dCache and Tokens

Bernardino Spisso

INFN
NAPOLI

Overview

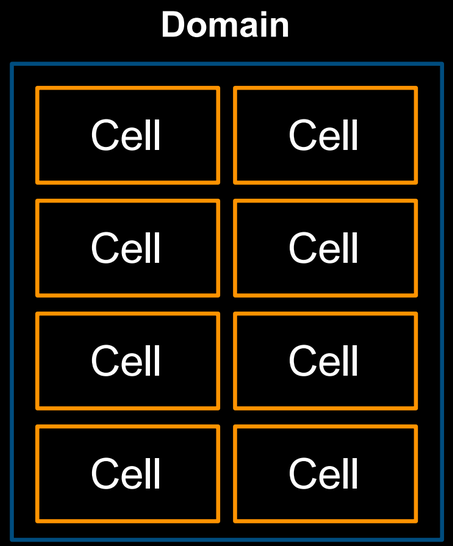
1. Getting to know dCache
2. OIDC Tokens
3. Data handling with token

dCache

Architecture I - Basic Blocks

dCache basic architectural element

- Performs specific tasks;
- May rely on other cells;
- High level tasks are obtained through the interaction between multiple cells



dCache high level element

- Runs a Java Virtual Machine instance equipped to run cells
- Can comunicate with other Domains through TCP;
- Does not share the JVM instance with other domains

dCache

Architecture II - Cells

Zookeeper

a distributed directory and coordination system. Allows domains to communicate and coordinate with each other

Pnfsmanager

manages the pNFS file system, the pNFS database and the metadata

Cleaner

periodically removes deleted file replica from the pools

Poolmanager

heart of the system. Handles read / write requests and manages the transfer of files between the user and the system

Spacemanger

Handles free space reservation. Relies upon Poolmanager and SRM

Pinmanager

Ensures the presence of file replica on disk. It is used to ensure that a certain number of replica are available

Billing

Built-in monitor system which provides an overview of activity and performance of doors and pools

GPlazma

authorisation and authentication. Credentials are collected by doors and sent to gplazma for authentication

Admin

Provides admin shell services that allow interaction with all cells within the dCache system

SRMmanager

The Storage Resource Manager, handles dynamic space allocation and file management

Transfermanager

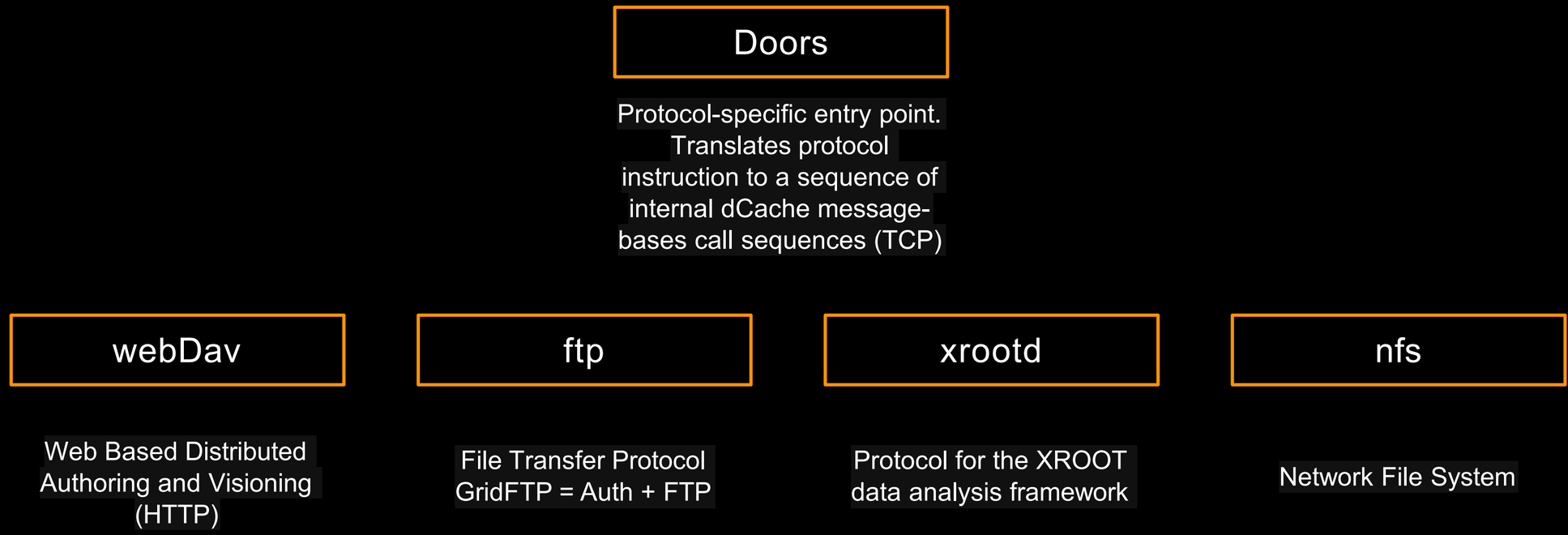
Handles 3rd party copy transfers initiated by SRM or WebDav

Pool

Handles data storage, performs checksums and data migrations

dCache

Architecture III - Communication and Transfers Cells



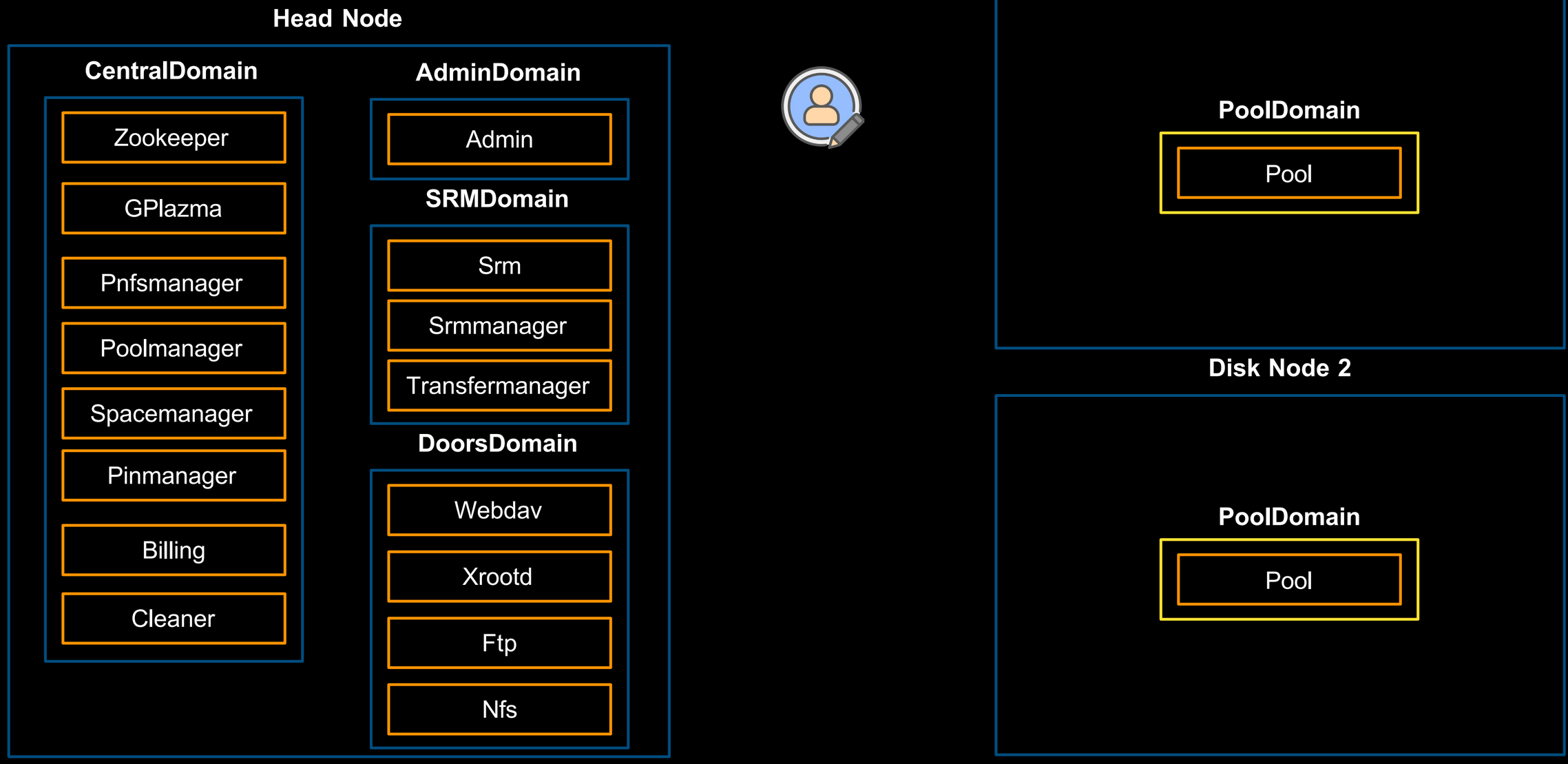
dCache

Architecture IV - Other Components

- Frontend -> Web page with monitoring services
- Namespace -> PostgreSQL server managing several databases:
  1. chimera
  2. space manger
  3. pin manager
  4. srm

dCache

Architecture V - Building Domains



dCache

Architecture VII - Layout Files, i.e. building domains

```
[centralDomain]
dcache.broker.scheme = core
[centralDomain/zookeeper]
[centralDomain/pnfsmanager]
[centralDomain/cleaner]
[centralDomain/poolmanager]
[centralDomain/spacemanager]
[centralDomain/pinmanager]
[centralDomain/billing]
[centralDomain/gplazma]
```

All Core cells are in the central Domain

dCache

Architecture VII - Layout Files, i.e. building domains

```
# Disknode dCache layout configuration
# ====================================

# disknodes needs to know how to contact central node
dcache.zookeeper.connection = t2-dpm-dome.na.infn.it:2181
```

Zookeeper connection

```
# project staticPool on filesystem /data/t2-disk01-static
[poolsDomain_${host.name}_staticPool]
[poolsDomain_${host.name}_staticPool/pool]
pool.name=staticPool_008
pool.tags=hostname=${host.name} poolgroup=staticPool
pool.path=/data/t2-disk01-static/dcache/staticPool_008
```

Pool Domains names must be unique
Pool Group is used for space reservation and authentication

dCache

Architecture VI - Authentication and Authorisation

The gPlazma interface (Grid-aware Pluggable AuthoriZation Management) allows to build complex authorisation logics through configurable plugins. Plugins are chained together in stacks.

```
# location:  gplazma.configuration.file=/etc/dcache/gplazma.conf

auth     optional     x509
auth     optional     voms
auth     optional     oidc

map      optional     multimap   gplazma.multimap.file=/etc/dcache/multi-mapfile.group
map      optional     vogroup    vo-group-path=/etc/dcache/vo-group.json
map      sufficient   multimap   gplazma.multimap.file=/etc/dcache/multi-mapfile.user
map      optional     vogroup    vo-group-path=/etc/dcache/vo-user.json
map      sufficient   multimap   gplazma.multimap.file=/etc/dcache/multi-mapfile.vo
map      sufficient   multimap   gplazma.multimap.file=/etc/dcache/multi-mapfile.unmapped

session  requisite  roles
session  sufficient omnisession
```

Auth plugins are used to read the user public and private credentials and ask some authority if those are valid to access the system

Map plugins map the user information obtained in the auth to UID and GIDs

Session plugins enrich the session with additional attributes
Roles: add the admin gid for any user who should have this capacity
Omnisession: provides session information to the user (home directory, root directory, etc)

dCache

Architecture VII - Authentication and Authorisation (The Stack)

Optional: the success or failure of this plug-in is only important if it is the only plug-in in the stack associated with this type

Required: Failure is fatal but only after all the other plugins in the stack have been invoked

Sufficient: success of this plug-in is enough to satisfy the authentication requirements. Failure is not fatal

Requisite: like required, but in this case control is directly returned to the door

```
# location: gplazma.configuration.file=/etc/dcache/gplazma.conf

auth     optional     x509
auth     optional     voms
auth     optional     oidc

map      optional     multimap   gplazma.multimap.file=/etc/dcache/multi-mapfile.group
map      optional     vogroup    vo-group-path=/etc/dcache/vo-group.json
map      sufficient   multimap   gplazma.multimap.file=/etc/dcache/multi-mapfile.user
map      optional     vogroup    vo-group-path=/etc/dcache/vo-user.json
map      sufficient   multimap   gplazma.multimap.file=/etc/dcache/multi-mapfile.vo
map      sufficient   multimap   gplazma.multimap.file=/etc/dcache/multi-mapfile.unmapped

session  requisite  roles
session  sufficient omnisession
```

X509 extract the X.509 certificate chains from the credentials of a user (certificate stored in: /etc/grid-security/certificates)

voms takes x509 certificates and tests them against trusted CAs. Verified certificates are passed along in the stack (voms are stored in: /etc/grid-security/vomsdir/)

Multimap dCache requires that authenticated credentials are mapped to posix style **username**, **uid** and **gid.** Records of the mappings are keeper in the gplazma.multimap.file. Multiple files can be configured to organise complex relations. Files are located in: /etc/dcache/

dCache

Architecture VIII - multi-map file

```
# multi-map file.group
#  ==================================

# ATLAS mapping of client groups/roles to gid numbers
fqan:/atlas                              gid:1000  group:writer
group:atlas_oidc                         gid:1000, true group:writer
username:atlas_oidc
oidcgrp:/atlas                           gid:1000
fqan:/atlas/Role=production              gid:1001
oidcgrp:/atlas/production                gid:1001
fqan:/atlas/usatlas/Role=production      gid:1002
oidcgrp:/atlas/usatlas/production        gid:1002

# BELLE mapping of client groups/roles to gid numbers
fqan:/belle                  gid:2000 group:writer
group:belle_oidc             gid:2000,true group:writer username:belle_oidc
oidcgrp:/belle               gid:2000

# DTEAM mapping of client groups/roles to gid numbers
fqan:/dteam                  gid:3000 group:writer
group:dteam_oidc             gid:3000,true group:writer username:dteam_oidc
oidcgrp:/dteam               gid:3000

# ESCAPE mapping of client groups/roles to gid numbers
fqan:/escape                 gid:4000 group:writer
group:escape_oidc            gid:4000,true group:writer username:escape_oidc
oidcgrp:/escape              gid:4000

# OPS mapping of client groups/roles to gid numbers
fqan:/ops                    gid:5000 group:writer
group:ops_oidc               gid:5000,true group:writer username:ops_oidc
oidcgrp:/ops                 gid:5000

# WLCG mapping of client groups/roles to gid numbers
fqan:/wlcg                   gid:6000 group:writer
group:wlcg_oidc              gid:6000,true group:writer username:wlcg_oidc
oidcgrp:/wlcg                gid:6000
```

```
# multi-map file.user
#  ==================================

"dn:/C=IT/O=INFN/OU=Personal Certificate/L=Napoli/CN=Alessandra Doria" username:alessandra_doria uid:11000
"dn:/C=UK/O=eScience/OU=Manchester/L=HEP/CN=james collinson" username:james_collinson uid:11001
"dn:/C=UK/O=eScience/OU=Manchester/L=HEP/CN=robert barnsley" username:robert_barnsley uid:11002
"dn:/C=UK/O=eScience/OU=Manchester/L=HEP/CN=rohini joshi" username:rohini_joshi uid:11003
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=afkiaras/CN=817926/CN=Aristeidis Fkiaras" username:afkiaras uid:11004
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=arturos/CN=679537/CN=Arturos Sanchez Pineda" username:arturos uid:11005
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=avendrel/CN=849027/CN=Alba Vendrell Moya" username:avendrel uid:11006
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=ddmadmin/CN=531497/CN=Robot: ATLAS Data Management" username:ddmadmin uid:11007
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=ewp2c01/CN=531497/CN=Robot: ESCAPE WP2 CERN 01" username:ewp_c___ uid:11008
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=ewp2c01/CN=817926/CN=Robot: ESCAPE WP2 CERN 01" username:ewp_c____1 uid:11009
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=jhover/CN=653174/CN=John Raymond Hover" username:jhover uid:11010
"dn:/DC=org/DC=terena/DC=tcs/C=ES/O=Port dInformacio Cientifica/CN=Agustin Bruzzese bruzzese@pic.es" username:agustin_bruzzese_bruzzese_pic_es uid:11011
"dn:/DC=org/DC=terena/DC=tcs/C=IT/L=Frascati/O=Istituto Nazionale di Fisica Nucleare/OU=Cloud/CN=escape-monitoring.cloud.cnaf.infn.it"
username:escape_monitoring_cloud_cnaf_infn_it uid:11012
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare - INFN/CN=Federica Agostini fagostin@infn.it" username:federica_agostini_fagostin_infn_it
uid:11013
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare - INFN/CN=Robot - Andrea Ceccanti aceccant@infn.it"
username:robot___andrea_ceccanti_aceccant_infn_it uid:11014
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Alessandra Doria adoria@infn.it" username:alessandra_doria_adoria_infn_it uid:11015
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Bernardino Spisso spisso@infn.it" username:bernardino_spisso_spisso_infn_it uid:11016
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Davide Michelino dmichelino@infn.it" username:davide_michelino_dmichelino_infn_it
uid:11017
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Elisabetta Vilucchi evilucch@infn.it" username:elisabetta_vilucchi_evilucch_infn_it uid:11018
"dn:/O=GRID-FR/C=FR/O=CNRS/OU=LAPP/CN=Stephane Jezequel" username:stephane_jezequel uid:11019
```

```
# Omnisession plugin (omnisession) configuration
# ========================================
#
# where "group:writer" have privileges to the whole namespace should be
# sufficient (fine grained permissions comes from user/group/ACLs).

group:writer         root:/ home:/
```

- OpenID Connect (OIDC): built on top of OAuth 2.0
- @OP style suffix for oidcgrp fields

dCache

Architecture VIII - multi-map file

```
# multi-mapfile.vo
# ======================================
#
# example for DTEAM:
#fqan:/dteam                    uid:1000 username:dteam
#username:dteam_oidc            uid:1001

# generated from imported config file

# ATLAS mapping of username resolved by vogroup plugin to uid
username:atlas                  uid:1000
#username:atlas_oidc            uid:1000
username:atlas_production       uid:1001
username:atlas_usatlas_production  uid:1002

# BELLE mapping of username resolved by vogroup plugin to uid
username:belle                  uid:2000
#username:belle_oidc            uid:2000

# DTEAM mapping of username resolved by vogroup plugin to uid
username:dteam                  uid:3000
#username:dteam_oidc            uid:3000

# ESCAPE mapping of username resolved by vogroup plugin to uid
username:escape                 uid:4000
#username:escape_oidc           uid:4000

# OPS mapping of username resolved by vogroup plugin to uid
username:ops                    uid:5000
#username:ops_oidc              uid:5000

# WLCG mapping of username resolved by vogroup plugin to uid
username:wlcg                   uid:6000
#username:wlcg_oidc             uid:6000
```

```
# multi-mapfile.unmapped
# ======================================
#
# example for DTEAM mapping for groups and roles with no explicit configuration
# (remove to grant access only to explicitly configured VO FQAN):
#fqan:/dteam                    uid:1000 gid:1000,true username:dteam

# generated from imported config file

# ATLAS default mapping for groups and roles with no explicit configuration
fqan:/atlas                     uid:1000 gid:1000,true username:atlas

# BELLE default mapping for groups and roles with no explicit configuration
fqan:/belle                     uid:2000 gid:2000,true username:belle

# DTEAM default mapping for groups and roles with no explicit configuration
fqan:/dteam                     uid:3000 gid:3000,true username:dteam

# ESCAPE default mapping for groups and roles with no explicit configuration
fqan:/escape                    uid:4000 gid:4000,true username:escape

# OPS default mapping for groups and roles with no explicit configuration
fqan:/ops                       uid:5000 gid:5000,true username:ops

# WLCG default mapping for groups and roles with no explicit configuration
fqan:/wlcg                      uid:6000 gid:6000,true username:wlcg
~
```

```
# vo-group.json

[
  {
    "fqan":  "/atlas",
    "mapped_gid":  1000
  },
  {
    "fqan":  "/atlas/Role=production",
    "mapped_gid":  1001
  },
  {
    "fqan":  "/atlas/usatlas/Role=production",
    "mapped_gid":  1002
  },
  {
    "fqan":  "/belle",
    "mapped_gid":  2000
  },
  {
    "fqan":  "/dteam",
    "mapped_gid":  3000
  },
  {
    "fqan":  "/escape",
    "mapped_gid":  4000
  },
  {
    "fqan":  "/ops",
    "mapped_gid":  5000
  },
  {
    "fqan":  "/wlcg",
    "mapped_gid":  6000
  }
]
```

```
# vo-user.json

[
  {
    "fqan":  "/atlas",
    "mapped_gid":  1000,
    "mapped_uname":  "atlas"
  },
  {
    "fqan":  "/atlas/Role=production",
    "mapped_gid":  1001,
    "mapped_uname":  "atlas_production"
  },
  {
    "fqan":  "/atlas/usatlas/Role=production",
    "mapped_gid":  1002,
    "mapped_uname":  "atlas_usatlas_production"
  },
  {
    "fqan":  "/belle",
    "mapped_gid":  2000,
    "mapped_uname":  "belle"
  },
  {
    "fqan":  "/dteam",
    "mapped_gid":  3000,
    "mapped_uname":  "dteam"
  },
  {
    "fqan":  "/escape",
    "mapped_gid":  4000,
    "mapped_uname":  "escape"
  },
  {
    "fqan":  "/ops",
    "mapped_gid":  5000,
    "mapped_uname":  "ops"
  },
  {
    "fqan":  "/wlcg",
    "mapped_gid":  6000,
    "mapped_uname":  "wlcg"
  }
]
```

dCache

Architecture VIII - multi-map file

```
# multi-map file.group
#  ==================================

# ATLAS mapping of client groups/roles to gid numbers
fqan:/atlas                                gid:1000  group:writer
group:atlas_oidc                           gid:1000, true group:writer
username:atlas_oidc
oidcgrp:/atlas                             gid:1000
fqan:/atlas/Role=production                gid:1001
oidcgrp:/atlas/production                  gid:1001
fqan:/atlas/usatlas/Role=production        gid:1002
oidcgrp:/atlas/usatlas/production          gid:1002

# BELLE mapping of client groups/roles to gid numbers
fqan:/belle                  gid:2000 group:writer
group:belle_oidc             gid:2000,true group:writer username:belle_oidc
oidcgrp:/belle               gid:2000

# DTEAM mapping of client groups/roles to gid numbers
fqan:/dteam                  gid:3000 group:writer
group:dteam_oidc             gid:3000,true group:writer username:dteam_oidc
oidcgrp:/dteam               gid:3000

# ESCAPE mapping of client groups/roles to gid numbers
fqan:/escape                 gid:4000 group:writer
group:escape_oidc            gid:4000,true group:writer username:escape_oidc
oidcgrp:/escape              gid:4000

# OPS mapping of client groups/roles to gid numbers
fqan:/ops                    gid:5000 group:writer
group:ops_oidc               gid:5000,true group:writer username:ops_oidc
oidcgrp:/ops                 gid:5000

# WLCG mapping of client groups/roles to gid numbers
fqan:/wlcg                   gid:6000 group:writer
group:wlcg_oidc              gid:6000,true group:writer username:wlcg_oidc
oidcgrp:/wlcg                gid:6000
```

```
# multi-map file.user
#  ==================================

"dn:/C=IT/O=INFN/OU=Personal Certificate/L=Napoli/CN=Alessandra Doria" username:alessandra_doria uid:11000
"dn:/C=UK/O=eScience/OU=Manchester/L=HEP/CN=james collinson" username:james_collinson uid:11001
"dn:/C=UK/O=eScience/OU=Manchester/L=HEP/CN=robert barnsley" username:robert_barnsley uid:11002
"dn:/C=UK/O=eScience/OU=Manchester/L=HEP/CN=rohini joshi" username:rohini_joshi uid:11003
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=afkiaras/CN=817926/CN=Aristeidis Fkiaras" username:afkiaras uid:11004
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=arturos/CN=679537/CN=Arturos Sanchez Pineda" username:arturos uid:11005
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=avendrel/CN=849027/CN=Alba Vendrell Moya" username:avendrel uid:11006
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=ddmadmin/CN=531497/CN=Robot: ATLAS Data Management" username:ddmadmin uid:11007
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=ewp2c01/CN=531497/CN=Robot: ESCAPE WP2 CERN 01" username:ewp_c___ uid:11008
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=ewp2c01/CN=817926/CN=Robot: ESCAPE WP2 CERN 01" username:ewp_c____1 uid:11009
"dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=jhover/CN=653174/CN=John Raymond Hover" username:jhover uid:11010
"dn:/DC=org/DC=terena/DC=tcs/C=ES/O=Port dInformacio Cientifica/CN=Agustin Bruzzese bruzzese@pic.es" username:agustin_bruzzese_bruzzese_pic_es uid:11011
"dn:/DC=org/DC=terena/DC=tcs/C=IT/L=Frascati/O=Istituto Nazionale di Fisica Nucleare/OU=Cloud/CN=escape-monitoring.cloud.cnaf.infn.it"
username:escape_monitoring_cloud_cnaf_infn_it uid:11012
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare - INFN/CN=Federica Agostini fagostin@infn.it" username:federica_agostini_fagostin_infn_it
uid:11013
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare - INFN/CN=Robot - Andrea Ceccanti aceccant@infn.it"
username:robot___andrea_ceccanti_aceccant_infn_it uid:11014
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Alessandra Doria adoria@infn.it" username:alessandra_doria_adoria_infn_it uid:11015
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Bernardino Spisso spisso@infn.it" username:bernardino_spisso_spisso_infn_it uid:11016
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Davide Michelino dmichelino@infn.it" username:davide_michelino_dmichelino_infn_it
uid:11017
"dn:/DC=org/DC=terena/DC=tcs/C=IT/O=Istituto Nazionale di Fisica Nucleare/CN=Elisabetta Vilucchi evilucch@infn.it" username:elisabetta_vilucchi_evilucch_infn_it uid:11018
"dn:/O=GRID-FR/C=FR/O=CNRS/OU=LAPP/CN=Stephane Jezequel" username:stephane_jezequel uid:11019
```

```
# Omnisession plugin (omnisession) configuration
# ========================================
#
# where "group:writer" have privileges to the whole namespace should be
# sufficient (fine grained permissions comes from user/group/ACLs).

group:writer          root:/ home:/
```

- OpenID Connect (OIDC): built on top of OAuth 2.0
- @OP style suffix for oidcgrp fields

dCache

Architecture VIII - multi-map file

```
# multi-map file.group
#   ==================================

# ATLAS mapping of client groups/roles to gid numbers
fqan:/atlas                                gid:1000   group:writer
group:atlas_oidc                           gid:1000, true group:writer
username:atlas_oidc
oidcgrp:/atlas                             gid:1000
fqan:/atlas/Role=production                gid:1001
oidcgrp:/atlas/production                  gid:1001
fqan:/atlas/usatlas/Role=production        gid:1002
oidcgrp:/atlas/usatlas/production          gid:1002
```

```
# multi-map file.user
#   ==================================

"dn:/C=IT/O=INFN/OU=Personal Certificate/L=Napoli/CN=Alessandra Doria"
username:alessandra_doria uid:11000
```

```
# Omnisession plugin (omnisession) configuration
# ========================================
#
# where "group:writer" have privileges to the whole namespace should be
# sufficient (fine grained permissions comes from user/group/ACLs).

group:writer          root:/ home:/
```

Predicates:

- **Dn**: matches user's Distinguished Name, obtained by authenticating via X509;
- **Group**: matches user's group;
- **Dfqan**: matches user's VOMS FQAN;
- **Oidc**: matches an OpenID-Connect subject claim from a specific OP;
- **Oidcgrp**: matches an OpenID-Connect groups claim value;
- **Username**: matches the username of a user

User principals may match multiple line in the configuration. Attributes override each other, first win.

Fully Qualified Authorities, Groups, Username and odcgrp are associated to gid

dCache

Architecture IX - PoolManager Service

The heart of a dCache System is the poolmanager. When a user performs an action on a file - reading or writing - a transfer request is sent to the dCache system. The poolmanager then decides how to handle this request

PSU (Pool Selection Unit): responsible for finding the set of pools which can be used for a transfer request.

- We can adjust the transfer conditions by telling the PSU which pools are permitted for which type of file requests.
  The PSU generates a list of allowed storage pools for each request, which are then dynamically used by the PoolManager.

  Link: consists of a set of unit groups and a list of pools. Each link contains one or more unit groups (conditions).
  Each group contains several units and the unit group is matched if at least one unit within the group is matched.
  All group units have to be matched by the transfer request to initiate a transfer.

  - Network (-net) — IP address and a net mask. It is satisfied if the request is coming from an IP address with the subnet given by the address/netmask pair
  - Protocol (-protocol) — Name of the protocol and version number. It is satisfied if the request is coming from the correct protocol
  - Storage Class (-store) — Storage Class. It is satisfied if the request file has this storage class
  - Cache Class (-dcache) — Cache Class. It is satisfied if the cache class of the requested file agrees with it

Type of Transfer: four possible attributes (-readpref, -writepref, -p2ppref, -cachepref). A value of zero disables this type of transfer, any positive assert preference.
A negative value for -p2ppref sets it equal to -readpref.

Pool Groups: (pgroup) pools can be grouped together in pool groups. If the -dynamic parameter is used, pools are added dynamically on the basis of a pool tag

Link Groups: (linkGroup) are used by the SRM SpaceManager to make reservations against space. A link group is a collection of links and each link pointing to the associated pool groups. Each link group knows about the size of its available space (SUM of pools). Has five boolean properties
(replicaAllowed, outputAllowed, custodiaAllowed, onlineAllowed, nearlineAllowed)

dCache

Architecture IX - PoolManager Service

```
# PoolManager configuration in /var/lib/dcache/config/poolmanager.conf
#  =====================================
#
psu create unit -store *@*
psu create unit -net 0.0.0.0/0.0.0.0
psu create unit -net ::/0
psu create unit -protocol */*

psu create ugroup any-protocol
psu addto ugroup any-protocol */*

psu create ugroup any-store
psu addto ugroup any-store *@*

psu create ugroup world-net
psu addto ugroup world-net 0.0.0.0/0.0.0.0
psu addto ugroup world-net ::/0

psu create pgroup spacemanager_poolGroup_StaticEscape -dynamic -tags=poolgroup=StaticEscape
psu create pgroup spacemanager_poolGroup_lnf-volatile -dynamic -tags=poolgroup=lnf-volatile
psu create pgroup spacemanager_poolGroup_roma1-volatile -dynamic -tags=poolgroup=roma1-volatile
psu create pgroup spacemanager_poolGroup_staticPool -dynamic -tags=poolgroup=staticPool

psu create link default-link any-protocol any-store world-net
psu set link default-link -readpref=10 -writepref=10 -cachepref=10 -p2ppref=-1

psu create link spacemanager_link_StaticEscape any-protocol any-store world-net
psu set link spacemanager_link_StaticEscape -readpref=10 -writepref=10 -cachepref=0 -p2ppref=-1
psu addto link spacemanager_link_StaticEscape spacemanager_poolGroup_StaticEscape

psu create link spacemanager_link_lnf-volatile any-protocol any-store world-net
psu set link spacemanager_link_lnf-volatile -readpref=10 -writepref=10 -cachepref=0 -p2ppref=-1
psu addto link spacemanager_link_lnf-volatile spacemanager_poolGroup_lnf-volatile

psu create link spacemanager_link_roma1-volatile any-protocol any-store world-net
psu set link spacemanager_link_roma1-volatile -readpref=10 -writepref=10 -cachepref=0 -p2ppref=-1
psu addto link spacemanager_link_roma1-volatile spacemanager_poolGroup_roma1-volatile

psu create link spacemanager_link_staticPool any-protocol any-store world-net
psu set link spacemanager_link_staticPool -readpref=10 -writepref=10 -cachepref=0 -p2ppref=-1
psu addto link spacemanager_link_staticPool spacemanager_poolGroup_staticPool
```

Units

unit groups

pool groups

Links

```
psu create linkGroup spacemanager_linkGroup_StaticEscape
psu set linkGroup custodialAllowed spacemanager_linkGroup_StaticEscape true
psu set linkGroup replicaAllowed spacemanager_linkGroup_StaticEscape true
psu set linkGroup nearlineAllowed spacemanager_linkGroup_StaticEscape true
psu set linkGroup outputAllowed spacemanager_linkGroup_StaticEscape true
psu set linkGroup onlineAllowed spacemanager_linkGroup_StaticEscape true
psu addto linkGroup spacemanager_linkGroup_StaticEscape spacemanager_link_StaticEscape

psu create linkGroup spacemanager_linkGroup_lnf-volatile
psu set linkGroup custodialAllowed spacemanager_linkGroup_lnf-volatile true
psu set linkGroup replicaAllowed spacemanager_linkGroup_lnf-volatile true
psu set linkGroup nearlineAllowed spacemanager_linkGroup_lnf-volatile true
psu set linkGroup outputAllowed spacemanager_linkGroup_lnf-volatile true
psu set linkGroup onlineAllowed spacemanager_linkGroup_lnf-volatile true
psu addto linkGroup spacemanager_linkGroup_lnf-volatile spacemanager_link_lnf-volatile

psu create linkGroup spacemanager_linkGroup_roma1-volatile
psu set linkGroup custodialAllowed spacemanager_linkGroup_roma1-volatile true
psu set linkGroup replicaAllowed spacemanager_linkGroup_roma1-volatile true
psu set linkGroup nearlineAllowed spacemanager_linkGroup_roma1-volatile true
psu set linkGroup outputAllowed spacemanager_linkGroup_roma1-volatile true
psu set linkGroup onlineAllowed spacemanager_linkGroup_roma1-volatile true
psu addto linkGroup spacemanager_linkGroup_roma1-volatile spacemanager_link_roma1-volatile

psu create linkGroup spacemanager_linkGroup_staticPool
psu set linkGroup custodialAllowed spacemanager_linkGroup_staticPool true
psu set linkGroup replicaAllowed spacemanager_linkGroup_staticPool true
psu set linkGroup nearlineAllowed spacemanager_linkGroup_staticPool true
psu set linkGroup outputAllowed spacemanager_linkGroup_staticPool true
psu set linkGroup onlineAllowed spacemanager_linkGroup_staticPool true
psu addto linkGroup spacemanager_linkGroup_staticPool spacemanager_link_staticPool
```

Link Groups

dCache

Architecture X - Space Reservation

Space reservation guarantees the the request amount of storage space type is made available by the storage system.

- Retention Policy: describes the quality of the storage service
  - OUTPUT: output files are alowed
  - REPLICA: lower quality, only one copy is stored
  - CUSTODIAL: higher quality, storage on TAPE
- Access Latency: describes the data availability
  - NEARLINE: data is allowed to migrate to permanent media. Retrieving data may result in delays due to the transfer from permanent media
  - ONLINE: data is readily available allowing for faster access (guaranties a copy on disk)

```
psu create pgroup spacemanager_poolGroup_StaticEscape -dynamic -tags=poolgroup=StaticEscape

psu create link spacemanager_link_StaticEscape any-protocol any-store world-net
psu set link spacemanager_link_StaticEscape -readpref=10 -writepref=10 -cachepref=0 -p2ppref=-1
psu addto link spacemanager_link_StaticEscape spacemanager_poolGroup_StaticEscape

psu create linkGroup spacemanager_linkGroup_StaticEscape
psu set linkGroup custodialAllowed spacemanager_linkGroup_StaticEscape true
psu set linkGroup replicaAllowed spacemanager_linkGroup_StaticEscape true
psu set linkGroup nearlineAllowed spacemanager_linkGroup_StaticEscape true
psu set linkGroup outputAllowed spacemanager_linkGroup_StaticEscape true
psu set linkGroup onlineAllowed spacemanager_linkGroup_StaticEscape true
psu addto linkGroup spacemanager_linkGroup_StaticEscape spacemanager_link_StaticEscape
```

```
# perform space reservation
ssh -p 22224 -l admin localhost
\c SrmSpaceManager
reserve space -owner=/atlas/Role=production -desc=atlas-static-escape
              -lg=spacemanager_linkGroup_staticEscape 5TB
```

```
# SpaceManagerLinkGroupAuthorizationFile
#  ===================================
# location: spacemanager.authz.link-group-file-name=/etc/dcache/LinkGroupAuthorization.conf
#
# example
#LinkGroup spacemanager_linkGroup
#*/Role=*

spacemanager_linkGroup_StaticEscape
/atlas/Role=*
/escape/Role=*

spacemanager_linkGroup_lnf-volatile
/atlas/Role=*

spacemanager_linkGroup_roma1-volatile
/atlas/Role=*

spacemanager_linkGroup_staticPool
/escape/Role=*
/belle/Role=*
/atlas/Role=*
```

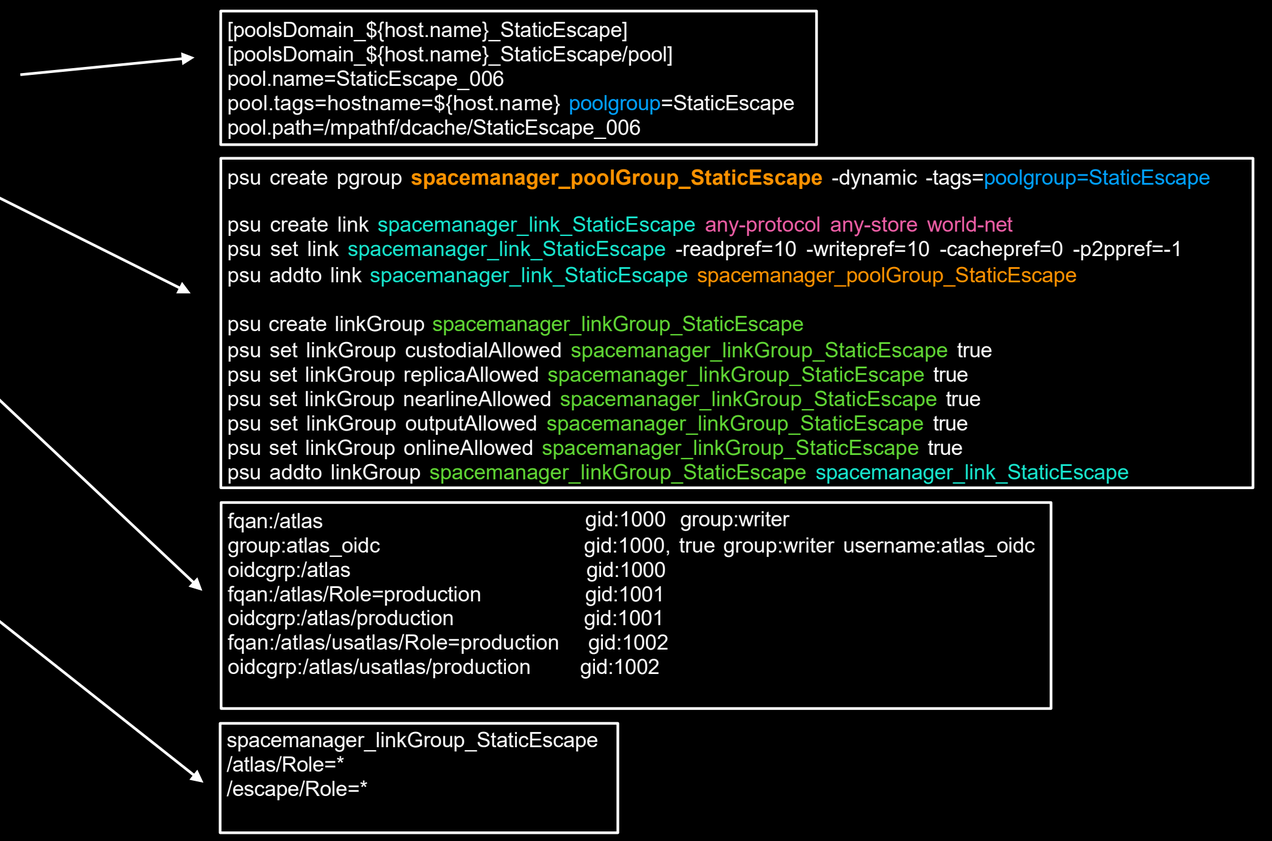
Coming back to authorisation

We are allowing users with a specific role to make reservations on given linkgroups, and this will have dedicate storage pools

dCache

Wrapping Up Authorisation and Space and Pool Reservation

1. We have created in the layout file at least a Pool with a given pool tag
2. We have created the linkGroup in the poolmanager.conf with the associated pool tag
3. We have associated gids to a given VO in the multi-map file
4. We have given to the VO space reservation permissions in the SpaceManagerLinkGroupAuthorizationFile and reserved some space through the admin interface



dCache

Architecture XI - Configuration

```
# dCache configuration in /etc/dcache/dcache.conf
# =====================================
#
# If you have a big pool or many pools per domain, you probably want more
# memory in the java process than the default 512m. 1024m ought to be enough
# for a single 10TiB pool, but you probably need 2048m for a 20TiB pool. As a rule
# of thumb, use 512m + 512m for every 10 TiB of pool (rounded to a nice value).

# Tape write pools are usually not big and you can use 1024m. A large file system
# cache is more useful on such pools.dcache.java.memory.heap = 4096m

# The default is 512m, but in particular with xrootd this isn't quite enough.
#
# There is no reason to scale this by pool size - how much is required depends more on
# access patterns. For Alice tape write pools this MUST be at least 2048m.
dcache.java.memory.direct = 2048m

# dCache services configuration
dcache.layout = layout-${host.fqdn}

# database configuration
#dcache.db.host = localhost
#dcache.db.user = dcache
#dcache.db.password =

# Diskonly storage
#       https://dcache.org/old/manuals/Book-7.2/config-SRM.shtml#utilization-of-space-reservations-for-data-storage
dcache.default-retention-policy=REPLICA
dcache.default-access-latency=ONLINE

# enable ACL support
pnfsmanager.enable.acl = true

# Explicit port configuration for LAN/WAN access with GridFTP and HTTP protocol
dcache.net.wan.port.min = 20000
dcache.net.wan.port.max = 25000
# LAN port range for internal pool to pool communication
dcache.net.lan.port.min = 33115
dcache.net.lan.port.max = 33145
# use same ports for all protocols (GridFTP, WebDAV, xroot)
pool.mover.xrootd.port.min = ${dcache.net.wan.port.min}
pool.mover.xrootd.port.max = ${dcache.net.wan.port.max}

# Allow Let's encrypt certificates that doesn't provide CRLs
#dcache.authn.hostcert.verify=true
#dcache.authn.crl-mode = IF_VALID

# BDII Glue Info Provider
# Besides these two basic configuration option it's necessary to update also
# /etc/dcache/info-provider.xml and install bdii packages for details see
#     https://dcache.org/old/manuals/Book-7.2/config-info-provider.shtml
info-provider.site-unique-id=GOCDB_SITE_NAME
info-provider.se-unique-id=dcache.example.com

# EGI StAR (APEL Storage Accounting)
#star.gid-mapping = 2000=/atlas, 1010=/dteam, 1020=/wlcg

# By default, dCache auto initializes pool directories. To avoid
# accidentally initializing a pool in a mount point, we configure all
# pools to wait for the pool file system to be mounted.
pool.wait-for-files = ${pool.path}/data

# All pools are tagged by the FQDN of the host. dCache uses this to avoid
# replicating files to pools on the same host.
pool.tags = hostname=${host.fqdn}

# To be able to use xroot-tpc with EOS it is necessary to add "unix" (now enabled by default)
#pool.mover.xrootd.tpc-authn-plugins=gsi,unix
#pool.mover.xrootd.plugins=gsi,unix

# pool startup optimization
#       https://indico.desy.de/event/25462/contributions/57176/attachments/36943/46186/dcache-project-whatsnew.pdf
pool.limits.scan-threads=8
pool.plugins.meta.db!je.checkpointer.wakeupInterval = 60 s
pool.plugins.meta.db!je.checkpointer.bytesInterval = 0
pool.plugins.meta.db!je.cleaner.wakeupInterval= 0 s
pool.plugins.meta.db!je.log.fileCacheSize=1024

# Enable writing with xroot protocol (be aware that xroot protocol doesn't
# protect established connection from packet content changes on the wire
# unless you enable integrity validation or TLS)
xrootd.authz.write-paths = /
```

dCache

Shift Toward Token-Based Authentication

- Simplifies user interaction and supports web standards.
- Improves interoperability.
- dCache supports OAuth2/OIDC for modern workflows.
- OIDC is a simple identity layer on top of OAuth 2.0.
- Allows clients to verify end-user identity via Identity Provider (IdP).

dCache

Where OIDC Fits in dCache

- gPlazma module handles identity mapping and authorization.
- User → IdP → dCache

dCache

Components of Token Processing

- User logs in via IdP (e.g., WLCG IAM, Keycloak, CILogon).
- IdP issues JWT (JSON Web Token) that includes user claims.
- dCache verifies the token's signature and maps identity.

- User → IdP → dCache Doors → gPlazma → Namespace/Pool.

- The gplazma plugins provides the support for authentication using OpenID credentials and mapping a verified OpenID credential to dCache specific username, uid and gid.

dCache

gPlazma Plugin Stack

- gPlazma supports a plugin-based stack for authN/Z.
- Common OIDC config in /etc/dcache/gplazma.conf:

```
auth optional oidc
map optional multimap
session requisite omnisession
```

- Oidc plugin authenticates tokens.
- Multimap maps claims to local principals.
- Omnisession enforces roles and access rights.

dCache

gPlazma Plugin Stack

```
auth     optional    voms
auth     optional    oidc
auth     optional    kpwd

map      optional    multimap gplazma.multimap.file=/etc/dcache/multi-mapfile.role
map      sufficient  multimap gplazma.multimap.file=/etc/dcache/multi-mapfile.vorole
map      optional    multimap gplazma.multimap.file=/etc/dcache/multi-mapfile.group
map      sufficient  multimap gplazma.multimap.file=/etc/dcache/multi-mapfile.vo
map      sufficient  multimap gplazma.multimap.file=/etc/dcache/multi-mapfile.oidc

account  requisite  banfile

session  requisite  roles
session  sufficient omnisession
```

dCache

Registering an IdP

- In dcache.conf or in the layout file, can be defined trusted OIDC providers:

  gplazma.oidc.provider!wlcg.issuer=https://wlcg.cloud.cern.ch

- Set audience, allowed claims, and issuer validation.
- Use discovery endpoint for dynamic metadata.

dCache

Registering an IdP

[centralDomain/gplazma]

gplazma.oidc.provider!atlas_new = https://atlas-auth.cern.ch/ -profile=wlcg -prefix=/home/atlas -authz-id="uid:1000 gid:1000 username:atlas_oidc"

gplazma.oidc.provider!datacloud-tb = https://iam-aa.wp6.cloud.infn.it/ -profile=wlcg -prefix=/home/datacloud -authz-id="uid:8000 gid:8000 username:datacloud_oidc"

gplazma.oidc.provider!belle = https://iam-belle.cloud.cnaf.infn.it/ -profile=wlcg -authz-id=username:belle_oidc_with_storage_scope -prefix=/home/belle/

gplazma.oidc.audience-targets = 5ee10e60-1dd9-44dd-8f02-3b04fe1f71fc 66b64328-d8be-4e97-93c6-85cc1818c30b https://wlcg.cern.ch/jwt/v1/any https://t2-dcache-02.na.infn.it roots://t2-dcache-02.na.infn.it:1094 t2-dcache-02.na.infn.it

dCache

Mapping Users

The OAuth2 Provider (OP) key issues an “access token” that allows anyone with that token to access

cat /etc/dcache/multi-mapfile.oidc

```
op:datacloud-tb    uid:8000 gid:8000,true username:datacloud_oidc
op:poc-icsc        uid:8003 gid:8003,true username:poc-icsc_oidc
op:escape          uid:5000 gid:5000,true username:escape_oidc
op:atlas           uid:1000 gid:1000,true username:atlas_oidc
op:atlas_new       uid:1000 gid:1000,true username:atlas_oidc
```

dCache

Access rights /etc/dcache/omnisession.conf

```
#DATACLOUD
username:datacloud-tb                read-only root:/  home:/home/datacloud/
username:datacloud-tb_production       root:/ home:/home/datacloud/

username:datacloud_oidc              root:/ home:/home/datacloud/

#poc-icsc
username:poc-icsc                  read-only root:/  home:/home/poc-icsc/
username:poc-icsc_production          root:/ home:/home/poc-icsc/

username:poc-icsc_oidc               root:/ home:/home/poc-icsc/


#ATLAS
username:atlas                     root:/ home:/home/atlas
username:atlas_role                  root:/ home:/home/atlas
username:atlas_oidc                  root:/ home:/home/atlas

# fallback
group:writer        root:/ home:/
```

dCache

Admin Tools and Monitoring

- Enable logging: /var/log/dcache/<gplazma_cell_name>.log
- Use JWT debugging tools (jwt.io, curl) to inspect tokens.
- Monitor token failures, mapping errors, login attempts.

dCache

Configure an OIDC Client

An OIDC client is an application registered with an Identity Provider (IdP) that supports OpenID Connect, such as IAM, Keycloak, Auth0, Google, etc.
To Register the client with the IdP you'll need to provide:

- Client name: descriptive name
- IdP URI: e.g. https://atlas-auth.cern.ch/ or one compatible with your CLI/browser
- Scopes: openid, profile, email, and often offline_access (for refresh tokens)
- Client ID and Secret: generated by the IdP (secret might not be required for public clients)

dCache

oidc-agent

oidc-agent is a command-line utility to manage OIDC tokens, supporting various auth flows.

- Installation -> dnf install oidc-agent
- Add an account -> oidc-gen myclient

Follow the prompts to configure: client ID, issuer URL, etc.

- Start the agent -> eval `oidc-agent` -> oidc-add myclient
- Get the token -> oidc-token myclient

This returns an OIDC access token.

dCache

Use GFAL with OIDC Token

GFAL2 supports OIDC through bearer tokens. You can pass the token via environment variable or CLI.

With gfal-copy, gfal-ls, etc. set the environment variable:

```
export BEARER_TOKEN=$(oidc-token myclient)
```

Or use it in commands:

```
gfal-ls --bearer $BEARER_TOKEN https://your.storage.server/path
gfal-copy --bearer $BEARER_TOKEN file.txt https://your.storage/path/file.txt
```

Or use a .token file, save token to a file:

```
oidc-token myclient > /tmp/mytoken
```

```
gfal-copy --bearer "$(cat /tmp/mytoken)" file.txt ...
```

dCache

Security Considerations

- Token lifetime is typically short (e.g., 1 hour). After expiration, a new token is required.

- Try to not store tokens or refresh tokens in world-readable locations.
  Consider setting restrictive file permissions:

```
chmod 600 /tmp/mytoken
```