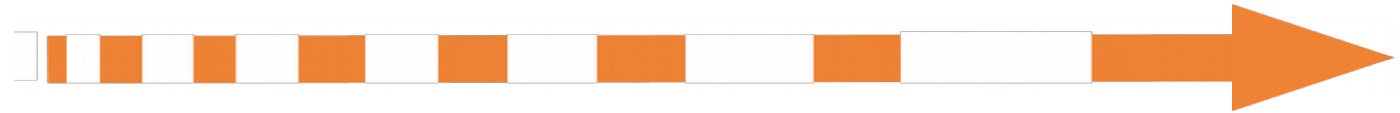


# XDC (eXtreme DataCloud)



Data Management for extreme scale computing



Paul Millar

[paul.millar@desy.de](mailto:paul.millar@desy.de)

... with thanks to the  
many people who  
contributed material.

**dCache workshop**  
21<sup>st</sup> – 22<sup>nd</sup> May 2019, Madrid Spain



eXtreme DataCloud is co-funded by the Horizon2020  
Framework Program – Grant Agreement 777367  
Copyright © Members of the XDC Collaboration, 2017-2020

<https://indico.desy.de/indico/event/22170/>

# XDC Objectives

- ✕ The eXtreme DataCloud is a software development and integration project
- ✕ Develops **scalable** technologies for federating storage resources and managing data in highly distributed computing environments
  - ➡ Focus efficient, policy driven and Quality of Service based DM
- ✕ The targeted platforms are the current and next generation e-Infrastructures deployed in Europe
  - ➡ European Open Science Cloud (EOSC)
  - ➡ The e-infrastructures used by the represented communities

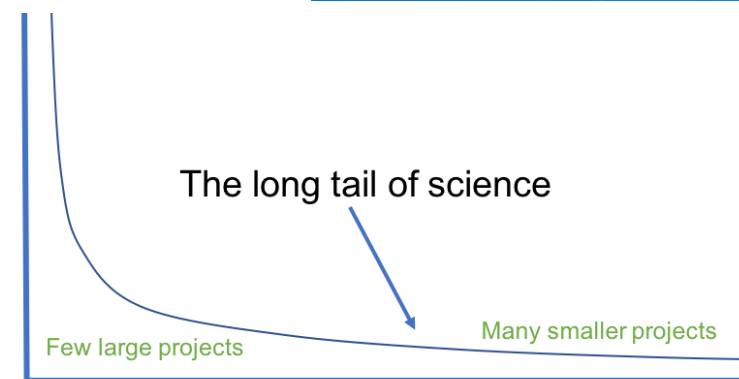
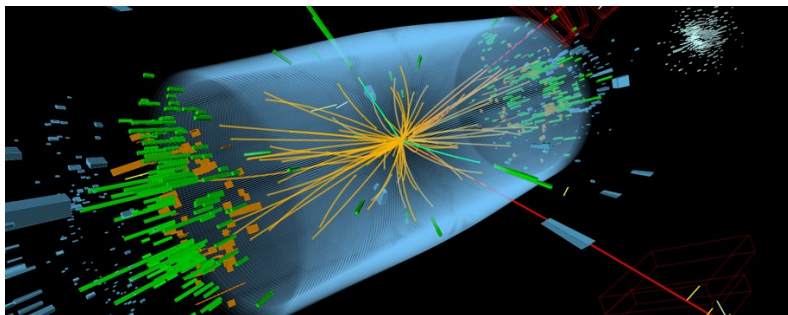
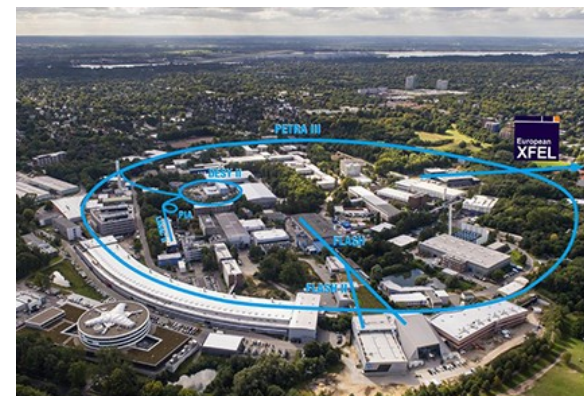
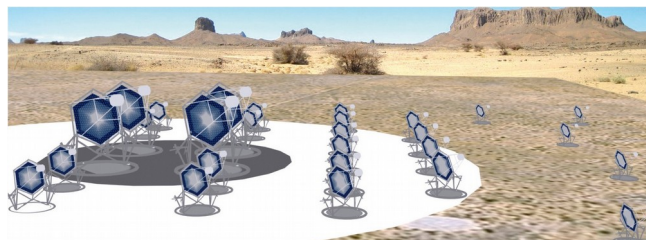
# XDC Consortium

ID	Partner	Country	Represented Community	Tools and system
1	INFN (Lead)	IT	HEP/WLCG	INDIGO-Orchestrator, INDIGO-CDMI(*)
2	DESY	DE	Research with Photons (XFEL)	dCache
3	CERN	CH	HEP/WLCG	EOS, DYNAFED, FTS
4	AGH	PL		ONEDATA
5	ECRIN	[ERIC]	Medical data	
6	UC	ES	Lifewatch	
7	CNRS	FR	Astro [CTA and LSST]	
8	EGI.eu	NL	EGI communities	



- ✂ 8 partners, 7 countries
- ✂ 7 research communities represented + EGI
- ✂ XDC Total Budget: 3.07M€
- ✂ XDC started on Nov 1<sup>st</sup> 2017 – will run for 27 months until Jan 31<sup>st</sup> 2020

# Represented research communities



# XDC Technical Topics

## ✂ Intelligent & Automated Dataset Distribution

- ➡ Orchestration to realize a policy-driven data management
- ➡ Data distribution policies based on Quality of Service (i.e. disks vs tape vs SSD) supporting geographical distributed resources (cross-sites)
- ➡ Software lifecycle management

## ✂ Data pre-processing during ingestion

## ✂ Data management based on access patterns

- ➡ Move to 'glacier-like' storage unused data, move to fast storage "hot" data
  - ➡ at infrastructure level

## ✂ Smart caching

- ➡ Transparent access to remote data without the need of a-priori copy

## ✂ Metadata management

## ✂ Sensitive data handling

- ➡ secure storage and encryption



# XDC Technical Topics

## ✂ Intelligent & Automated Dataset Distribution

**QoS talk, next**

- ➡ Orchestration to realize a policy-driven data management
- ➡ Data distribution policies based on Quality of Service (i.e. disks vs tape vs SSD) supporting geographical distributed resources (cross-sites)
- ➡ Software lifecycle management

## ✂ Data pre-processing during ingestion

**... more in a moment**

## ✂ Data management based on access patterns

- ➡ Move to 'glacier-like' storage unused data, move to fast storage "hot" data
  - ➡ at infrastructure level

## ✂ Smart caching

**... "XDC smart caching", tomorrow**

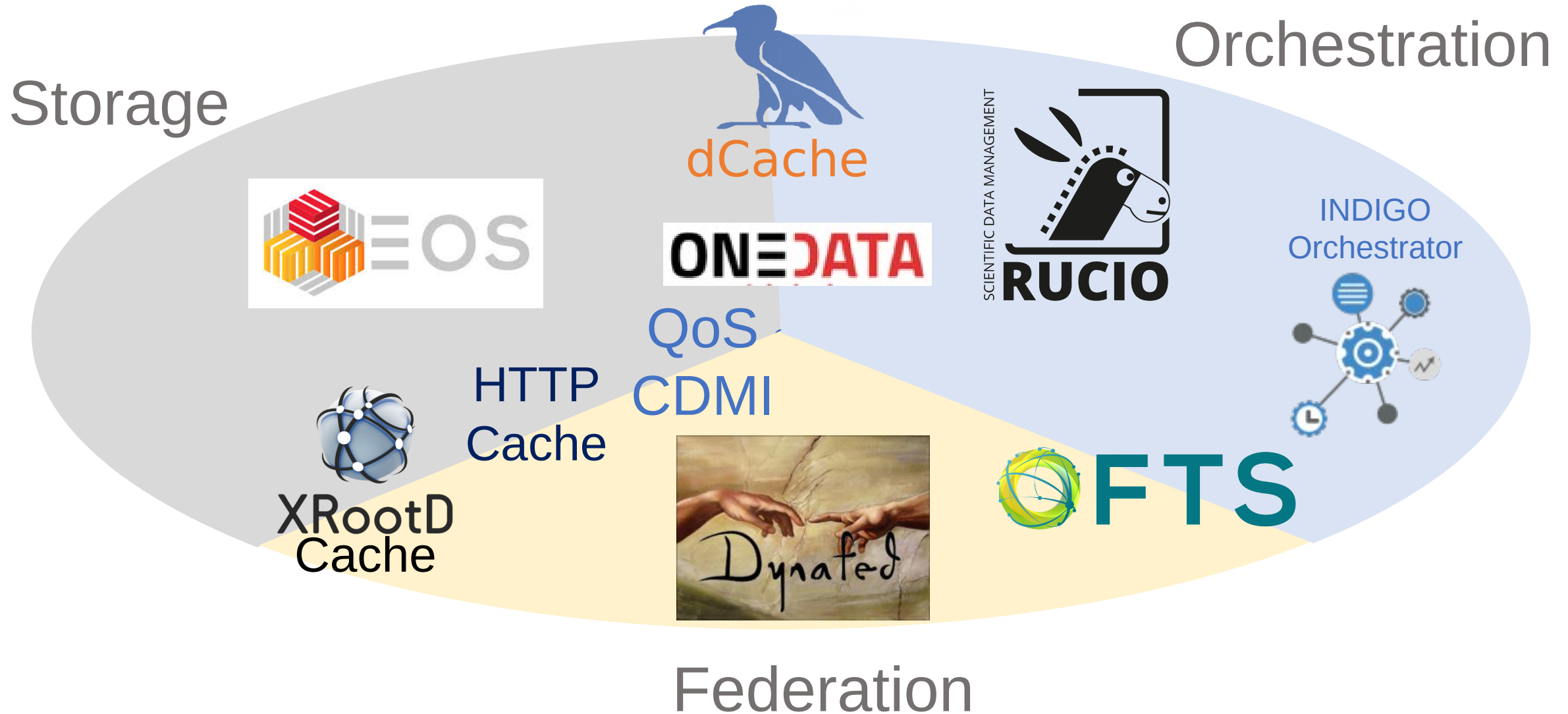
- ➡ Transparent access to remote data without the need of a-priori copy

## ✂ Metadata management

## ✂ Sensitive data handling

- ➡ secure storage and encryption

# Production Level Components

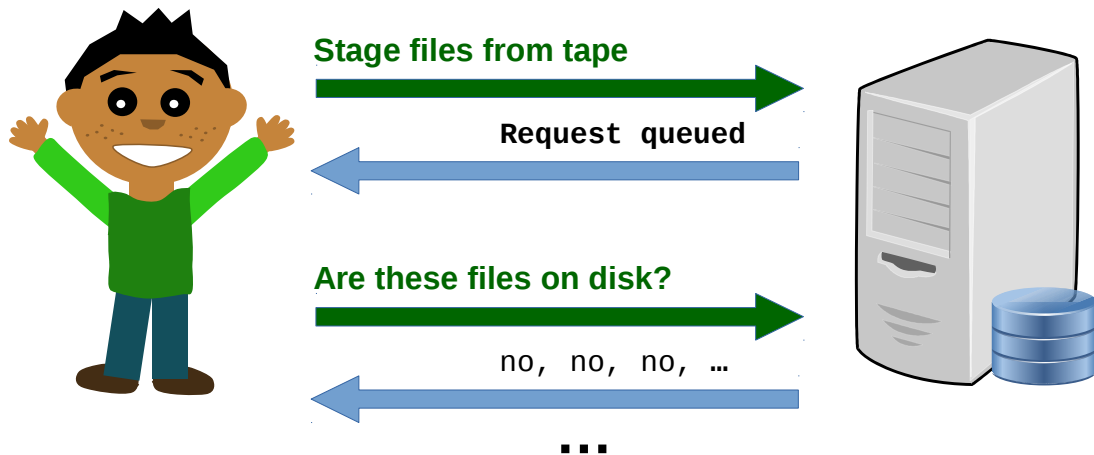
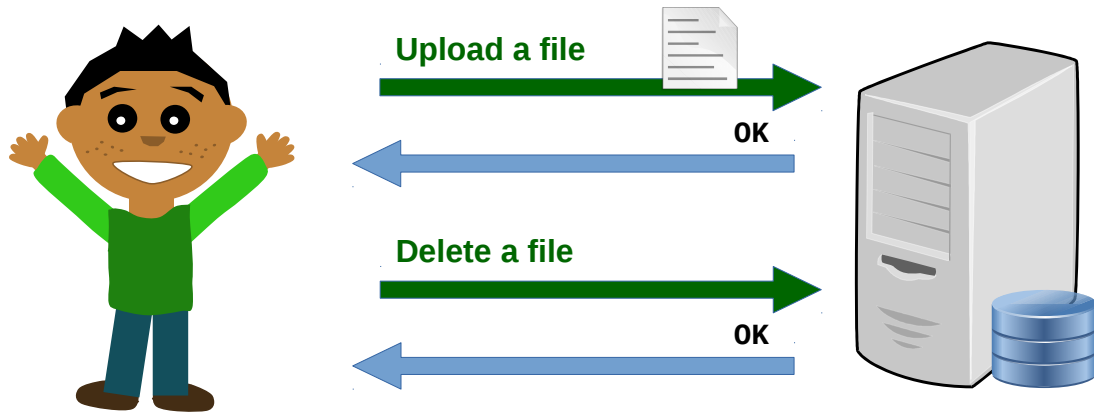


# Software improvements

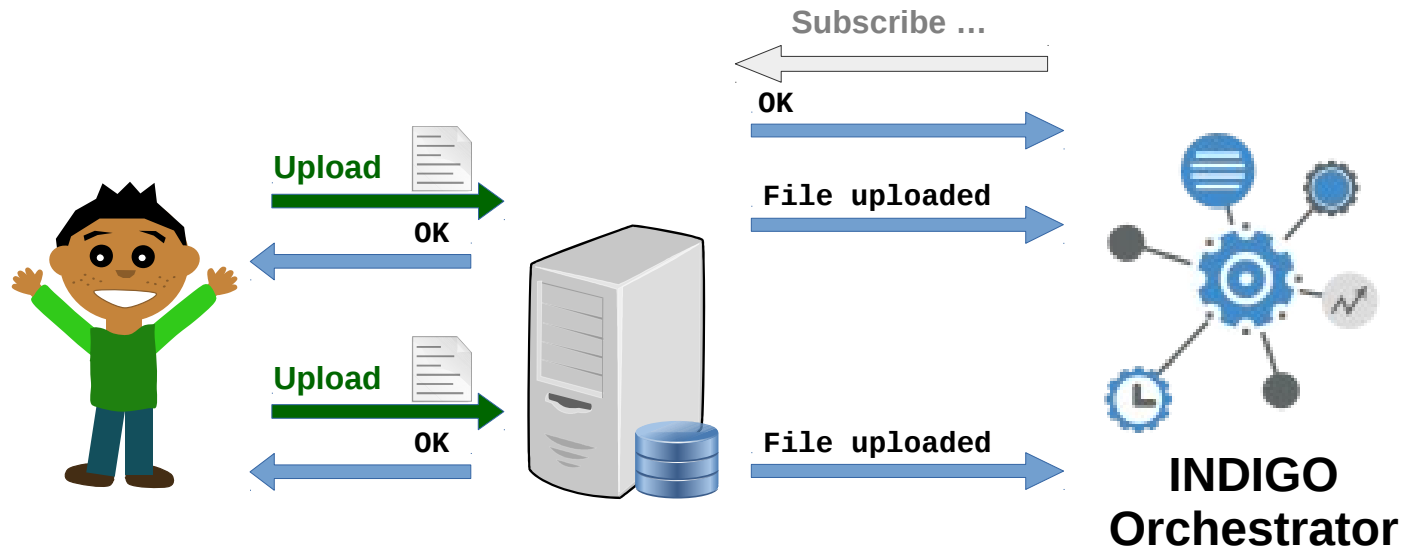




# Storage Events: new way to interact



# New solutions to old problems

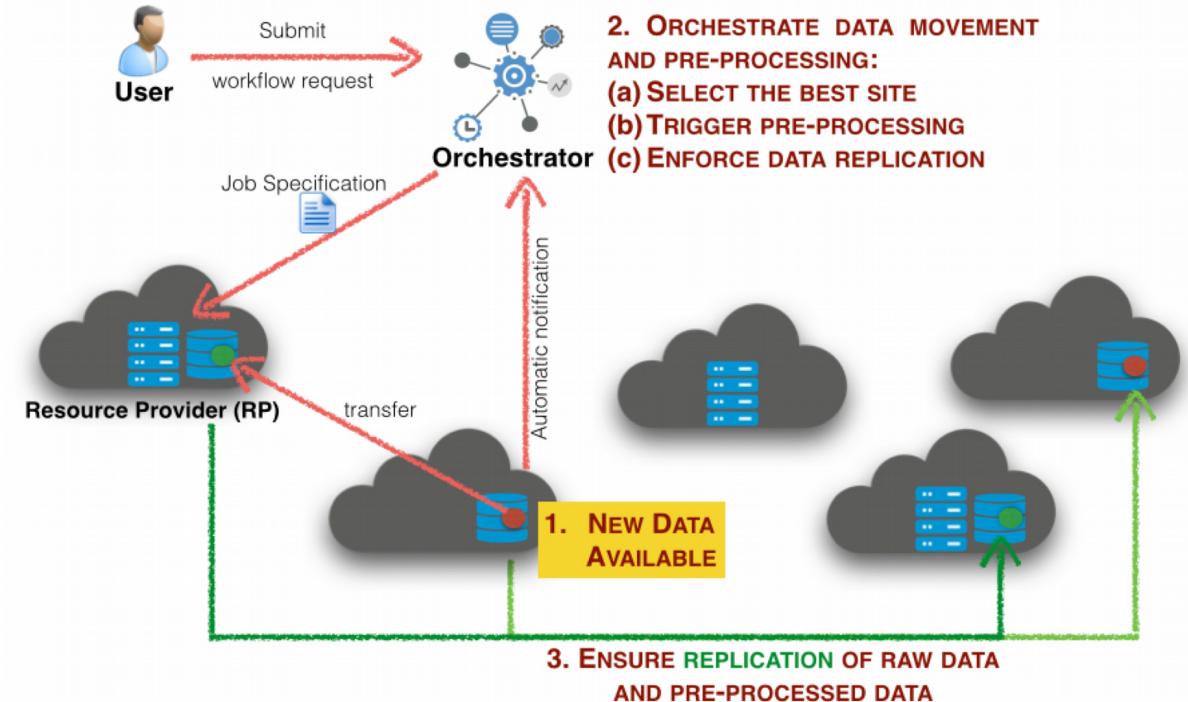
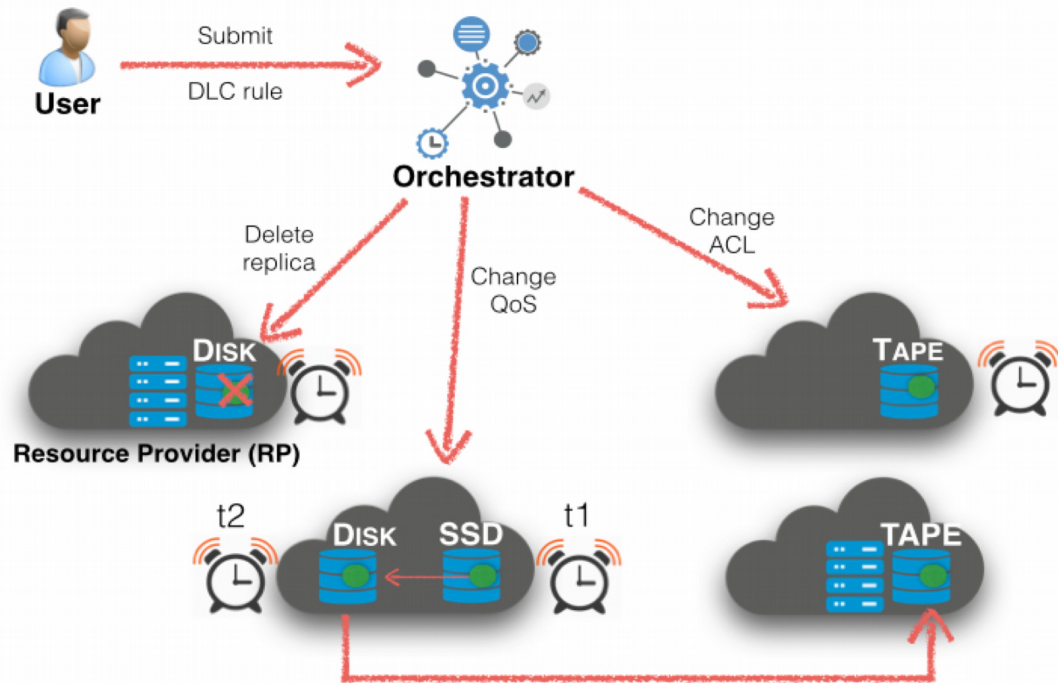


- User- and internally triggered events:
  - Data uploaded
  - Data deleted/renamed/moved
  - Tape flush/stage operations
- Uses: update catalogue, metadata extraction, data normalisation, build derived data, ...
- Two event systems:
  - Site integration (Kafka)
  - Per user events (SSE/inotify)

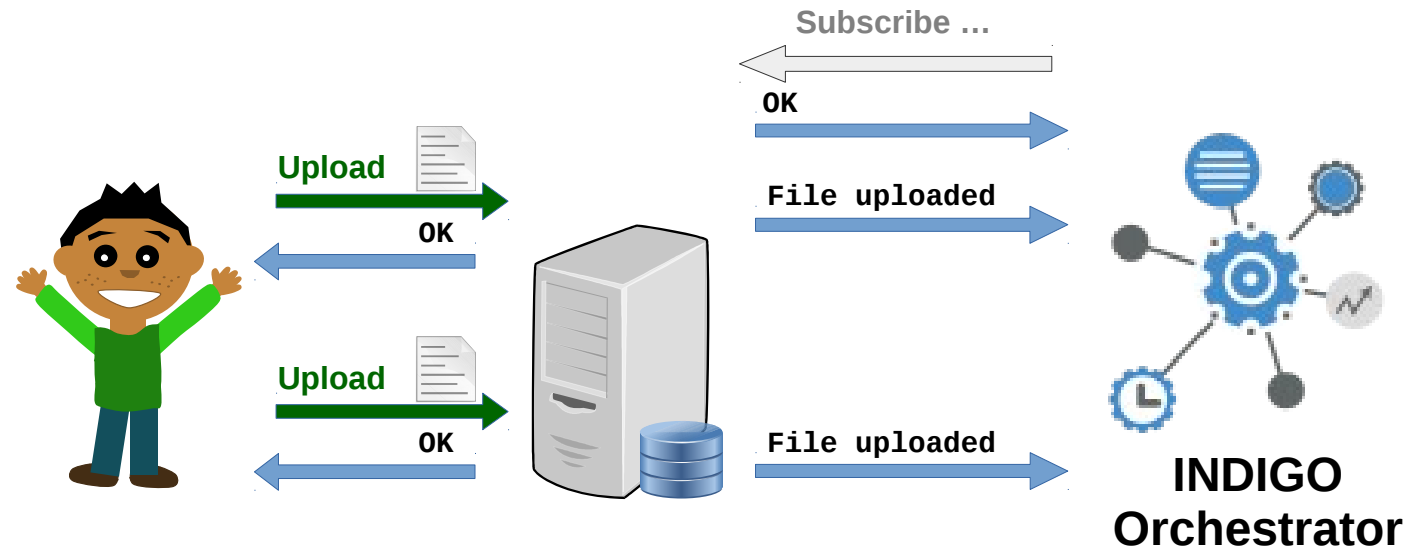


# INDIGO Orchestrator

# Data lifecycle and data management

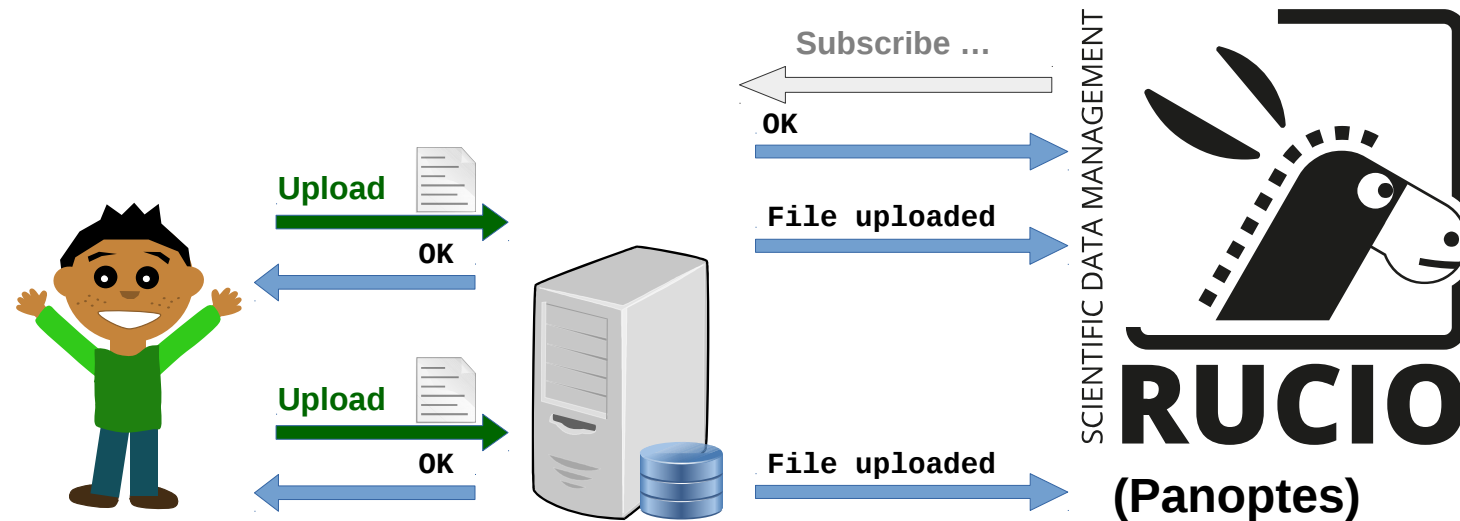


# New solutions to old problems





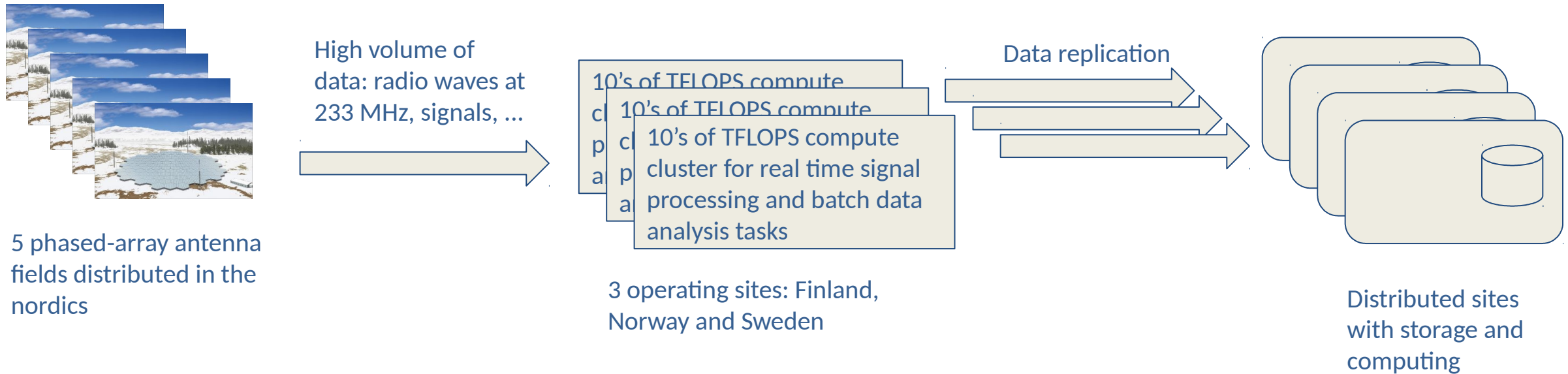
# New solutions to old problems



# EISCAT\_3D use case

Next generation radar for 3D monitoring of the atmosphere and ionosphere

## Data-intensive instruments generates a high volume of data

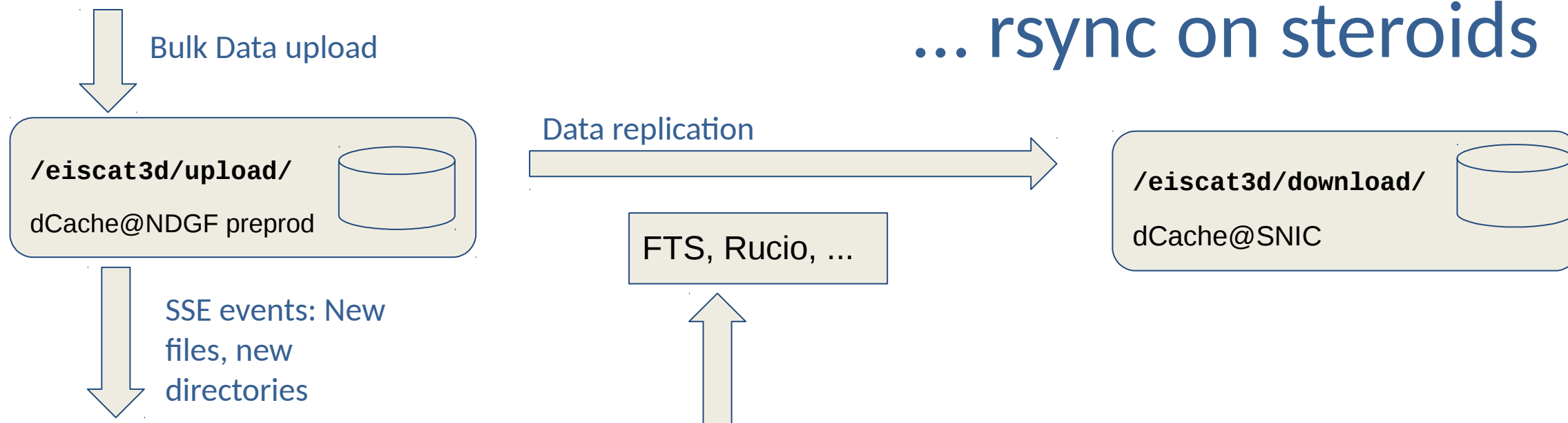


- Researchers need to analyse data and share their results
- How can we automate the data replication ? sync with third party system ,e.g., data management tools and catalogs ?

Side from Vincent Garonne

# Panoptes project: Automatic replication

## ... rsync on steroids



### Panoptes service

- Detect new files and register them into a third party system for bulk replication and data sharing:
  - FTS ✓
  - Rucio ✗

```
> dcache-admin \
--url https://preprod-srm.ndgf.org:3880 \
--x509_proxy /opt/rucio/tools/x509up sync \
--root_path /pnfs/ndgf.org/data/ \
--source https://preprod-srm.ndgf.org/eiscat3d/upload/ \
--destination https://gsiftp.swestore.se/snic/eiscat3d/test \
--fts_host https://rizzo.ndgf.org:8446
```

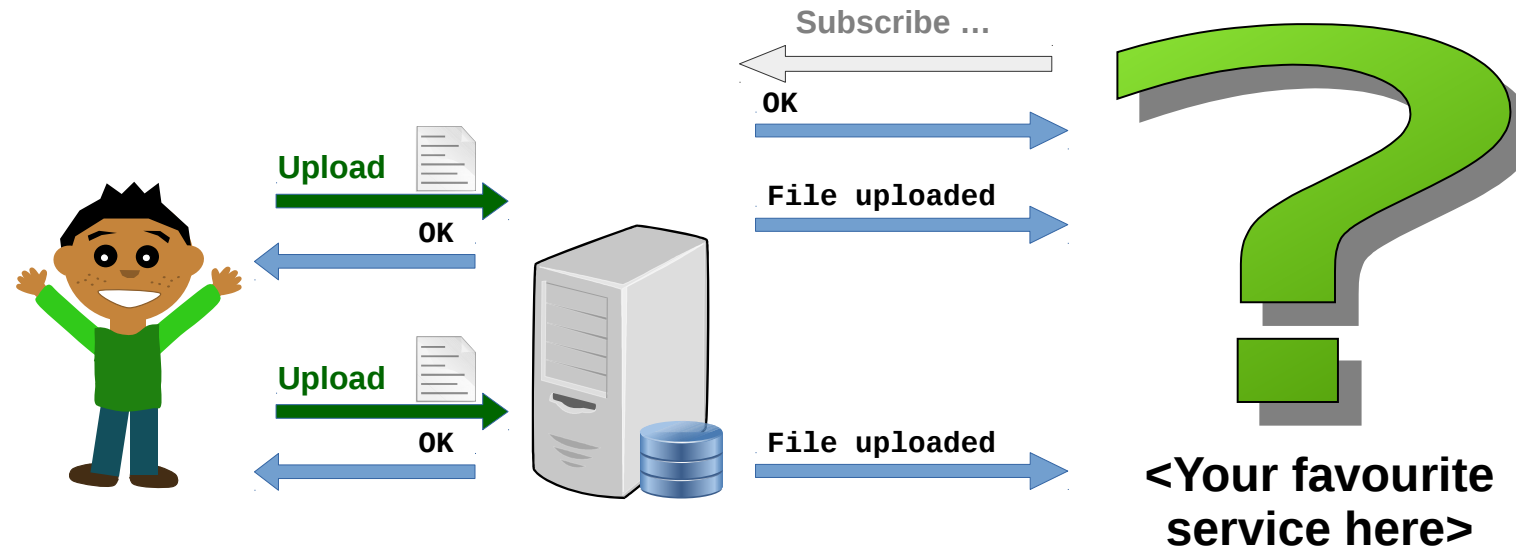
<https://github.com/vingar/dcacheclient>

<https://pypi.org/project/dcacheclient/>

<https://hub.docker.com/r/vingar/dcacheclient>

Side from Vincent Garonne

# New solutions to old problems



# Sneaky demo here







# Dynafed integration with OIDC

## ✂ Integration of support for Dynafed in two relevant OAuth roles

### ➡ Resource server

- ➡ Dynafed will accept access tokens (such as those issued by IAM) and authorise the client on this basis

### ➡ Client

- ➡ Dynafed can invoke the Authorization Code Grant flow if accessed with a browser, redirecting the end user to IAM in order to obtain the relevant credentials

### ➡ NB - Dynafed doesn't speak OAuth to the federated storage systems

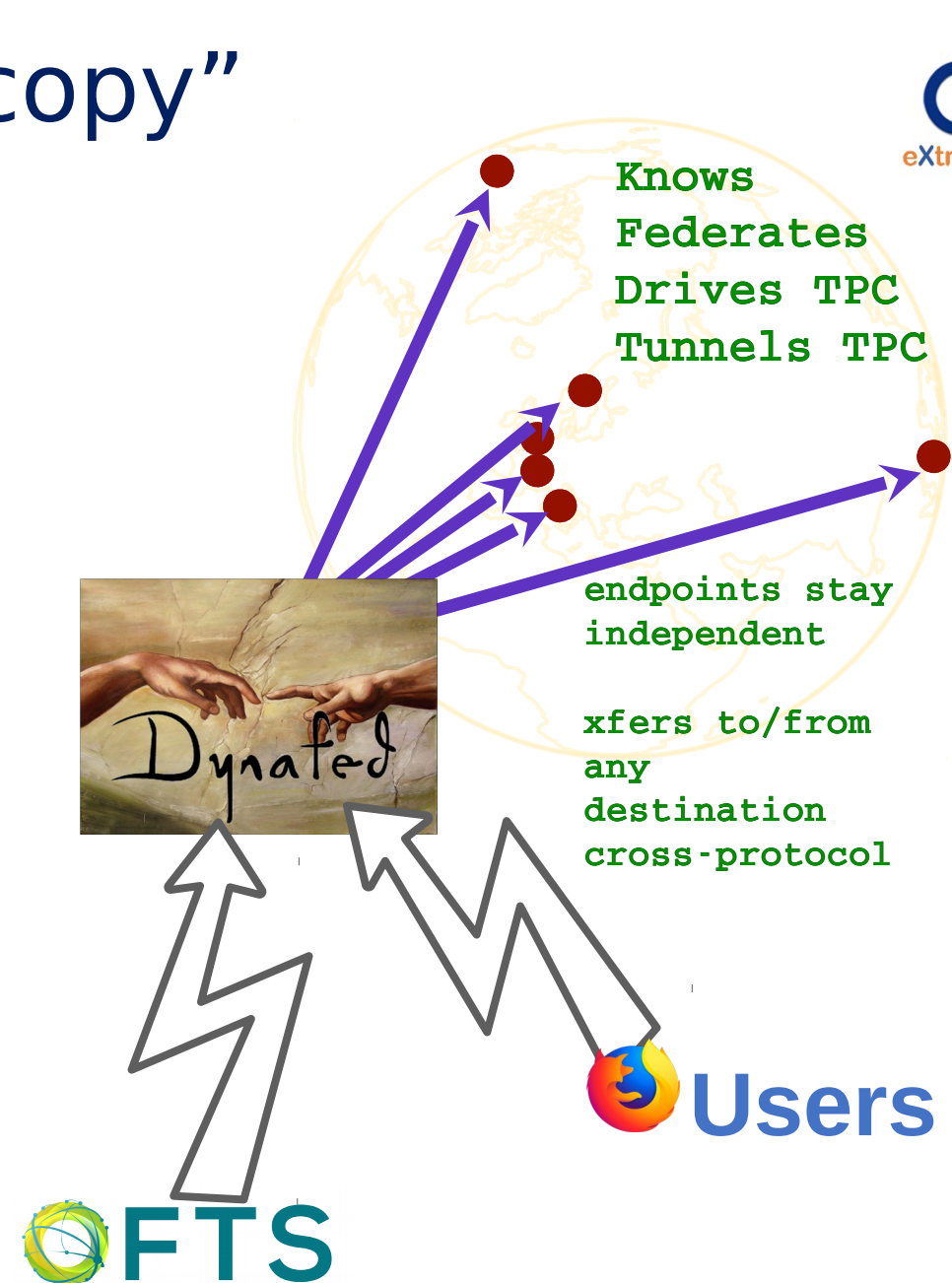
## ✂ Orchestrator integration

- ➡ Orchestrator now integrates with Dynafed, using an access token, so Dynafed is operating as a resource server

# Dynafed “Fourth party copy”

✂ Dynafed can now operate in "fourth party copy" mode.

- ➡ In this scenario it can be instructed to copy files between storage systems, streaming the data through itself.
- ➡ The reference scenario here is Dynafed colocated with cloud storage, allowing data ingestion or export to/from storage systems that don't support TPC.





# FTS integration with OIDC: status

- ✗ FTS Auth/Authz historically done only with X509 proxy certificates and VOMS groups/Roles
  - ➡ not user-friendly
  - ➡ X509 delegation needed
- ✗ 2 types of OIDC integrations implemented
  - ➡ Directly accepting access tokens from users via CLI/REST API
  - ➡ Redirect WebFTS users to IAM in order to acquire a token and using it via the FTS REST API
- ✗ Tokens are used both to authenticate to FTS and to the storages
  - ➡ dCache and Storm are supporting OIDC for now
  - ➡ **X509 delegation is not needed anymore!** (both to FTS and to storages)

# FTS integration with OIDC: plans

- ✂ Understand how to handle groups/roles for certain REST operations
  - ➡ With X509 they are based on VOMS groups/roles
- ✂ Extend REST operations to non-X509 identities
  - ➡ User banning now is based on the X509 User DNs
- ✂ Integration of a Token Translation Service
  - ➡ Present a token – get an X509 certificate
  - ➡ Needed for EOS in XDC, but of course for all the other storages which do not support OIDC yet
    - ➡ Needed also to use other protocols than HTTP

# FTS QoS extension: status and plans

## ✂ FTS extension to steer file's QoS via CDMI interface

- ➡ supported by dCache

## ✂ Status

- ➡ FTS QoS job submission implemented, resulting in a QoS query
  - ➡ Gfal2 CDMI extension implemented (incl python bindings)

## ✂ Plans

- ➡ Full integration of QoS logic
- ➡ Transfer/Transition logic
  - ➡ Use existing multi-hop logic to serialise transfer-then-QoS-transition
  - ➡ Separate QoS daemon under development
- ➡ Validate integration of all QoS methods in gfal2
- ➡ Definition of FTS QoS interface for Rucio/Orchestrator
- ➡ Classification of QoS classes for XDC



Thanks for listening!

Backup slides

# EOS - Working with an imported file (backup slide)

✕ MGM keeps track of additional information, such as *lpath* and *ctime*.

