

dCache

- sync'n share
- Software defined storage (QoS)



INDIGO DataCloud

INDIGO DataCloud

Patrick Fuhrmann

On behalf of the project team



What is this about ?



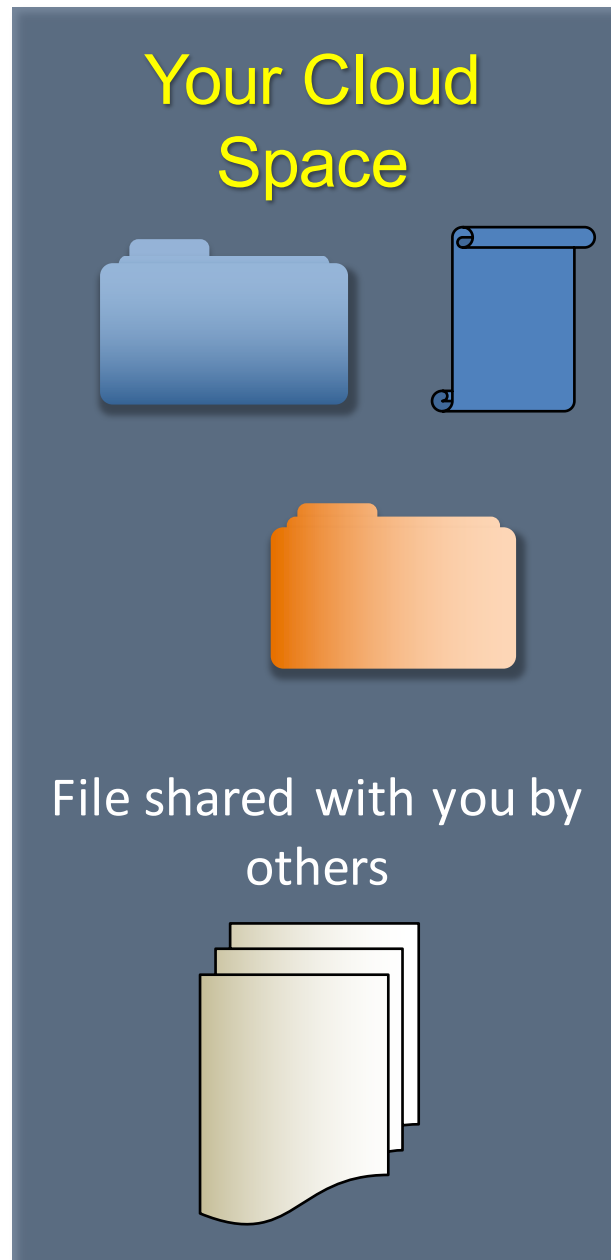
It's about on how modern scientists (people) want to generate , analyse, manage, access and share their data individually or via a jobs, using federated compute and storage infrastructures.

Sharing requirements from DESY users



- Accessing data anytime from everywhere
- Fine grained sharing with individuals and groups.
- Sharing via intuitive Web 2.0 mechanisms (Apps or Browser)
- Sharing with ‘public’ with or w/o password protection
- Sharing of free space (upload)
- Expiration of shares

And the sharing part



Share files/folders with individuals



Share files/folders with 'groups'



Share with 'public' with and w/o password
(Shares can expire)



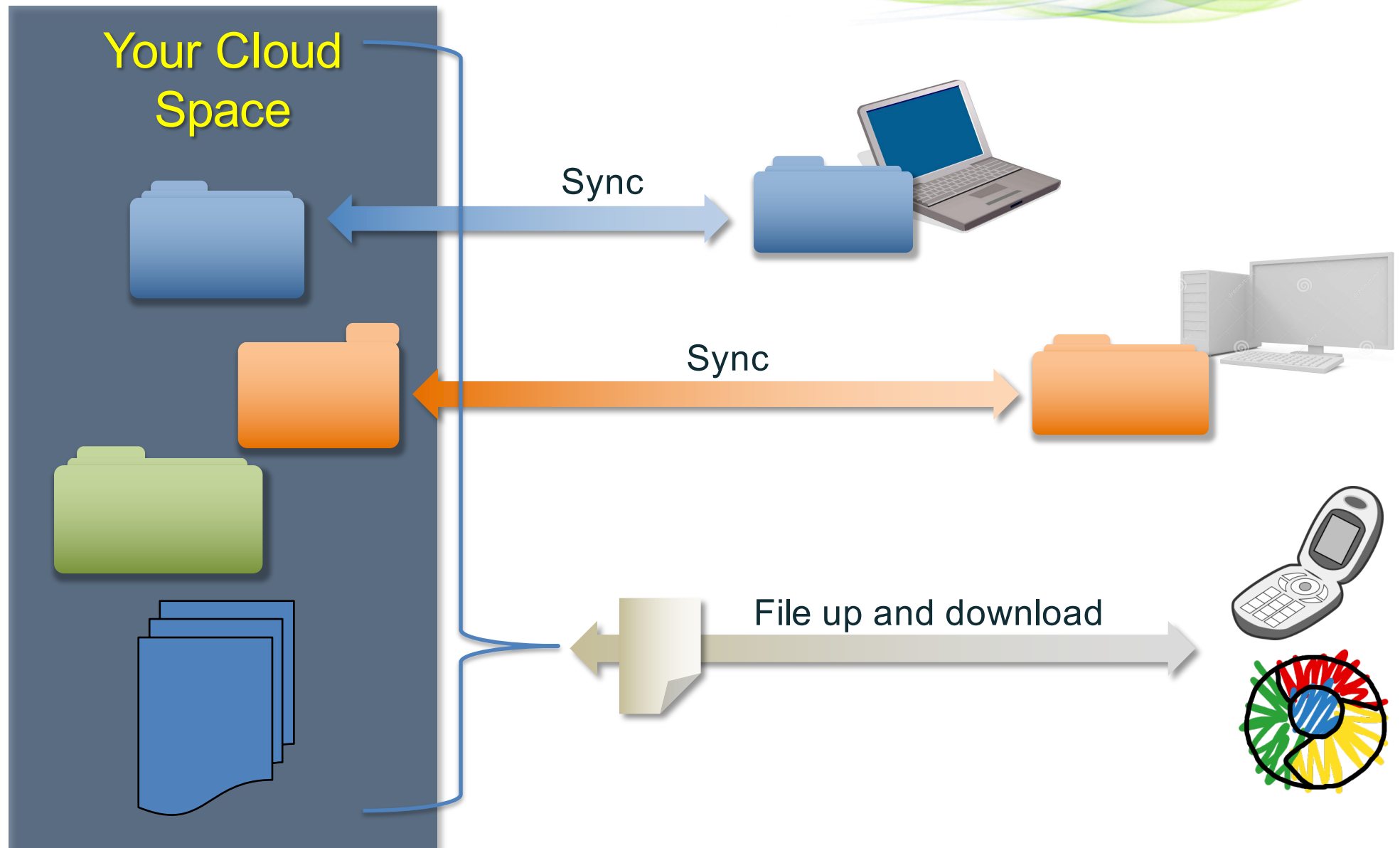
Share space(s) with others for upload



Others sharing data with you (in your home)

And the sync'ing

How does that look like

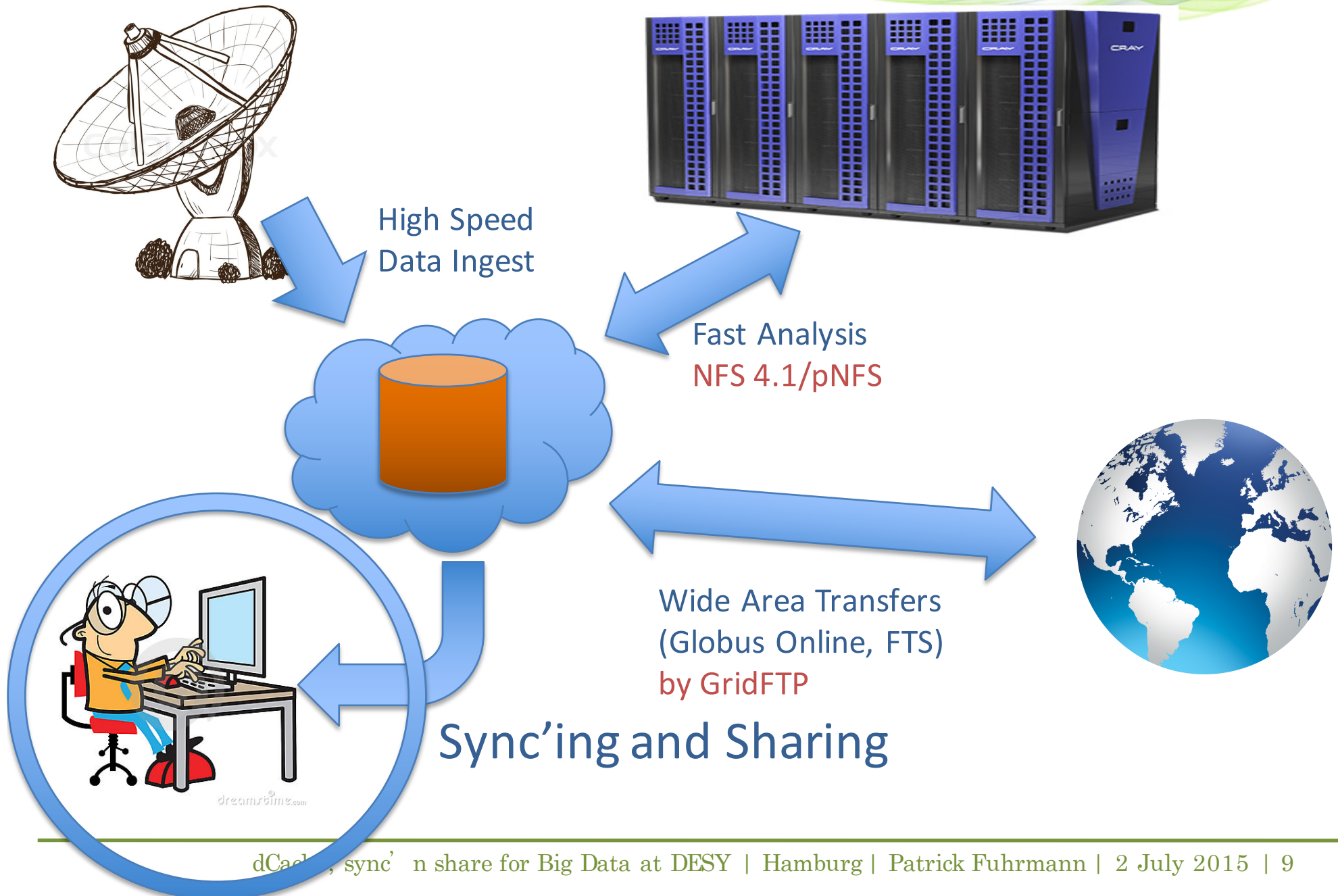


- Request for *unlimited, indestructible storage*.
- Request for *different quality of services* (SLA), coming with different price tags and controlled by customer.
 - *Data Loss Protection* (non-user introduced), e.g.:
 - One copy.
 - Two copies on independent systems.
 - Two copies in different buildings.
 - Two copies at different sites (e.g. Hamburg and Zeuthen)
 - Some of above plus 'n' tape copies.
 - *Access latency* and max data rate, e.g.:
 - Regular sync and web access.
 - Worker-node access: High throughput
 - Low latency (e.g. on SSD) for HPC.
- User defined *Data Life Cycle*
 - Move data to tape after 'n' months.
 - Remove from random access media after 'm' months.
 - Make public after 'x' month.
 - Remove completely after 'y' months.
- Controlled by Web or API (*Software defined storage*)

And not to forget

- Access to the same data via different transport mechanisms
 - GridFTP for wide area bulk transfers
 - http/WebDAV for Web applications
 - NFS 4.1/pNFS for low latency, high speed access (e.g. HPC)
- Access with different credentials
 - Username / password
 - X509 Certificates
 - SAML (Single Sign On)
 - Open ID Connect (See Anupam 14:15 today)
 - Kerberos
 - Macaroons (bearer tokens)

Scientific Data Flow



To make a long story short :

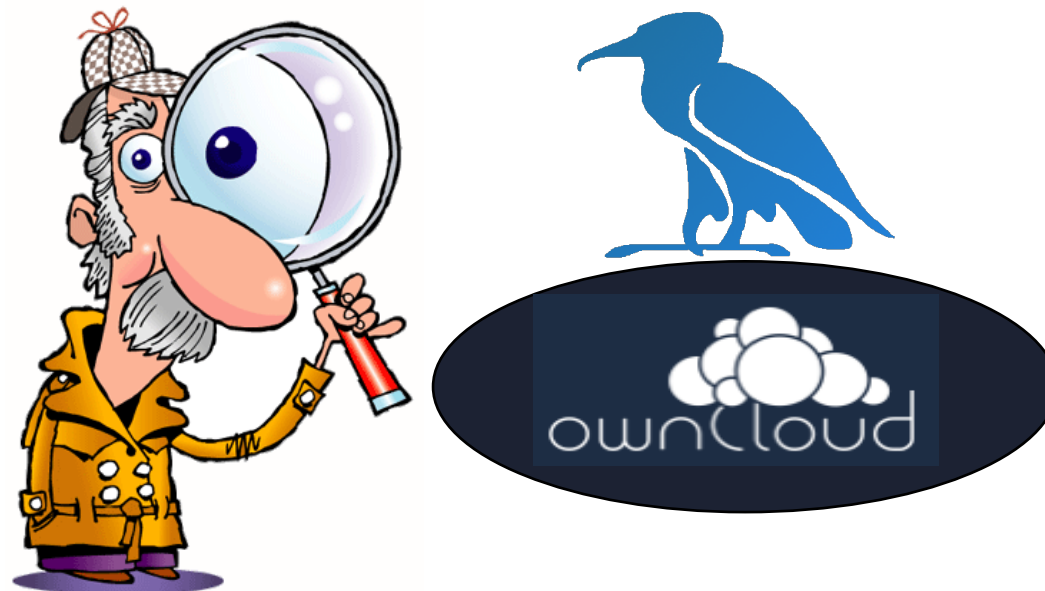
– Sync and Share part : Decided to go for OwnCloud

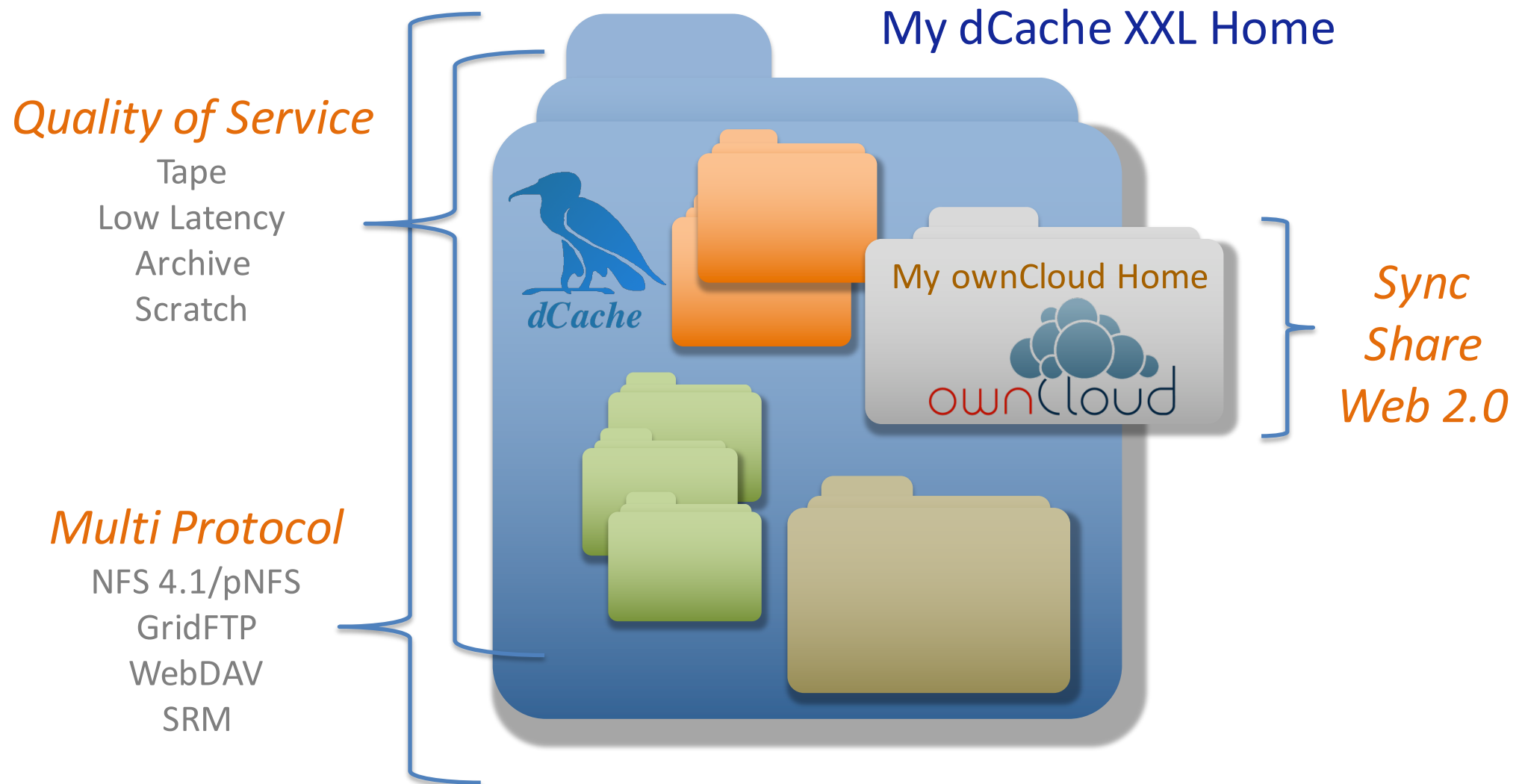
- They provide essentially what we need.
- Most sites decided to go for OwnCloud, so we were hoping to benefit from synergy effect.
- They initiated ‘federated OwnCloud’ structure via countries which we found promising

– Software Defined Storage and Quality : dCache

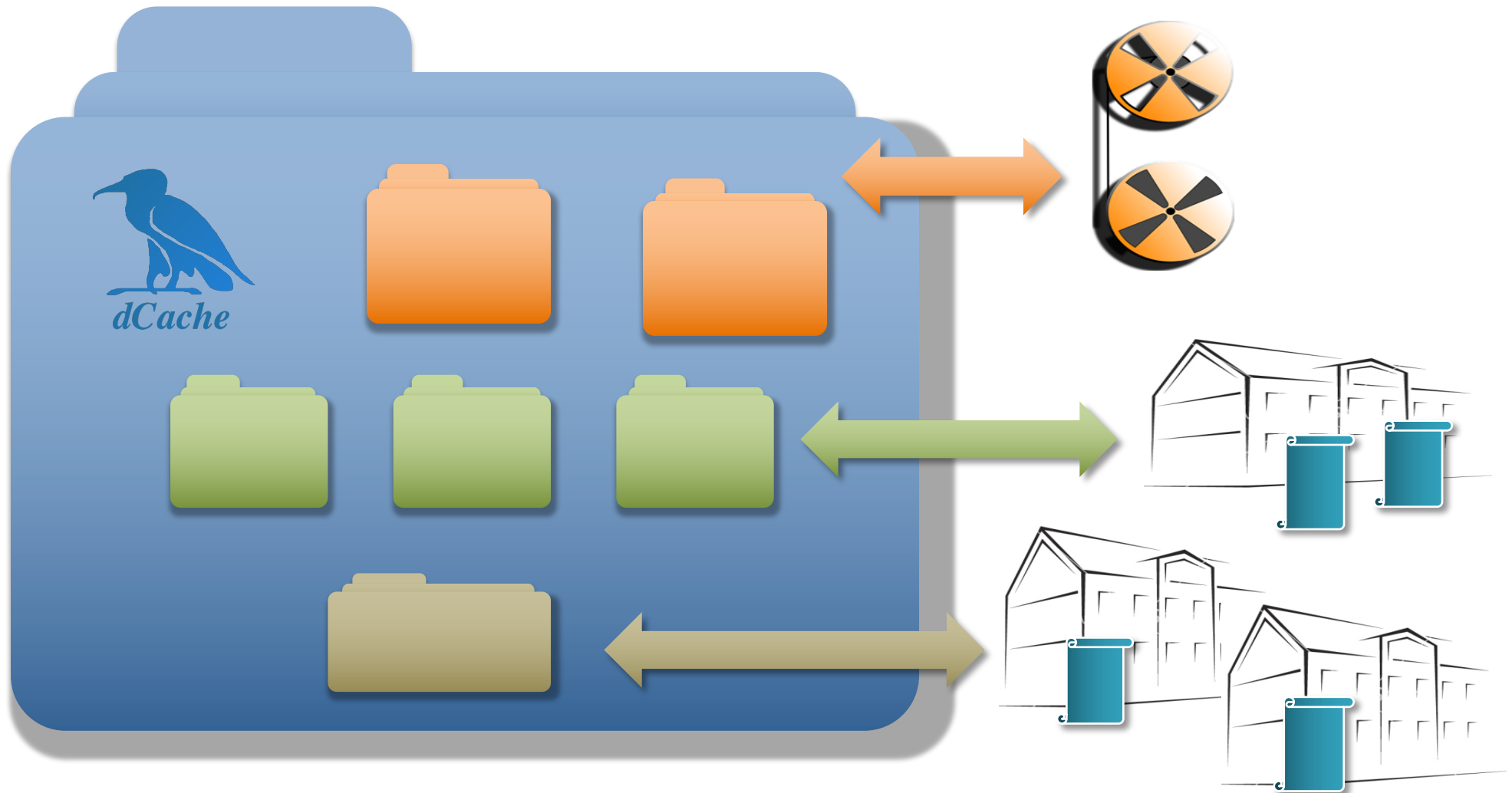
- Surprise surprise
- That’s our jobs for over a decade
- Multi Access Protocol
- Multi Tier (Software defined storage)
- Multi Authentication

Now ... how does this look like

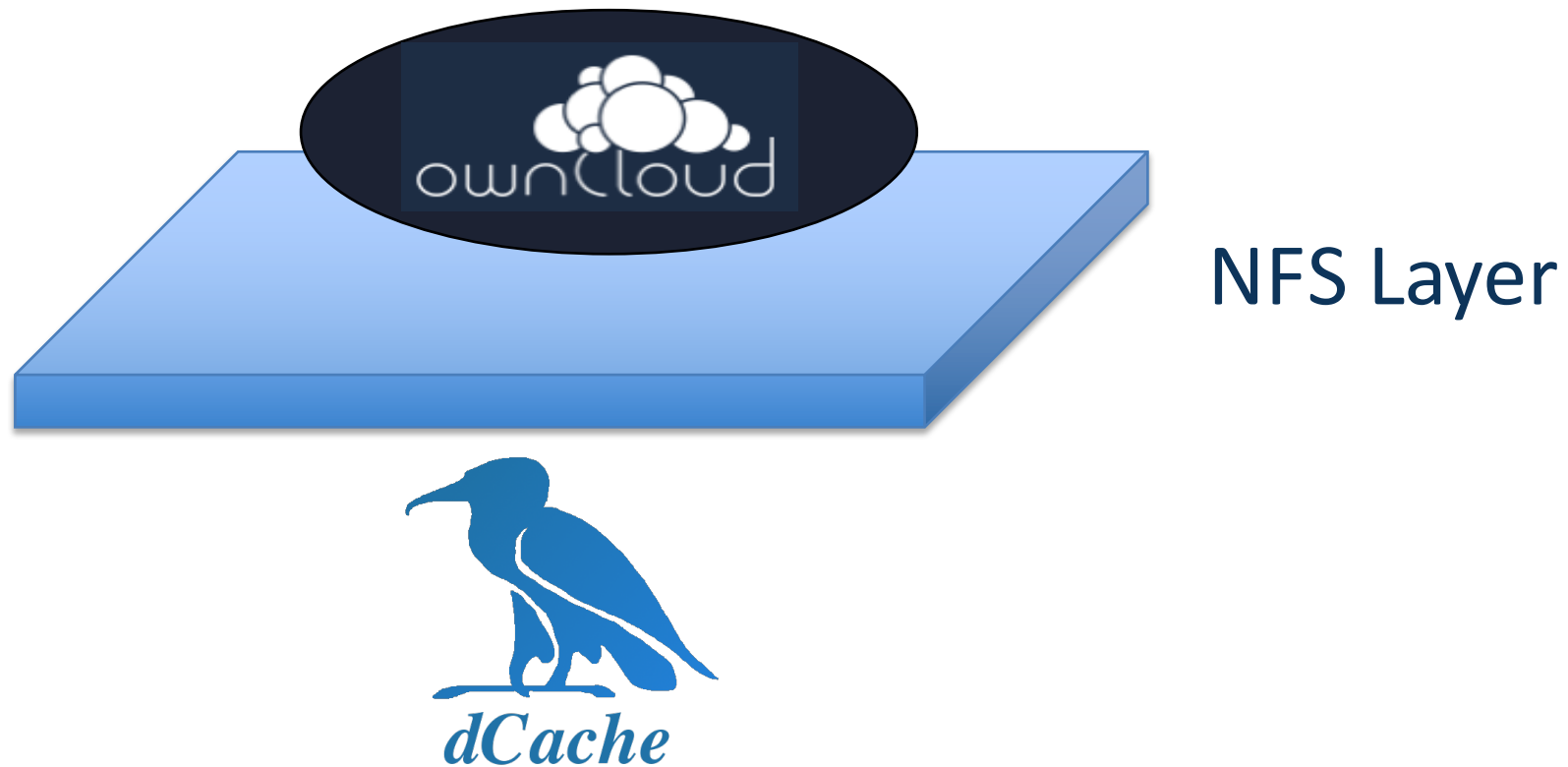




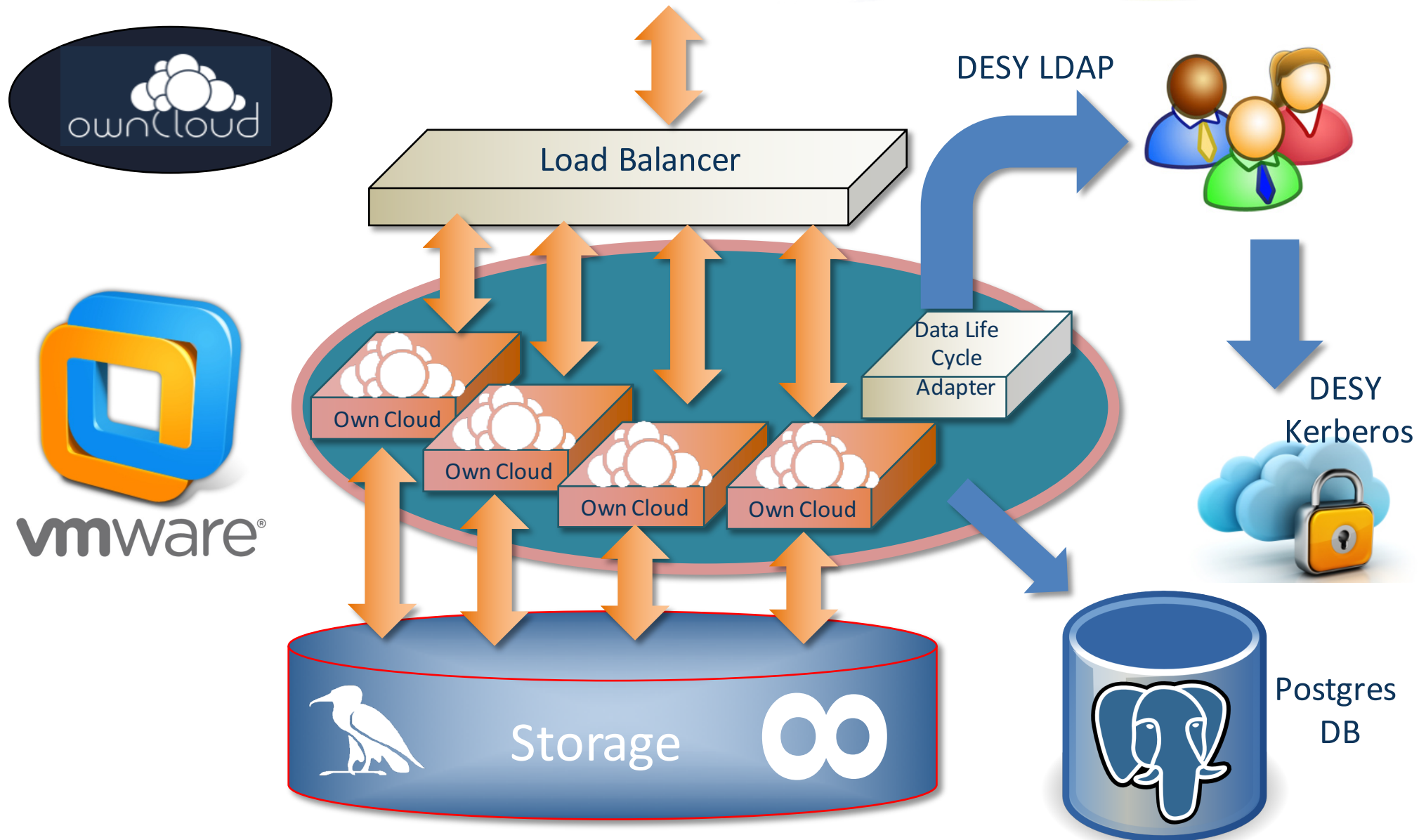
My dCache XXL Home



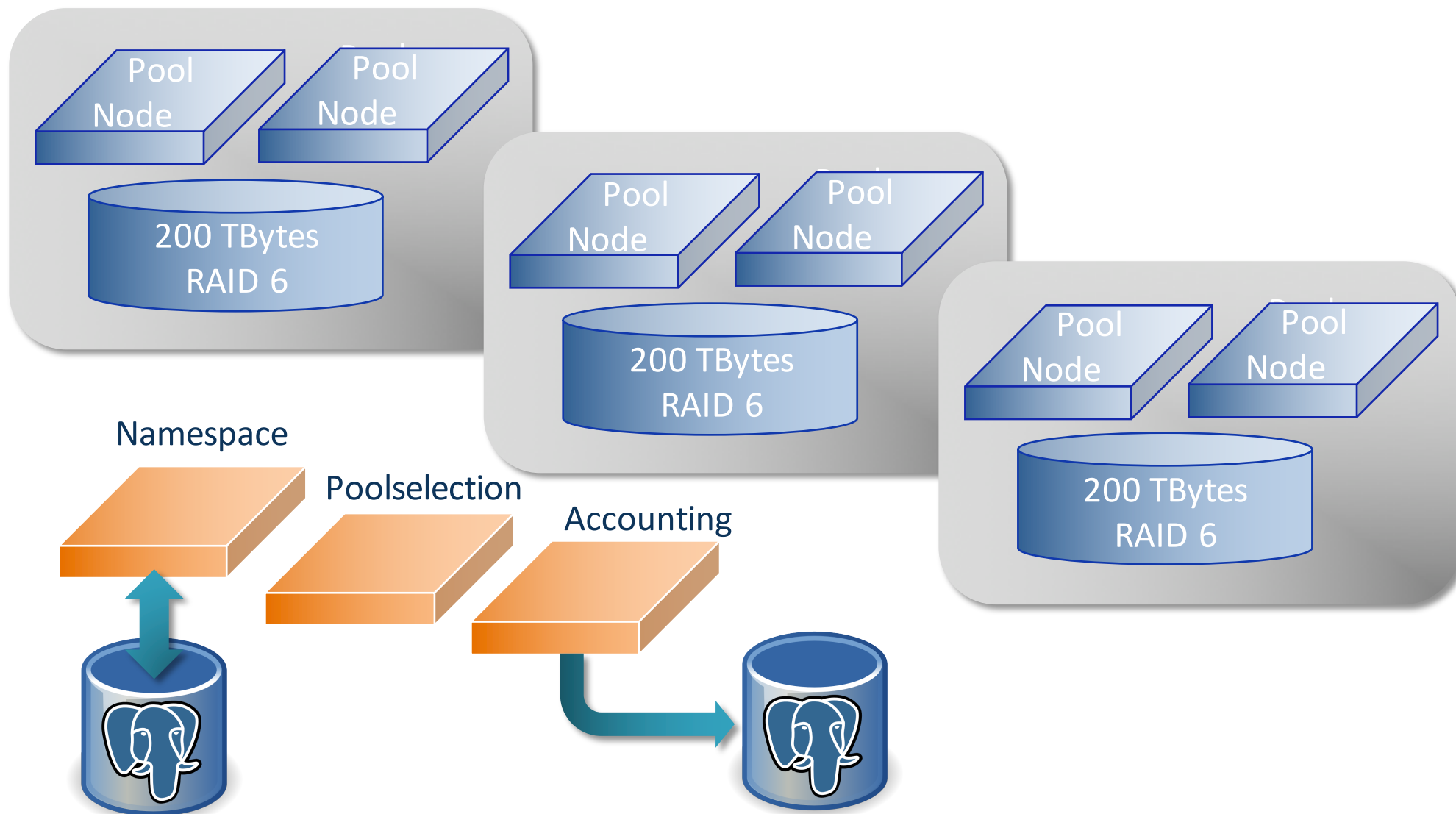
Simplified Technical Perspective



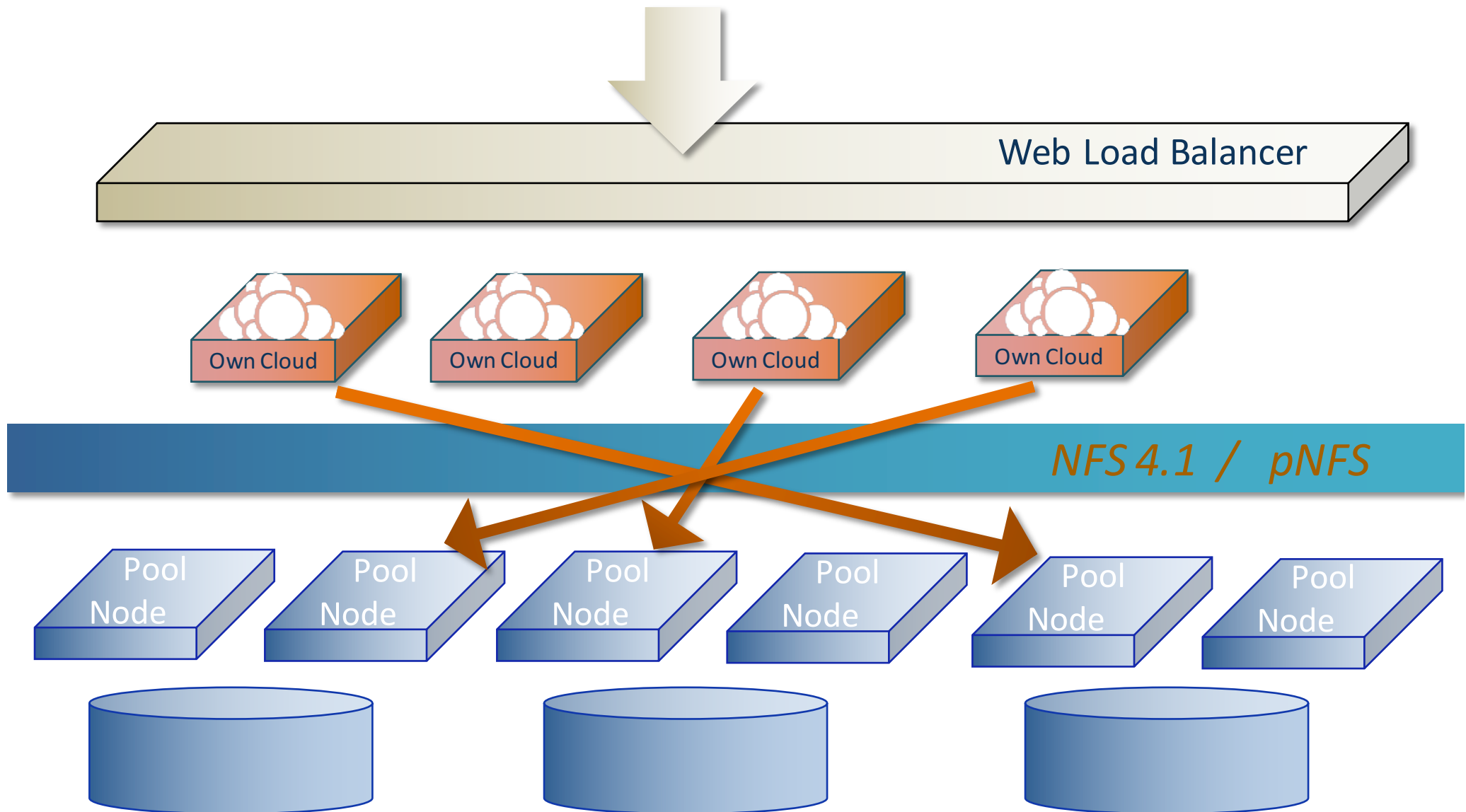
The Own Cloud Part

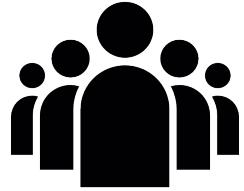


The dCache part



The horizontal scaling





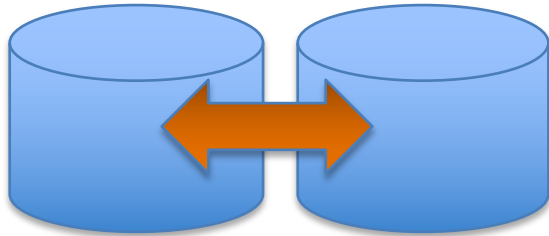
560 Active Users (not just registered)



20 M Files



1.6 M Directories



- 30 TB in total, mostly small files
- QoS : 2 replica, replica manager
- No tape copies yet

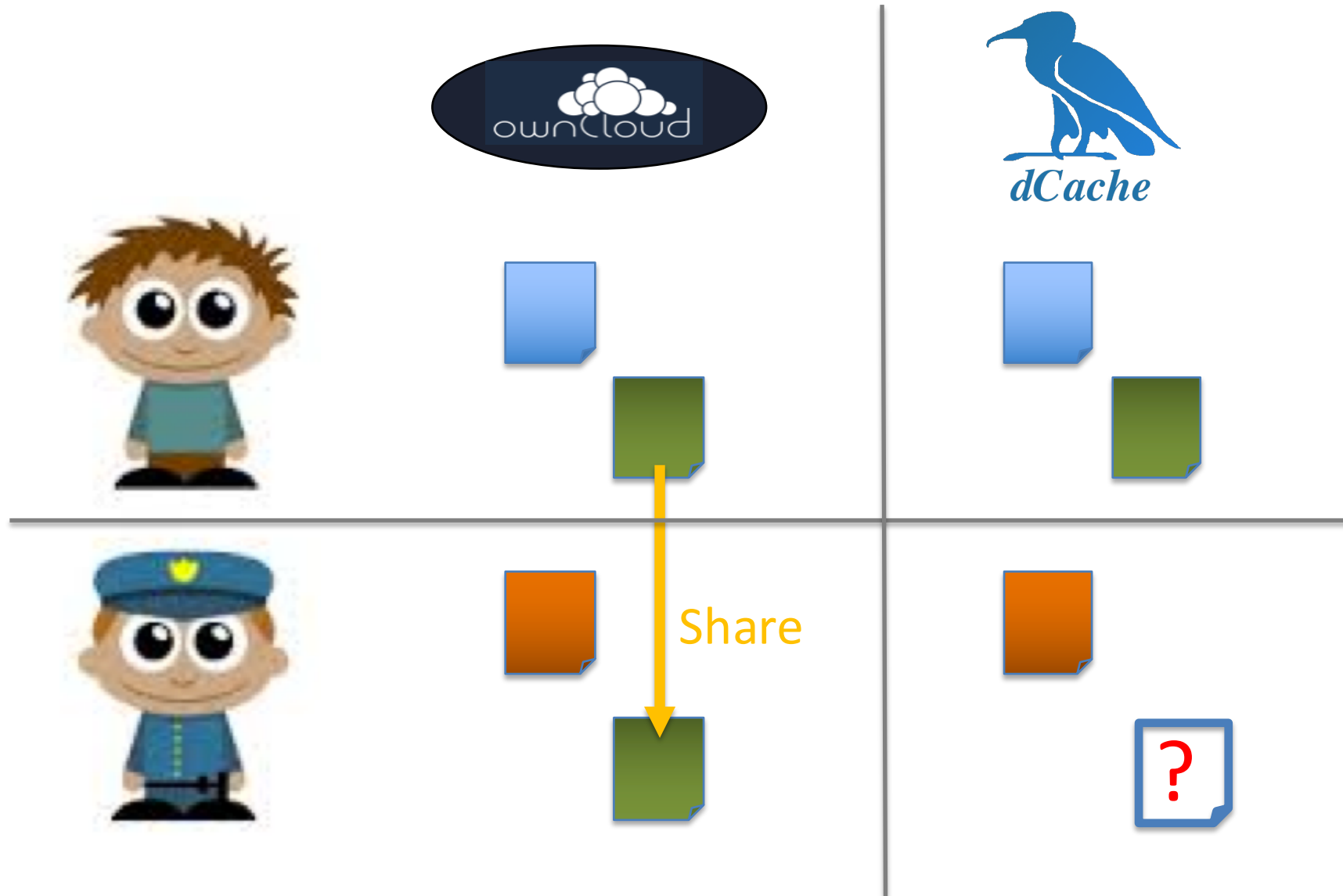
Things are running just fine

- If OwnCloud is the only access to dCache

However, if we allow NFS (etc) to the same space

- Writing OwnCloud, reading NFS
 - no problem
- Writing NFS (etc) , reading OwnCloud
 - OwnCloud sync script needs to be called
 - Past : only available per user, not per directory
 - For some users this becomes a pain
 - Better now with OwnCloud 9

Even bigger issue : “shares”



Problem

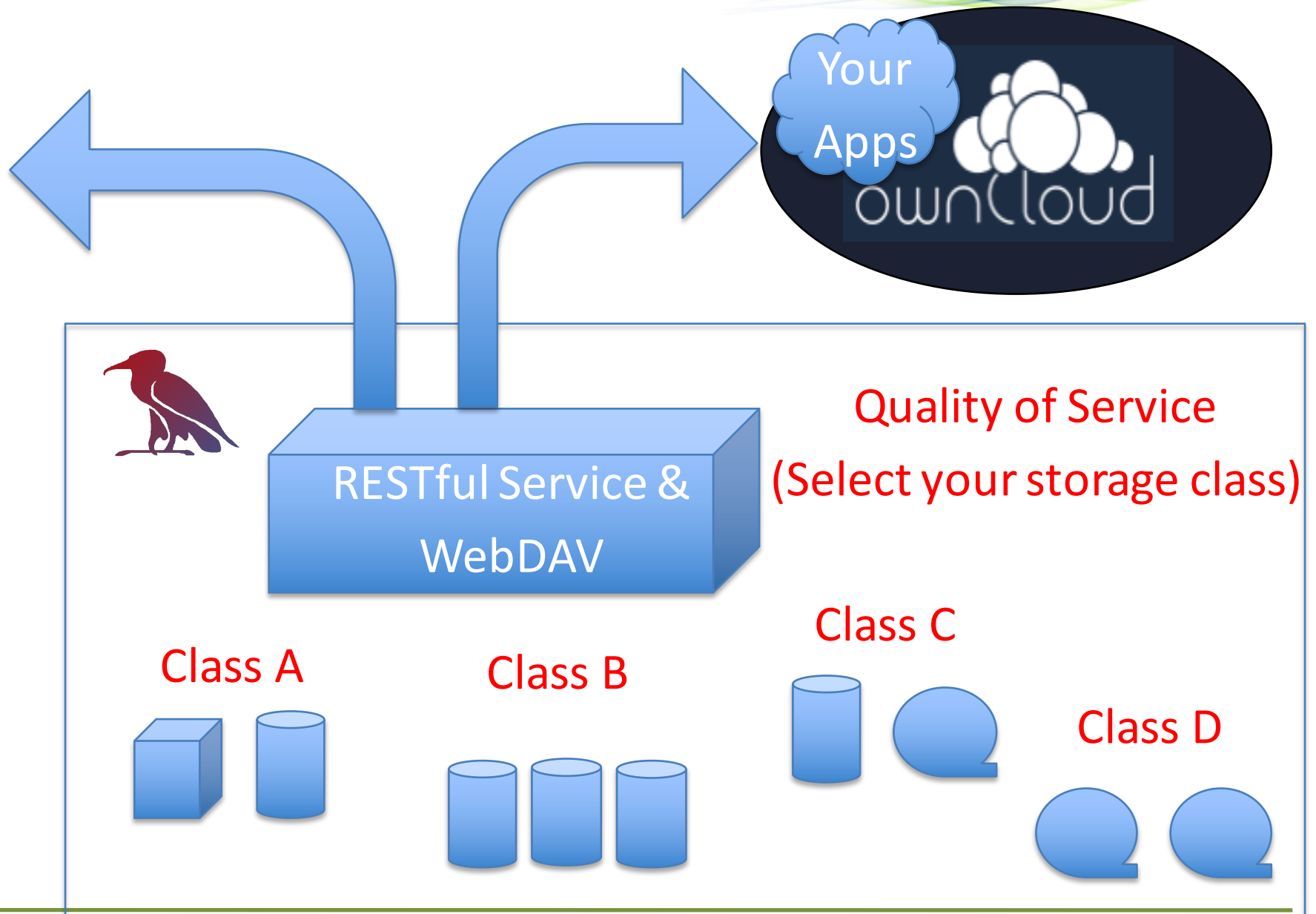
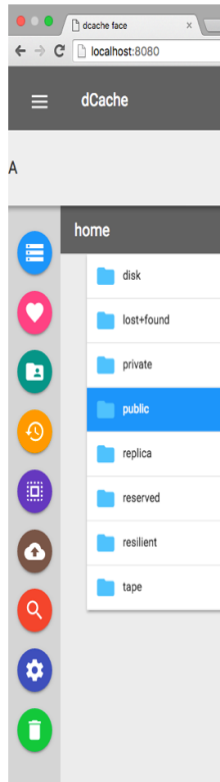
- Two sources of the file system view
 - dCache and OwnCloud
- Synchronization is possible not that great
- Can dCache become the actual only source of both systems ?
- Yes, BUT :
 - dCache needs the concept of a “share”
 - Similar to file, dir, hard link, soft link
 - Issues with ‘ACL’ : on share or share-target ?
- We’ll very likely implement that concept

Advantage

- Having that solved (the part with the share)
- And taking the new RESTful service (see Marinas talk)
 - You can either use FEMI's GUI
 - Or write your own OwnCloud app
- To perform “Data Management” or QoS on your data yourself.
 - Specify Storage Quality (and price) : Disk, Tape and number of copies (resilient manager)
 - Flush a file to tape (and save space) or bring-online in advance of running a job on that data.

New approaches in dCache

dFace



dCache spec for Dummies

dCache.org 

NFS/pNFS

httpWebDAV

gridFTP

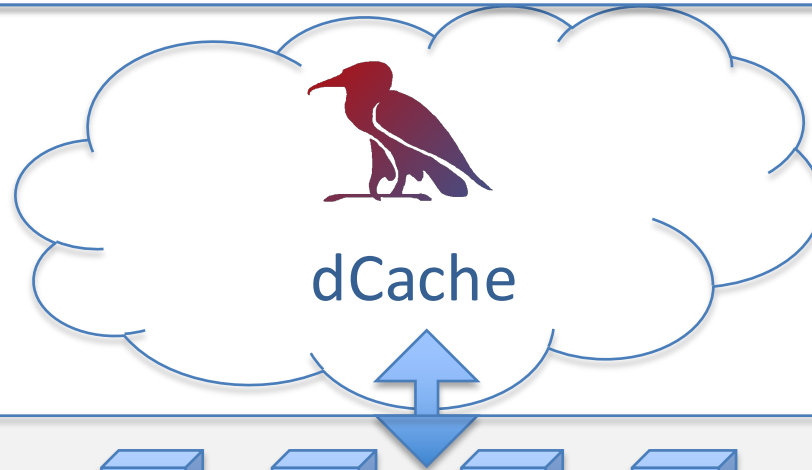
xRootd/dCap



Protocol and Authentication Engines

Virtual File-system Layer

Media Transfer Engine
and Pool
Management



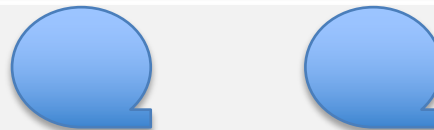
SSDs



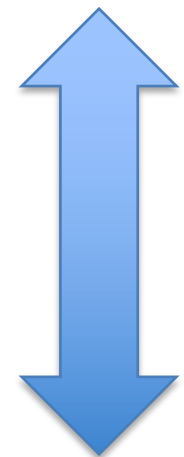
Spinning Disks



Tape, Blue Ray ...



Automatic
and
Manual
Media
transitions



The END

further reading
www.dCache.org