DISTRIBUTED "CLOUD" STORAGE SERVICES

Giacinto DONVITO

INFN-Bari

OUTLINE

- Storage services in EGI Fed Cloud
    - Technological solutions
    - How it works
- AAI
- First experience on using EGI Cloud Services
- Other "cloud-like" Storage development
- Open Issues

STORAGE SERVICES IN EGI FED CLOUD

- Two types of approach:
  - Object Storage
    - Remotely exploitable
    - PUT/GET calls via HTTP/HTTPS
  - Block Storage:
    - Standard Disk Device
    - The users can format the device depending on his/her needs
    - Fully posix compliant
    - Usable locally only at a given cloud site
- There is also ephemeral space on the virtual machine itself

OBJECT STORAGE IN EGI FED CLOUD

- The Object storage services are accessible via X509/VOMS

- Usually implemented with CDMI or Swift native APIs

- This kind of services could be easily exploited by libraries (programmatic APIs)

    - "curl"

- Mostly OpenStack Swift based

- Swift is also able to implement "sort-of" encryption

- It is possible to both stream the file or just read few chunks

- Also partial upload is possible

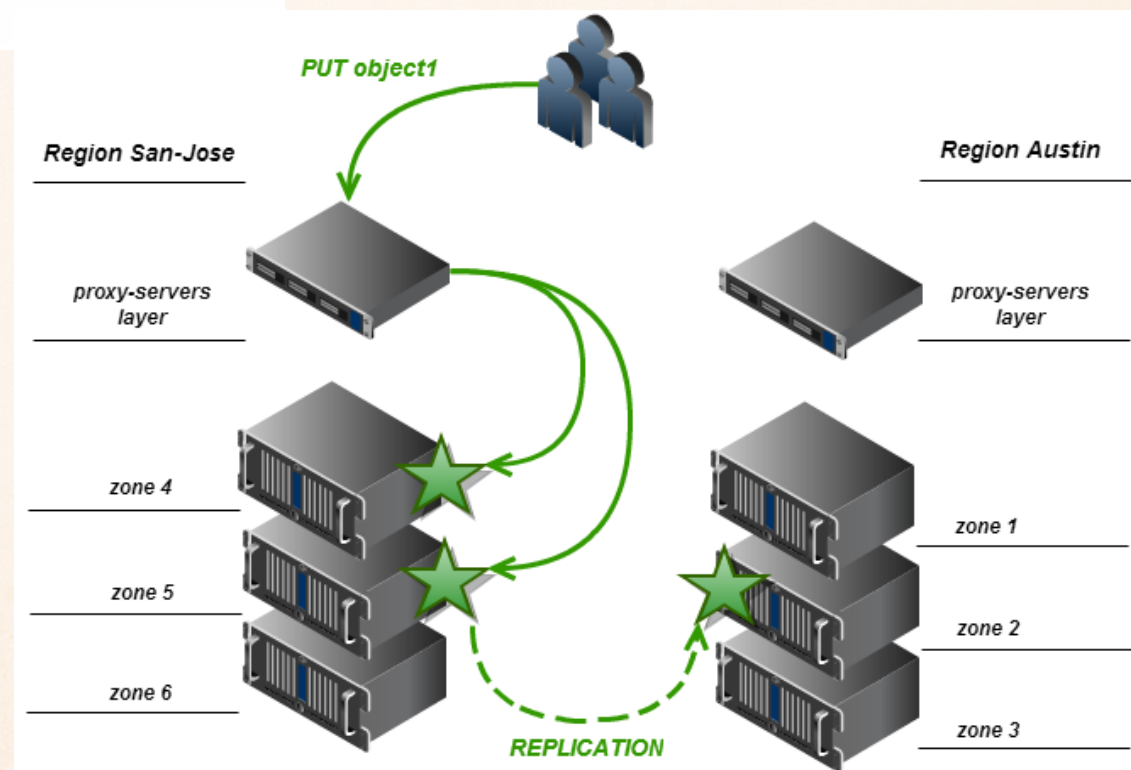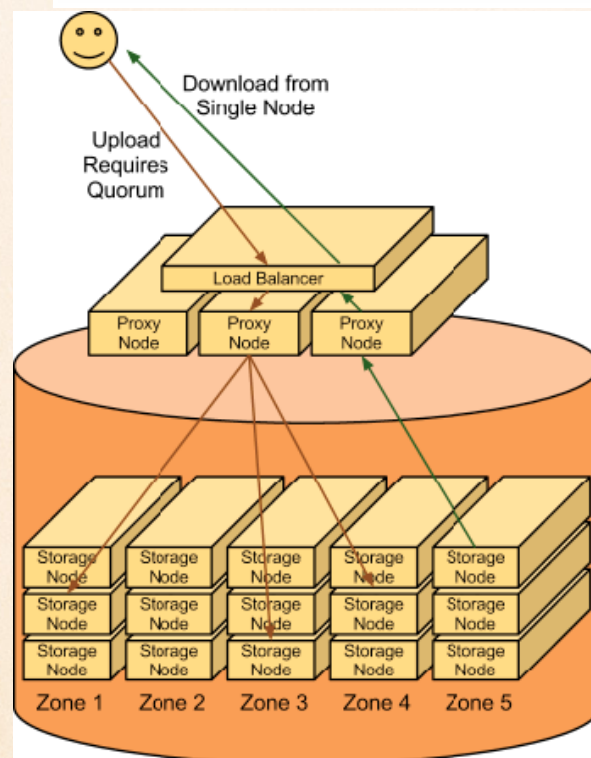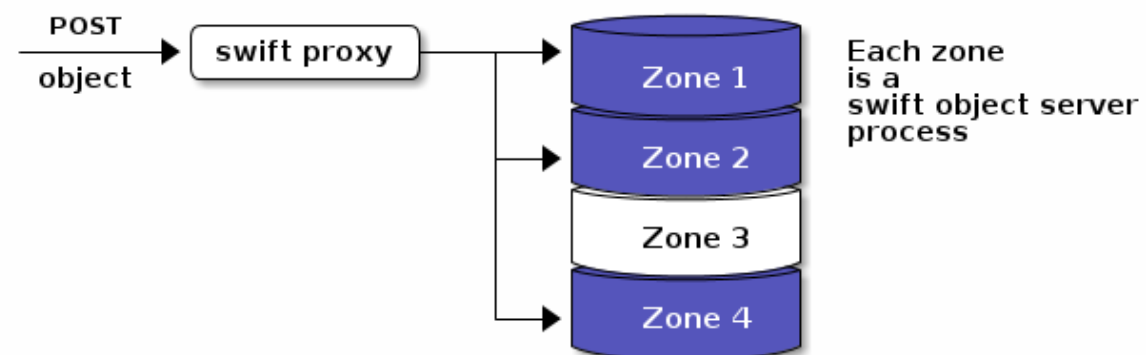# OBJECT STORAGE IN EGI FED CLOUD

- The object could be "enriched" with the use of metadata

- Data are organized per containers and objects

- This storage space is often configured with a software replica

  - 2-3 copies on single disk without raid

- It is possible to share data with other users:

  - ACLs could be arranged per objects or containers

  - or made the data publicly available

- Only the real used space is accounted

- Provide a virtually un-limited amount of storage

The Swift example

OBJECT STORAGE IN EGI FED CLOUD

- [root@pccms64 ~]# ldapsearch -x -H ldap://egee-bdii.cnaf.infn.it:2170 -b o=glue | grep -i cdmi | grep GLUE2EndpointID | grep -v "dn:"

- GLUE2EndpointID: https://server2-epsh.unizar.es/swift/v1_CDMI_1.0.1_X509-VOMS

- GLUE2EndpointID: https://server3-eupt.unizar.es/swift/v1_CDMI_1.0.1_X509-VOMS

- GLUE2EndpointID: http://cdmi.cloud.gwdg.de:4001:_CDMI_1.0.1_X509-VOMS

- GLUE2EndpointID: https://storage-service01.example.org:8080_CDMI_1.0.1_X509-VO

- GLUE2EndpointID: https://storage1.ui.savba.sk:8080_CDMI_1.0.1_X509-VOMS

- GLUE2EndpointID: https://storage2.ui.savba.sk:8080_CDMI_1.0.1_X509-VOMS

- GLUE2EndpointID: https://swift.zam.kfa-juelich.de:8888/cdmi/_CDMI_1.0.1_X509-V

- GLUE2EndpointID: https://prisma-swift.ba.infn.it:8080_CDMI_1.0.1_X509-VOMS

- GLUE2EndpointID: https://okeanos-cdmi.hellasgrid.gr/pithos/_CDMI_1.0.2_X509-VO

OBJECT STORAGE IN EGI FED CLOUD

- First of all you have to create a Token:
  - `curl -k --cert /tmp/x509up_u50865 -d '{"auth":{"voms": true, "tenantName": "EGI_FCTF"}}' -H "Content-type: application/json" https://egi-cloud.zam.kfa-juelich.de:5000/v2.0/tokens`
  - `export TOKEN="MIIRZAY…………………CRv-UVQ=="`

- List all the available container:
  - `curl --insecure -X GET -H 'X-Auth-Token:'$TOKEN https://swift.zam.kfa-juelich.de:8888/cdmi/AUTH_df37f5b1ebc94604964c2854b9c0551f/`

- Create a new container:
  - `curl --insecure -X PUT -H 'X-Auth-Token:'$TOKEN https://swift.zam.kfa-juelich.de:8888/cdmi/AUTH_df37f5b1ebc94604964c2854b9c0551f/giacinto_test/`

- Upload a file to an existing container:
  - `curl -k -X PUT -H 'X-Auth-Token:'$TOKEN -H 'Content-Type: application/cdmi-object' -H 'Accept: application/cdmi-object' https://swift.zam.kfa-juelich.de:8888/cdmi/AUTH_df37f5b1ebc94604964c2854b9c0551f/giacinto_test/test.txt -T ./testfile.txt`

- And read it back:
  - `curl --insecure -X GET -H 'X-Auth-Token:'$TOKEN https://swift.zam.kfa-juelich.de:8888/cdmi/AUTH_df37f5b1ebc94604964c2854b9c0551f/giacinto_test/test.txt -o test.test.txt`
  - `curl -k https://swift.zam.kfa-juelich.de:8888/cdmi/AUTH_df37f5b1ebc94604964c2854b9c0551f/giacinto_test/test.txt`

OBJECT STORAGE IN EVERYDAY LIFE
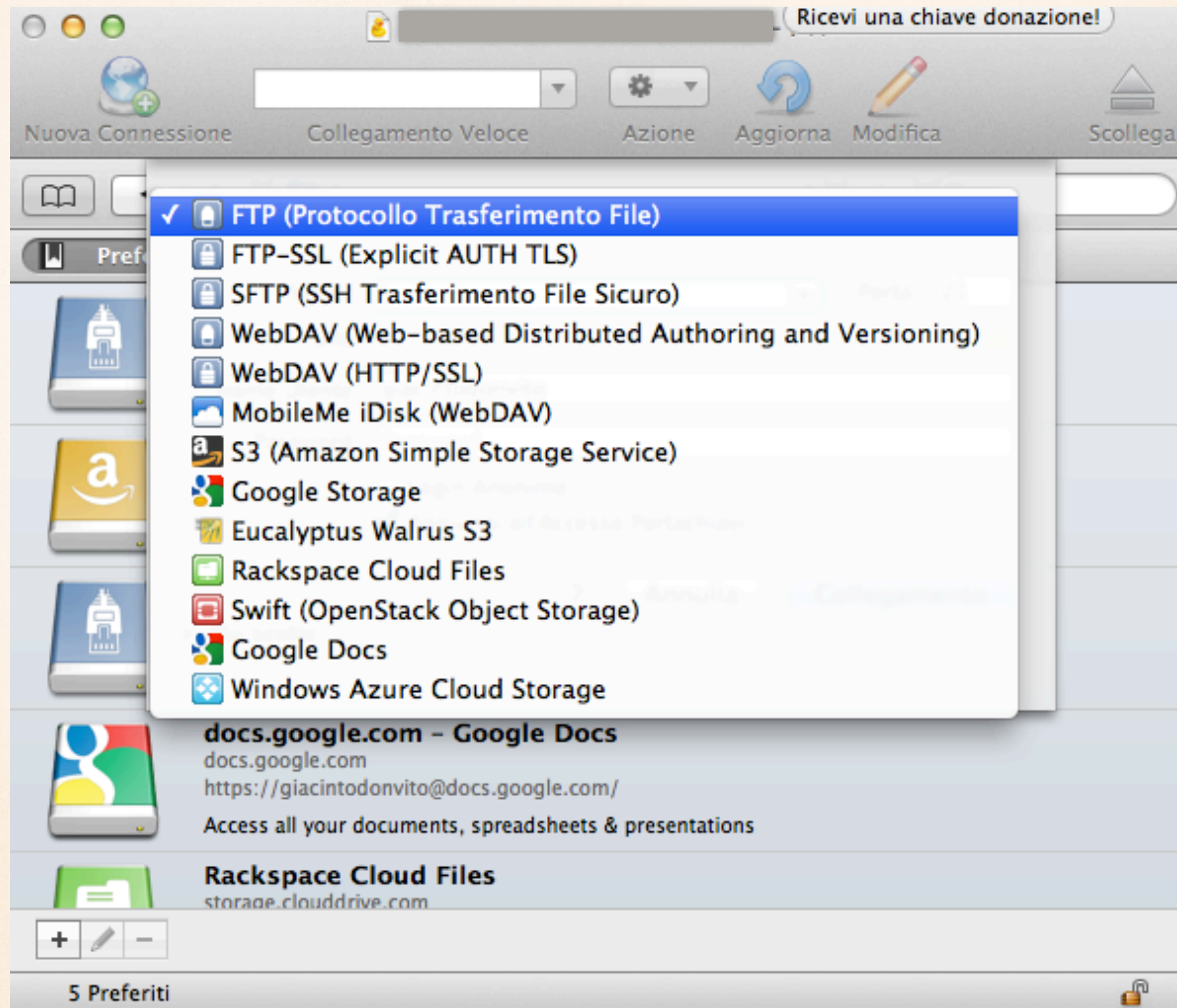
BLOCK STORAGE IN EGI FED CLOUD

- The block storage provide a Virtual Machine with additional disk space

    - As attaching an USB drive o a remote virtual machine

- It is possible to use standard posix I/O

- It is possible to exploit this space for any kind of data (RDMS, NoSQL, etc)

- This is strictly linked to a given site, could not be moved out of it

- It is possible to manage it via OCCI interface as for the virtual machine

    - The same X509/VOMS authorization mechanism

- This storage space survive to the Virtual Machine

BLOCK STORAGE IN EGI FED CLOUD

- The disk space is accounted for the entire block storage device, regardless how much of it is currently in use within the VM

- Depending on the storage implementation at a given site:

  - There is a limit on the number of block storage devices you can attach on a VM

  - There is a limit to the maximum size of such virtual disks

    - usually no more than 2-5TB

- Each block storage could be attached to only one VM at the time

- Several different implementation techniques available, both proprietaries or OpenSource. Practically transparent for the users.

  - Open: CEPH, GlusterFS, simple iSCSI on LVM

  - Proprietary: Netapp, DELL, IBM, etc

BLOCK STORAGE IN EGI FED CLOUD

- First of all you need to create the device:

  - ```
    occi -e <site_occi_endpoint> --auth x509 --user-cred <proxy_certificate> --voms --
    action create --resource storage -t occi.storage.size='num(<storage_size_in_gb>)',
    occi.core.title= <storage_resource_name> http://site.occi.endpoint/storage/
    <storage_resource_id>
    ```

- Then you can attach to a VM

  - ```
    occi -e <site_occi_endpoint> --auth x509 --user-cred <proxy_certificate> --voms --
    action link --resource /compute/<vm_id> --link /storage/<storage_resource_id>
    ```

- Or create a new VM with the disk attached

  - ```
    occi -e <site_occi_endpoint> --auth x509 --user-cred <proxy_certificate> --voms --
    action create --resource compute [...] --link /storage/<storage_resource_id>
    ```

- The disk could be detached to a VM and attached to another one

- Or could be deleted

- Object Storage:

  - Not many site already providing it

  - Quite good performance with limited client-side effort
    - 20-30MB/s WAN (single stream, simple client, insecure)

  - noticeable overhead for small files:
    - about 2.3 secs per file over WAN

  - It is possible to exploit it with widely known client (curl, wget) and without VOMS certificate
    - Using temporary tokens

- ❖ Block Storage:

  - ❖ Most of the sites support this storage type

  - ❖ The performance largely depend on the site implementation but could be as good as a local disk

    - ❖ it is not difficult to run at 100MB/s for sequential I/O on a single block device

  - ❖ Quite straightforward to support legacy application with permanent storage

    - ❖ Very well suited for databases, application server, etc

STORAGE IN EGI FED CLOUD

Block storage

/home

Object Storage

HTTP/REST
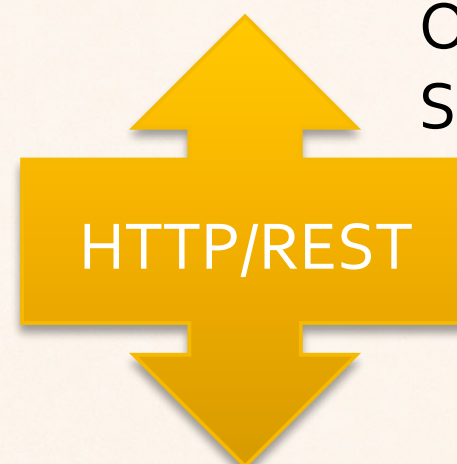
Reliable, secure, high-performance *file transfer* and *replication*

- **"Fire-and-forget" transfers**

- **Automatic fault recovery**

- **Seamless security integration**

- **Powerful GUI and APIs**

2 Globus moves and syncs files

Data Source → Data Destination

1 User initiates transfer request

3 Globus notifies user
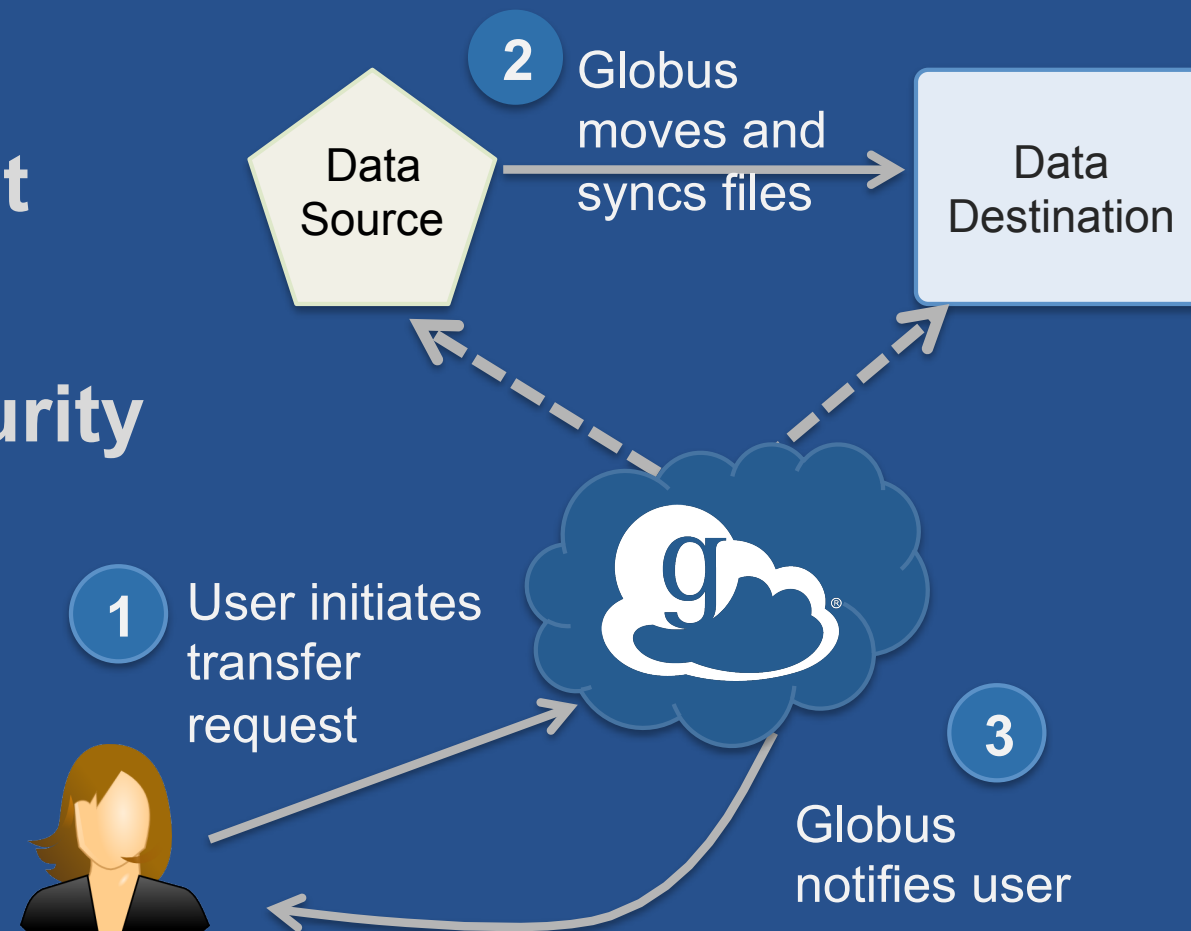
Simple, secure *sharing* off existing storage systems

- **Easily share large data with any user or group**

- **No cloud storage required**

2 Globus tracks shared files; no need to move files to cloud storage!

Data Source

1 User A selects file(s) to share, selects user or group, and sets permissions

3 User B logs in to Globus and accesses shared file
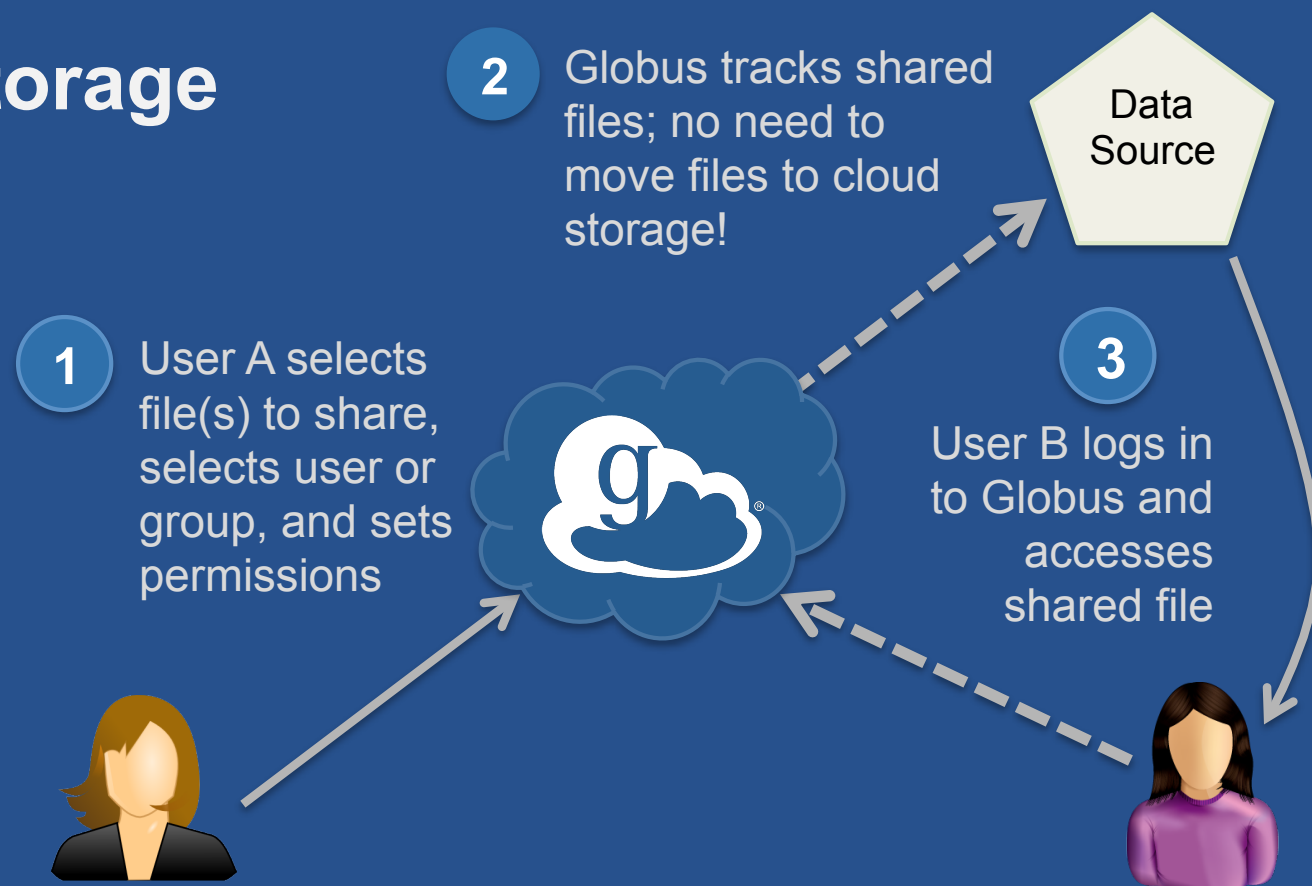
What is WebFTS?

- Web based tool to transfer files between grid/cloud storages

- Modular protocol support
  - gsiftp, http(s), xrootd and srm
  - Cloud extensions: dropbox, CERNBox

- Development funded by

Success Stories

- WebFTS has been successfully tested to transfer from/to:

    - **EUDAT B2Stage** (iRODS DSI)
    - Any gsiftp/webdav/xrootd aware grid storage (**DPM, dCache, Castor, EOS, Storm**)
    - **HPC Titan** @ Oak Ridge National Lab (ongoing)
        - https://www.olcf.ornl.gov/titan/

- Under evaluation by **LHCb**

Transfer interface

Dropbox plugin

- Server side the development of a plugin for the metadata management and I/O operations was needed:

  - FTS REST integrates the plugin to perform metadata management operations

  - FTS3 server uses the plugin to perform the transfers:

    - GridFTP <-> dropbox
    - Http(s) <-> dropbox

Links

- Online service accessible:
  - https://webfts.cern.ch ← try now!
  - User certificate in your browser

- User guide, F.A.Q:
  - Online guided-tour
  - http://fts3-service.web.cern.ch/documentation/webfts

- Official support & code
  - fts-support@cern.ch
  - https://github.com/cern-it-sdc-id/webfts

STILL OPEN ISSUES

- Object Storage:
  - You have to look for resources
    - into the global BDII
  - No federation among different instance on different sites
  - No simple (neither) automatic way of moving files from one storage to another

- Block Storage:
  - Not possible at the moment to move a block from one cloud instance to another
  - No "automation" features available at the moment, via OCCI, for building a complex template comprising VM and Disk

- There are others tools that on the long-term could be useful in the cloud environment, but development are still needed