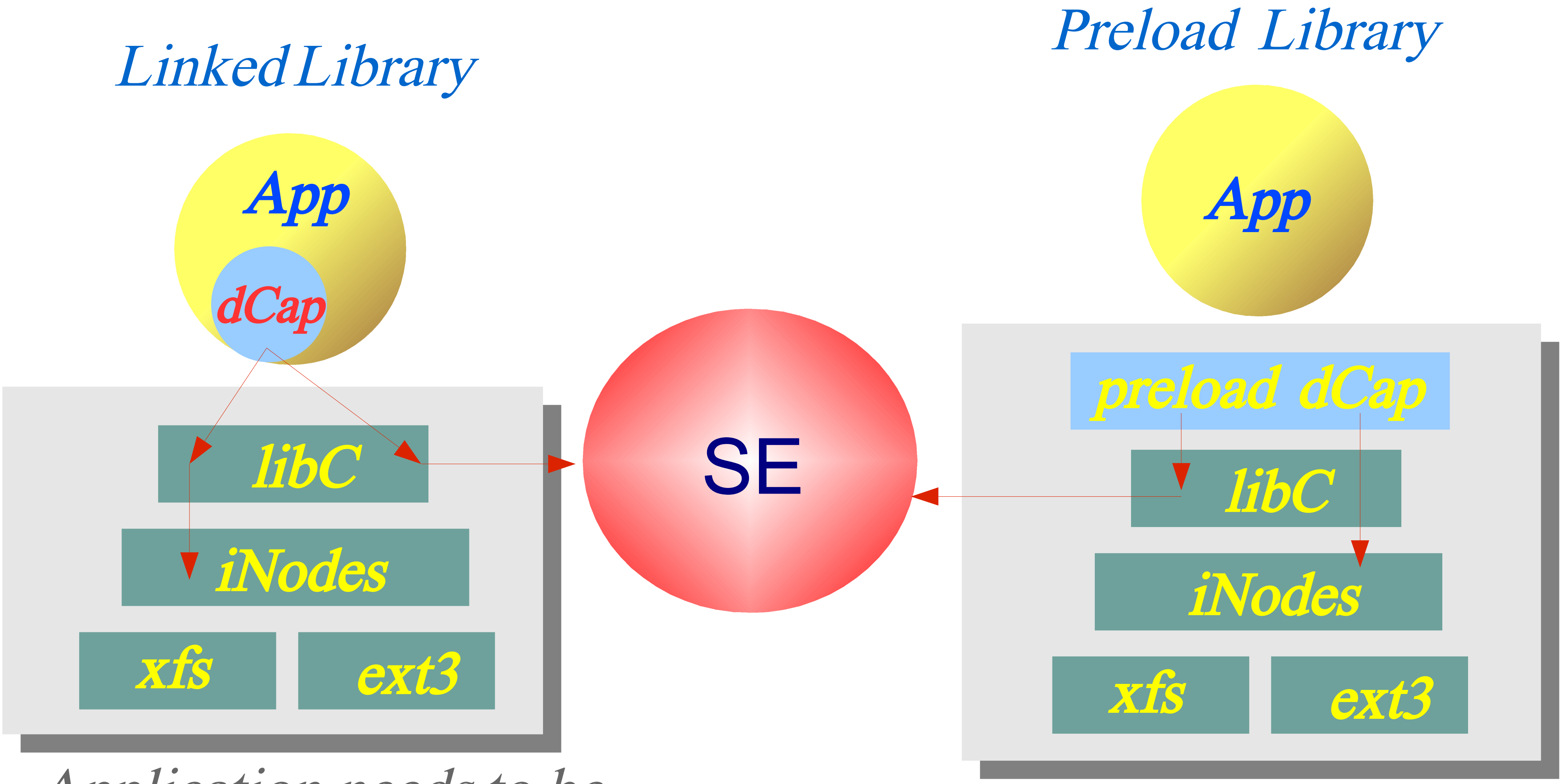
Weakpoints :

Http(s) not really supported

Security might not be sufficient

“Posix like” is NOT posix (file system driver)

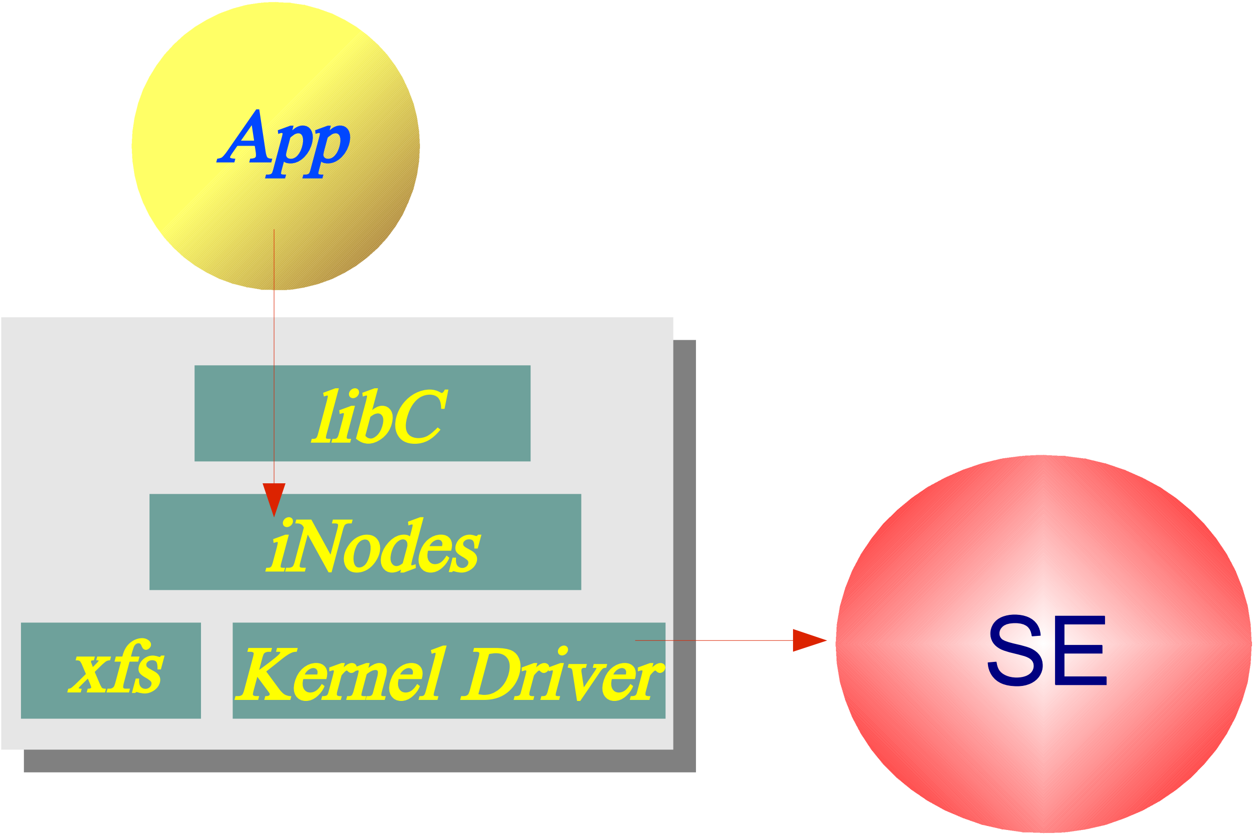
Posix like is NOT posix



Application needs to be linked with the dCap library.

Application stays unchanged but doesn't work in all cases. (Static linked, Some C++ apps.)

And this is real posix



Application doesn't need to be changed.
Kernel driver comes with OS.
But dCache.org doesn't want to write/support kernel drivers.

Solution is on the way....

NFS 4.1

And another project : NFS 4 within CITI

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.

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, gpfs(IBM), Sun, EMC,Panasas, netApp, Lustre (Sun), dCache*

exactly what our clients need !!!

NFS 4.1 : More details

- dCache is invited to the regular bakeathons.
- CITI, IBM and others are working on Linux client implementation
- A stable client implementation is essential for industry to sell their products. -> we profit.
- Bakeathon last week : except for sparse files, the dCache server could interact reliably with all client implementations.
- Currently, NFS4.1 is only available as a special pool within dCache.
- We are currently refurbishing the generic pool in order to integrate NFS4.1.

Whyis NFS 4.1 : project perspective

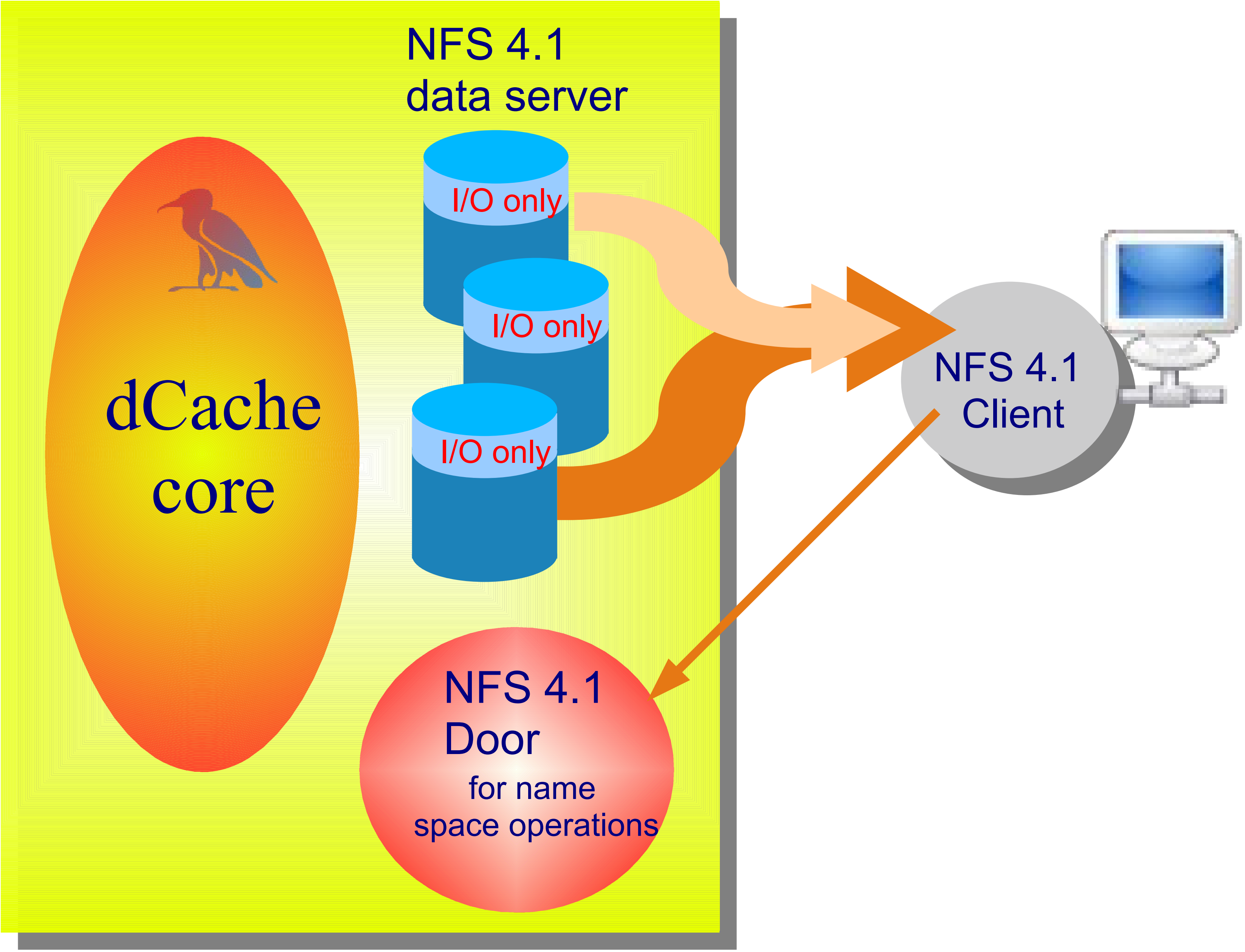
dCache.ORG

- 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 LAN protocol zoo (dcap,rfio,xroot)

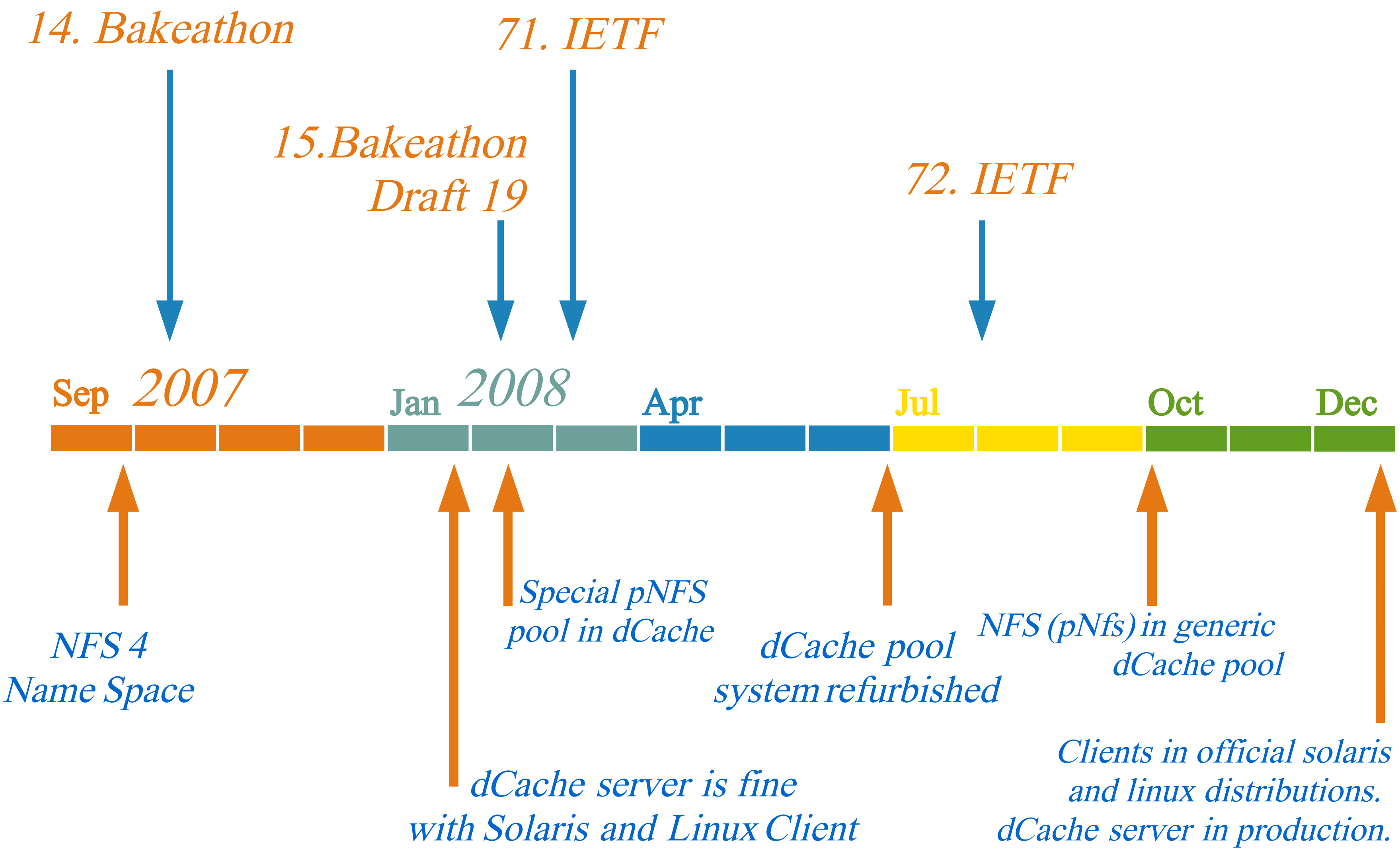
Whyis NFS 4.1 : technical perspective

- 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 : technically



NFS 4.1 in dCache : time-line



14. Bakeathon

15.Bakeathon
Draft 19

71. IETF

72. IETF

Sep *2007*

Jan *2008*

Apr

Jul

Oct

Dec

NFS 4
Name Space

dCache server is fine
with Solaris and Linux Client

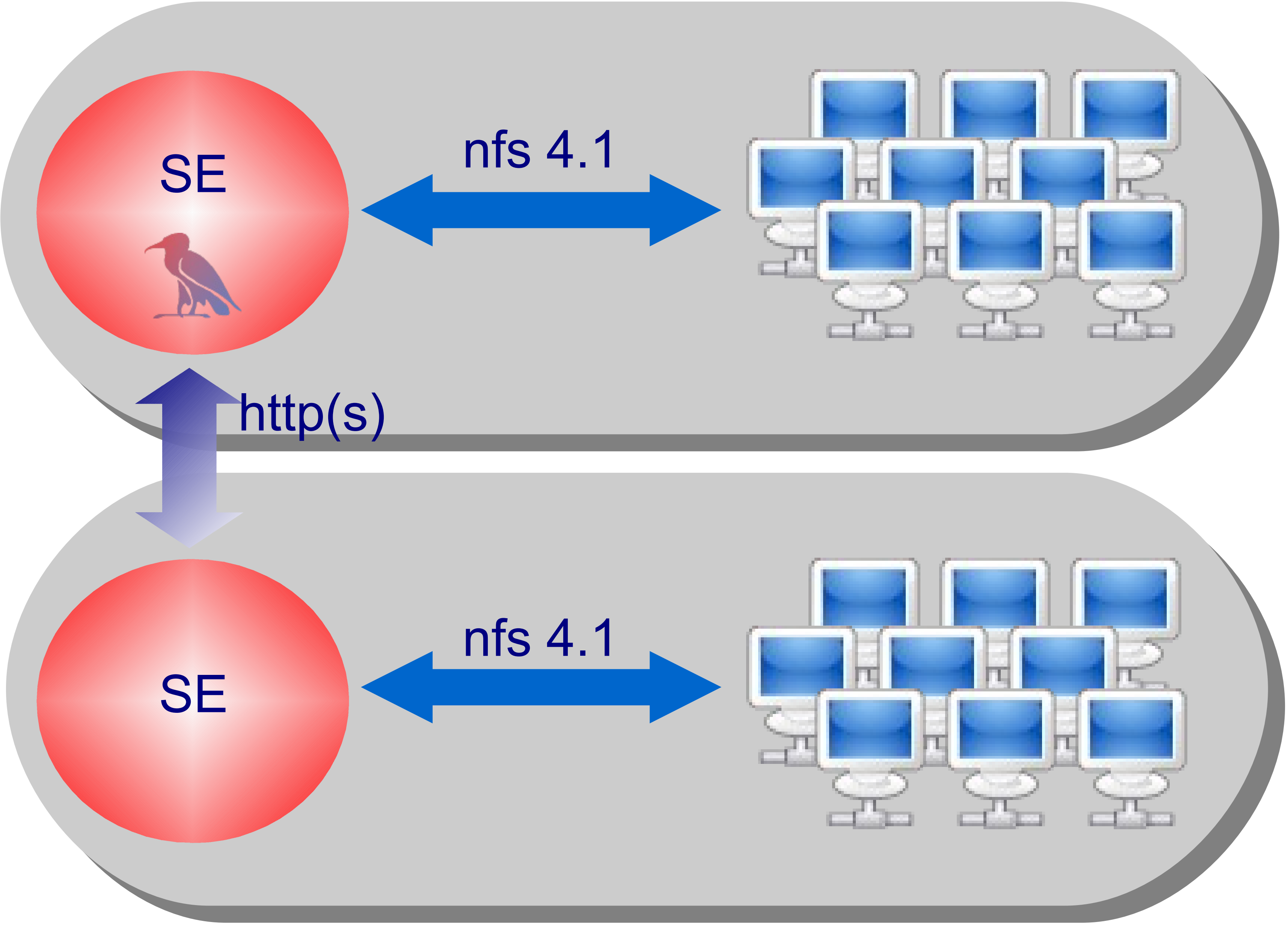
Special pNFS
pool in dCache

dCache pool
system refurbished

NFS (pNfs) in generic
dCache pool

Clients in official solaris
and linux distributions.
dCache server in production.

Goal : Industry standards in HEP ?



Summary

- NFS 4.1 (pNFS) is just an additional protocol for dCache
- NFS 4.1. simplifies LANPosix access to dCache.
- Applications don't need special treatment any more
- NFS4.1/dCache is attractive for non HEP communities.
- We expectproduction system end of 2008
- BUT : Success resp acceptance not guarantied yet.

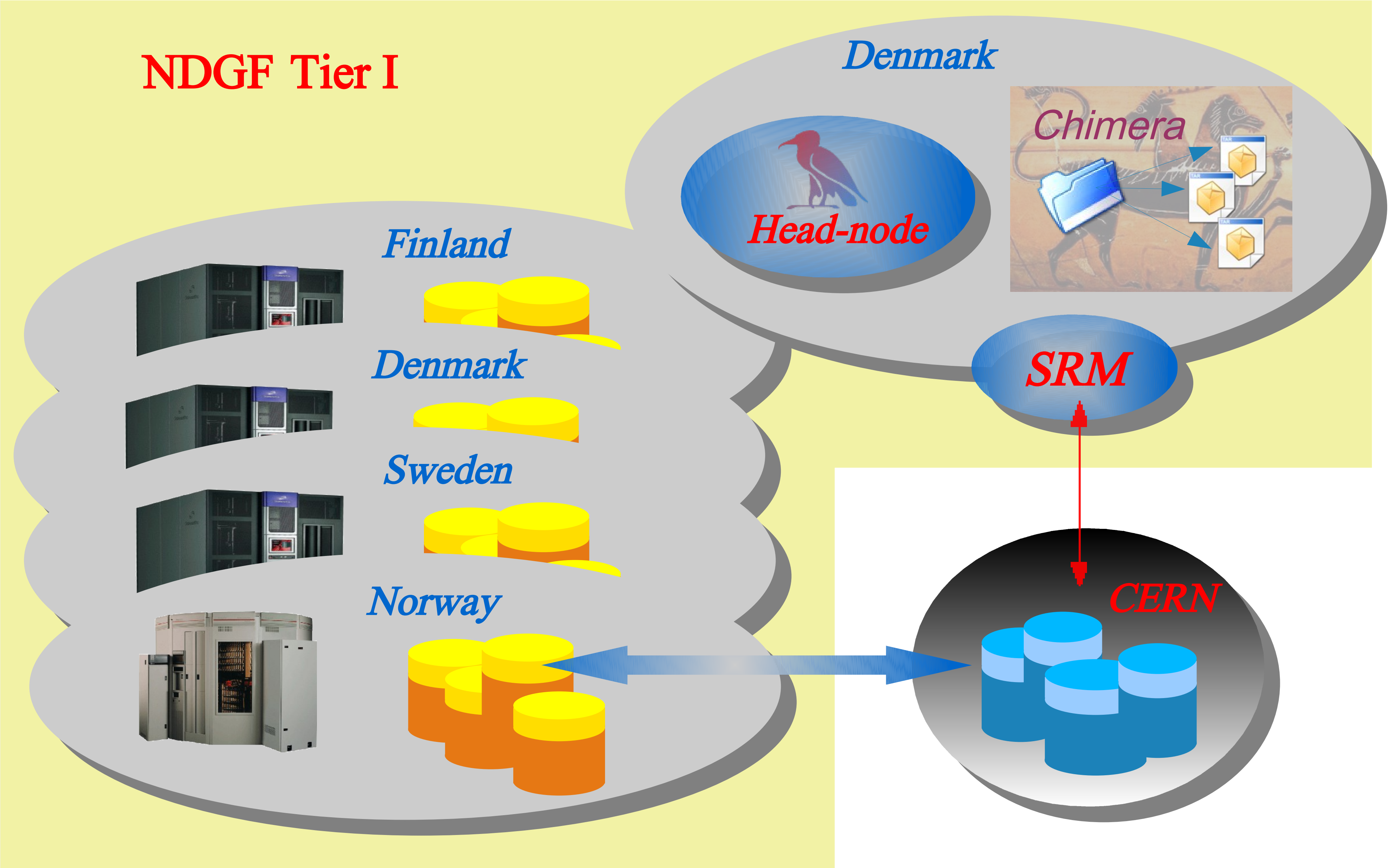
Further reading

www.dCache.ORG

www.citi.umich.edu/projects/nfsv4/

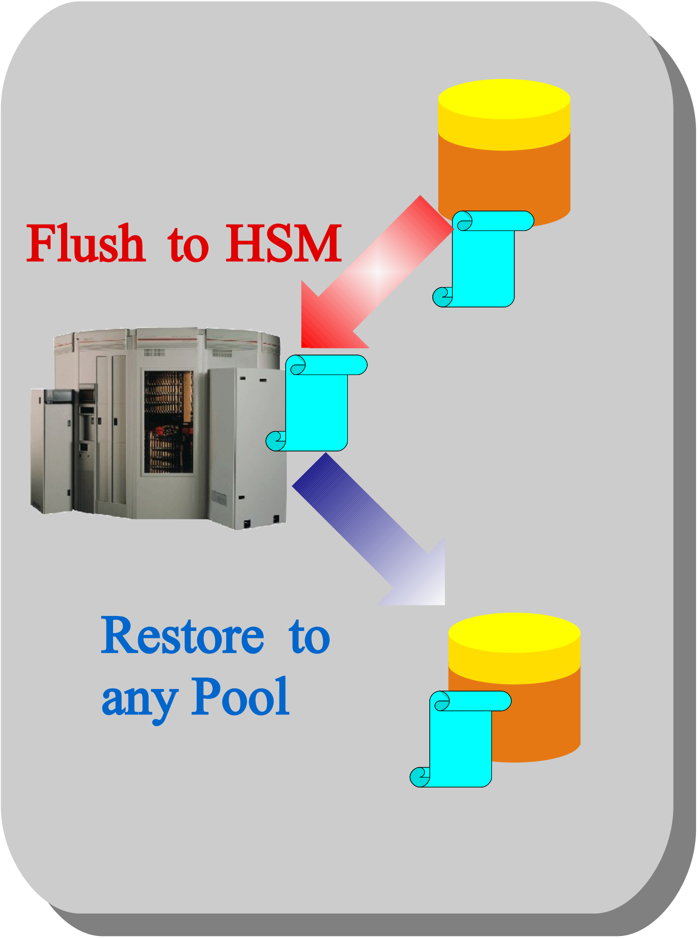
Some more hot topics

The NDGF Challenge : gsiFtp Protocol Version II

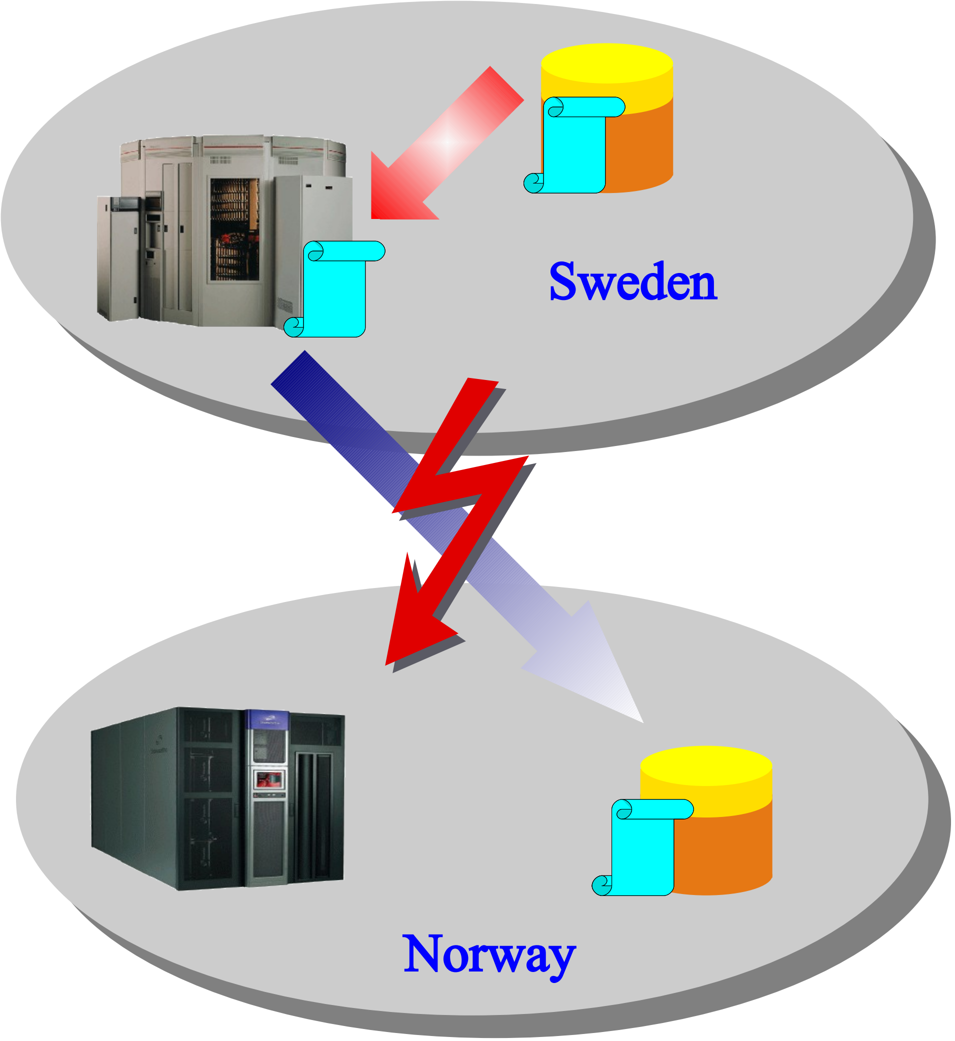


The NDGF Challenge : Multi Site HSMsupport

Single Site approach



Multi Site approach



Not all pools can access all HSM systems

The wonderful world of

SRM2.2

Only if there is a lot of time left

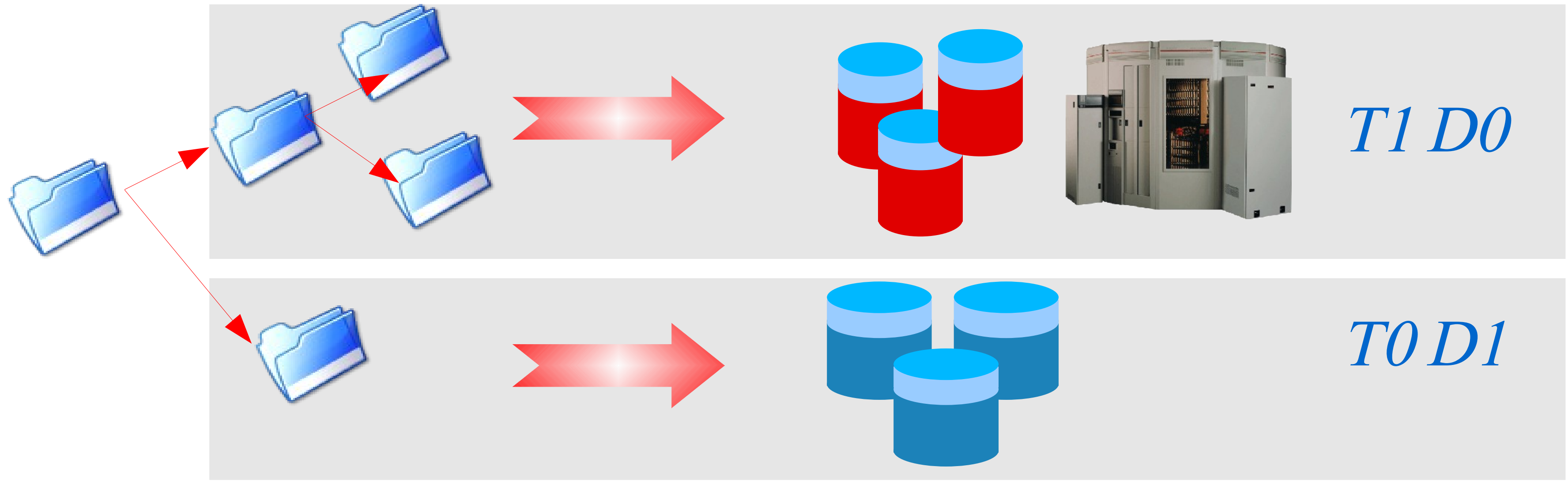
SRM2.2

The SRM in dCache supports

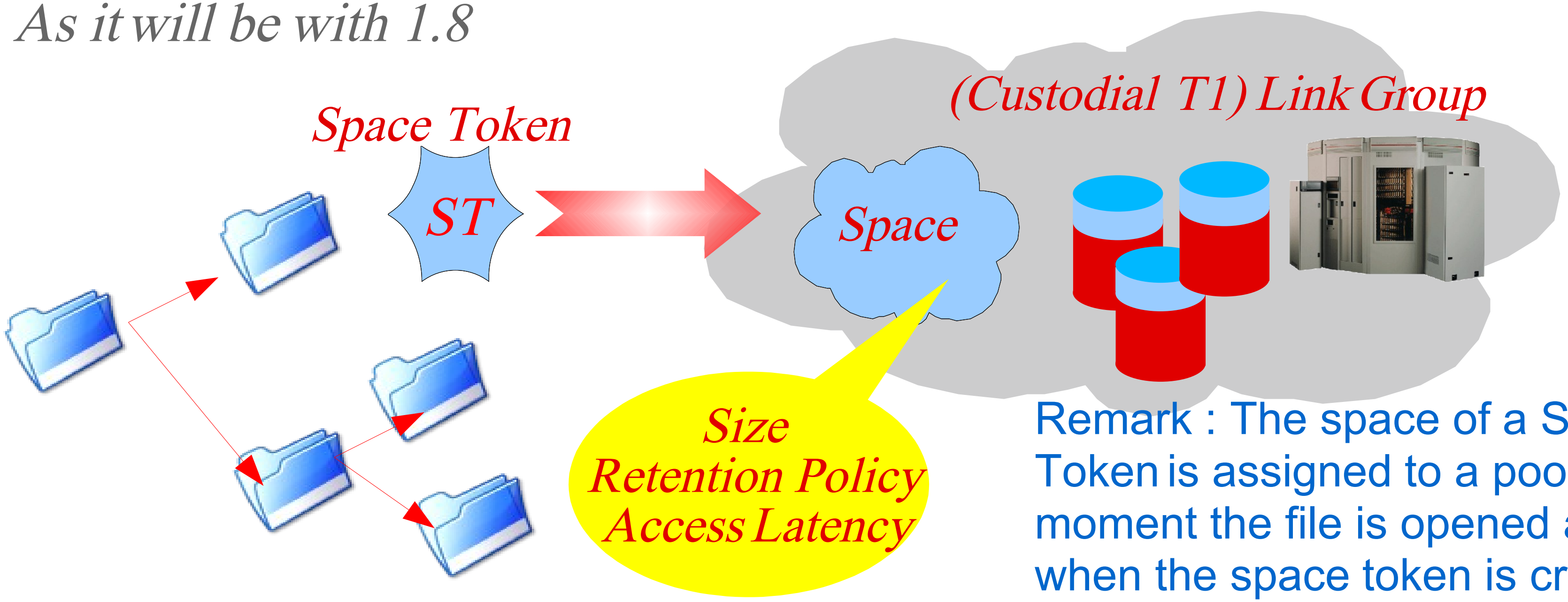
- *CUSTODIAL (T1Dx)*
- *NON-CUSTODIAL (T0D1)*
- *Dynamic Space Reservation*
- *late pool binding for spaces*
- *and more*

SRM2.2 (The space token)

As it used to be (<= 1.7)



As it will be with 1.8



Remark : The space of a Space Token is assigned to a pool at the moment the file is opened and not when the space token is created.