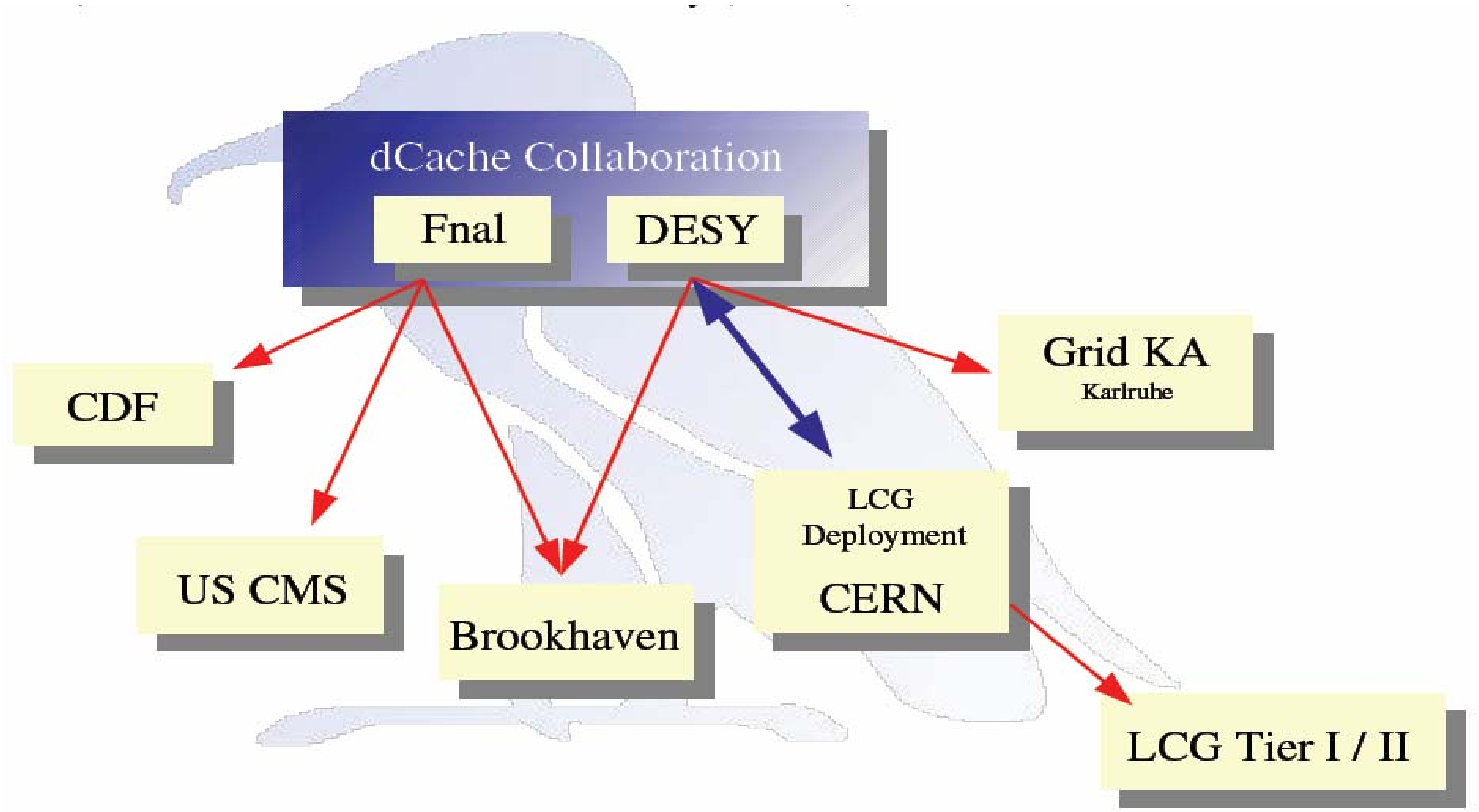
dCache Status and Plans – Proposals for SC3

Michael Ernst

For the dCache Team

LCG Data Management Workshop

dCache is a joint effort between DESY and Fermilab



Agenda

- Architecture and Components
- dCache Functionality Layers and Basic Design
- Data Access Methods
- HSM Interface
- Pool Selection Mechanism
- SRM/dCache as LCG SE
- Installation & Management
- dCache Support Model
- Plans
  - Extensions to Information Provider to support Job & Data Co-Scheduling
  - "The Mirror Cache"

dCache – The Architecture and Components

- Name Space uniquely represented within single file system tree
  - Strictly separates filename space of data repository from physical location
  - File Namespace in DB & accessible to application by NFS, SRM, GridFTP etc.
  - Replicas of given file may exist on multiple storage nodes and MSS for e.g. load balancing (pool-to-pool transfers)
- Scalable Architecture
  - Fully distributed Architecture w/ Autodiscovery for Components
  - Integrates Heterogeneous Disk Storage and Server Technology with multiple hundred individual (commodity) nodes
  - Automatic load balancing by cost metric and inter pool transfers
  - Multiple Distributed Data Access Points (pluggable Door/Mover pairs) supporting different Standards for Data Access
  - dCache distributes files autonomously across Disk Servers
    - Selection depends on available space and server load
  - Fine-grained steering directives to control data flow and utilization of Storage Resources (Pool Attraction)

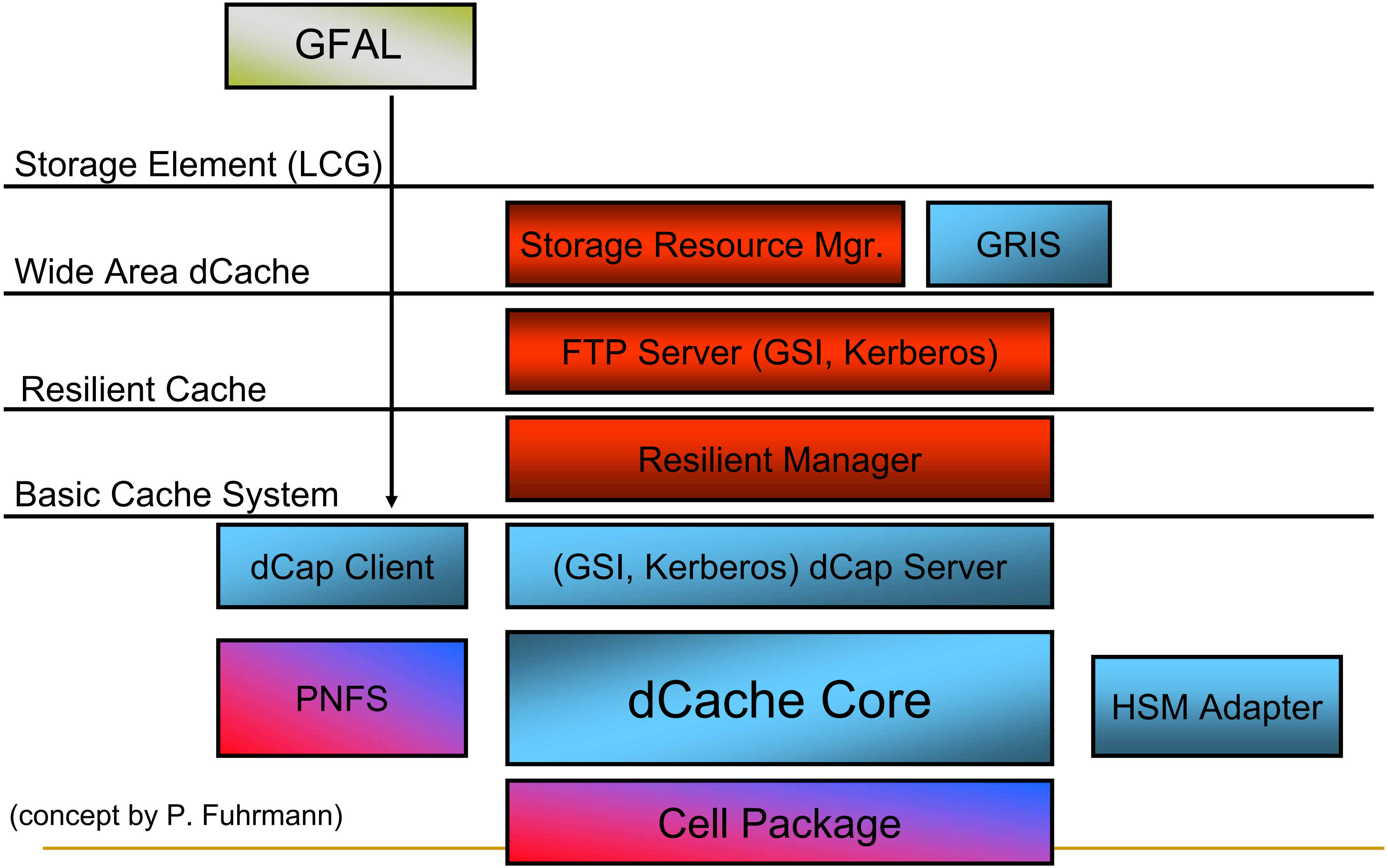
dCache – The Architecture and Components

- Scalable Architecture (cont'd.)
  - Configurable as Disk-only and as part of a Storage Hierarchy (e.g. with Tape Back-end)
  - Automatic Migration and Staging between dCache and underlying Storage System
  - Very Flexible and Modular MSS interface - Integration done for Enstore, TSM, OSM and HPSS
  - Resiliency Manager allows to automatically create and maintain a configurable number of copies of a given file on different storage nodes – eases maintenance (adjusts replica count on scheduled pool maintenance) and improves availability in case of pool failures
- Management
  - Rich set of Admin commands coming with individual dCache Modules (Cells)
  - CLI (via SSH) and GUI (Web interface) allow to navigate through and login to the Modules (e.g. SRM, GridFTP door, PoolManager, Storage Pools)
    - Allow to add/remove components to/from the active system
    - To customize/tune the system while in operation

Resilient dCache

- Developed by Fermilab
- Goal is managed reliable storage without a tape backend
- Reliability is achieved through replication
- Expect pools to go in/out of service and files are replicated as this happens. Can also schedule pools offline to smooth replication process
- Active replica checksum comparison with replacement when necessary
- User web interface

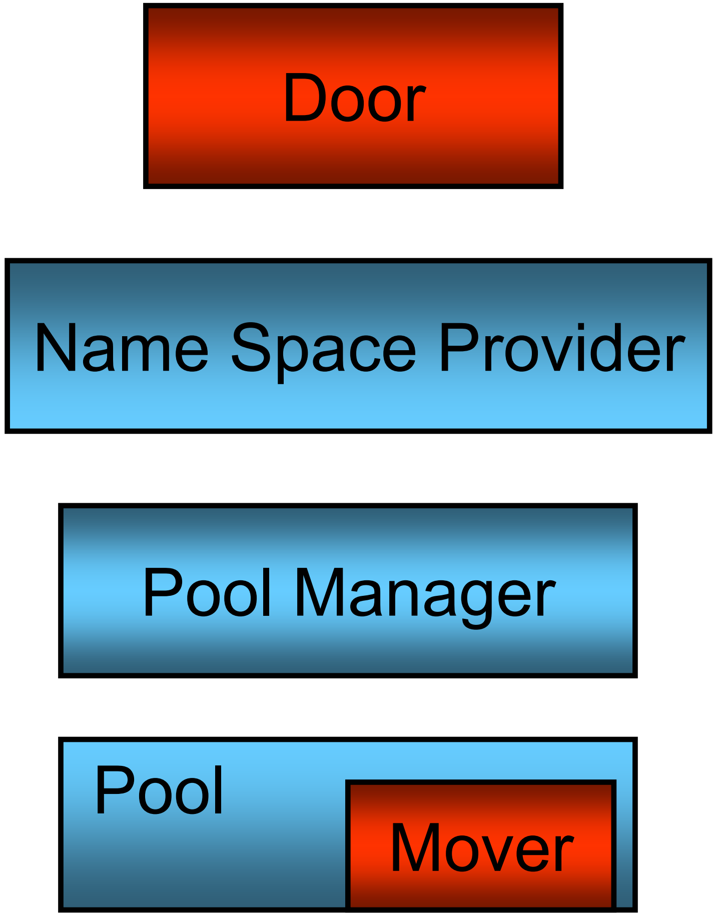
dCache Functionality Layers



(concept by P. Fuhrmann)

dCache Basic Design

Components involved in Data Storage and Data Access



Door

- Provides specific end point for client connection
- Exists as long as client process is alive
- Client’s proxy used within dCache

Name Space Provider

Interface to a file system name space

- Maps dCache name space operations to filesystem operations
- Stores extended file metadata

Pool Manager

Performs pool selection

Pool

Mover

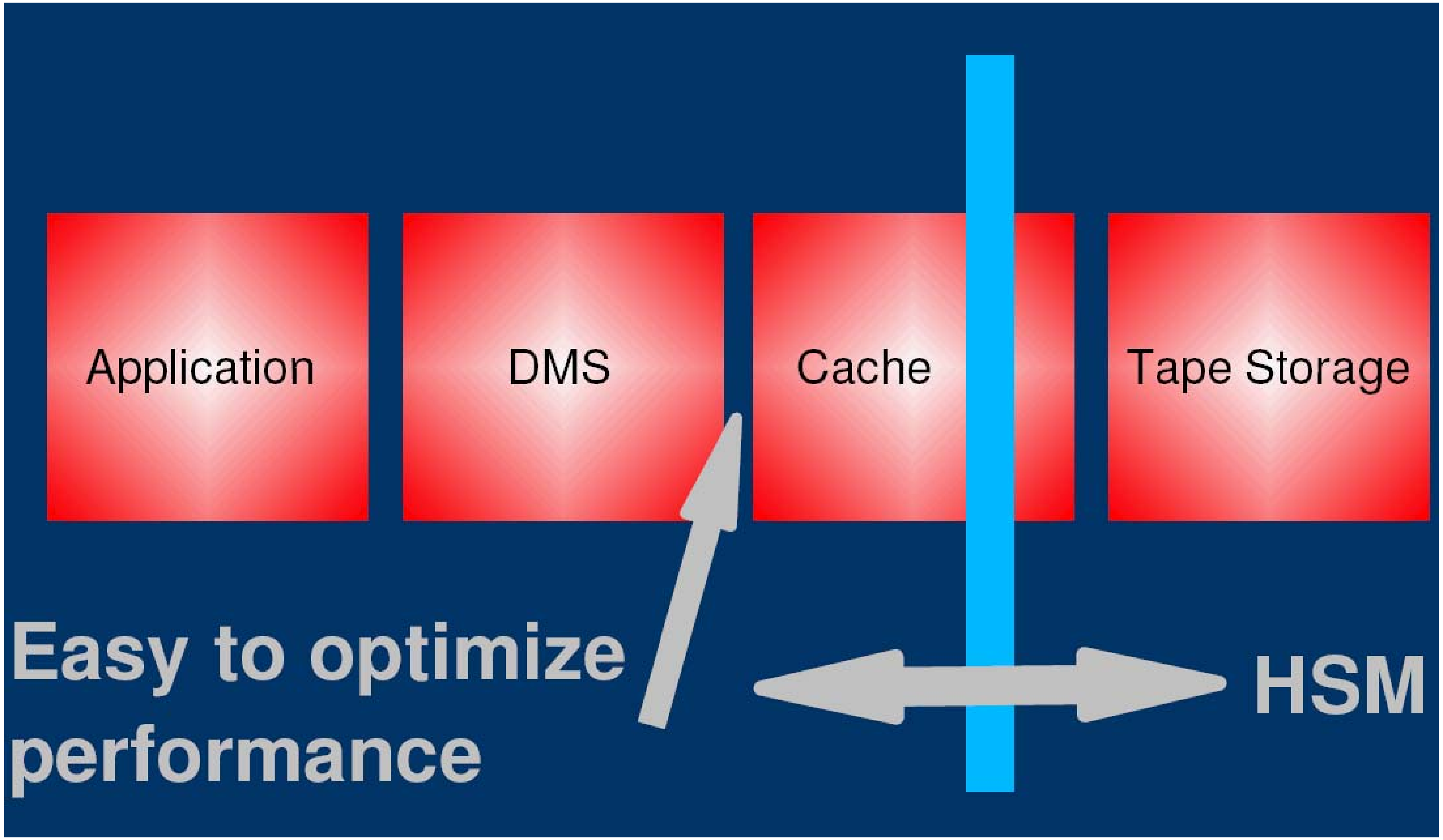
- Data repository handler
- Launches requested data transfer protocols
- Data transfer handler (gsi)dCap, (Grid)FTP, http, HSM hooks

(concept by P. Fuhrmann)

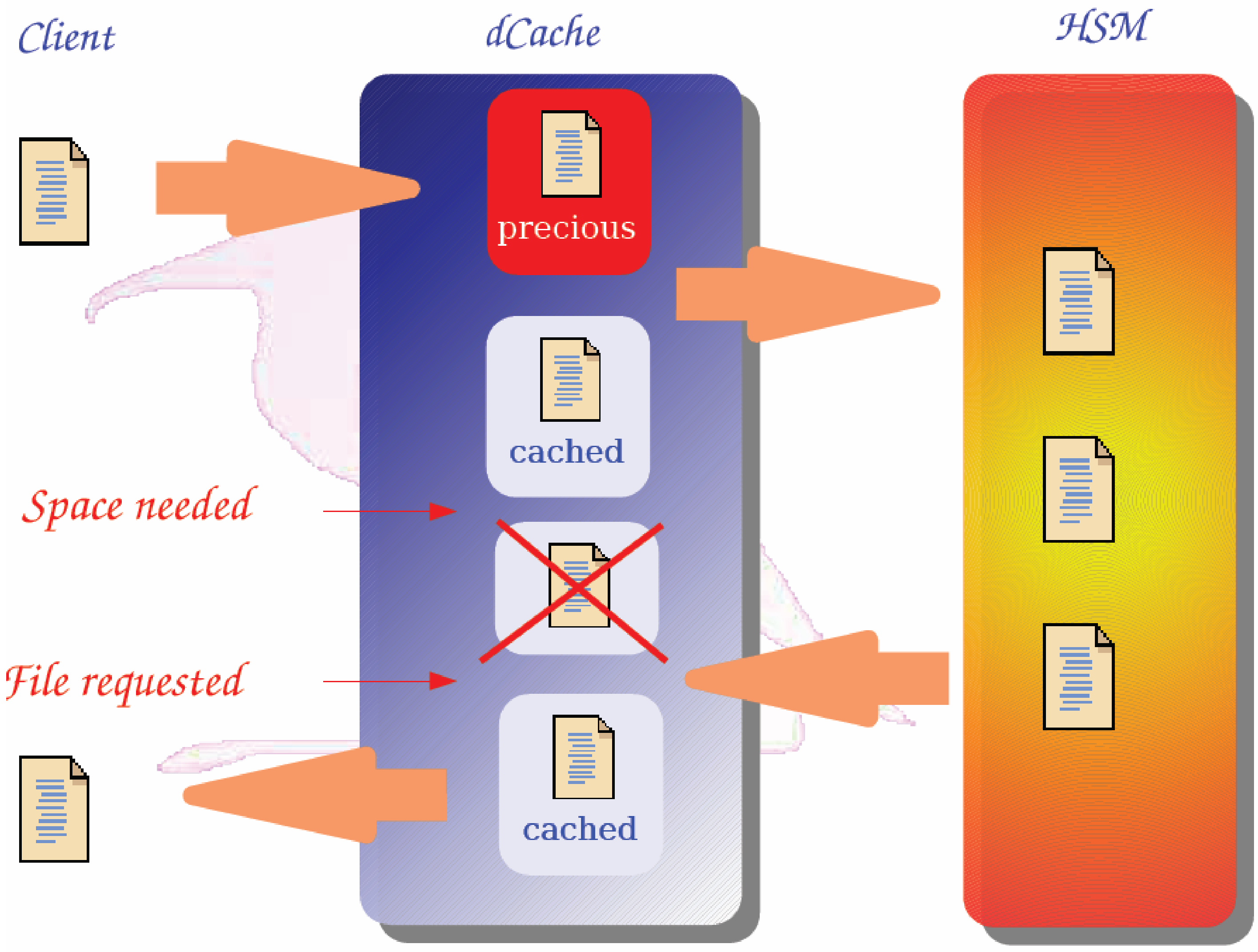
Data Access Methods

- dCap – the native protocol
  - Library for regular file access via POSIX calls
    - Can be linked against applications and available as preload lib
  - Supports pluggable security mechanisms
    - Implemented are GssApi (Kerberos, GSI) and ssl
  - Adds resiliency by protecting clients from intermediate network and storage node failures (lib reconnects)
  - Provides URL style addressing (alternative to mounting the pnfs namespace provider)
- FTP – support for multiple dialects
  - GridFTP with GSI security
  - GssFTP with Kerberos security
- Other Protocols can be easily integrated through well defined, well documented Interface

Where starts your HSM ?



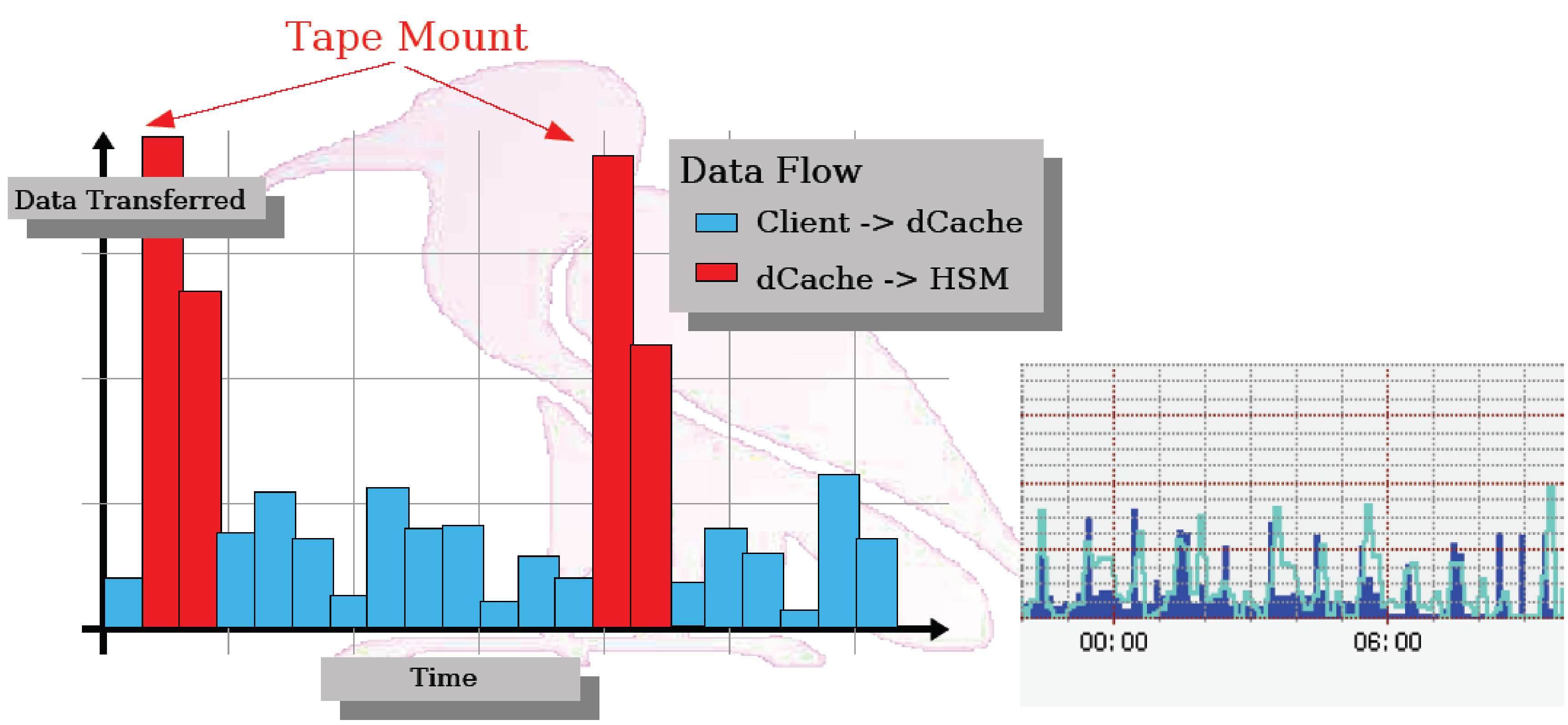
HSM Interface



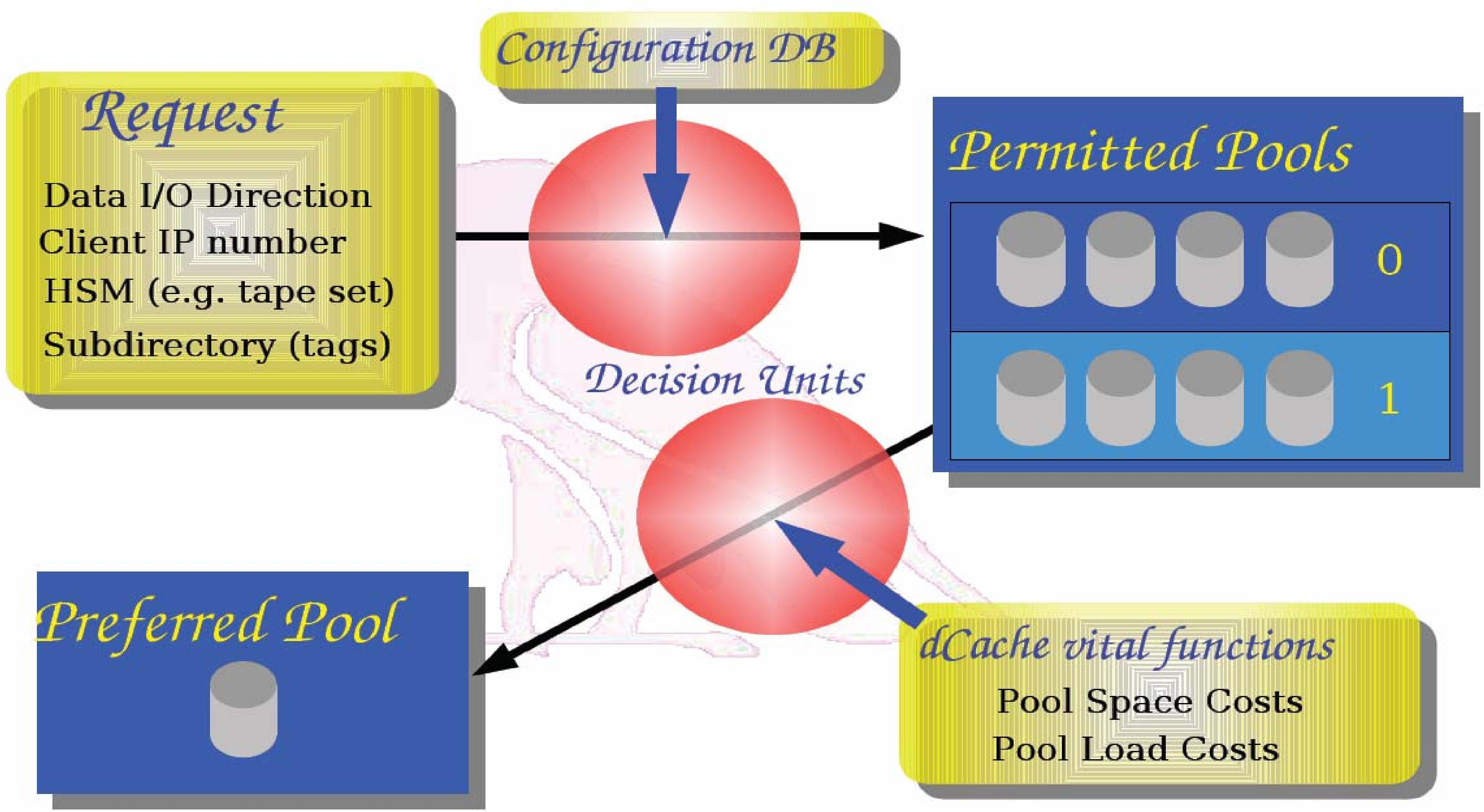
HSM Interface

- Very flexible Interface to various MSSs (scripts)
- Precious data is separately collected per storage class
- Each storage class queue has individual steering parameters for HSM flush operation
  - Elapsed time a file is allowed to be precious per storage class
  - Total data volume that is precious per storage class
  - Maximum number of precious files per storage class
- Maximum number of concurrent HSM flush operations is configurable
- Multiple HSM instances and HSM classes are simultaneously supported

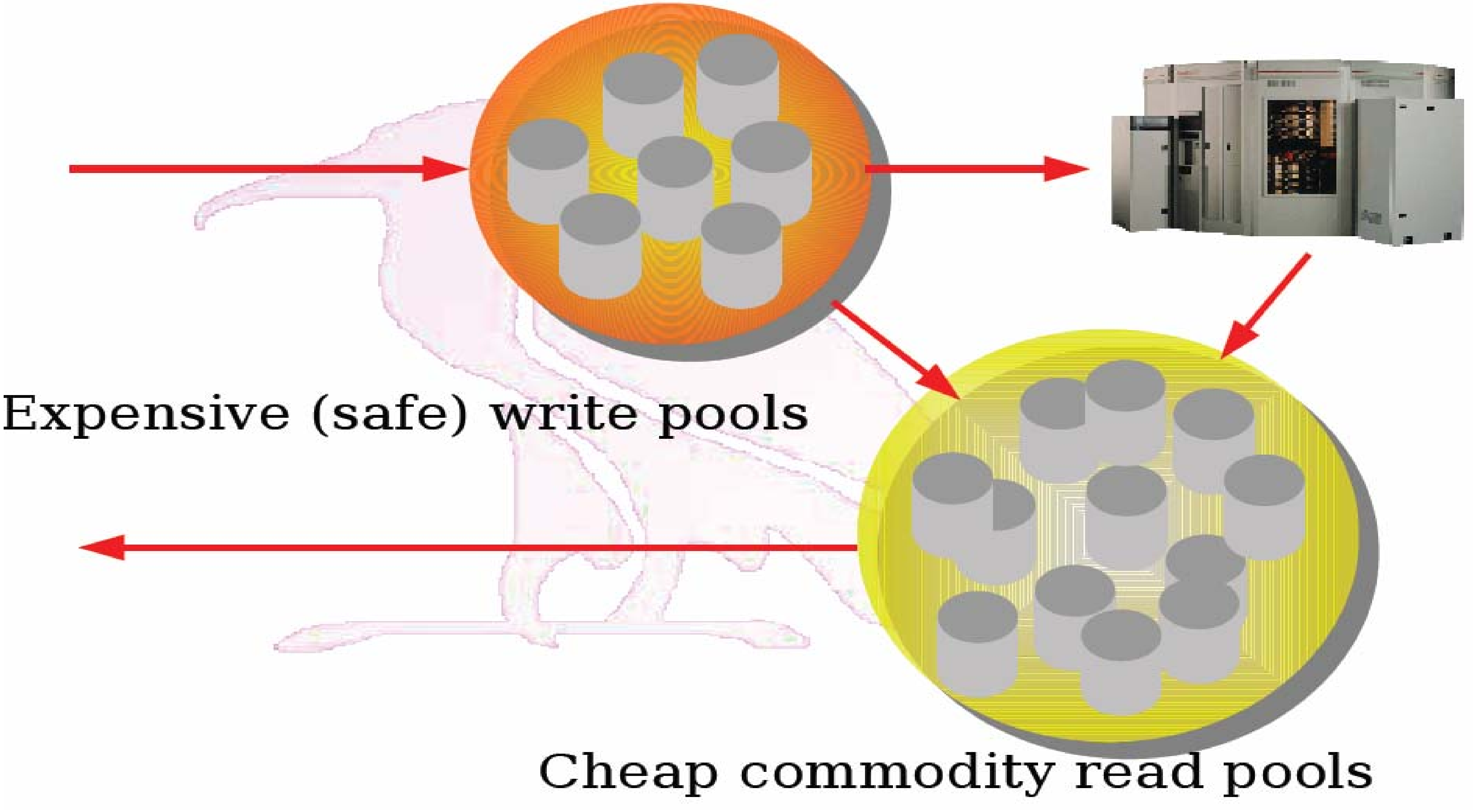
Deferred HSM flush



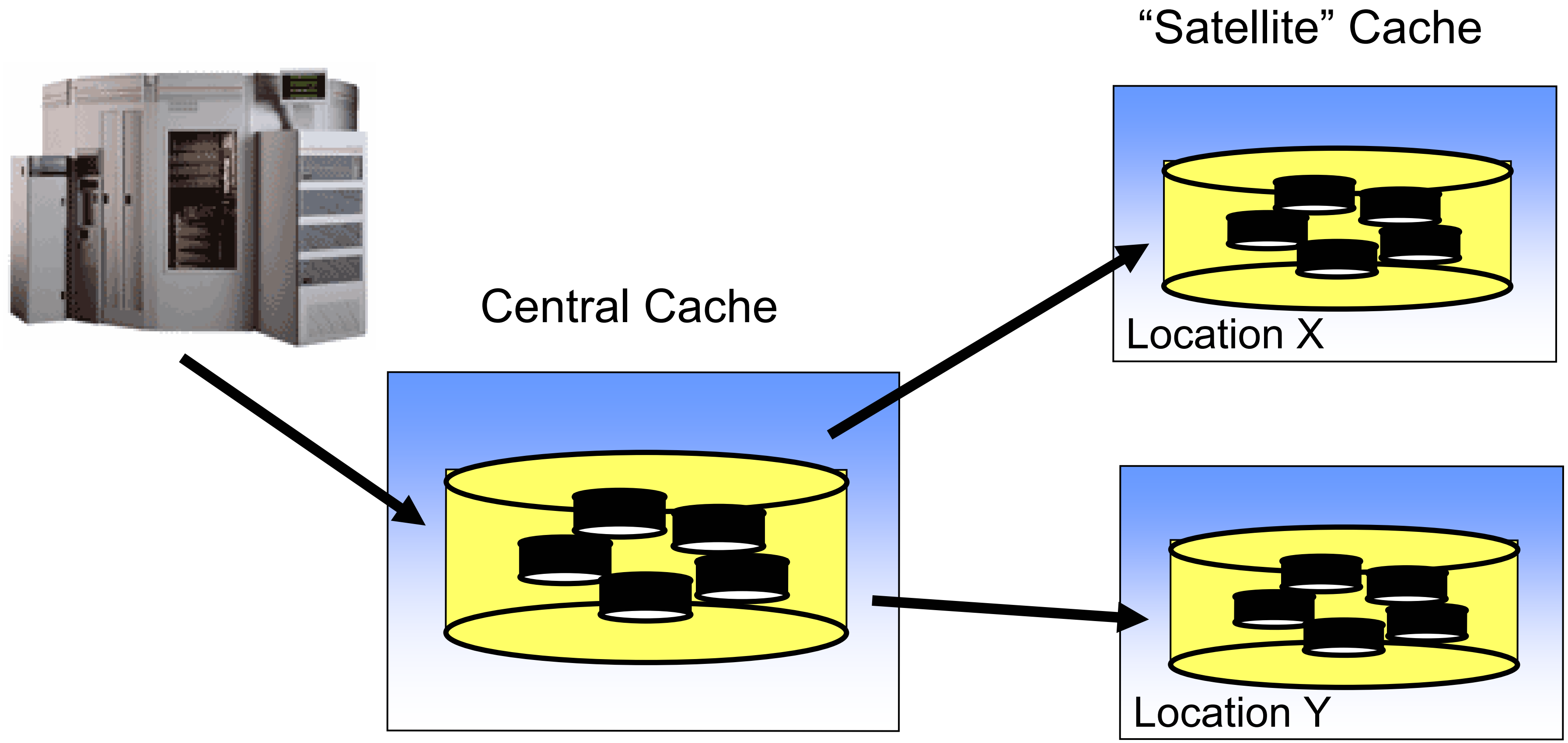
Pool Selection Mechanism



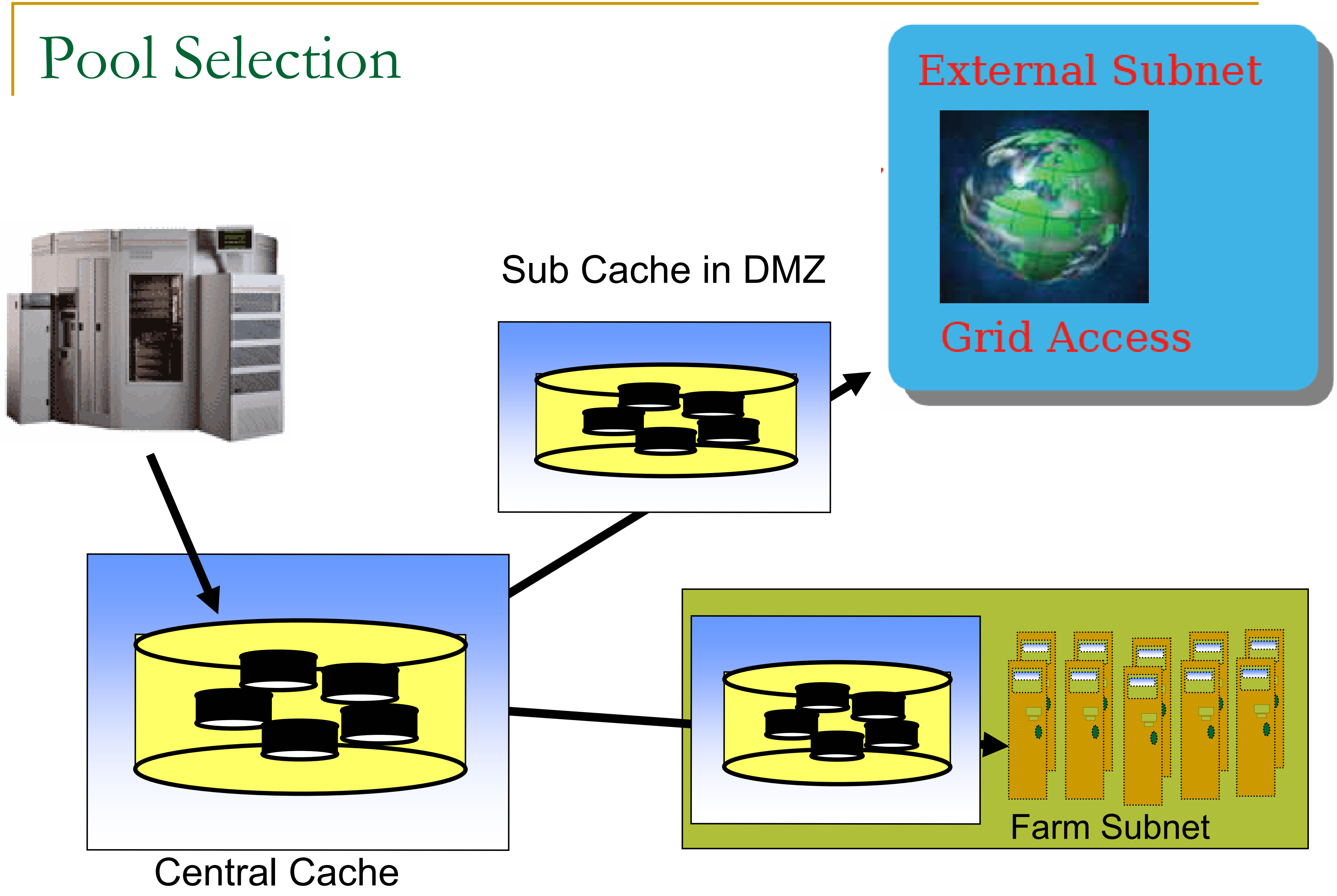
Pool Selection



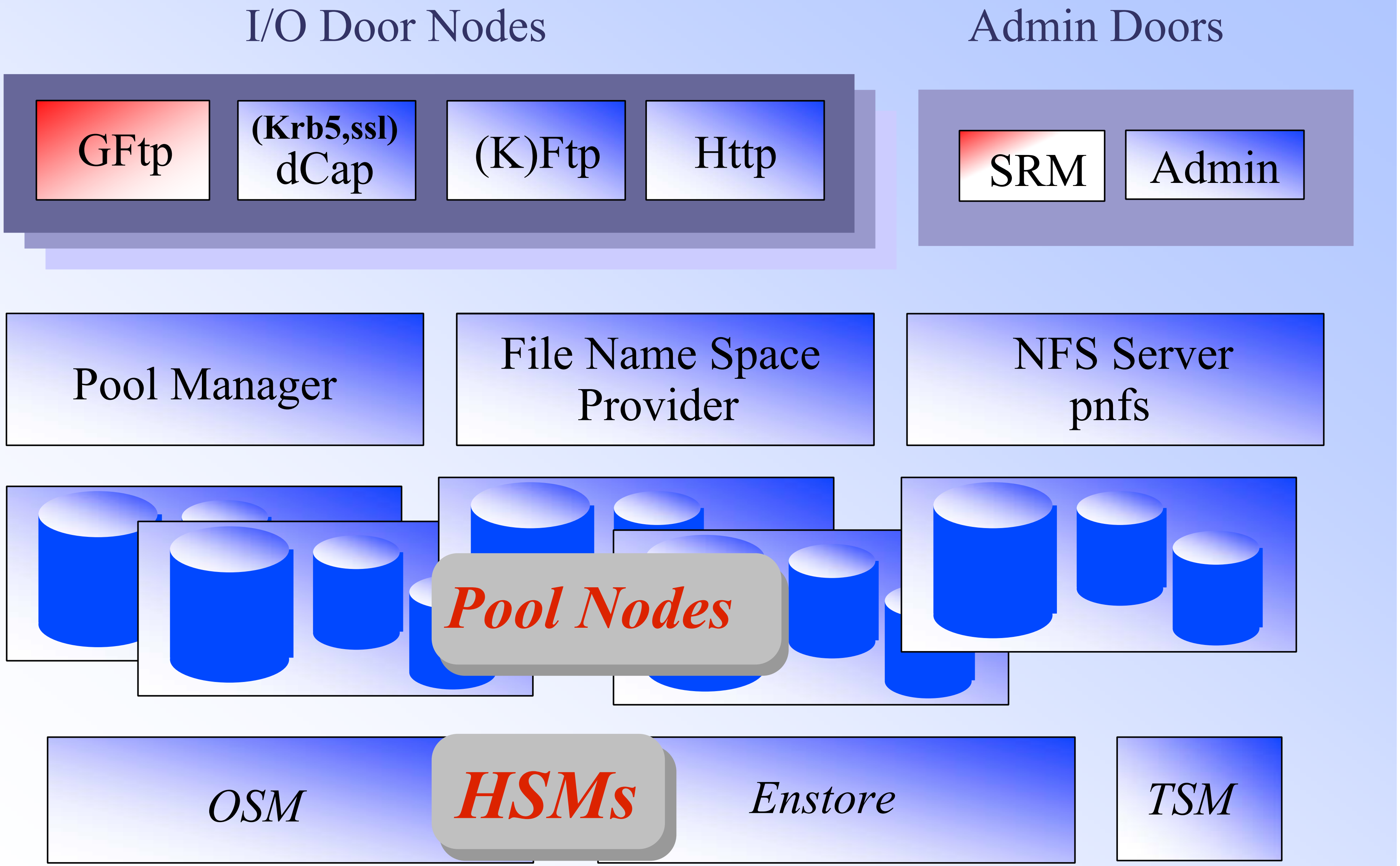
Pool Selection



Pool Selection



dCache Components



I/O Door Nodes

GFtp

(Krb5,ssl) dCap

(K)Ftp

Http

Admin Doors

SRM

Admin

Pool Manager

File Name Space Provider

NFS Server pnfs

Pool Nodes

HSMs

OSM

Enstore

TSM

Data Grid Functionality – dCache as LCG SE

- Aiming at Seamless Integration with LCG Infrastructure
  - SRM related aspects covered in Timur's talk
  - Information Provider (GRIS - currently fairly detached, working on full integration w/ dCache system)
  - Full compliance with LCG demanded functionalities & supplied Client Utilities, and other SE Implementations
    - Primarily concerns SRM, GridFTP, dCap (GFAL)
  - Working with the LCG GDT on improving packaging & documentation, installation procedure and manageability to make dCache a suitable solution for any kind of LHC relevant Computing Facility

Installation & Management

- Few RPMs (<= 4) to install
  - PNFS (Namespace Manager)
  - dCache Core (for Admin, Pool, Door node)
  - dCache Optional S/W (SRM, GridFTP, gsidcap)
    - SRM needs PostgreSQL DB
  - dCache Client
- Central Configuration provides suitable defaults & needs only minor customization
- Auto Configuration to add & remove pools
  - Disk pools/nodes request to add themselves to dCache
  - Allows simple reconfiguration of the Disk Pools
  - Administrator can temporarily remove pools from dCache if a disk has crashed and is being repaired
  - PoolManager takes out a Pool automatically in case it doesn't respond any longer
- Medium Tier-2 site w/o tape backend and ~20 pool nodes needs approx. 2h to install dCache from scratch
  - Upgrade is usually much faster

Installable Server Components

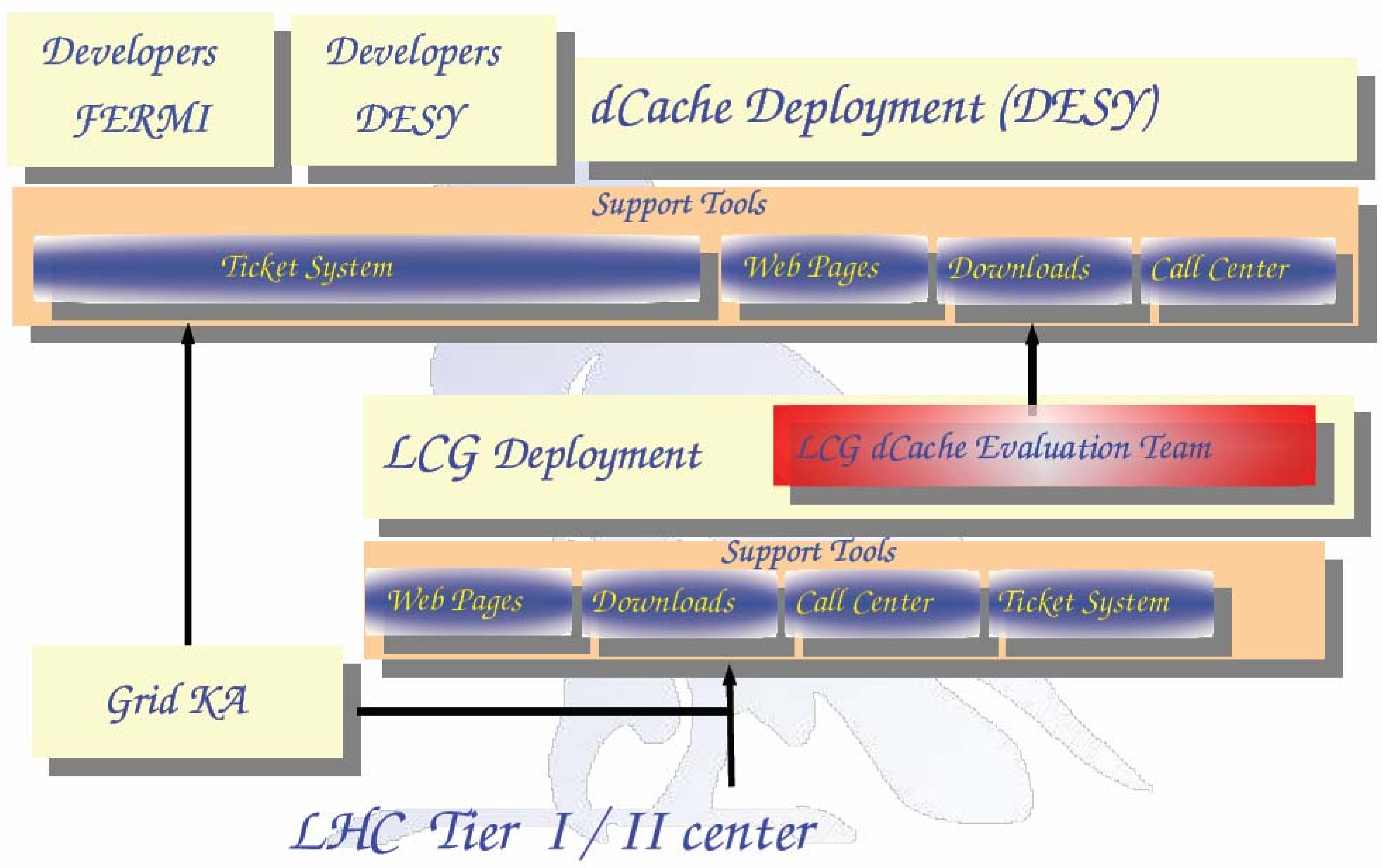
- PNFS Name Space Manager
  - Choice of two Databases
    - GDBM (default)
    - Postgres (not part of the dCache distribution)
- Admin Node
  - Provides all central Management Components
    - LocationManager, PoolManager, httpd,
  - Can have Storage Pools (small configurations)
  - Can have Access Points for a variety of Management (SRM) and Data Transfer Protocols (automatically inserted/removed)
- Pool Node
  - Provides Storage Space managed by dCache
  - Automatically inserted/removed in default configuration
  - Can have Access Points (automatically inserted/removed)
- Door Node
  - Provides Access Points for SRM and a variety of Data Transfer Protocols

dCache Licensing

Note: The following applies to dCache only. The SRM as it is distributed by Fermilab has different terms

- Current Conditions (work in progress)
  - Free, non-exclusive license to use dCache w/o limitations
  - The source code is available to selected partners (to be approved by dCache Collaboration)

LCG - dCache Support Model



Developers FERMI

Developers DESY

dCache Deployment (DESY)

Support Tools

Ticket System

Web Pages

Downloads

Call Center

LCG Deployment

LCG dCache Evaluation Team

Support Tools

Web Pages

Downloads

Call Center

Ticket System

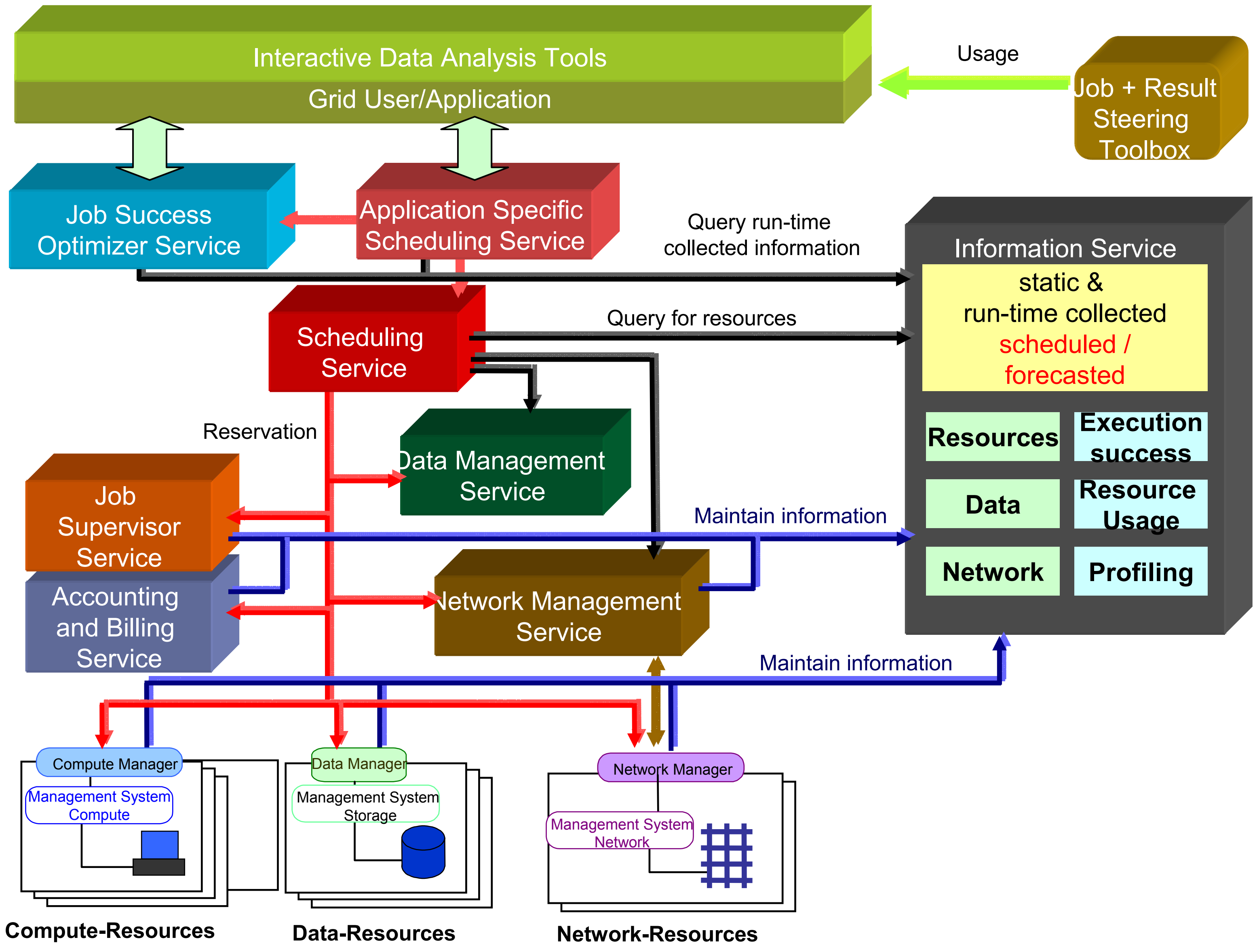
Grid KA

LHC Tier I / II center

Make Storage a seamlessly integrated Resource

Particularly interesting for access to data on "Permanent Storage" (Tape based MSS)

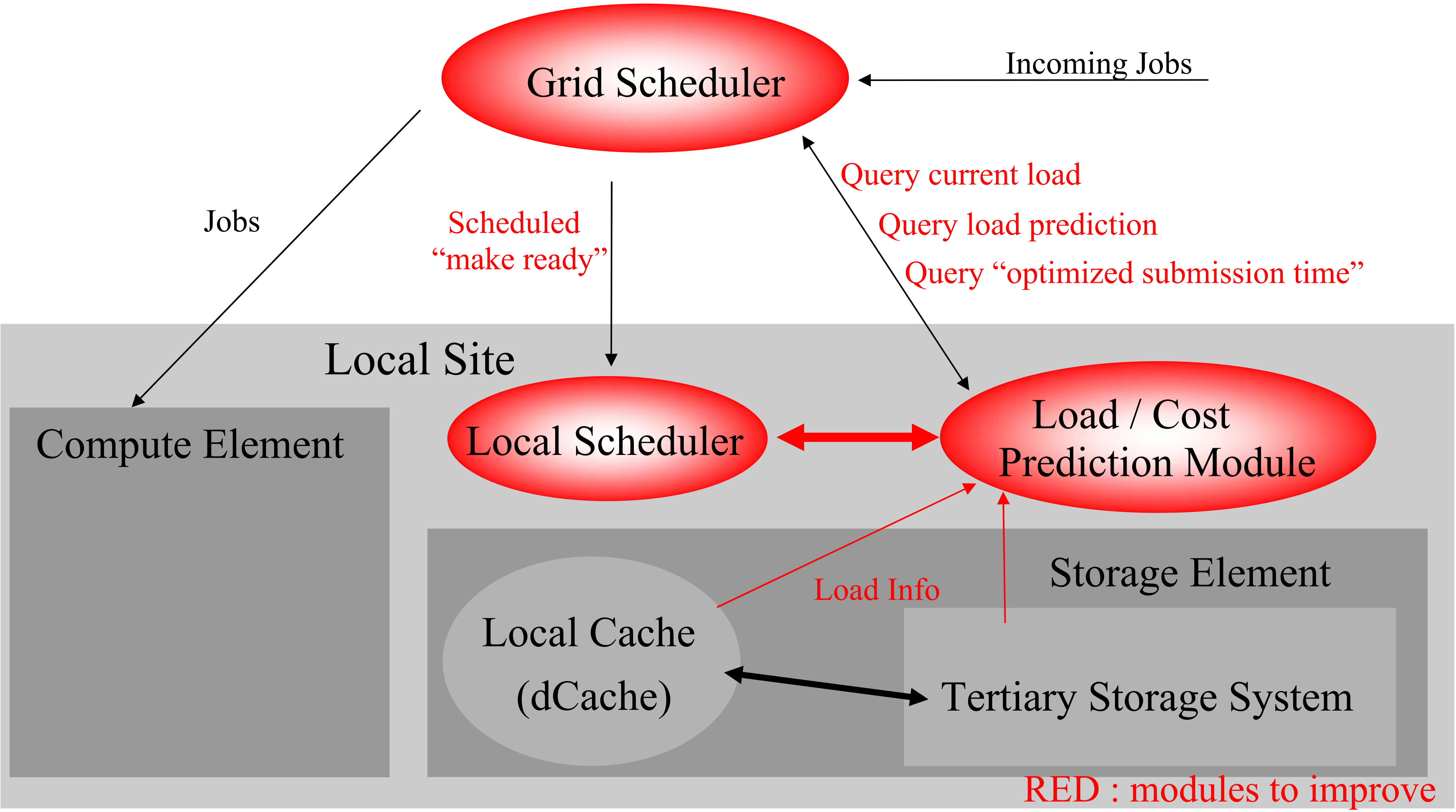
- Access to information about the location of data sets
- Information about access and transfer costs
- Scheduling of data transfers and data availability
  - optimize data transfers w.r.t. storage space, data access costs etc.
- Perfectly fits into general grid scheduling:
  - access to similar services
  - interaction necessary


Interactive Data Analysis Tools
Grid User/Application
Usage
Job + Result Steering Toolbox
Job Success Optimizer Service
Application Specific Scheduling Service
Query run-time collected information
Information Service
static & run-time collected scheduled / forecasted
Scheduling Service
Query for resources
Reservation
Data Management Service
Resources
Execution success
Data
Resource Usage
Network
Profiling
Job Supervisor Service
Maintain information
Accounting and Billing Service
Network Management Service
Maintain information
Compute Manager
Management System Compute
Data Manager
Management System Storage
Network Manager
Management System Network
Compute-Resources
Data-Resources
Network-Resources

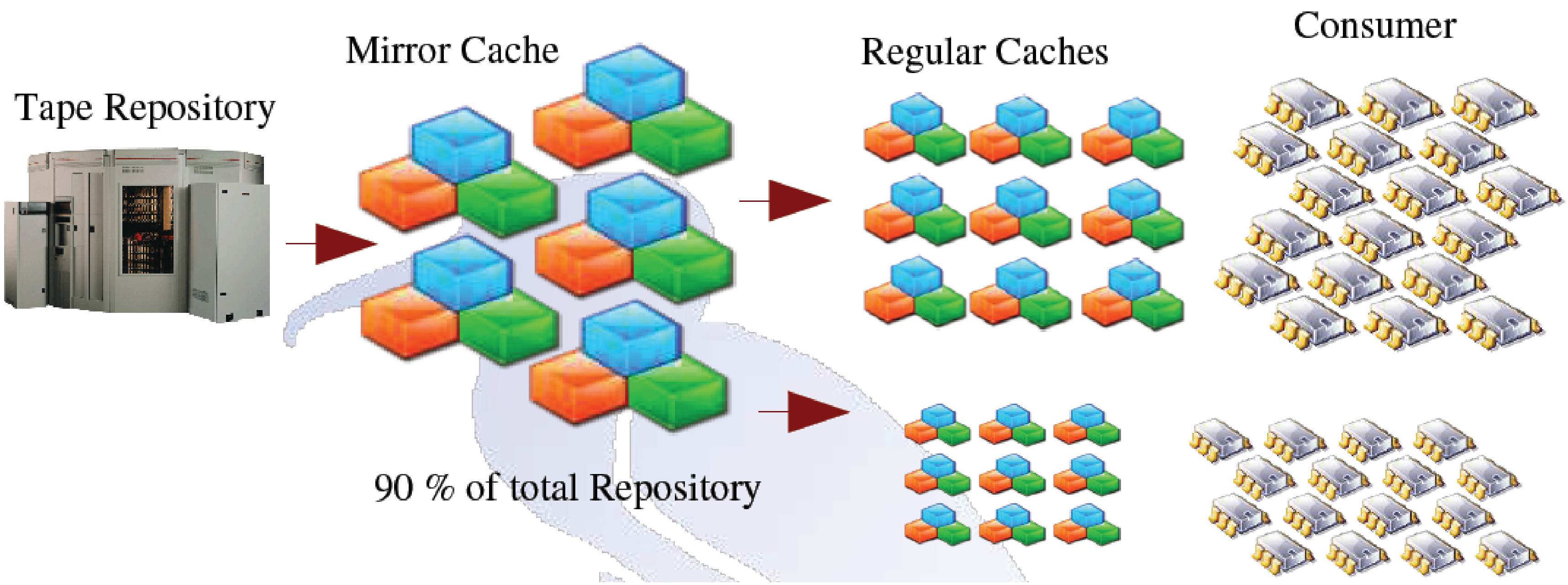
Extensions to SE Information Providers

- A “Cost Prediction Module” calculates load development of the local Storage Element (SE) and makes the information available using an agreed upon schema (i.e. GLUE, additional dynamic records req.)
- By using this Information the local SE can predict the time required to make requested data sets available, and publish it to the Grid Scheduler
- The local SE can calculate a point in time a collection of data sets can be optimally made available
- This approach allows to only schedule jobs to the Compute Element (CE) when the associated SE has flagged all required data sets online

Extensions to improve Job/Data Co-Scheduling



"The Mirror Cache"



- Almost all Tape Data on Mirror Cache
- Mirror Cache built from low cost Components
- Mirror Cache adds another level in Hierarchy (no cartridge handling penalties)
- Managed number of high performance streams between Mirror & Regular Cache
- Mirror Cache disks spin only if accessed
- Tape is treated as Archive