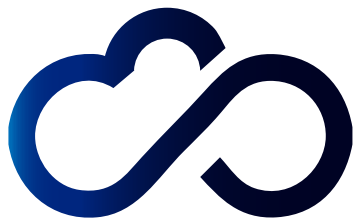


dCache-view

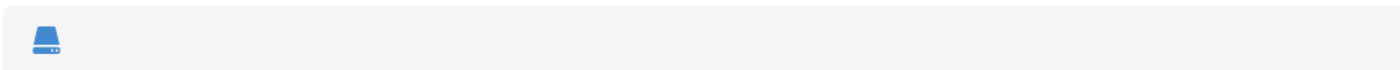
Olufemi S. Adeyemi









On behalf of the project team



INDIGO DataCloud



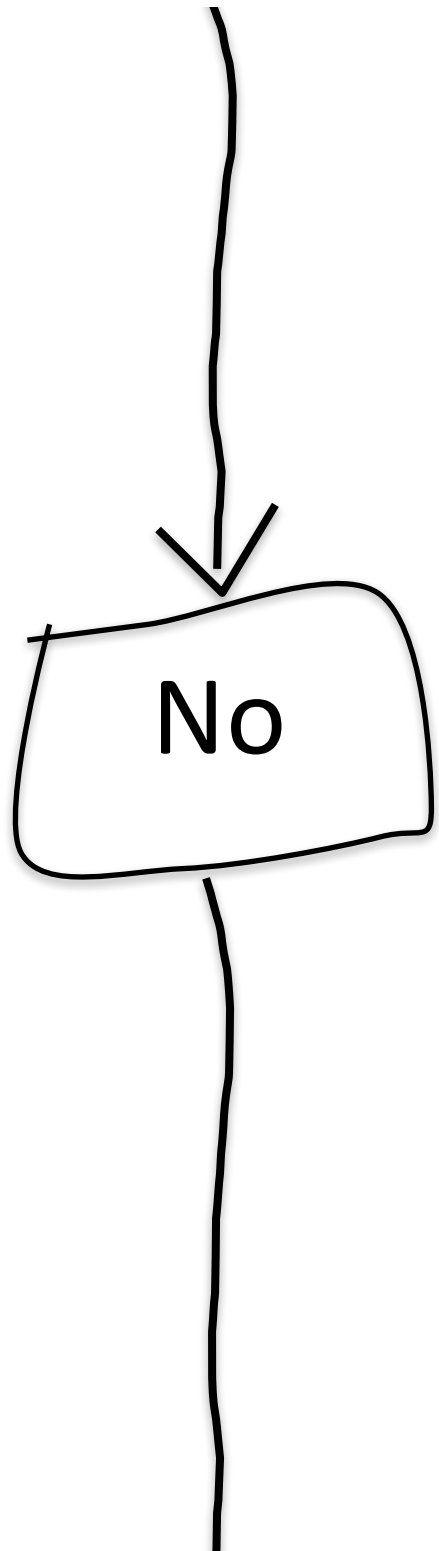


Name	Size	Last Modified
 disk		Sat Apr 09 13:04:02 CEST 2016
 lost+found		Sat Apr 09 13:03:46 CEST 2016
 private		Sat Apr 09 13:04:02 CEST 2016
 public		Sat Apr 09 13:04:02 CEST 2016
 replica		Sat Apr 09 13:04:02 CEST 2016
 reserved		Sat Apr 09 13:04:02 CEST 2016
 resilient		Sat Apr 09 13:04:02 CEST 2016
 tape		Sat Apr 09 13:04:02 CEST 2016

Useful flow chart



Should I continue to
use the **webdav**
webpage?





But what if ...





**OMG
No!**

Use the “new” user interface



Noticeable Changes



1. The Name

The name

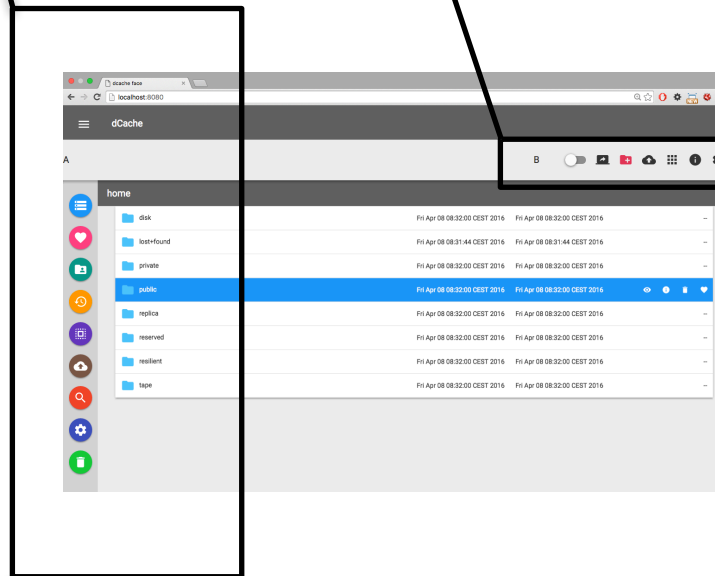
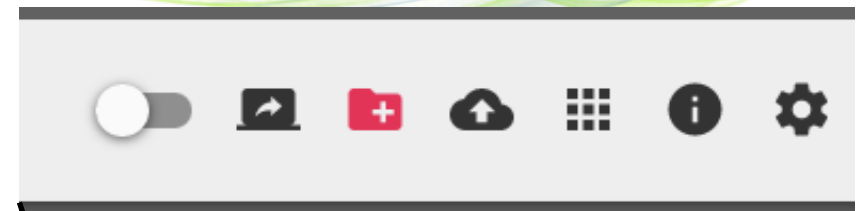
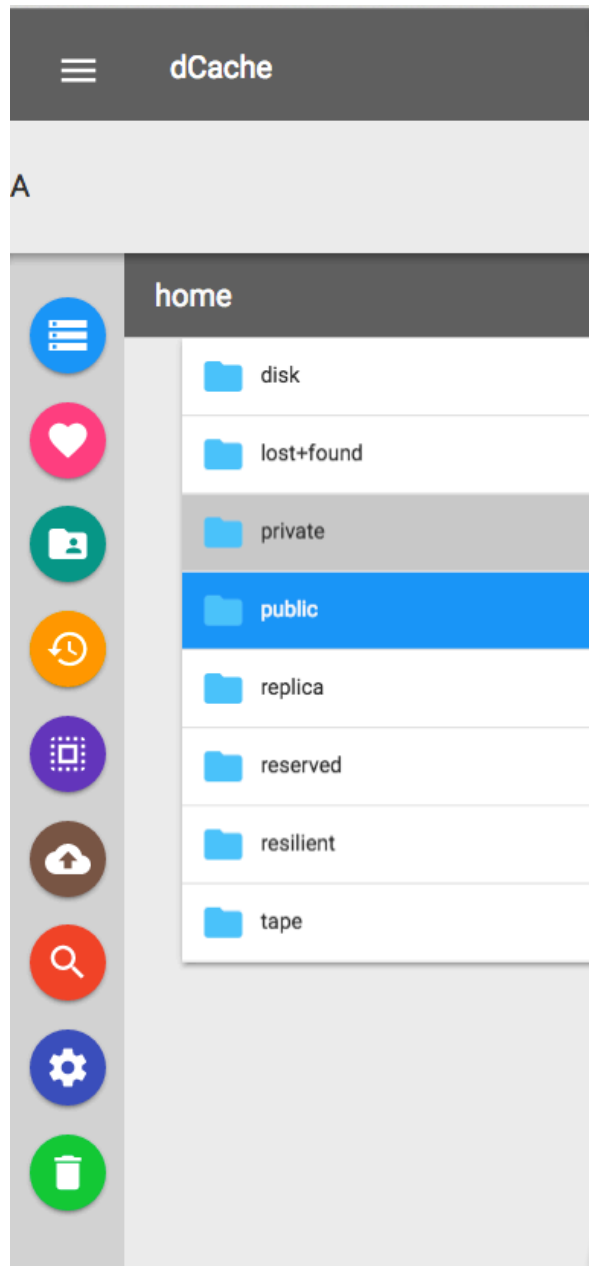


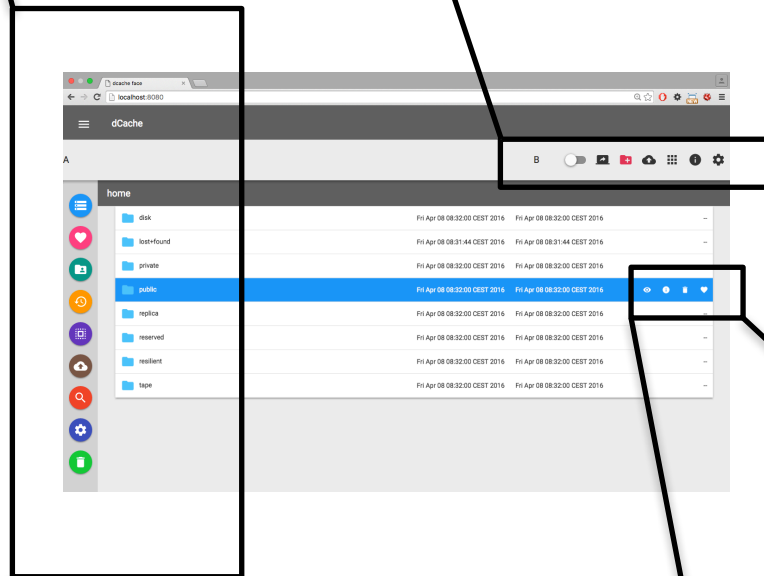
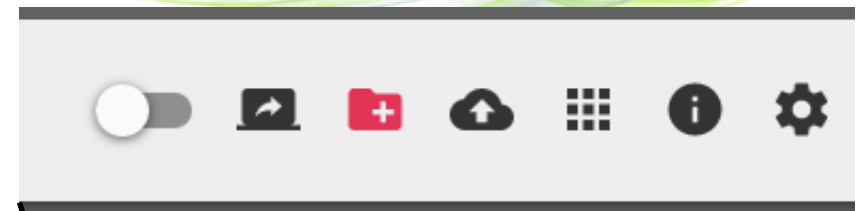
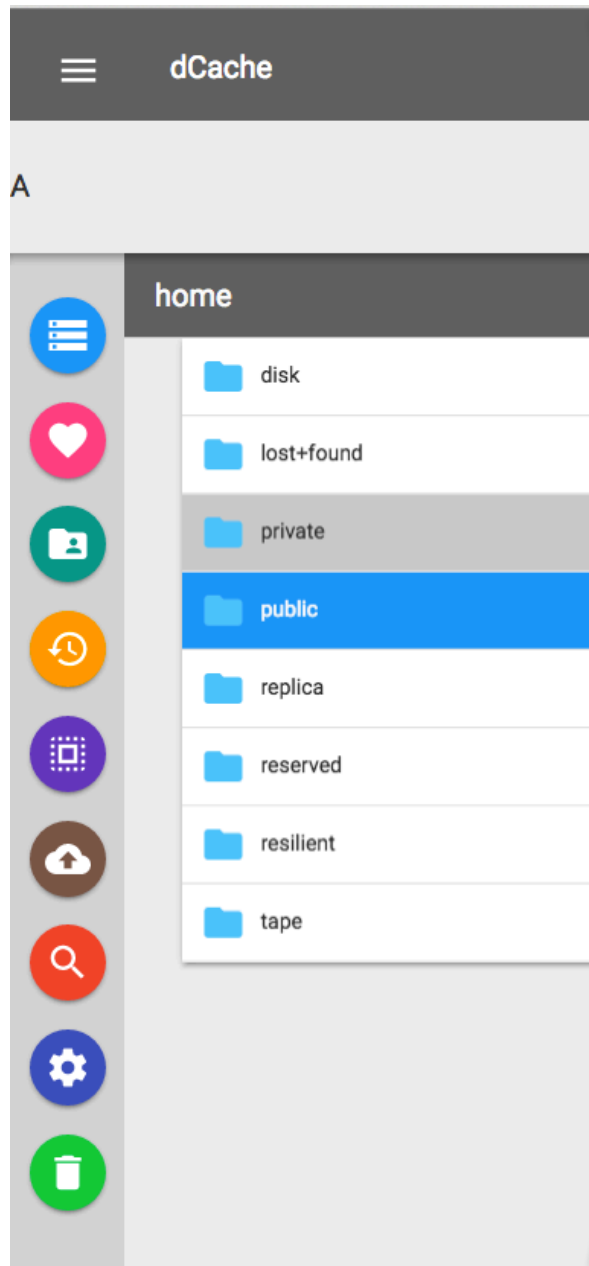
dCache ~~Face~~ = dFace


Noticeable Changes



2. Cleaner Interface













 This image cannot currently be displayed.

Hover context menu

system-test (built from a4d77e2) log in

dCache.org Root
 + ↑ i

Type	Name	Creation time	File location	Size
	disk	5/28/2017, 2:05:54 PM		--
	lost+found	5/28/2017, 2:05:51 PM		--
	private	5/28/2017, 2:05:54 PM		--
	public	5/28/2017, 2:05:54 PM		--
	replica	5/28/2017, 2:05:54 PM		--
	reserved	5/28/2017, 2:05:54 PM		--
	resilient	5/28/2017, 2:05:54 PM		--
	tape	5/28/2017, 2:05:54 PM		--

3. In-house Third Party Components

See: <https://github.com/dcache-elements>

dcache-namespace Element

Represent the namespace operation

HTML → `<dcache-namespace id="namespace"></dcache-namespace>`

JavaScript

```
let namespace = document.createElement('dcache-namespace');
let listResult;
namespace.auth = window.btoa('anonymous:nopassword');
namespace.promise.then((req)=>{
    listResult = req.response;
}).catch((err)=>{
    console.log(err.message);
});
namespace.ls({
    url: 'https://prometheus.desy.de:3880',
    path: '/Users',
    children: true
});
```

dcache-view is part of frontend service

Add frontend service to a dCache domain

```
[SomeDomain/frontend]
frontend.dcache-view.org-name=DESY
frontend.dcache-view.endpoints.webdav=
    https://example.org:3769/
frontend.authz.anonymous-operations=
    (one-of?NONE|READONLY|FULL)
frontend.authn.protocol= https
frontend.authn.basic=true
```


Features



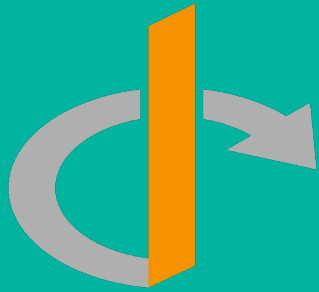
- Login – with username and password
- List directories – including home and root
- Rename
- Download
- Move
- Create
- File metadata

Newly added Features

- Context menu
- OpenID Connect
- Bulk Operation (?)
- User Info
- Upload
- Drag and Drop
- Quality of Service:
 - Get backend info
 - Change

Highlighting Some of the features

{ 1 }



OpenID Connect



See **Anupam Ashish** last year talk for details

<https://indico.desy.de/contributionDisplay.py?contribId=8&confId=13786>



Brief Intro. to OpenID Connect

*OpenID Connect is a simple identity layer on top of **OAuth 2.0** protocol (Authorisation framework).*

It allows Clients to verify the identity of the End-User based on the authentication performed by the an Authorisation Server.



OpenID Connect with dCache - Configuration

1. Register (or set-up) OpenID Provider
2. webdav

➤ `webdav.oidc.client.ids!provider.hostname`

```
webdav.oidc.client.id!accounts.google.com = <client id>
```

➤ `webdav.oidc.client.secrets!provider.hostname`

```
webdav.oidc.client.secret!accounts.google.com = <client secret>
```

3. dcache-domain -> gPlazma

```
dcache.oidc.hostnames = accounts.google.com
```

4. gplazma.conf

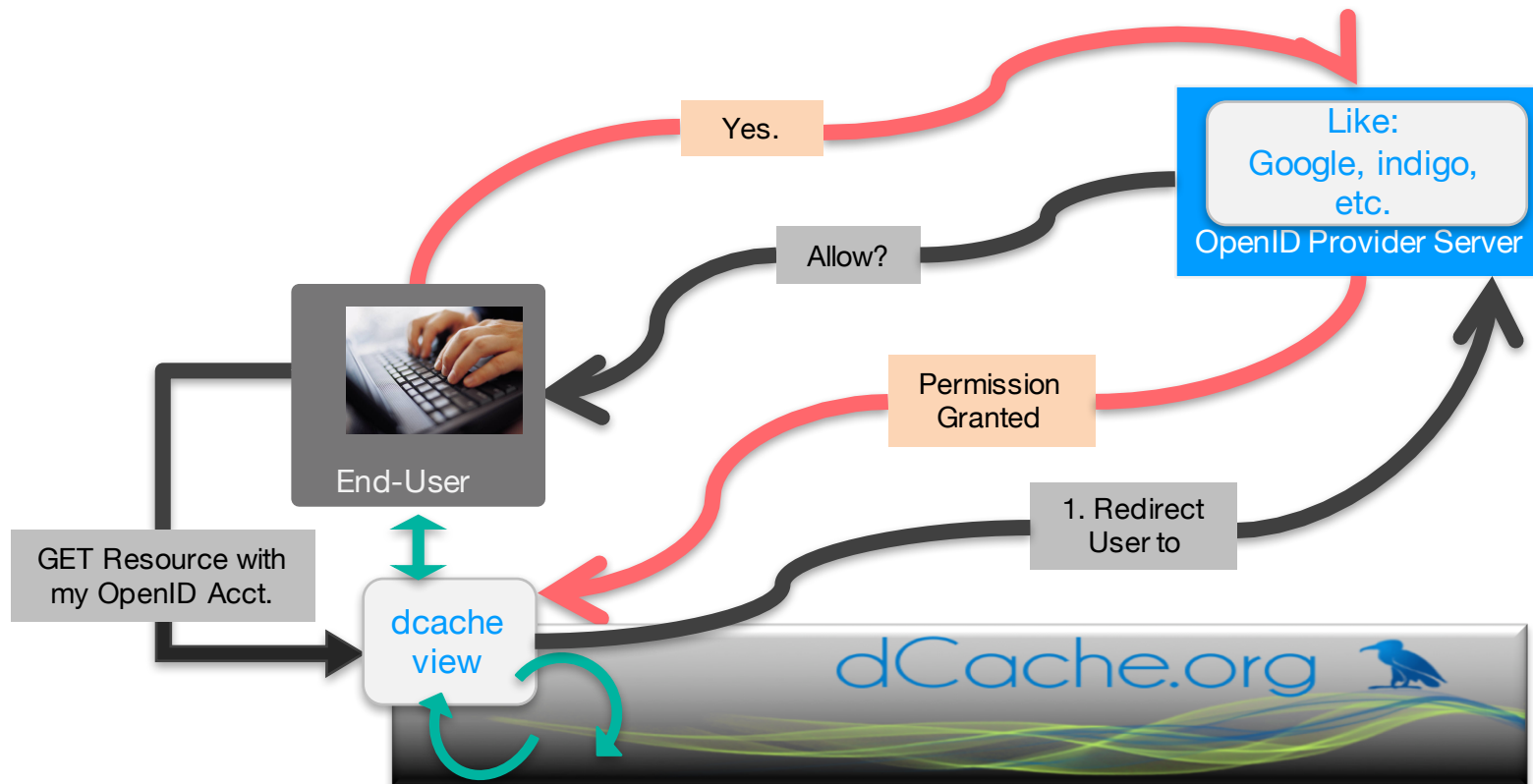
```
auth optional oidc  
map optional multimap
```

➔ Read more:

<https://github.com/dCache/dcache/wiki/OpenID-Connect-with-dCache>



How does it work with dcache-view? - **Implicit Flow**





OpenID Connect & Frontend Service </Setup>

1. frontend.dcache-view.oidc-provider-name-list

```
frontend.dcache-view.oidc-provider-name-list=google ebay
```

2. frontend.dcache-view.oidc-client-id-list

```
frontend.dcache-view.oidc-client-id-list=<clientID1> <clientID2>
```

3. frontend.dcache-view.oidc-Authz-endpoint-list

```
frontend.dcache-view.oidc-Authz-endpoint=<client_EndPoint1>  
<client_EndPoint2>
```




</Demo>

**Authentication & Authorisation
with
OpenID Connect**

{2}



Uploads and Downloads in dCache-View



Files Upload and Download in dcache-View

At the moment, dCache-view uses the webdav door for downloading and uploading files.

Hence, both need to know how to talk to each other



Configuration for Download and Upload

1. Frontend Service Configuration

- `frontend.dcache-view.endpoints.webdav`

```
frontend.dcache-view.endpoints.webdav=  
http://webdav.example.org:3335/
```

2. Webdav Service Configuration

- `webdav.allowed.origins`

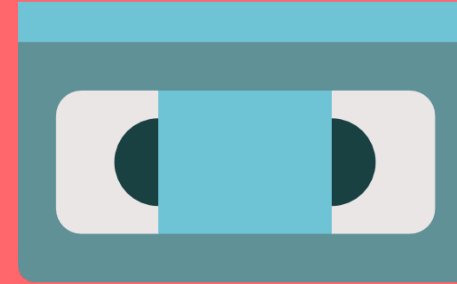
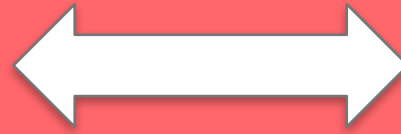
```
webdav.allowed.origins=  
http://dcache-view.example.org:3335
```

Cross-Origin Resource Sharing (CORS)



</Demo>

- 1. Uploads Files** *(using a button)*
- 2. Uploads Files** *(using a drag and drop)*
- 3. Downloads**



</Demo>

Change Quality of Service

in dCache-View



See Marina Sahakyan talk for details

</Demo>

O t h e r S t u f f

in dCache-View

<thank-you />