



# *dCache, the Update*

*Patrick Fuhrmann*

*for the dCache people*





*Responsibility, dCache*

Patrick Fuhrmann      Rob Kennedy

*Responsibility, SRM*

Timur Perelmutov

*Core Team (Desy and Fermi)*

Jon Bakken

Mathias de Riese

Micheal Ernst

Alex Kulyavtsev

Birgit Lewendel

Dmitri Litvintsev

Tigran Mrktchyan

Martin Radicke

Neha Sharma

Vladimir Podstavkov

*External*

*Development*

Nicolo Fioretti, BARI

Abhishek Singh Rana, SDSC

*Support and Help*

Maarten Lithmaath, CERN

Owen Synge, RAL





## dCache, the Idea

The story so far ...

what's new , what's coming next

### **Feature Track**

*SRM V2*

*Replica Manager*

*Information Provider*

*File Hopping*

*Smart Tape System flushing*

*Generic Tertiary Storage Interface*

*dCache partitioning*

*VOMS integration*

### **Performance Track**

*medium term strategy*

*Multiple I/O queues*

*Multiple Pnfs (file systems)*

*Chimera*





Distributed Peta Byte Disk Store with single rooted filesystem providing posix like and wide area access protocols.

Distributed cache system to optimize access to Tertiary Storage Systems

Grid Storage Element coming with standard data access protocols, Information Provider Protocols and Storage Resource Manager.





*The story so far ...*





## *Basic Specification*

Single 'rooted' file system name space tree

File system names space view available through an nfs2/3 interface

Data is distributed among a huge amount of disk servers.

Supports multiple internal and external copies of a single file

Supports 'posix like' (authenticated) access as well as various FTP dialects and the Storage Resource Manager Protocol.





## *Configuration*

Fine grained configuration of *pool attraction scheme*.  
(*write pools, subnet, directory tree, storage info*)

Pool to pool transfers on configuration of *forbidden transfers*

Fine grained tuning : Space vs. Mover cost preference

## *Tertiary Storage Manager connectivity*

Automatic HSM migration and restore

HSM dCache interface by script (shell, perl ...)

Convenient HSM connectivity for  
enstore, osm, tsm and HPSS





## *Scalability*

Distributed Movers AND Access Points (Doors)

Automatic load balancing using cost metric and inter pool transfers.

Pool 2 Pool transfers on pool hot spot detection

Handles bunch requests by fast pool selection unit







## *Configuration*

Fine grained configuration of *pool attraction scheme*.  
(*write pools, subnet, directory tree, storage info*)

## *Tertiary Storage Manager connectivity*

Automatic HSM migration and restore

HSM dCache interface by script (shell, perl ...)

Convenient HSM connectivity for  
enstore, osm, tsm, preliminary for Hpss by BNL.





## *Administration*

Using standard 'ssh' protocol for administration interface.

First version of graphical interface available for administration

Powerful command-set per module





*What's new ?*

*What's coming next ?*





# *Feature Track*

*SRM V2*

*Replica Manager*

*Information Provider*

*File Hopping*

*Smart Tape System flushing*

*Generic Tertiary Storage Interface*

*dCache partitioning*

*VOMS integration*





*Please refer to poster by Alexander Kulyavtsev*

## **Resilient dCache**

- Controls number of copies for each dataset in dCache
- Makes sure  $n < \text{copies} < m$
- Adjusts replica count on pool failures
- Adjusts replica count on scheduled pool maintenance
- Makes use of local disk space when running on farm nodes
- doesn't work with HSM back-end yet

## **Improvements**

- File copy operations (pool to pool) will be controlled by the PoolManager
- Pool Manager rules are honored (including : don't copy to same host/store)
- Pool Manager cost metrics is honored





*Please refer to talk by Timur Perelmutov*

## **Storage Resource Manager V2**

- Data Transfer and Directory methods/function available  
(Compatibility tests starting right after chep06)
- Explicit Space Management is in development phase





*Please refer to talk by Abhishek Singh Rana*

gPLAZMA (grid-aware PLuggable AuthoriZation MAnagement): Introducing RBAC (Role Based Access Control) Security in dCache

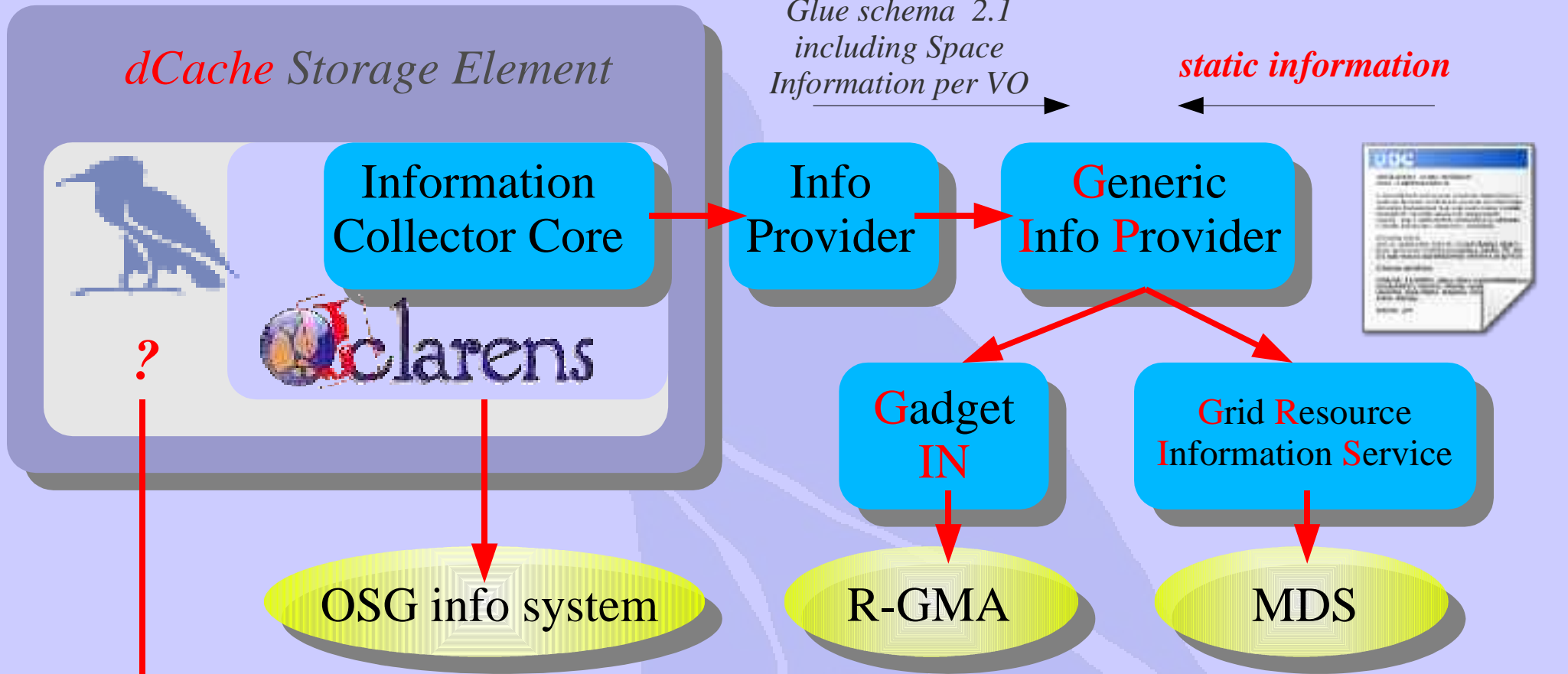
## **Virtual Organization Membership Service Integration**

- Pluggable authorization
- Backward compatible to current system.
- Storage Authorization Call-outs for SRM and GridFtp (dCap pending)
- uid/gid mapping still done locally (not a VOMS request)
- Code done, going into test phase this days.





by courtesy of Nicolò Fioretti



with many thanks to Jean-Philippe Baud and Lawrence Fields

D-Grid Info system

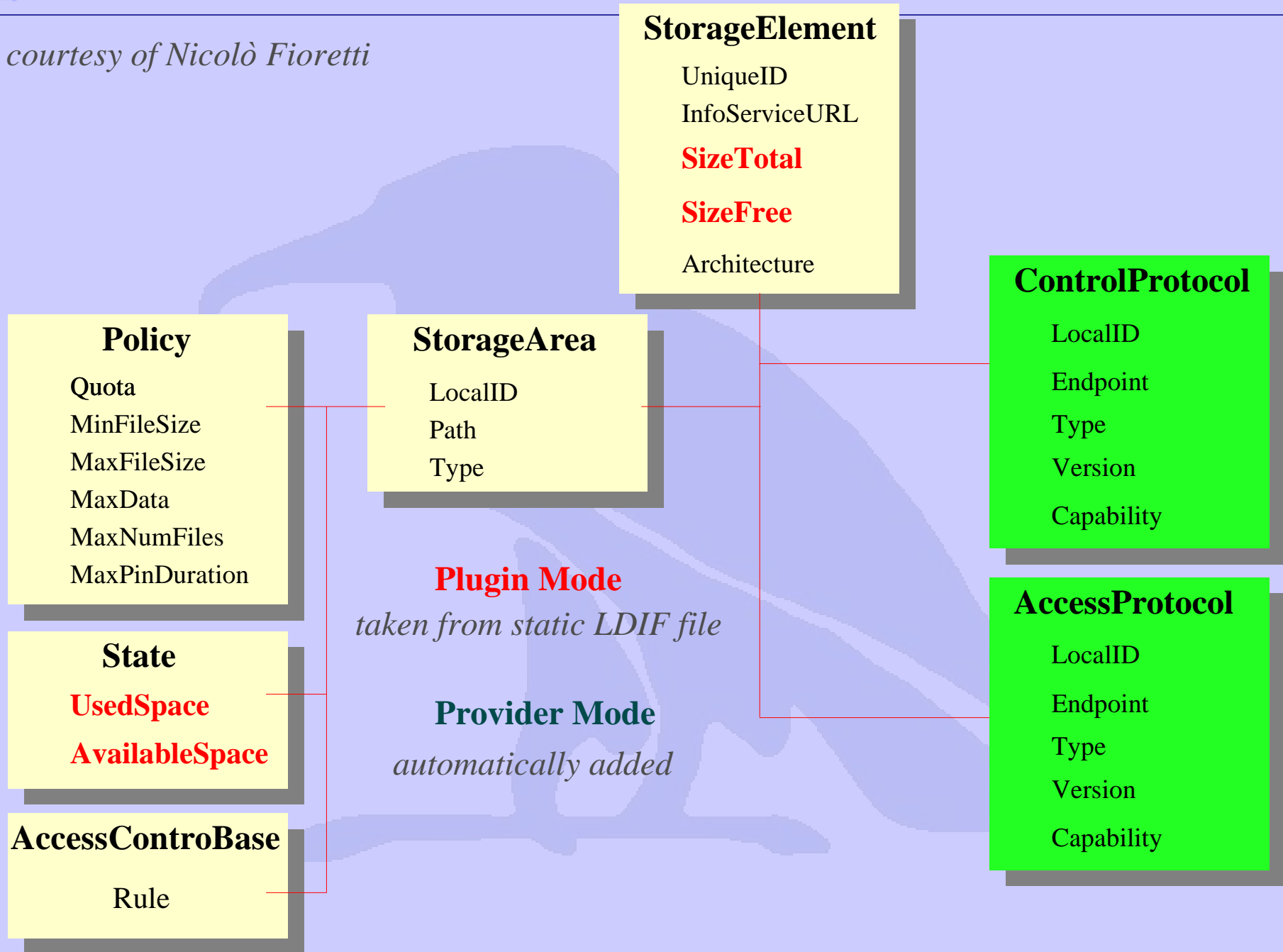
Poster : A computational and data scheduling architecture for HEP applications  
 by Martin Radicke and Lars Schley

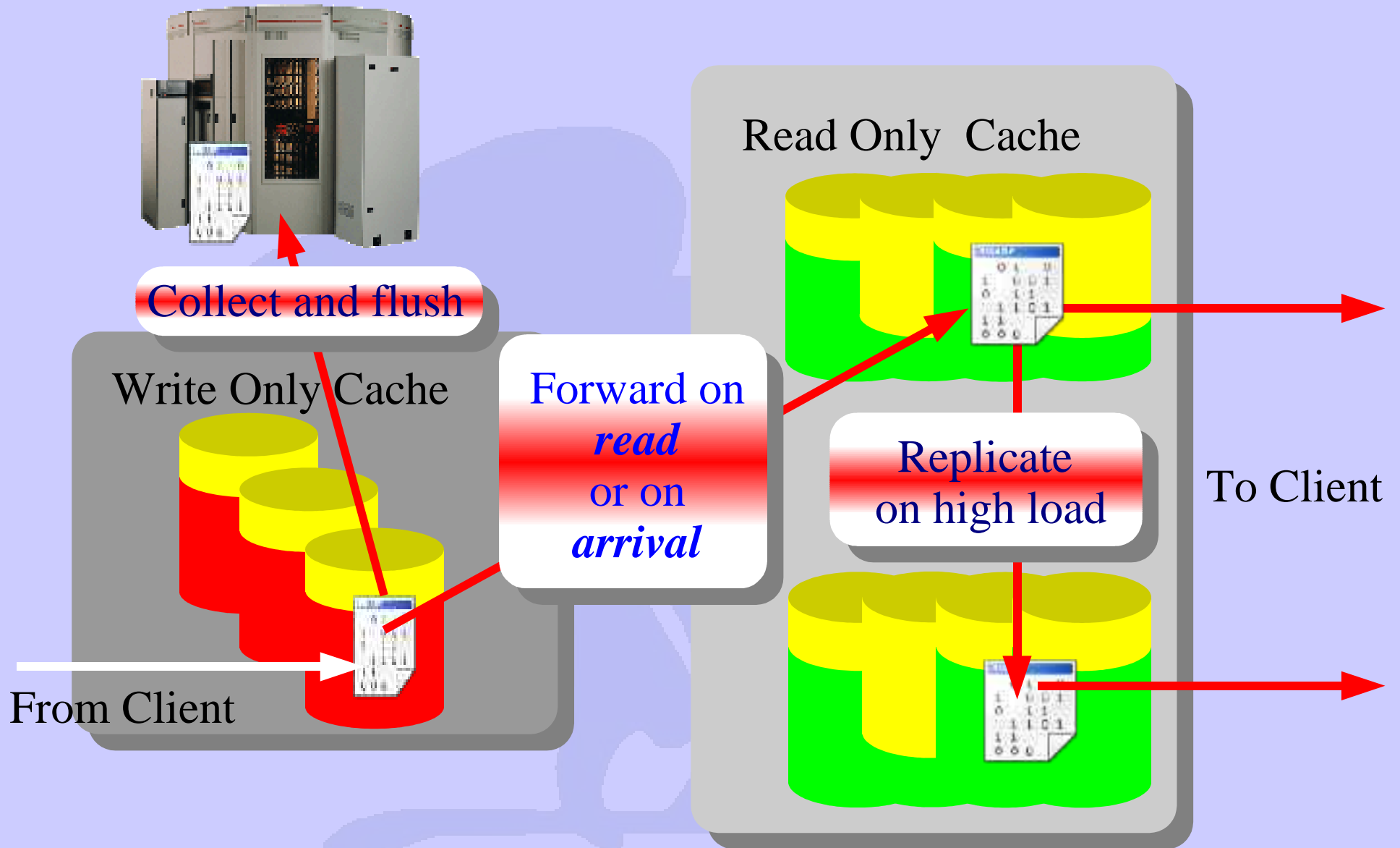






by courtesy of Nicolò Fioretti



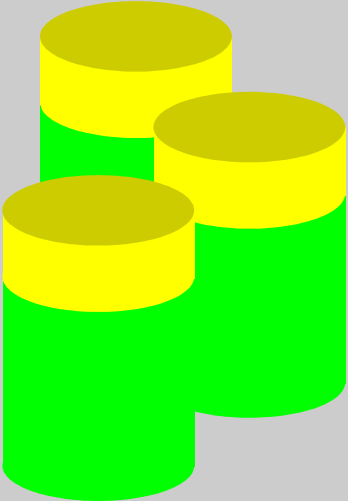


even more dynamic when using API

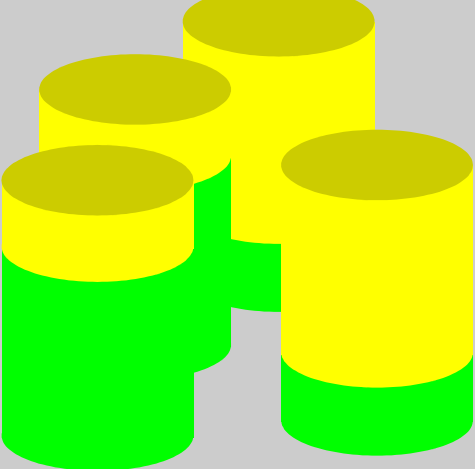




*One dCache instance, but pool-groups with different characteristics*



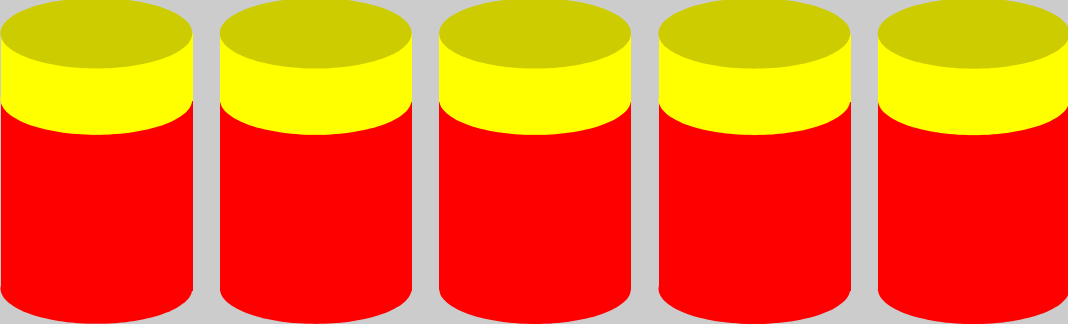
*prefer balanced  
movers*



*prefer filling empty pools  
first*



- *small file set*
- *high throughput*
- *all files on all pools*



***Resilient Pool Group,***  
*file multiplicity centrally  
managed,  
no HSM back end*





## Client Attributes

*Ip Subnet (Host)*

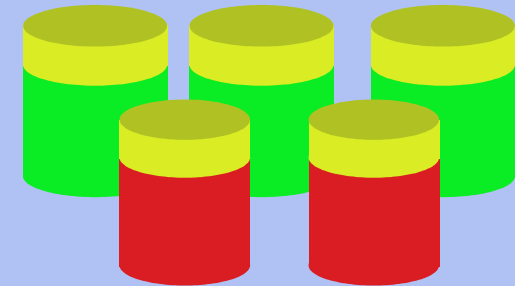
*Storage Class (Directory Tree)*

*Data flow direction (put,get)*

Link



## Set of pools



*Current Link Attributes :*

*readpref, writepref, cachepref*

*Future Link Attributes :*

*p2p-pref*

*Thresholds : Idle, p2p, ...*

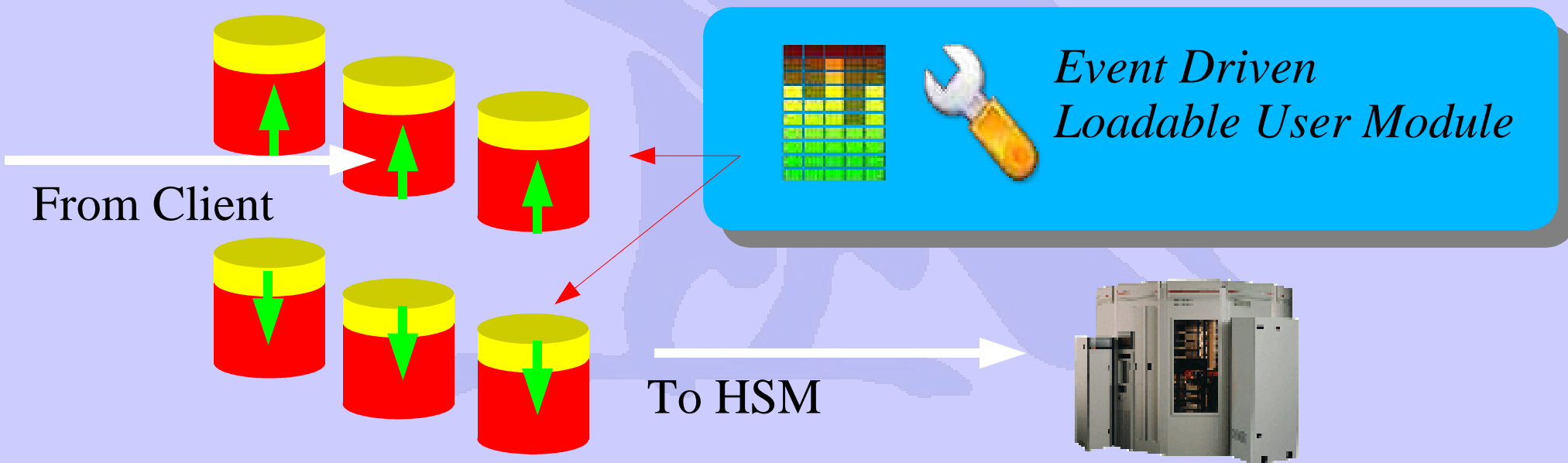
*cpu-factor and space-factor*





*Centrally controlled flushing of precious datasets to back-end tape system(s)*

- determines number of concurrent streams to HSM system.
- never allow client data flowing onto pools, while same pool is flushing.
- never allow pool flushing, while data coming in from clients.
- event driven API for fine grained customization of HSM flush.
- allows to globally limit the number of stream to a storage class (hpss, tsm)





*Incoming GET request*

gsiFtp  
dCap



Pre/Post  
Stage Event  
Handler

Prepare HSM  
for  
transfer

*Select stage pool*

*Do the staging*

File not  
in Cache

Delayed  
Staging  
(Collect  
stage  
requests)

PoolManager

PoolManager

Pool

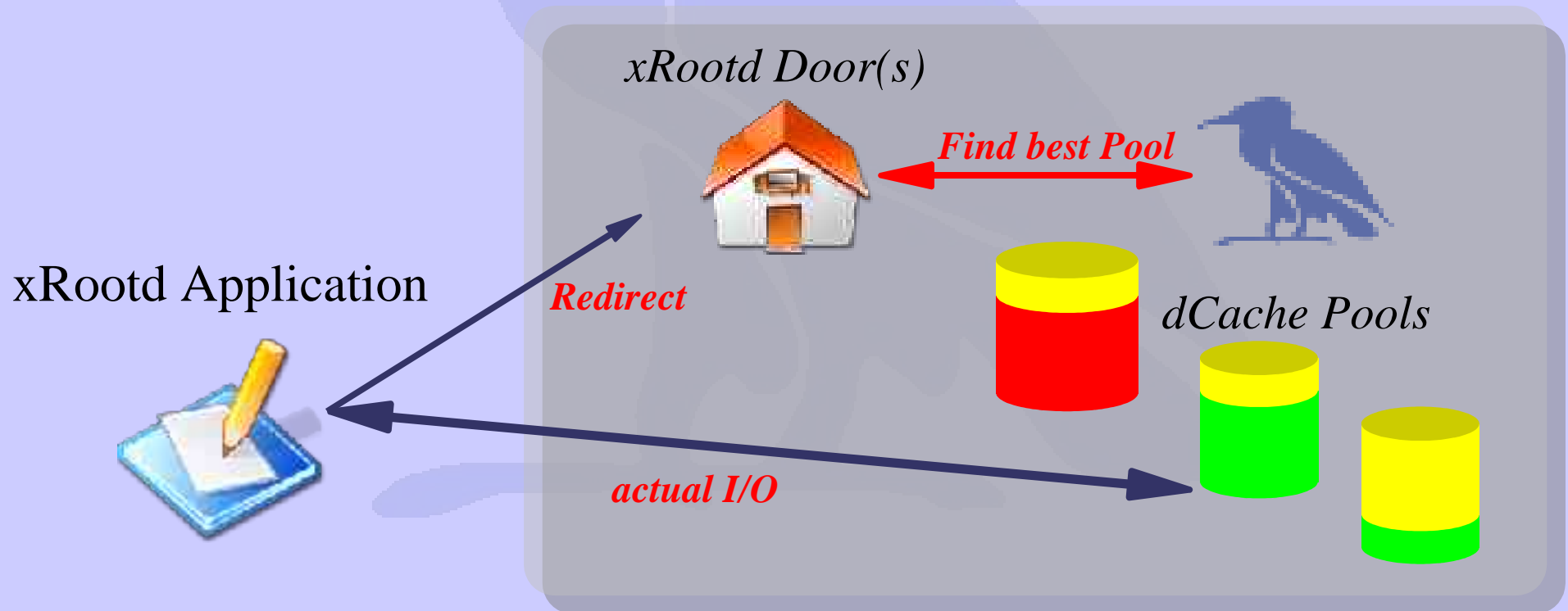
*File In Cache*





*xrootd implemented :*

- *regular I/O ok, including forwarding*
- *no asynchronous mode yet*
- *no security yet*
- *we need help from experts, specs sometimes not clear enough*





# *Performance Track*

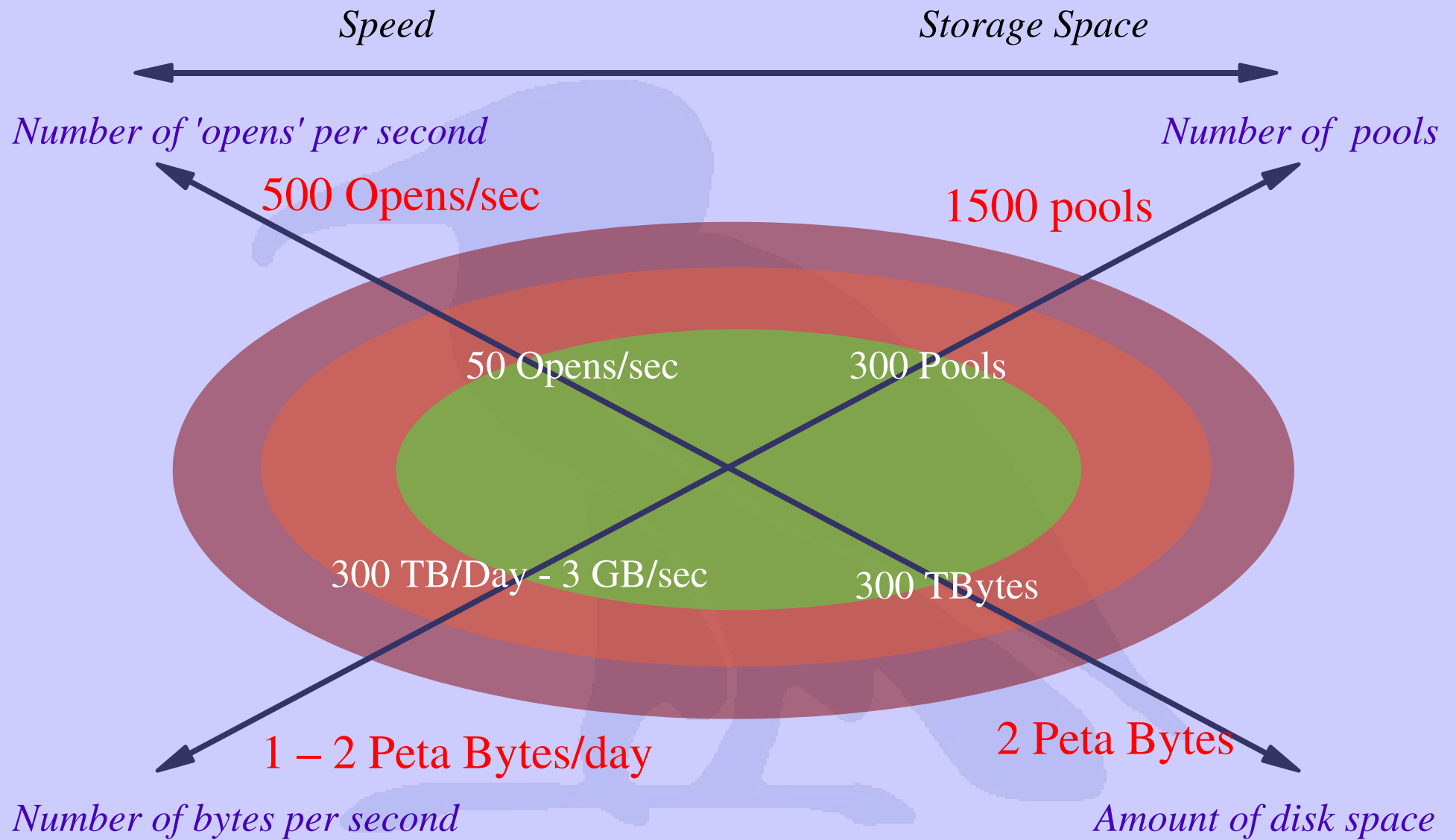
*Multiple I/O queues*

*Multiple Pnfs (file systems)*

*Chimera*









Mover Queue(s)

dCap

gridFtp

xRootd

Application

dccp

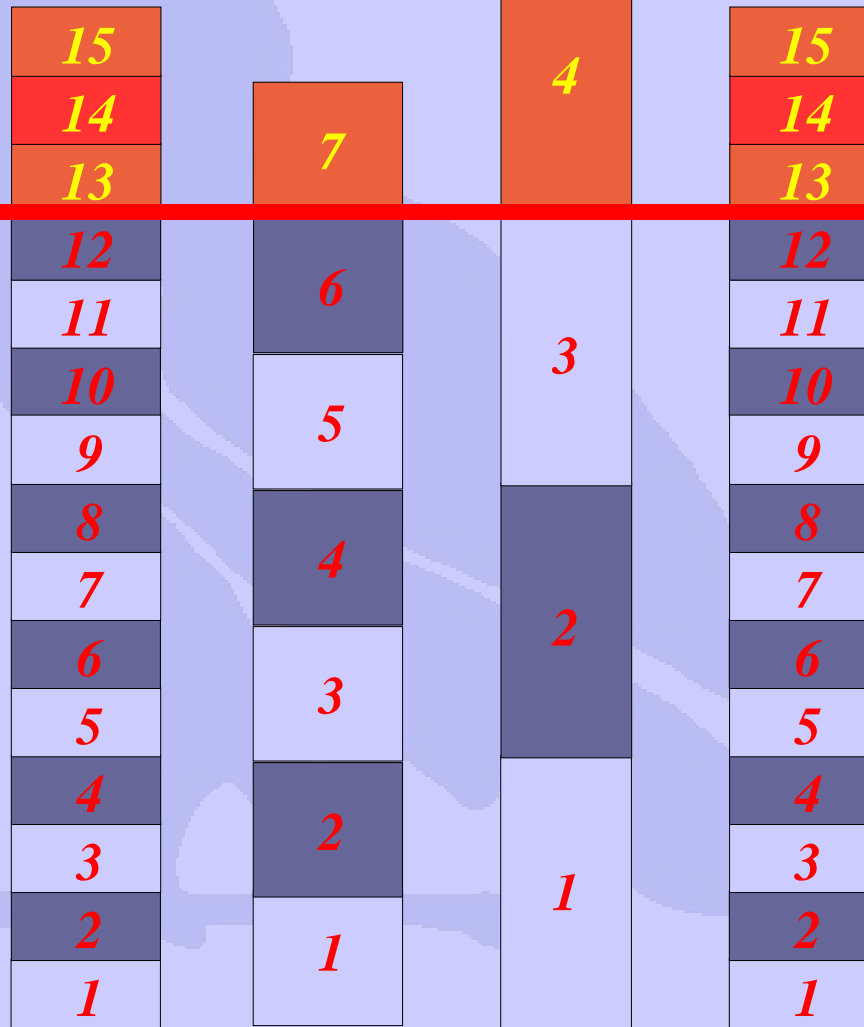
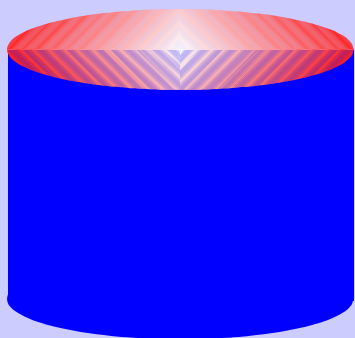
Application

Cost = 1

Waiting

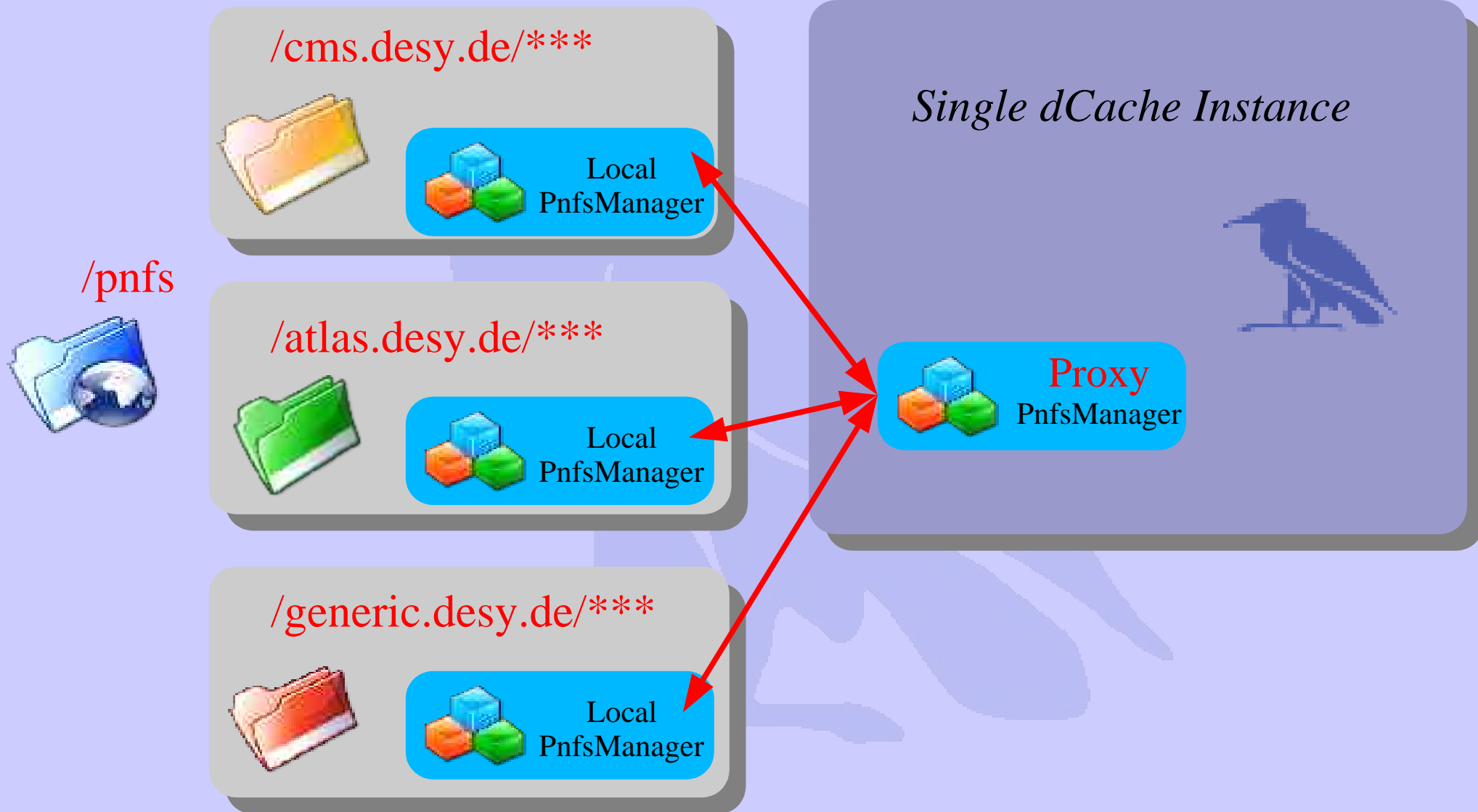
Active

Pool



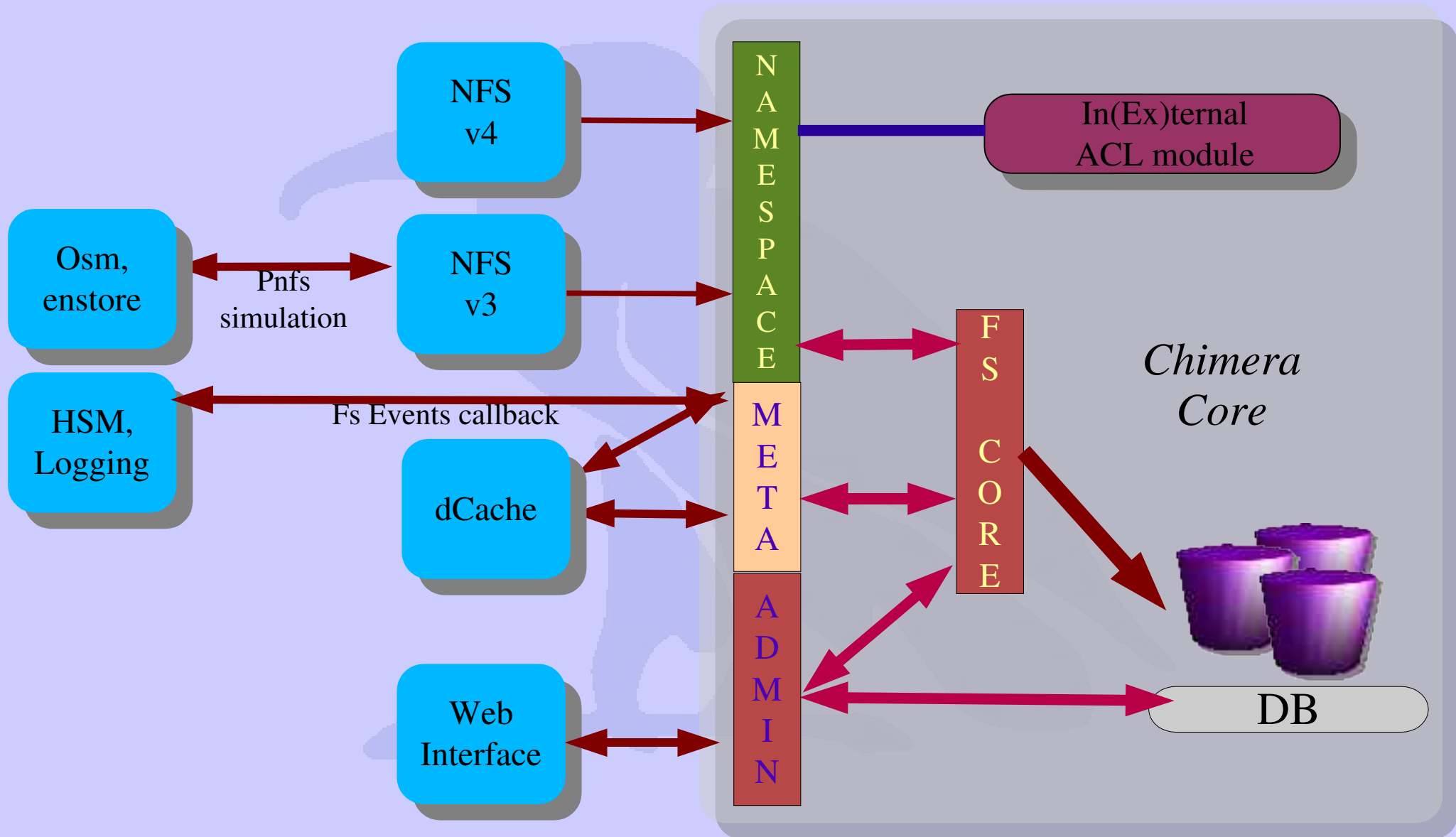


### Separate Pnfs Server





by courtesy of Tigran Mkrtchyan





dCache, the Book

*www.dCache.ORG*

need specific help for you installation or help  
in designing your dCache instance.

*support@dCache.ORG*

dCache user forum

*user-forum@dCache.ORG*

