

# What's New Since 2018

Lea Morschel for dCache Team  
dCache User Workshop 2019, Madrid, Spain



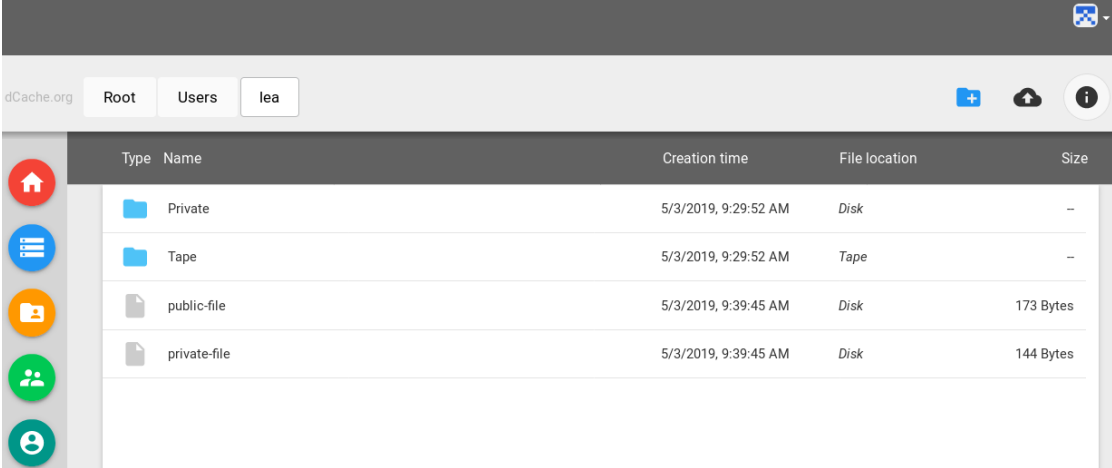
HELMHOLTZ

RESEARCH FOR  
GRAND CHALLENGES

- Official Oracle support for Java 8 ends January 2019
  - dCache 5 **can run with Java 11** (4.2 upcoming)
  - dCache 6+ exclusively on JRE 11
- New JDK licensing model from Oracle started on 16th April
  - Two editions of JVM: **OracleJDK** and **OpenJDK**
  - Using OracleJDK in production requires **commercial subscription**
  - dCache is developed and tested using OpenJDK
  - No OracleJDK subscription? Switch to OpenJDK!

- **dCache-View** replaces the old intermediate 'webadmin' interface (removed)
- Control visibility of operations exposing file metadata  
`frontend.authz.unlimited-operation-visibility = false`
- Bad authentication treated as the anonymous user:  
`dcache.enable.authn.anonymous-fallback-on-failed-login = true`
- File sharing using macaroons (or generated link, QR code)

# Frontend – Example



The screenshot displays the dCache.org web interface. At the top, there are navigation tabs for 'Root', 'Users', and 'lea'. Below the tabs is a table listing files and folders. The table has columns for Type, Name, Creation time, File location, and Size. On the left side, there is a vertical sidebar with five circular icons: a red home icon, a blue list icon, an orange folder icon, a green people icon, and a teal user icon.

Type	Name	Creation time	File location	Size
Folder	Private	5/3/2019, 9:29:52 AM	Disk	-
Folder	Tape	5/3/2019, 9:29:52 AM	Tape	-
File	public-file	5/3/2019, 9:39:45 AM	Disk	173 Bytes
File	private-file	5/3/2019, 9:39:45 AM	Disk	144 Bytes

- Direct command execution capability

```
> ssh -p 22224 localhost -l admin "\c PoolManager; rc ls"
```

- Added kerberos authentication for admin interface

- Enable specific authentication mechanisms

```
admin.ssh.authn.enabled = kerberos,password,publickey
```

```
admin.ssh.authn.kerberos.keytab-file = /etc/krb5.keytab
```

- Access log for user activity will be available from 5.2

- REST interface includes pluggable framework to support events
- Uses SSE for delivery
- Includes management framework for...
  - Creating SSE endpoint (*channel*)
  - Discovering supported event kinds
  - Managing which kinds are of interest (*subscribe* to event types)
- Notifications about changes in dCache namespace via interface modelled after *inotify(7)* API

# Events – Example Playground

- **Events Playground** in each dCache instance at

`<dcache-server>:3880/events.html`

## Events Playground

This page uses your web-browser's built-in support for Server-Sent Events (SSE) to receive events from dCache. It provides a simple demonstration to show events are working as expected.

### 1. Authentication

Enter your dCache username and password.

Name:  Password:

### 2. Channel

Each client needs exactly one channel to subscribe to events.

Create

Destroy

[https://prometheus.desy.de:3880/api/v1/events/channels/pSjjcKsZ\\_x4cForLUK7Bdg](https://prometheus.desy.de:3880/api/v1/events/channels/pSjjcKsZ_x4cForLUK7Bdg)

# Events – Example Playground

## 3. Subscriptions

Subscriptions are what generate events. The `SYSTEM` eventType is automatically subscribed but one (or more) subscriptions are needed before the client receives events.

to  events, emitting the message  every   for  messages

The current subscriptions:

- inotify-1** watching /Users/Lea

## 4. Events

Here are the ten most recent events.

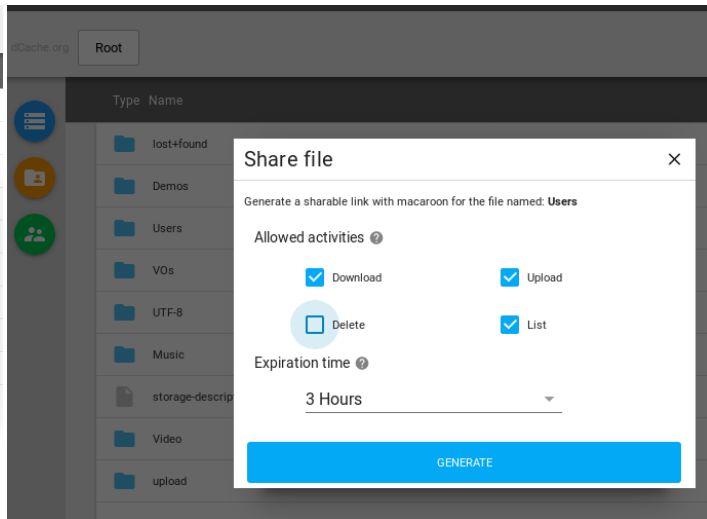
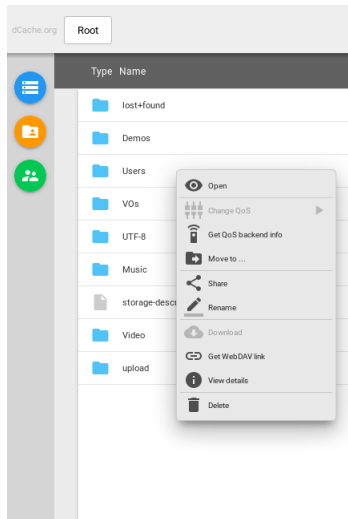
- 2019-05-14T11:59:10.398Z **SYSTEM** Subscription metronome-1 closed
- 2019-05-14T11:59:10.331Z **metronome-1** Hello 3
- 2019-05-14T11:59:07.347Z **metronome-1** Hello 2
- 2019-05-14T11:59:04.537Z **metronome-1** Hello 1
- 2019-05-14T11:59:01.343Z **SYSTEM** New subscription
- 2019-05-14T11:56:58.988Z **inotify-1** /Users/lea/Private/ IN\_CLOSE\_NOWRITE
- 2019-05-14T11:56:58.987Z **inotify-1** /Users/lea/Private/ IN\_OPEN
- 2019-05-14T11:56:54.941Z **inotify-1** /Users/lea/ IN\_CLOSE\_NOWRITE
- 2019-05-14T11:56:54.940Z **inotify-1** /Users/lea/ IN\_OPEN
- 2019-05-14T11:56:48.045Z **SYSTEM** New subscription



- Support for multiple Want-Digest request headers
  - Sender wants to receive an instance digest (checksum) on messages
  - Previously only first checksum request was considered, now all
- CORS support for HTTP requests for pools
- Enhanced Secure HTTP transfers over IPv6

- File sharing using macaroons
- Transfer data with remote site using X.509, authenticate locally with macaroon
- Ability to reject macaroons sent over unencrypted channel (default starting 5.1 is reject)
- Disabling basic authn no longer disables macaroons:  
`set webdav.authn.basic and frontend.authn.basic to false`

# WebDAV – Example



- Aim to provide alternatives to GridFTP
- HTTP
  - Source/ destination support
  - 3<sup>rd</sup> vendor HTTP server as destination
  - X.509, Macaroon and SciToken support
- XROOTD [4.2]
  - Source/ destination support
  - GSI authN and delegation
  - Interoperability with SLAC xrootd client server

- Handle pending transfers with admin commands

```
> transfer retry or > transfer forget
```

- No more 16-group limit for users! set NFS property:

```
nfs.idmap.manage-gids // default : false
```

→ If client uses AUTH\_SYS and has #groups>16, NFS door will query gPlazma to discover additional user groups

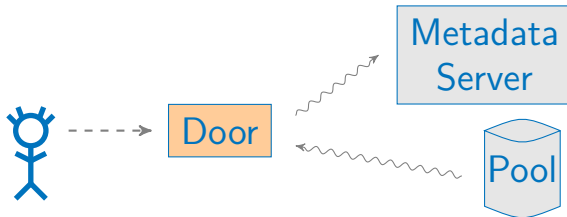
- Currently only LDAP is supported

- Pool decommissioning: [4.2]  
Notify clients about need to terminate remaining connections  

```
> pool reset id [-recall] <pool>
```
- Pool IO errors reported to door for pNFS clients using flex\_file layout: [5.0]

NFS4ERR\_NXIO → Client was unable to establish communication

NFS4ERR\_\* → Client established communication, returns errors



- Client can request inexistent checksum of a file not owned by him [4.2]
  - Checksum is then stored
- Ftp door access via anonymous FTP and encrypted FTPS (TLS) authentication [5.1]
  - At the moment just **client** ↔ **door** (control channel), not client ↔ pool (data channel) or proxying

- **info** admin command shows ASCII-art rendition of current data transfers  
→ Proxy status easier to discover and debug

- ASCII-Art can be written to log-file via

```
ftp.enable.log-aborted-transfers // default: false
```

```
pool.mover.ftp.enable.log-aborted-transfers // default: false
```



# Example Transfer Visualization I

- On-going transfer, TCP connections fully established

```
[sparkplug] (GFTP-sparkplug-AAVn7wdMZMg@dCacheDomain) admin > info
User Host : 127.0.0.1
```

```
[...]
```

```
Passive adapter:
```

```
Transfer mode: E
```

```
Listening on:
```

```
Client: /127.0.0.1:24141
```

```
Pool: /127.0.0.1:39629
```

```
Proxy status:
```

Client	Adapter	Pool
127.0.0.1:60930=====	127.0.0.1:24141	127.0.0.1:39629  =====127.0.0.1:49948
127.0.0.1:60932=====	127.0.0.1:24141	
127.0.0.1:60934=====	127.0.0.1:24141	
127.0.0.1:60936=====	127.0.0.1:24141	

```
TCP states: "=====" means Established
```

```
GFTP-sparkplug-AAVn7wdMZMg@dCacheDomain; p=GFTP-1;o=1000/0; [...]
```

- Active adapter, both client-adapter and adapter-pool are half-closed

```
[sparkplug] (GFTP-sparkplug-AAVn7w1hO5A@dCacheDomain) admin > info
  User Host : 127.0.0.1
```

```
[...]
```

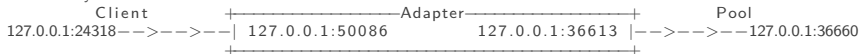
```
Active adapter:
```

```
  Listening for pool on: /127.0.0.1:36613
```

```
  Connecting to: 127.0.0.1:24318
```

```
  Streams: 1
```

```
  Proxy status:
```



```
TCP states: "-----" means Half-closed (arrows show open dirn)
```

```
GFTP-sparkplug-AAVn7w1hO5A@dCacheDomain; p=GFTP-1;o=1000/0; [...]
```

- Library updates led to VOMS certificate validation errors
  - Added refresh of certificates in trust anchor directory:  

```
gplazma.vomsdir.refresh-interval = 4  
gplazma.vomsdir.refresh-interval.unit = HOURS
```
  - Without this fix, VOMS plugin fails to validate VOMS certificates, rendering dCache non-operational
- Sites running dCache releases **4.2.14 – 4.2.22** are **strongly encouraged to upgrade!**

- *observer* role added (weaker read-only admin privileges) for `dcache-view` [5.0]
- With user's numerical UID gPlazma can query LDAP server for other attributes [4.2]  
`gplazma.ldap.try-uid-mapping // default: false`
- Support for SciToken authentication, improved OpenID Connect support [5.1]

- Fixed cause of possible data loss during removal of excess replicas [4.0]
- Starting from 5.2: Deleting → *soft remove*
  - Remove triggers **unsetting of sticky flag**
  - Non-sticky copies can be promoted to sticky instead of creating a new copy

- Resilience activity can be logged

```
dcache.log.level.resilience // off|error|warn|info|debug|trace|all
```

- After restart of pool, operations can be restarted without restarting resilience

```
> pool ctrl shutdown and > pool ctrl start
```

- Pool scans of excluded pools now generally not allowed

- Pool errors can be sent as an email alarm [5.0]
- Alarm `POOL_DEAD` to distinguish from `POOL_DOWN` and `POOL_DISABLED` [5.1]
- Always log why a replica state is changing [5.1]

- Dynamic pool groups based on tags [5.1]  
`> psu create pgroup -dynamic -tags=zone=A zone-A-pools`
- Inotify support for open/ read/ write/ close events [5.0]  
`pool.inotify-generation.enable = true`
- IO chunk size configurable (read and write) [5.0]  
`pool.mover.http.chunk-size = 8192`



- Previously NFS and WebDAV supported Kafka, now all doors

**general:** `dcache.enable.kafka // default: false`

**or:** `ftp.enable.kafka | dcap.enable.kafka | xrootD.enable.kafka`

- Adjust timeout for dCache attempting to send a Kafka event:

`dcache.kafka.maximum-block = 1`

`dcache.kafka.maximum-block.unit = SECONDS`

- Specify receiving topic name:

`dcache.kafka.topic = billing`

- Starting from 5.2 alarms can be sent to Kafka

- Major revision of documentation + new *UserGuide*
- MIME type support for scientific file formats HDF4, 5 and ROOT [5.0]
- XRootD: support for security level and signed hashes [5.0]
- Disallow tape staging for anonymous users [5.1]  
`poolmanager.authz.anonymous-staging // default: true`
- Multiple PnfsManagers can share the same cache [4.2]
  - HA-Nextcloud instance at DESY needs replicated Namespace

**Was anything missing  
that you expected to hear about?**

## Example: Events

- Request a channel

```
> curl -D- -E /tmp/x509up_u1000 -X POST <url>/events/channels/
```

- Enable receiving events

```
> curl -E /tmp/x509up_u1000 -H 'Accept: text/event-stream'  
<url>/events/channels/AgW3_ch4nn3ln4m3
```

- Create subscription to events from 'metronome'

```
> curl -D- -E /tmp/x509up_u1000 -X POST -H 'Content-Type:  
application/json' -d '{"delay": 2, "message": "event $count"}'  
<url>/events/channels/AgW3_ch4nn3ln4m3/subscriptions/metronome
```

- Delete subscription to stop receiving events:

```
> curl -D- -E /tmp/x509up_u1000 -X DELETE  
<url>/events/channels/AgW3_ch4nn3ln4m3/subscriptions/(continued->  
metronome/5ubss-c1pt-1d
```

# Example: Events – Part II

- Receive events:

```
> curl -E /tmp/x509up_u1000 -H 'Accept: text/event-stream'  
<url>/events/channels/AgW3_ch4nn3ln4m3
```

```
event: metronome
```

```
id: 0
```

```
data: {"event":"event 1",
```

```
"subscription":"<url>/events/channels/AgW3_ch4nn3ln4m3/(continued->  
subscriptions/metronome/5ubss-cr1pt-1d"}
```

```
event: metronome
```

```
id: 1
```

```
data: {"event":"event 2",
```

```
"subscription":"<url>events/channels/AgW3_ch4nn3ln4m3/(continued->  
subscriptions/metronome/5ubss-cr1pt-1d"}
```