dCache: challenges and opportunities when growing into new communities

Paul Millar

on behalf of
the dCache team

EMI is partially funded by the European Commission under Grant Agreement RI-261611

Contents

- **Orientation**: what is dCache?
- Storage for the **non-HEP user**.
- **Cella Nova**: the new storage
- Summary**.**

1. Orientation

What is dCache?

- Data **Storage system**
  - Upload files, get at uploaded bytes again
    - *Files can be deleted, renamed, moved but not updated or appended; subdirectories can be created, deleted, moved, renamed.*
  - Separates front-end nodes, storage nodes and namespace (makes it scale)
  - Supports multiple protocols: *FTP, HTTP/WebDAV, NFS 4.1, xrootd & *dcap.
  - Runs on multiple platforms (just needs a JVM)
- Many **advance features**
  - Fine-grain control over data placement (on write, on stage)
  - Supports pools that are read-only, write-only, stage-only or any combination thereof
  - Dynamic hot-spot replication
  - Supports tape back-ends
  - Can maintain redundant internal copies of data
  - Flexible approach for establishing users' identity
  - Supports data integrity assurance
  - Many aspects may be customised by writing plugins

  ... plus more ...

dCache evolution

- 2000–2005: the **site-centric era**

  Providing storage for local users. Users authenticate against site-local systems.

- 2005–2011: the **Grid era**

  deployed at sites throughout the world as a “Storage Element” using X.509 identification.

- 2011–... : **the SaaS era** “Storage as a Service”

  A single dCache can provide storage for multiple end-user groups, auto-provisioning users, who identify themselves in various ways, providing different qualities of service (**Amazon S3-like** service, **DropBox-like** service, **federated storage**, ...)

NB. these dates are very approximate

2. Storage for the non-HEP user.

LOFAR

LOFAR

- **Juelich** and **SARA** use dCache to provide storage for **LOFAR**
  - SARA currently provides ~**1PB** of storage
- Used for LOFAR's LTA: **long-term archive**
  - Data accessed using **SRM** + **GridFTP**, users identified with **X.509**
    - *No space tokens, but different QoS provided (d1t0, d1t1 and d0t1).*
  - SARA is investigating HTTP/WebDAV
    - *X.509 and username+pw authentication.*
- LOFAR have developed integration software
  - Generally treats EGEE/BiG Grid and Astro-WISE as separate domains
  - Metadata (hosted in Astro-WISE) is common and LTA data is accessible from both domains.
  - LOFAR users cope with (but don't like using) X.509 user certificates.
    - *Normal authentication is with LDAP*

XFEL example

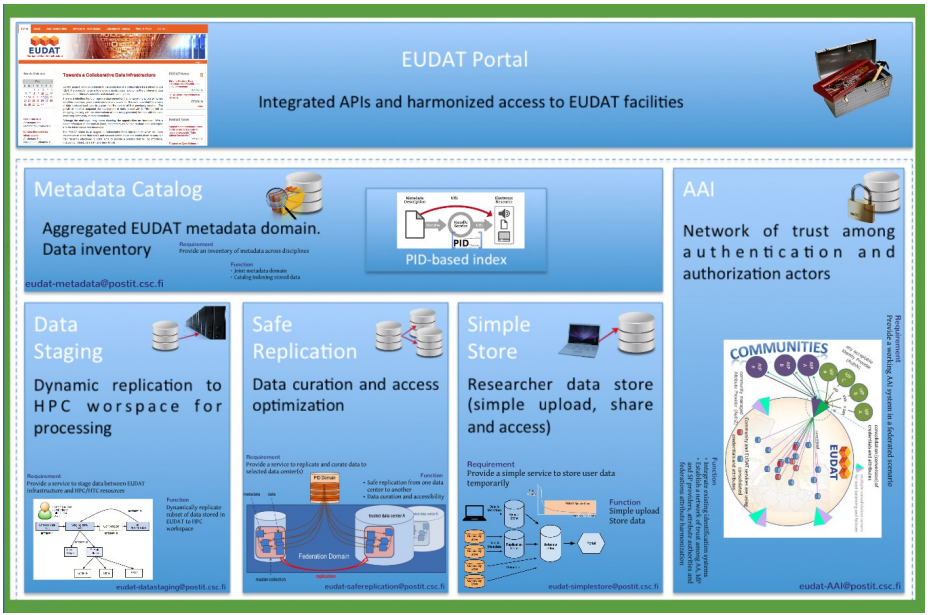
- Free electron laser facility
  - currently being built at DESY
- Software design is currently under development
  - dCache will be used, likely to provide archival storage
  - A potential barrier to broader use is end-user software's **write patterns** and possibly **immutability**.
- **Metadata** is key for most users' work-flow
  - Discovery of data is through the metadata
  - Metadata is held outside dCache
  - a web-portal to allow browsing and searching.
  - Web access is initially via portal, but redirected to dCache for accessing data.

EUDAT example

- Relatively **short project** (3 years)
  - Needs to take “read to use” software and deploy it, with minimum integration.
- User communities already have large amounts of data:
  - Software must work “along side” what already exists.
  - Unclear to what extent dCache will be used

    (Although SARA is a member)
  - … but their requirements are interesting.

EUDAT Core Services



Note how (in general) the underlying **storage isn't mentioned**, it's assumed. This relies on **easy integration** of storage with higher-level functionality

Slide adapted from Damien Lecarpentier's presentation, KE Research Data Working Group Meeting, Copenhagen, 14th August 2012

EUDAT: requirements

Service / Community	SR	DR	MD	SS	PID	AAI
CLARIN	X	+	X	X	+	X
ENES	X	X	X		+	X
EPOS	X	X			X	X
VPH	X	X			X	X
LifeWatch	X	+	X	+	+	X

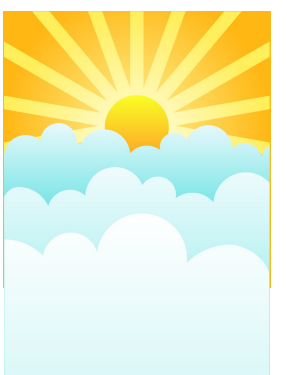
Note that **AAI** (Authentication) is a common requirement, and that all communities either require or are interested in **PID** (persistent identifiers).

Slide adapted from Damien Lecarpentier's presentation, KE Research Data Working Group Meeting, Copenhagen, 14th August 2012

Summary of friction points

- Evaluation stage:
  - Does a new project even know about dCache?
  - Do they understand dCache's flexibility?
- Missing features:
  - Missing functionality within dCache (e.g., mutability?)
  - Necessary “hooks” for easy integration with higher-level components
- Authentication and Identification management.
- Authorisation: if not based on filesystem permissions.
- Boundary activity: data ingest, egress and management.
- Desire for a “turn-key” solutions

3. Cella Nova: the new storage



Evaluation stage

- People can only evaluate what they know about
- How do people know to evaluate dCache?
  - Word-of-mouth
  - EMI, EGI, ScienceSoft, …
    - *Would it make sense for the EU to have a registry of EU-funded software projects?*
- But, as a general message:

  If you're building something that needs reliable, flexible, powerful storage, have a look at dCache.

  If you find a limitation, get in touch with the developers <support@dcache.org>; we might already be planning to working on it (or it might be easy to fix).

Identity management

- Federated Identity Management: **FIM**

  OpenID, SAML (“Shibboleth”), OAuth2, ...

- **gPlazma** is powerful enough to support all these

  It's use of **plugins** is ideal, just need to write the plugins :-)

- **HTTP access** need updating to provide new login possibilities

  OpenID login, Web-profile SAML, ...

- There is still a problem with **non-HTTP access**:

  Moonshot is most promising approach; it's also being investigated by other projects (Contrail, Eudat, ...)

- Need to handle **provisioning**: creating accounts automatically.

  Decommissioning is problematic — it's still generally an unsolved problem in FIM.

Authorisation

- Currently dCache support **UNIX permissions** and **NFS ACLs**
  - Users have a UID & GIDs
  - Permissions decided by ownership of files & directory and their modes.
  - ACLs allow a more flexible description.
- For grid users, their **DN** and **FQAN**(s) define their UID and GIDs.
  - Current mapping is somewhat awkward, but work is underway to fix this.
- Many projects have roughly **similar approach**:
  - User presents group-membership token(s), which are mapped to GIDs.
- Others projects may wish to make decisions completely outside of dCache
  - One approach is for users to supply an authz token with a request
  - Another approach is to call-out for each operation (e.g., XAML)
  - Some support already exists already, but not uniform and only for “the ALICE approach”.

Boundary activities

- Data **ingest** and **egress**:
  - More than just upload and download:
    - *Trigger activity when data is uploaded (e.g., update catalogue, extract metadata)*
    - *Trigger activity when data is downloaded (e.g., redacting or "anonymising")*
  - Should these activities happen inside dCache or outside, triggered by dCache?
- User may have **non-modifiable analysis application**
  - Can't modify (no source code) or don't want to modify
  - dCache's use of standard protocols (NFS, HTTP, WebDAV, FTP)
    - *Better chance of dCache being accessible to client's application.*
    - *Get the clients for free (or almost for free?)*
- Community comes with **additional protocol requirements**
  - Can add support for a new protocol.
- Management of data
  - dCache provides **SRM** as a standard management interface,
  - **Other interfaces** provide a subset of SRM functionality.
  - Does **user concepts** match **dCache management concepts**?

Integration

- Storage is a **minimum service**
  - Often, seen as some “hidden” back-end to higher-level functionality
- How much **functionality** should be in dCache?
  - Storing user-supplied metadata
    - *As RDF triple-store? With SPARQL end-point? With a reasoner? What complexity class?*
  - Persistent Identifiers?
  - How flexible should dCache be internally?
    - *Should it embed a domain-specific language triggered by activity within dCache?*
- Need to provide sufficient “hooks” to allow **easy integration** with higher-level services
  - What dCache activity should trigger these hooks?
  - Work on this already started within EMI
  - How should the reverse interaction (external systems triggering dCache activity) look like?

Summary

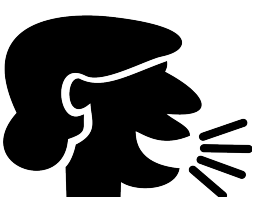
- Presented examples of **non-HEP communities** with strong data requirements
- Although dCache **is** being used by non-HEP users, there are points the **hinder their adopting** dCache
- We are working on these points, allowing people to better use dCache.

Thanks for listening ...

… and my thanks to **Ron Trompert** and **Shaun de-Witt** for their help and input.

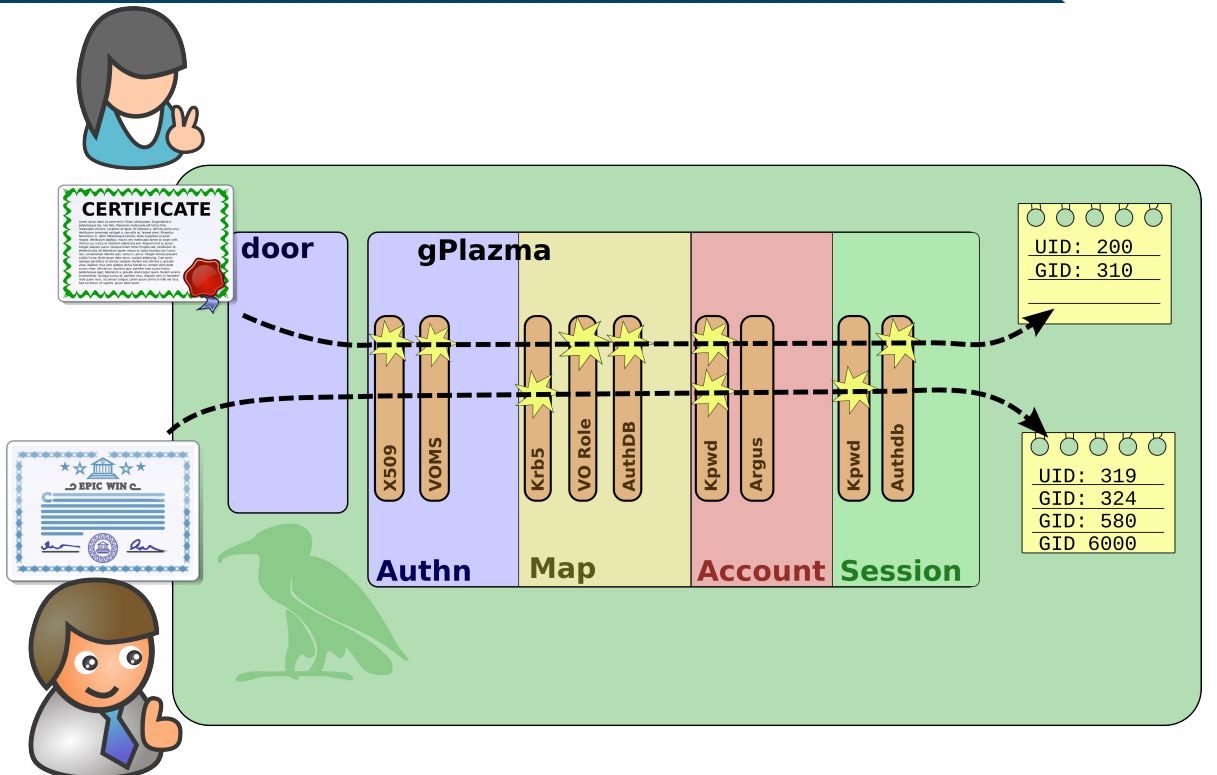
Questions?
Discussion?

www.dCache.org



Backup slides

gPlazma: new


CERTIFICATE
door
gPlazma
X509
VOMS
Krb5
VO Role
AuthDB
Kpwd
Argus
Kpwd
Authdb
Authn
Map
Account
Session
UID: 200
GID: 310
UID: 319
GID: 324
GID: 580
GID 6000
EPIC WIN

HTTP / WebDAV

- How do we support **non-HEP users**?
- dcap, SRM, rfio, xrootd
  - Nobody outside HEP has heard of these
- **HTTP** & **WebDAV**
  - Everyone has a web-browser
  - WebDAV is commonly available on platforms
  - Used by Microsoft's SkyDrive service
- Deployed **in production**: DESY, PIC, BNL, …

NFS v4.1 / pNFS

- Industry standard protocol:

  It is available **NOW**:

  RHEL/SL 6.x, Fedoria, Debian („Wheezy“), Ubuntu, Windows, Solaris, ...

- In **production** (at DESY) for over a year
- Fermi REX dept. evaluated dCache NFSv4.1 for Fermilab Intensity Frontier:

  „Results look promising, throughput scales well with number of pool nodes“

- Authn: trusted-host or Kerberos

NFS 4.1 with X.509

- HEP uses **X.509 client certificates** for authn and authz decisions.

  (everyone else is using Kerberos)

- NFS 4.1 doesn't support this, currently

  Linux has pluggable authn, so this is fix-able.

- Support need for HEP jobs to use NFS.
- **Collaborating with CERN/DPM** to solve this