

Continuous delivery @ NDGF

For the 14th international dCache workshop

Vincent Garonne
University of Oslo

On behalf of the project team



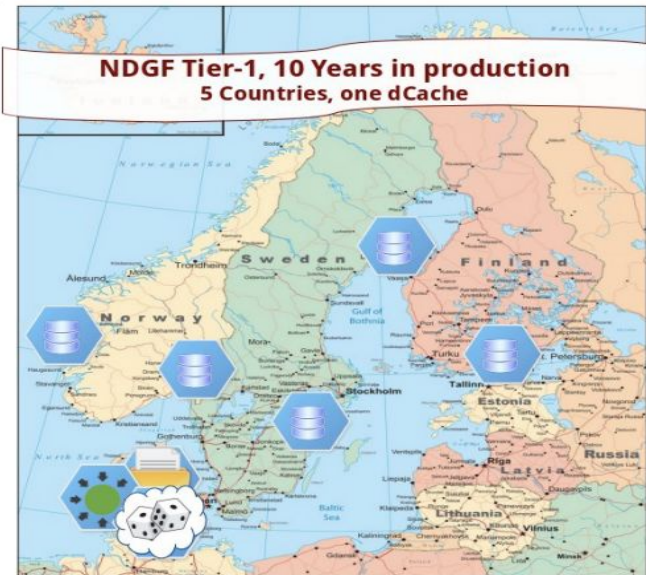
About the Nordic Center (NDGF)

Distributed centre:

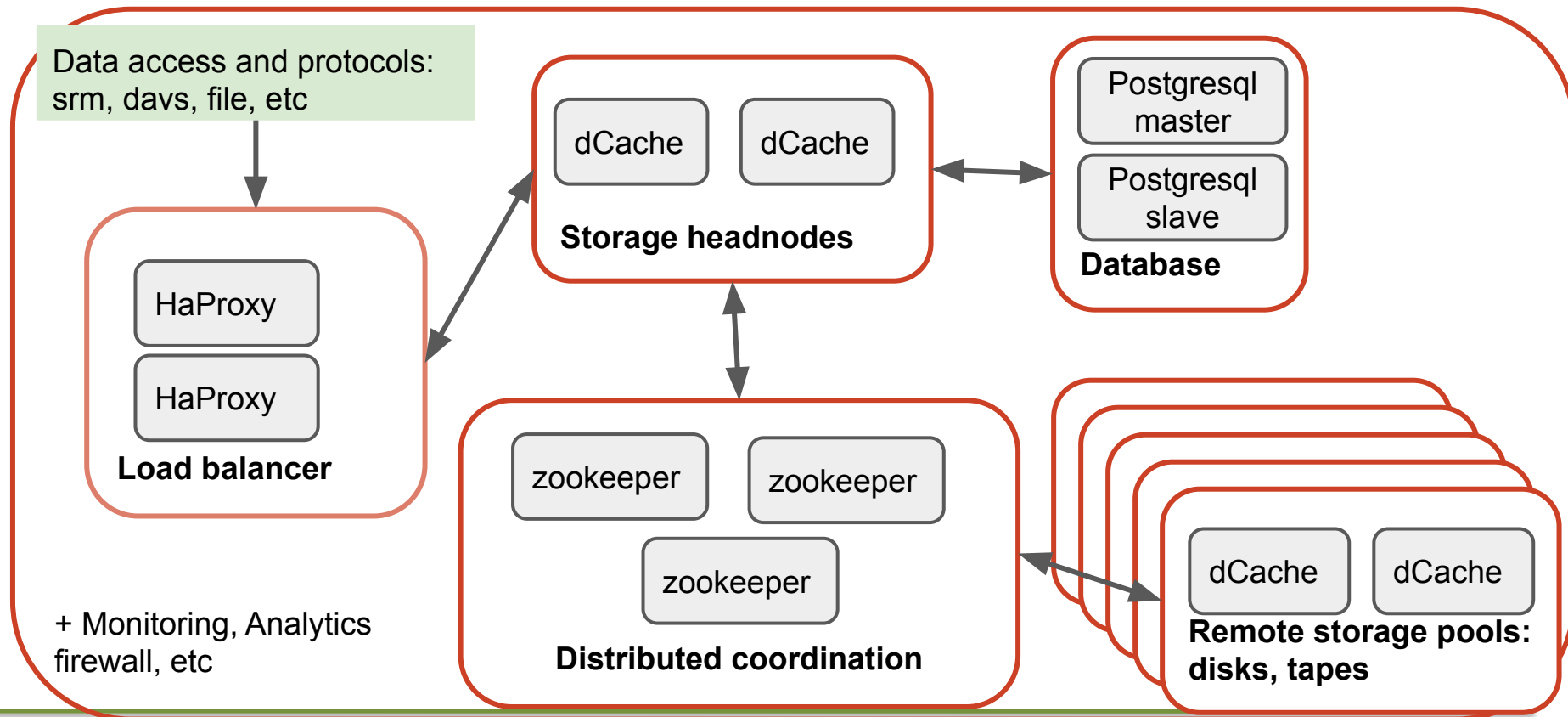
- 6 Nordic academics
- Support both for ATLAS and ALICE
- Distributed storage storage provided by dCache
- Data access for jobs provided by ARC

Multi-Site deployment

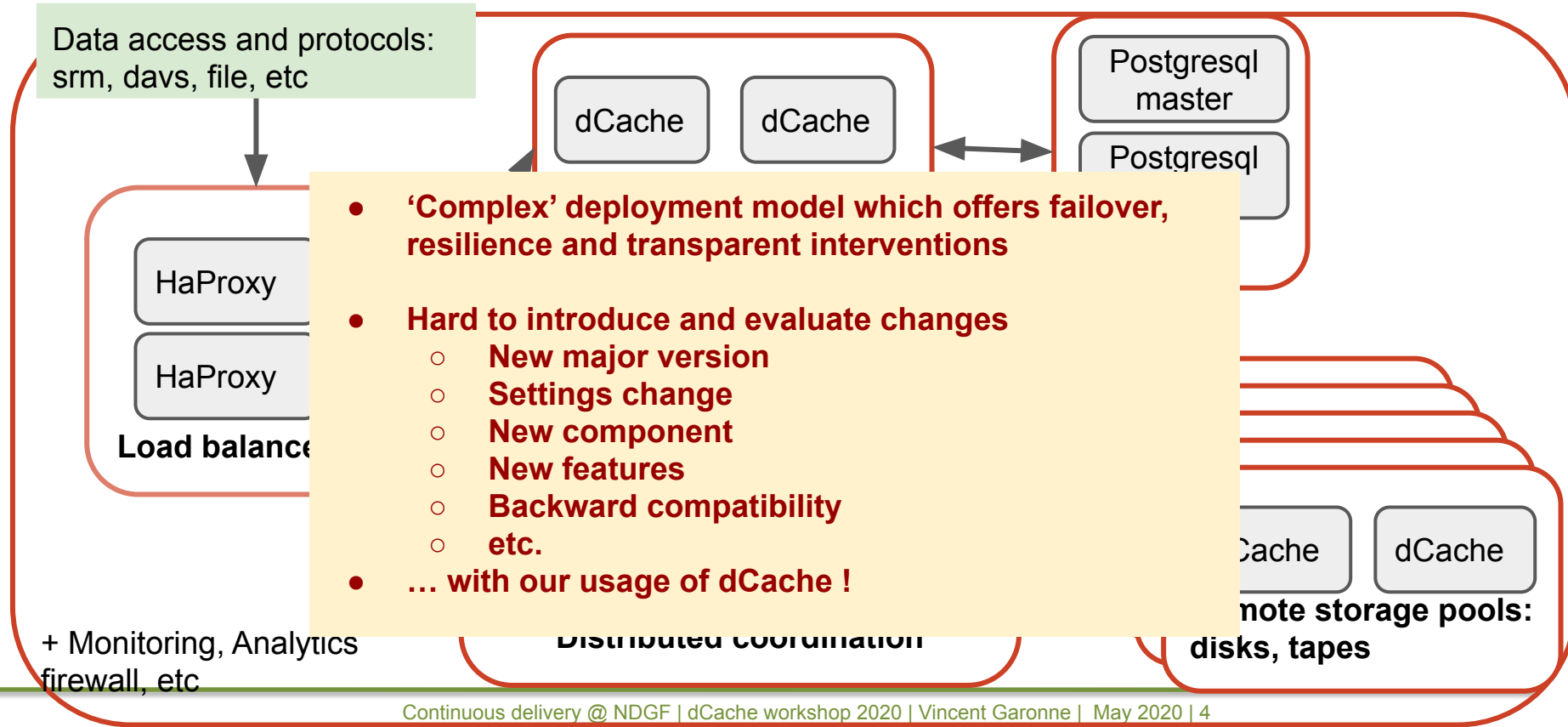
- Distribute data over multiple locations
- Multiple administrative domains
- Use available resources



Highly Available and Distributed 24/7 Storage



Highly Available and Distributed 24/7 Storage



Pre-production Instance

- The pre-production instance is the exact clone of the production deployment model
 - Haproxy, headnodes, database, remote disk pools, etc
- Load by the ATLAS DDM functional test, hammer clouds ARC jobs, Paul's dav smoke tests and probes: data access, copy, stats and deletion
- **Validation of changes before reaching production**
 - **Fix, detect and report issues as soon as possible**

Configuration management

- Same configuration management templates (roles and playbooks) than in production
- Multistage environments with Ansible
 - Distinct production and pre-production variables like database, software version, etc.
 - E.g., `dcache_version: 6.1.4`
`webdav.loginbroker.address:preprod-srm.ndgf.org`

➤ Example:

```
> time ansible-playbook -i environments/preproduction/ plays/dcache-headnode.yml
```

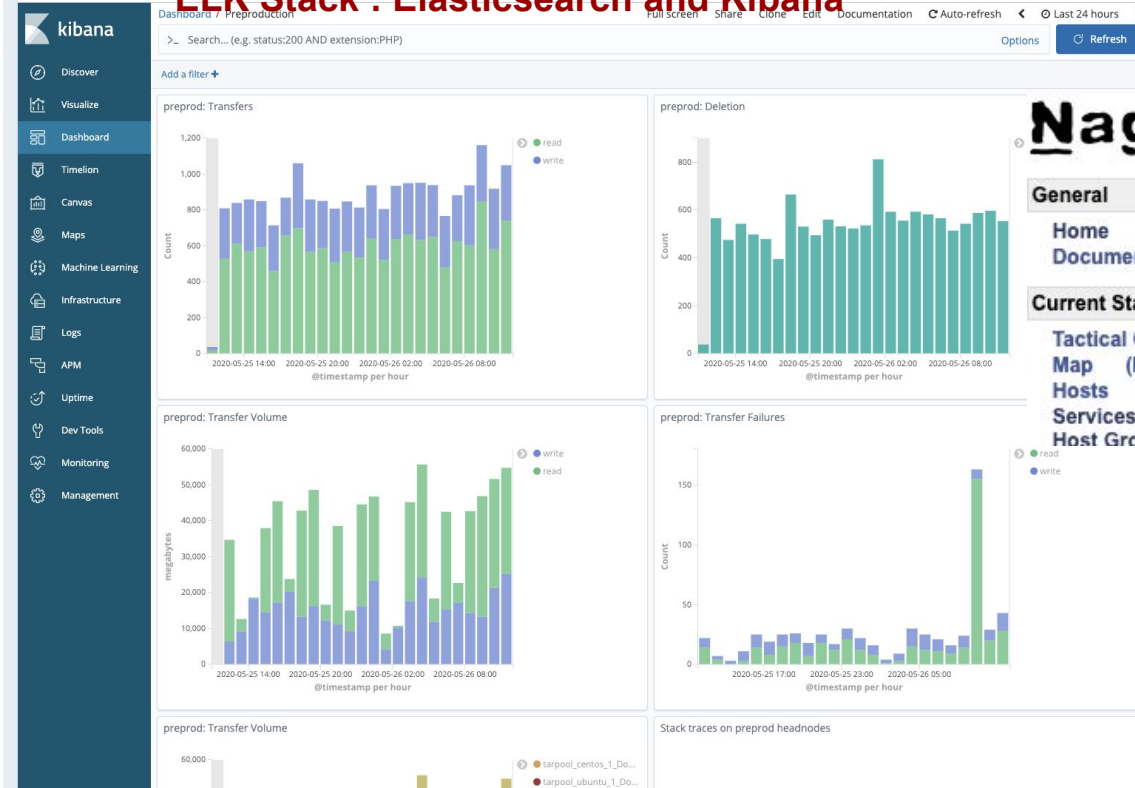
```
...
```

```
real: 2m29.479s
```

If fine, changes are applied the week after on `environments/production`

Monitoring

ELK Stack : Elasticsearch and Kibana



Infrastructure monitoring: Probes



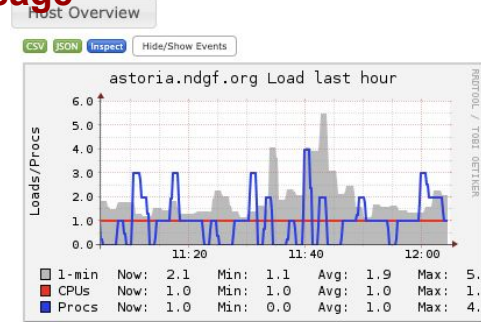
- General
- Home
- Documentation
- Current Status
- Tactical Overview
- Map (Legacy)
- Hosts
- Services
- Host Groups

preproduction nodes (preprod)

Host	Status	Services	Actions
astoria.ndgf.org	UP	8 OK	
cosgrove.ndgf.org	UP	8 OK	
labrea.ndgf.org	UP	8 OK	
preprod-srm.ndgf.org	UP	5 OK	
statler.ndgf.org	UP	16 OK	
waldorf.ndgf.org	UP	16 OK	

NDGF Grid > Monitor > astoria.ndgf.org

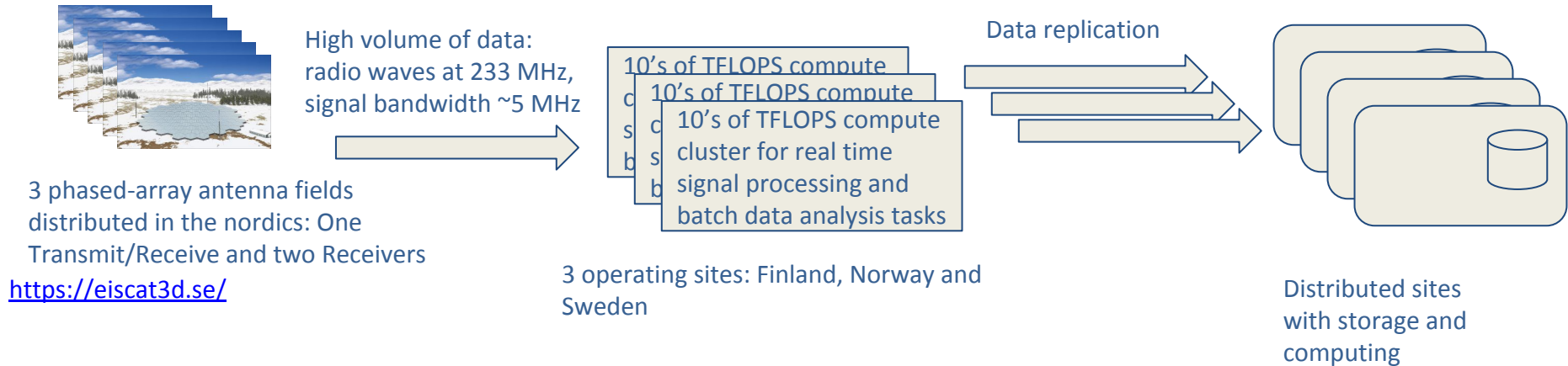
Infrastructure monitoring: Node usage



- Pre-production storage useful in many situations but non negligible investment
 - NDGF regularly updates to the latest version in production, e.g., 6.1.*
- We can explore with “agility” changes, new use cases and make evolved the stack together with ATLAS, WLCG, DOMA and other initiatives
 - E.g, Dav, Macaroon, OIDC, Storage events, etc
- Beyond LHC and NT1 support for others Data science application
 - Multi-experiment/VOs support
 - E.g., transfer tests with SKA@surf SARA, EISCAT_3D

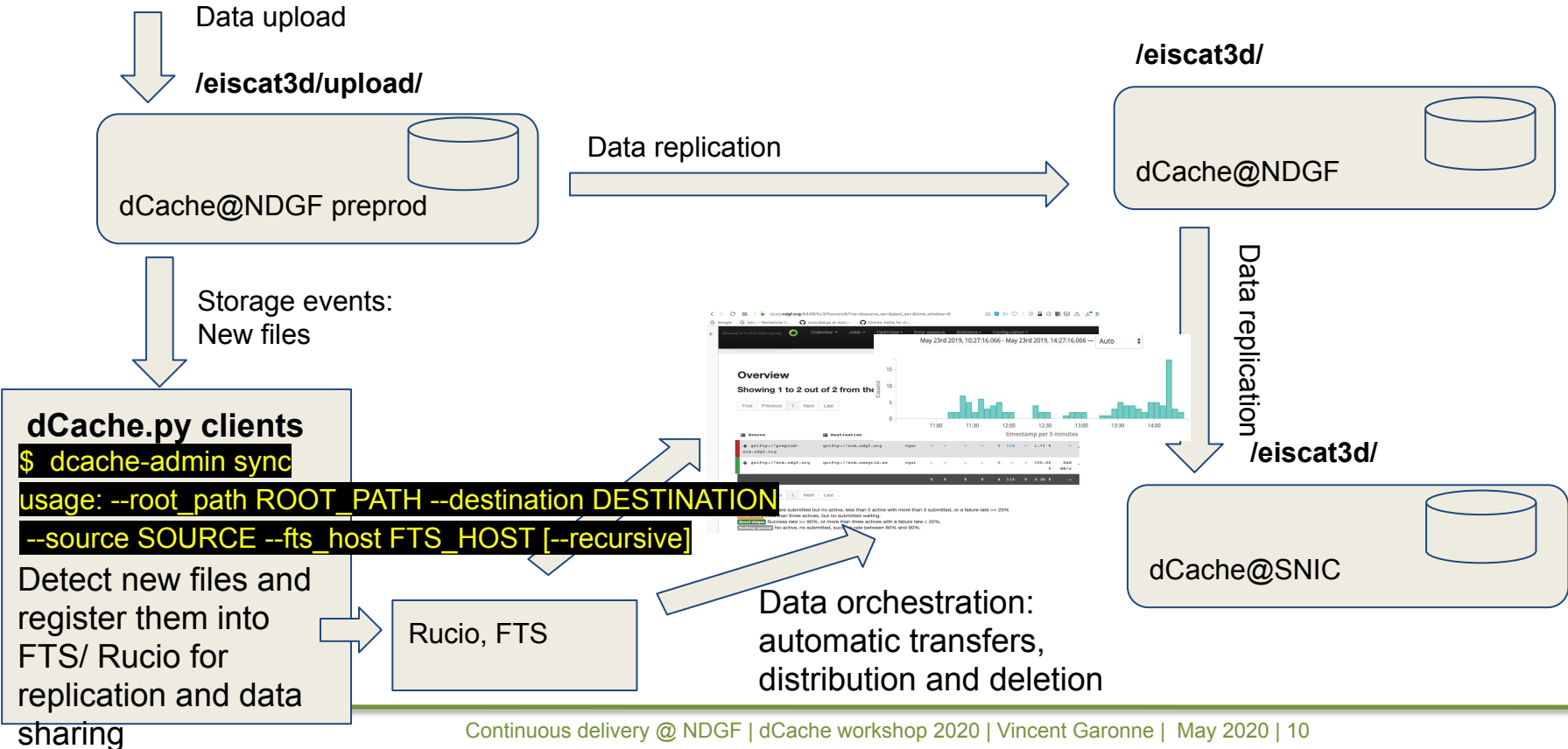
EISCAT_3D Use case: Next generation radar for 3D monitoring of the atmosphere and ionosphere

- Data-intensive instruments generates a high volume of data



- Researchers need to analyse data and share their results
- How can we automate the data replication ? sync with third party system ,e.g., data management tools and catalogs ?

Test Case: Automatic replication exercise



- Python client for the dCache frontend RESTful API:
<https://github.com/neicnordic/dcacheclient> <https://pypi.org/project/dcacheclient/>

- Resources:

complete print bash completion command

alarms The log of internal problems

billing The log of (significant) client activity

cells The running components within dCache

identity Information about users

namespace Files, directories and similar objects

poolmanager Data placement and selection decisions

pools File data storage

qos Managing how data is stored and handled

spacemanager Ensuring enough capacity for uploads

transfers The movement of data between dCache and clients

events Support for SSE clients receiving dCache events

sync Synchronise storage

dCache python Clients: Examples

➤ Installation:

```
pip install dcacheclient
```

➤ Configuration:

```
> cat .dcachecfg
```

```
[default]
```

```
url = https://preprod-srm.ndgf.org:3880
```

```
role = admin
```

➤ API Example:

```
>>> from dcacheclient.client import Client
```

```
>>> dcache = Client(url='https://srm.ndgf.org:3880')
```

```
>>> dcache.identity.get_user_attributes()
```

```
{'status': 'ANONYMOUS'}
```

➤ CLI Example:

```
> dcache-admin namespace getAttributes --pnfsid 00000C3690F0B2D490FB23316FDF61A681C
```

The END

further reading
www.dCache.org