

dCache, Sync-and-Share for Big Data

Paul Millar

dCache workshop Spring 2015





Need a sync-n-share service at DESY

- **Requirements:**

- Easy to use,

- Store everything at DESY,

- Integrate with existing infrastructure.

- Anticipated **future usage:**

- change data between syncing and non-syncing storage,

- like Amazon, provide different QoS with different costs,

- share data without syncing,

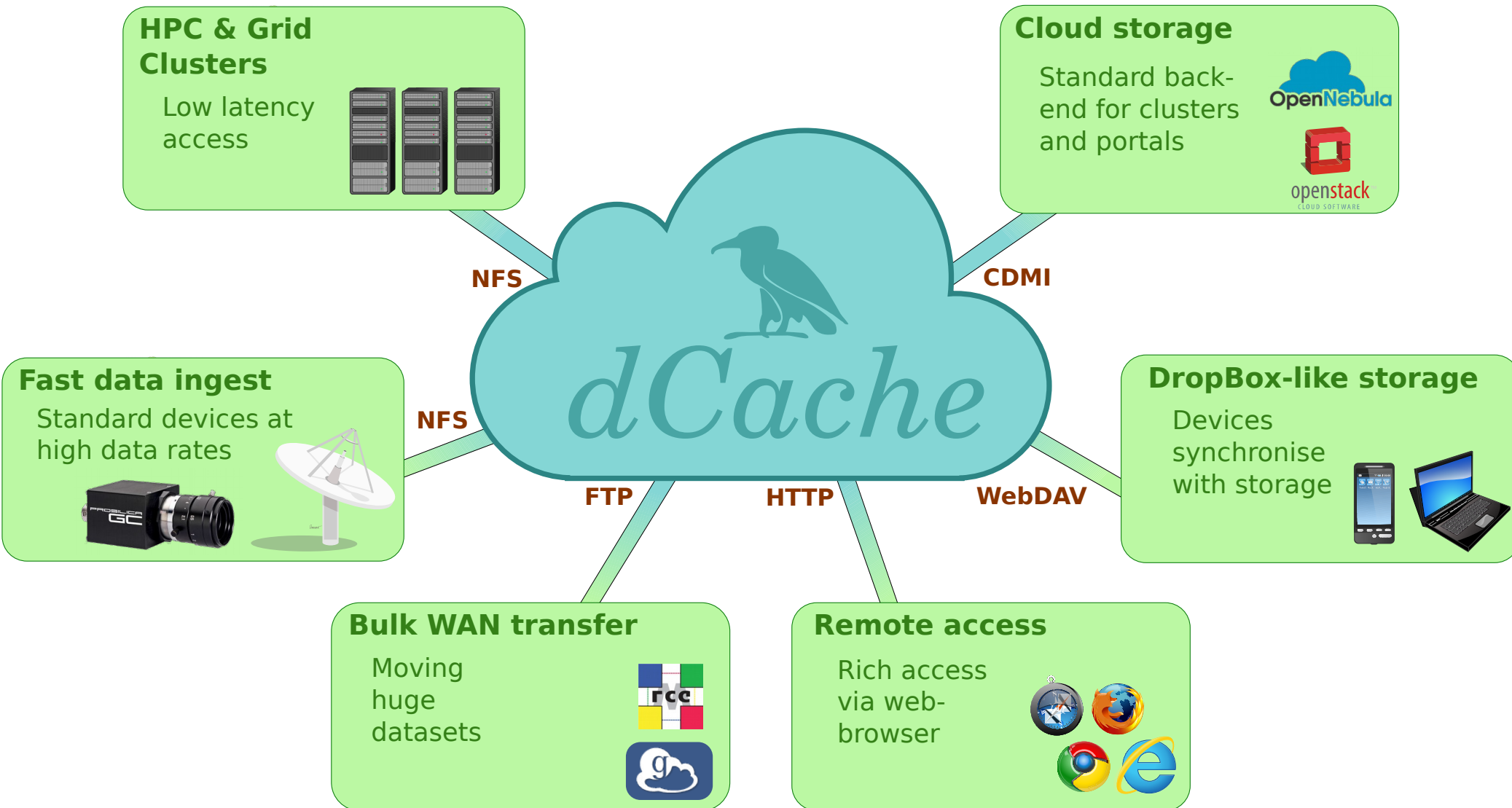
- 3rd party transfers between sites,

- direct access to sync space from compute facilities.

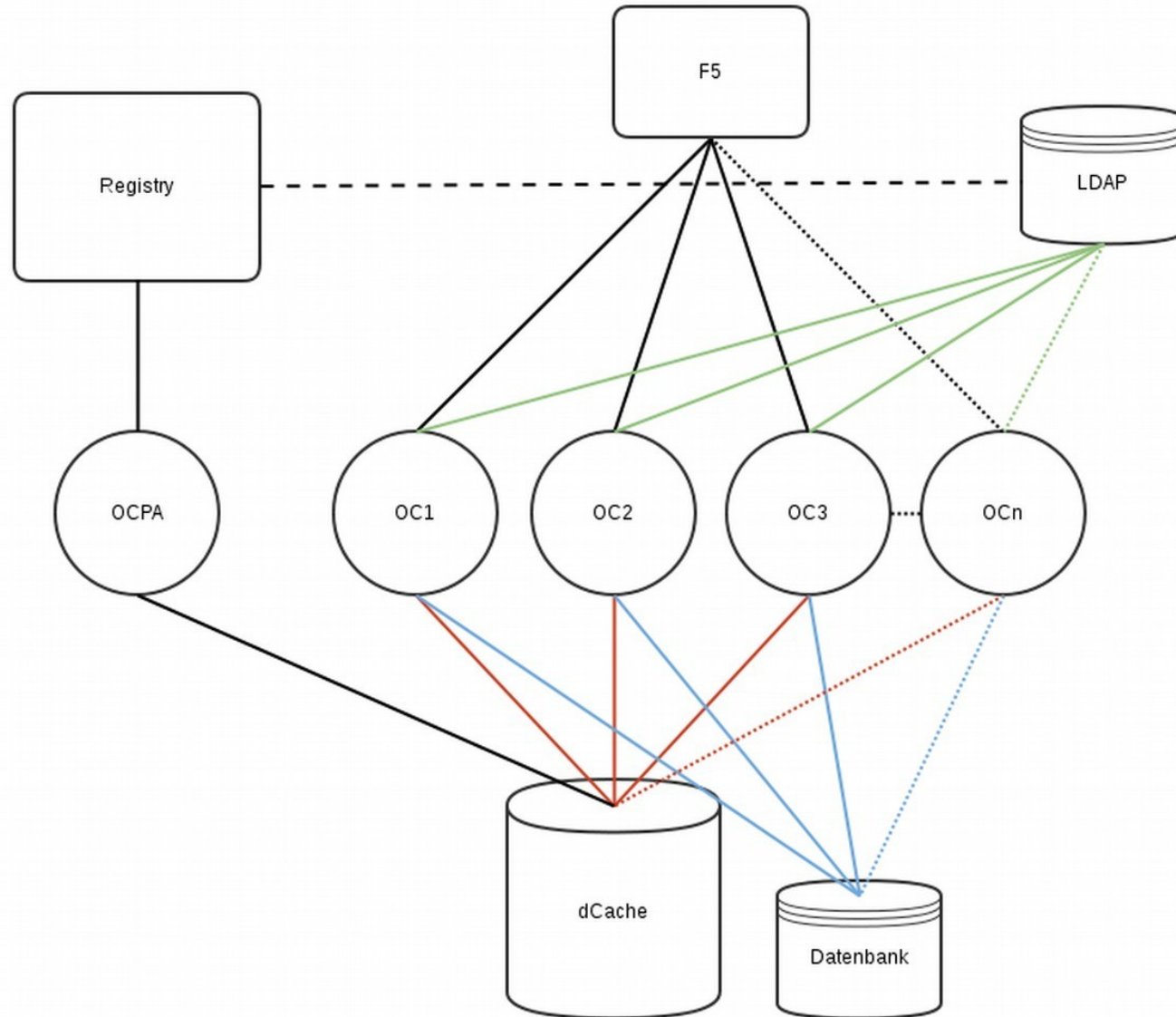
How we solved it at DESY

- Looked around, chose two open-source projects:
 - **dCache: powerful managed storage** system
 - Integration with scientific data life-cycle;
 - “Hot” data can be stored on SSDs, “cold” on cheaper HDDs, “archive” tape;
 - ... but no sync and share facilities.
 - **ownCloud: popular front-end**
 - Our collaborators adopting ownCloud makes it more attractive;
 - ... but assumes storage is managed.
 - Combining these two gives DESY the best of both worlds:
 - dCache is mounted on servers with **NFS v4.1/pNFS**, running community edition ownCloud.
 - Integrated with DESY Kerberos, LDAP and “Registry”.
-

The scientific cloud vision

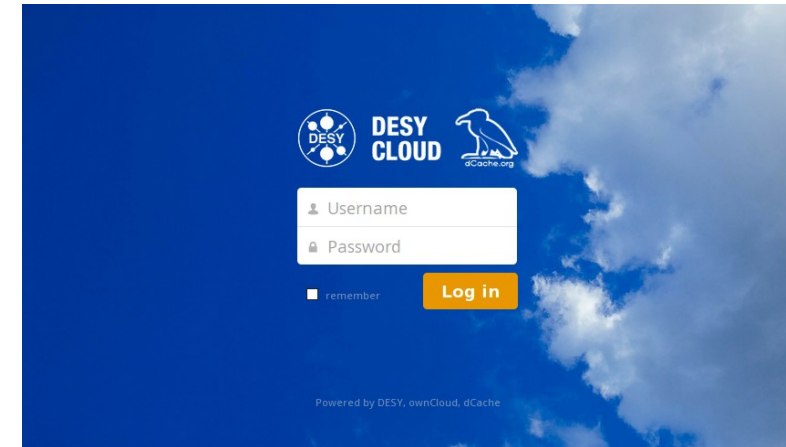


Integration within DESY infrastructure

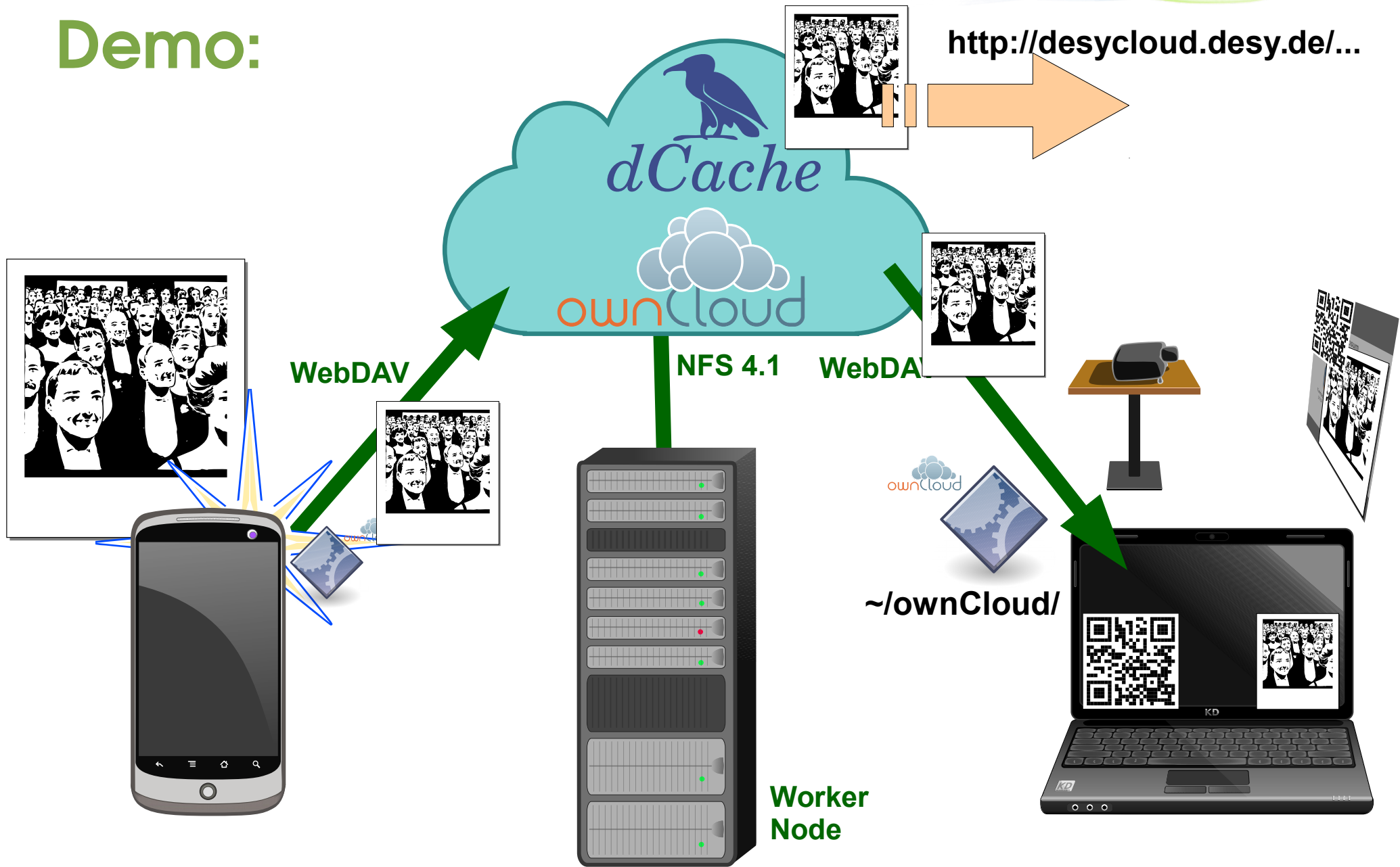


The DESY Cloud service

- Status: **production**(-ish), but only for a few groups:
219 users, 2×10^6 files, 2.4 TiB
- Required minor **patches** to ownCloud & dCache:
Changes always pushed into regular dCache releases; ownCloud 8 has all changes.
- Have a blueprint for **any site** to reproduce.



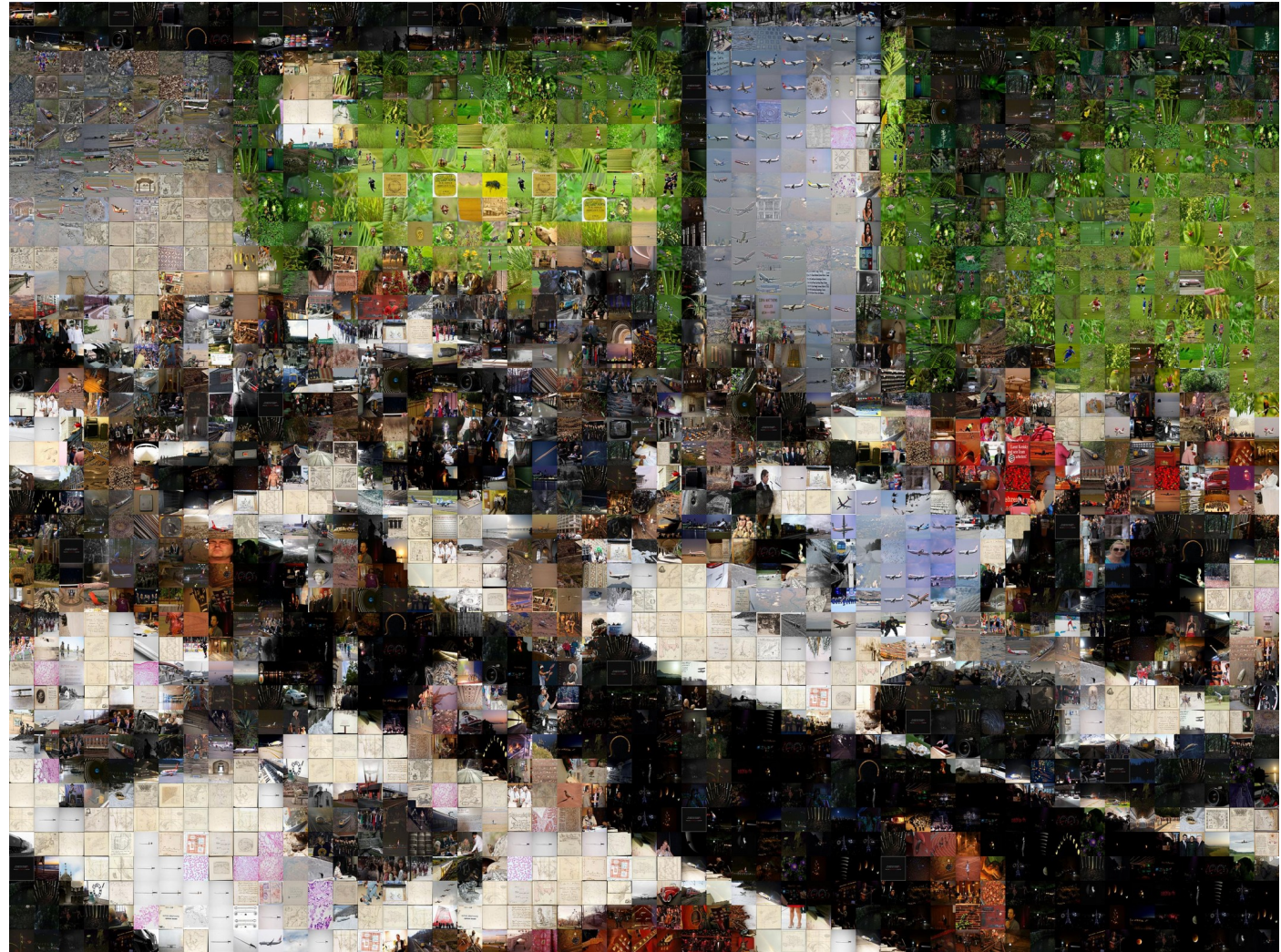
Demo:



Demo: sync-n-share



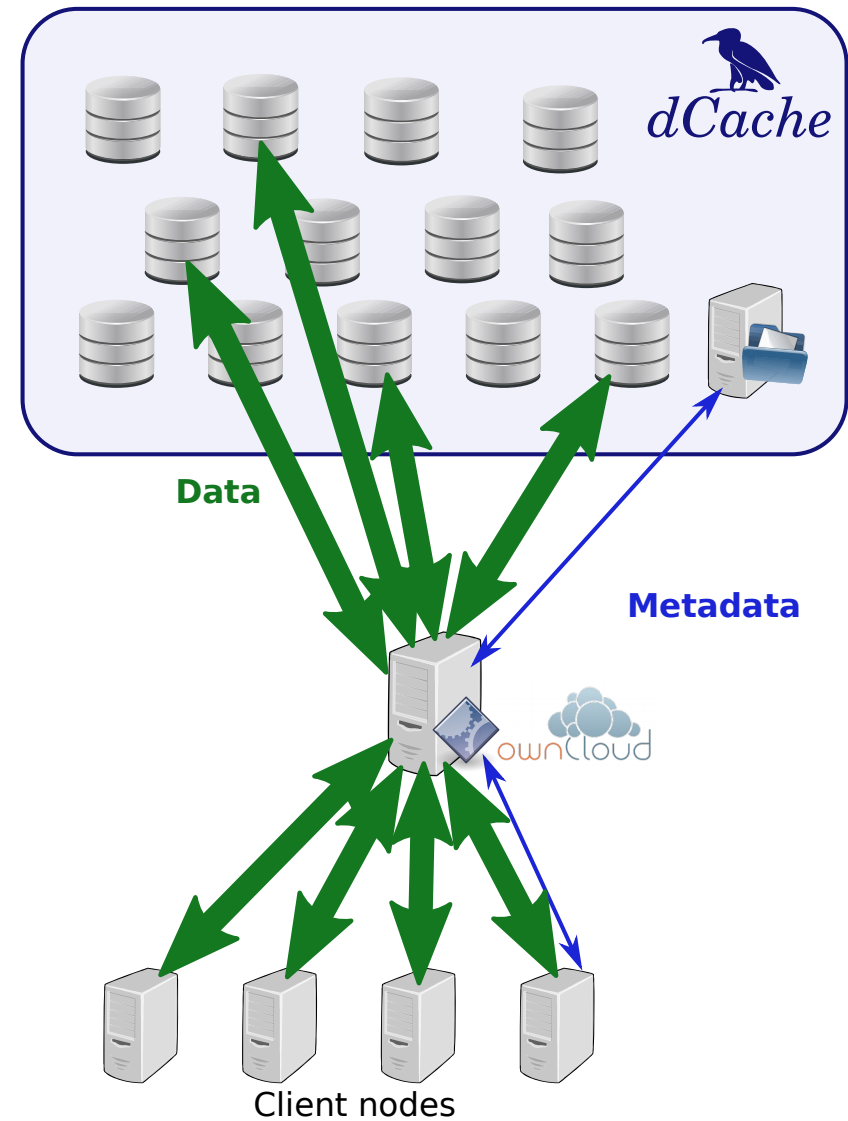
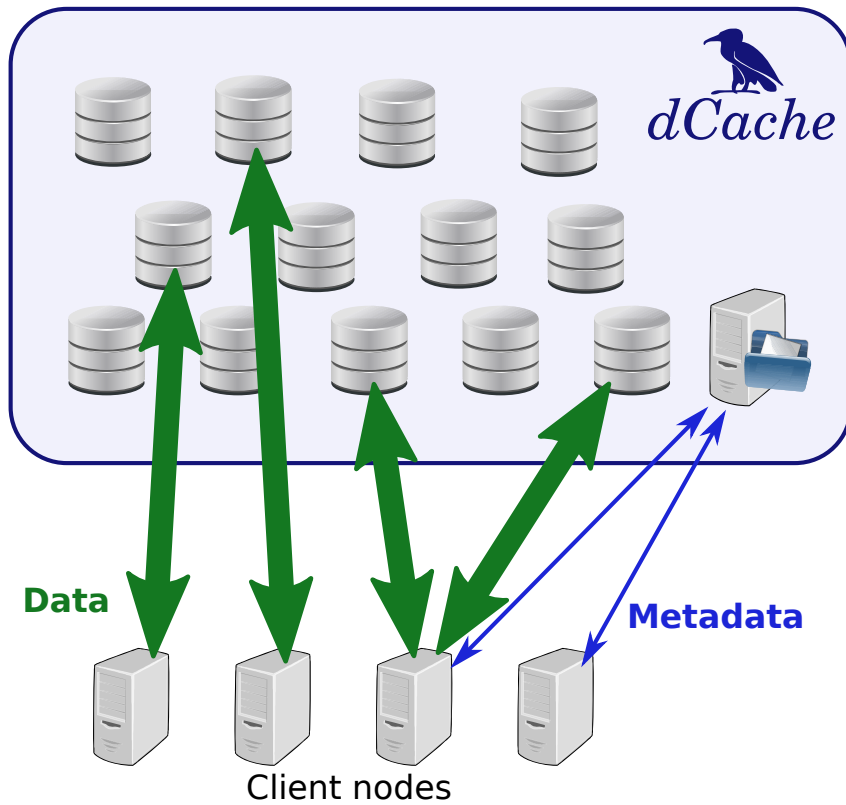
Demo: processed image, from WN



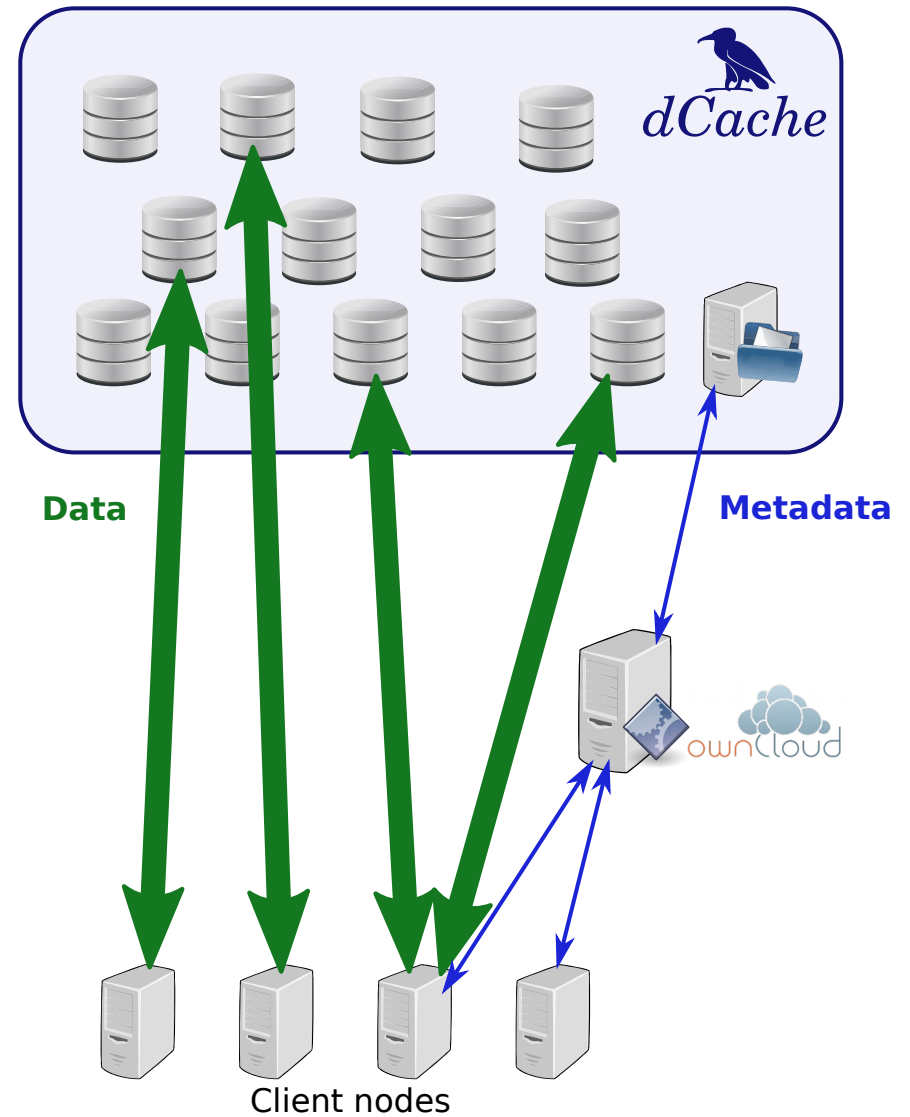
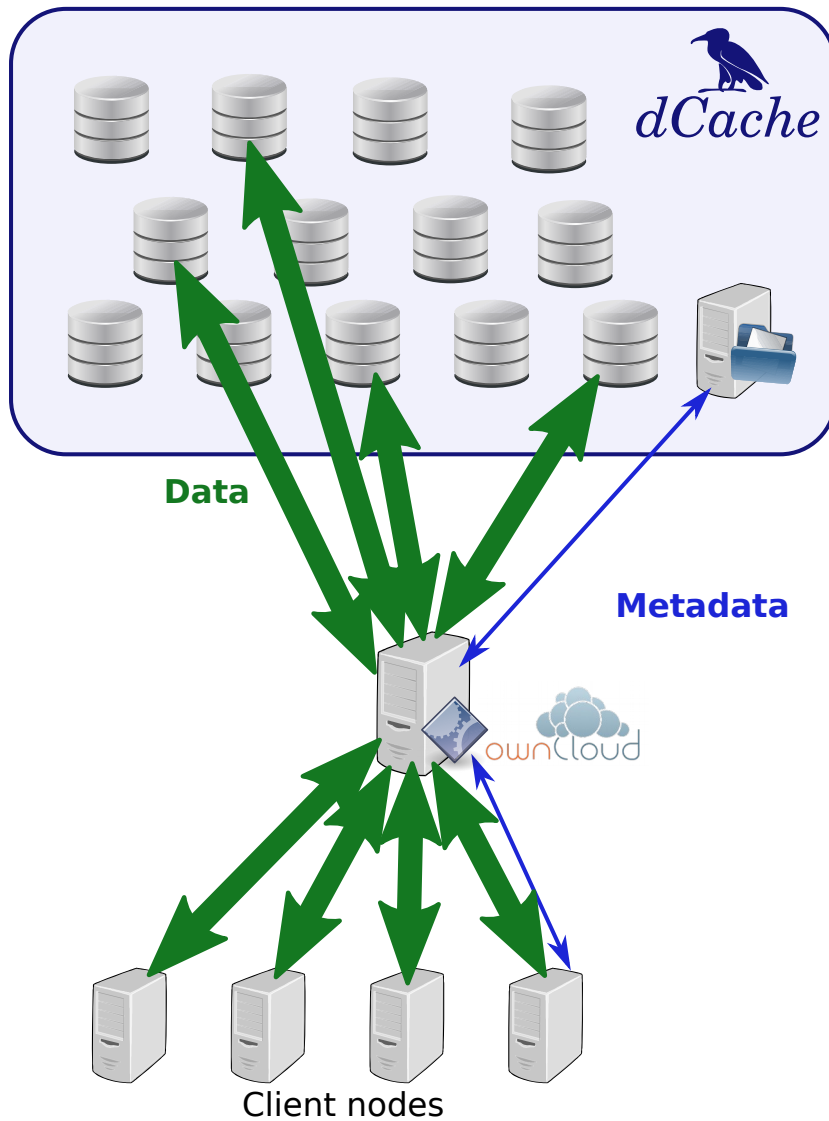
Development and future work

- Allow **direct access** to ownCloud files:
 - Supporting direct access from worker-nodes, 3rd party transfers, ...
 - Files in dCache need to be owned by the **user** (i.e., not user `owncloud`)
 - Couldn't fix ownCloud: work-around within dCache
 - **Consistency** between ACLs and shares:
 - dCache ACLs to honour ownCloud shares and vice versa
 - **Integrity**; e.g., propagate and handling checksums,
 - **Notification**: avoid client polling,
 - **Redirection** support for sync-client:
 - ownCloud server proxying data is bottleneck; want syncing to be more efficient by taking data from where its stored.
-

NFS v4.1/PNFS vs ownCloud (currently)



ownCloud: currently vs with redirect



Thanks for listening ... any questions?

Backup slides



Not just ownCloud ...

- dCache team hosted a **two-day workshop** with project- and technical-lead of DCORE
 - Provides cloud storage with features beyond ownCloud
 - Some “big name” customers
 - Initial “lite integration” by **December 2014**
(includes redirection support)
 - Then providing “tight integration” with shared namespace
-

Experience: problems with ownCloud

- If underlying FS disappears, **all sync-clients delete all data.**
 - If underlying FS returns **EIO** on read, sync-client creates 0-length file: **impossible to recover.**
 - Bulk delete through web interface is **unreliable** (under investigation).
 - Rename directory causes client to **delete all files and upload** them again.
 - Admin interface **awkward** with $O(5k)$ users.
-

Thinking about sync-and-share

- Like other systems, small fraction of data is “hot”
 - SSDs provide better performance, but can't afford only SSDs; nice to have system that places hot data on SSDs, cold data on HDD.
 - Amazon had a smart idea: allow people to choose how much to pay
 - Let users choose between Normal and Glacial QoS; e.g., disable sync for Glacial-like storage but allow access via web interface
-



**DESY
CLOUD**



 Username

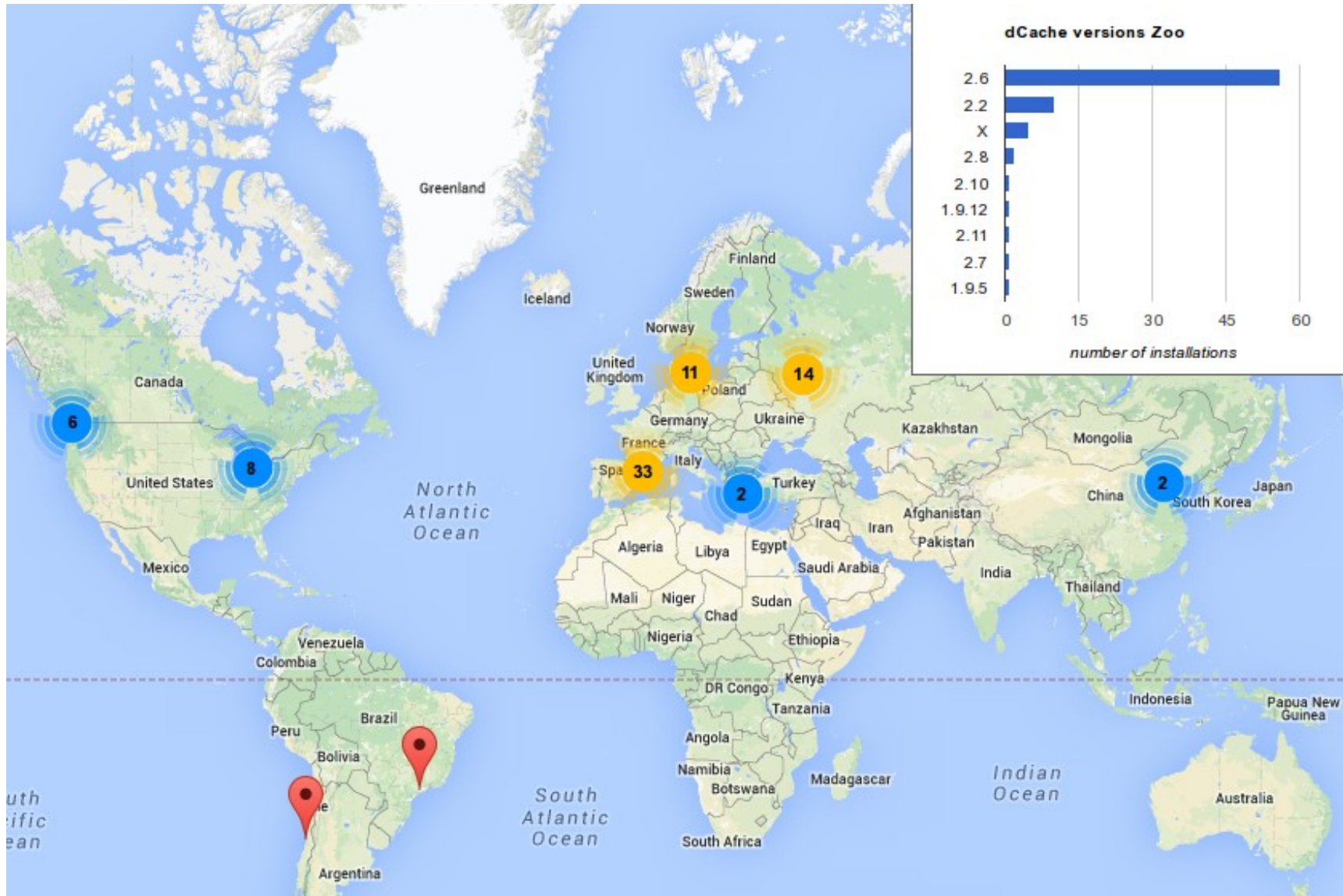
 Password

remember





Log in

Powered by DESY, ownCloud, dCache

WLCG dCache instances (only WLCG sites shown)



Over 10 years “Big Data” experience

Era	Disk cache	Grid Storage	Generic Storage	Cloud Storage
Additional Communities				
Additional Authentication	Trusted host	X.509, Kerberos	Username+PW	SAML, OpenID, OAuth, Token, ...