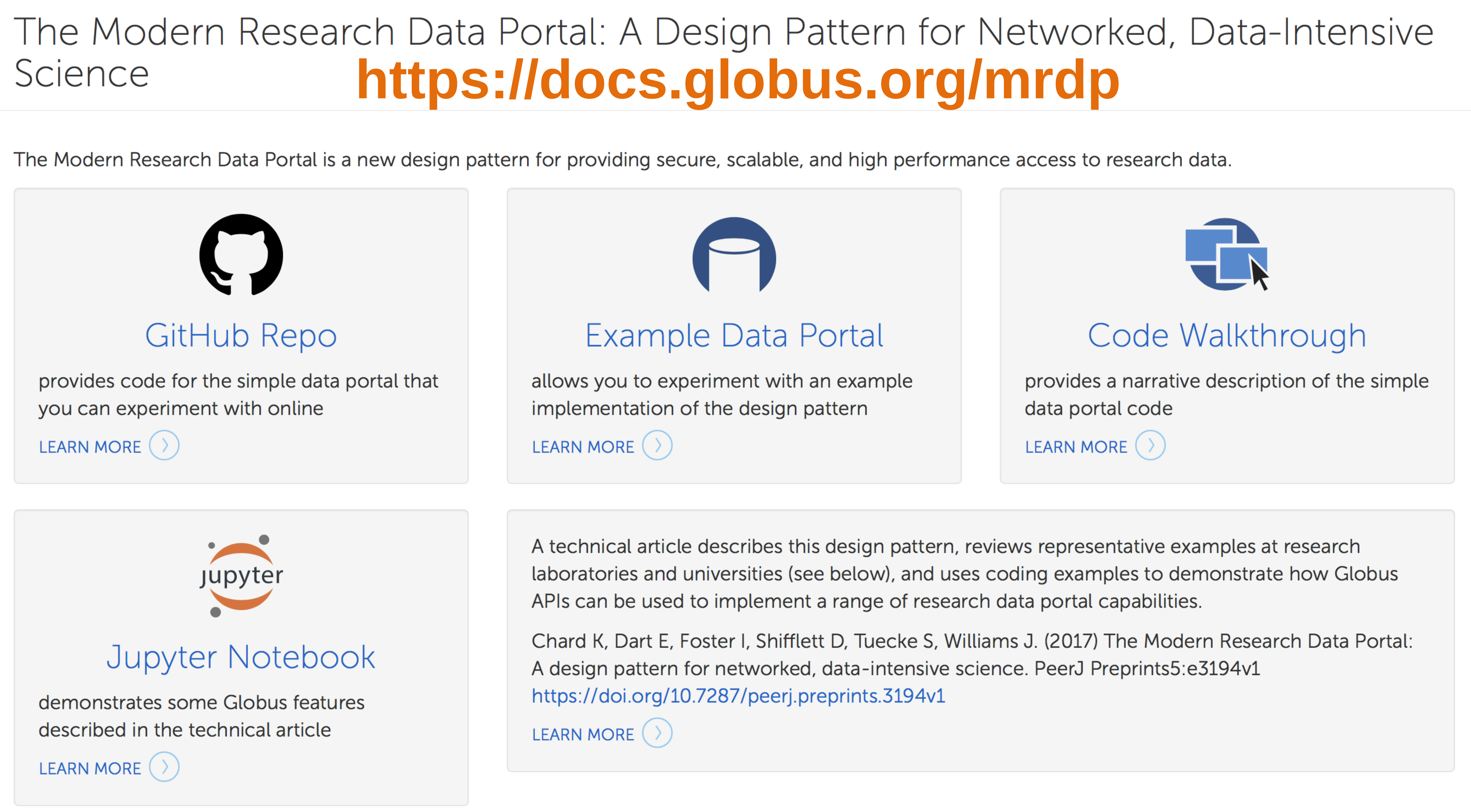
Building the Model Research Data Portal

Ian Foster

foster@uchicago.edu

with Kyle Chard, Eli Dart, Steve Tuecke, Jason Williams



The Modern Research Data Portal: A Design Pattern for Networked, Data-Intensive Science

https://docs.globus.org/mrdp

The Modern Research Data Portal is a new design pattern for providing secure, scalable, and high performance access to research data.

GitHub Repo

provides code for the simple data portal that you can experiment with online

LEARN MORE

Example Data Portal

allows you to experiment with an example implementation of the design pattern

LEARN MORE

Code Walkthrough

provides a narrative description of the simple data portal code

LEARN MORE

Jupyter Notebook

demonstrates some Globus features described in the technical article

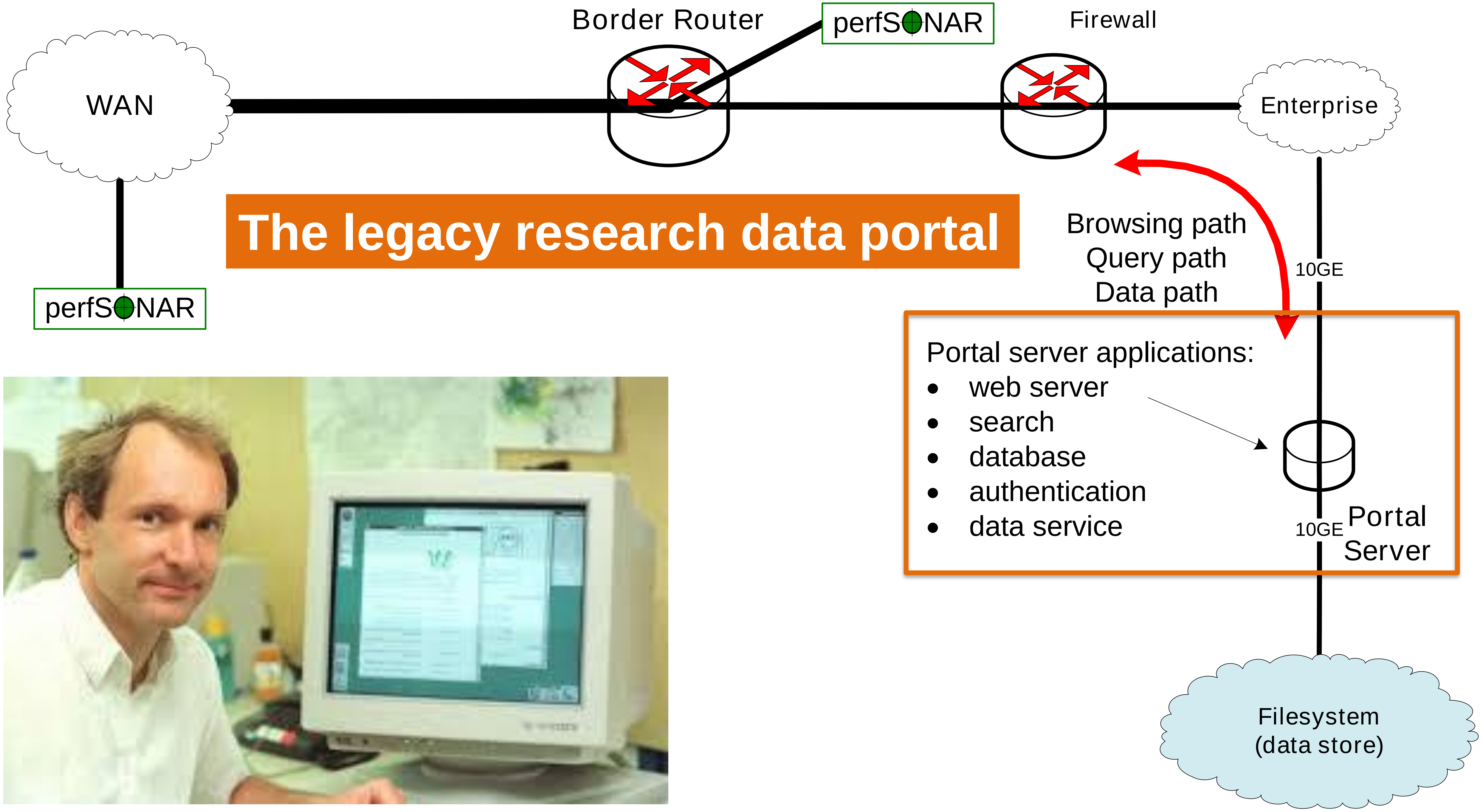
LEARN MORE

A technical article describes this design pattern, reviews representative examples at research laboratories and universities (see below), and uses coding examples to demonstrate how Globus APIs can be used to implement a range of research data portal capabilities.

Chard K, Dart E, Foster I, Shifflett D, Tuecke S, Williams J. (2017) The Modern Research Data Portal: A design pattern for networked, data-intensive science. PeerJ Preprints5:e3194v1 https://doi.org/10.7287/peerj.preprints.3194v1

LEARN MORE

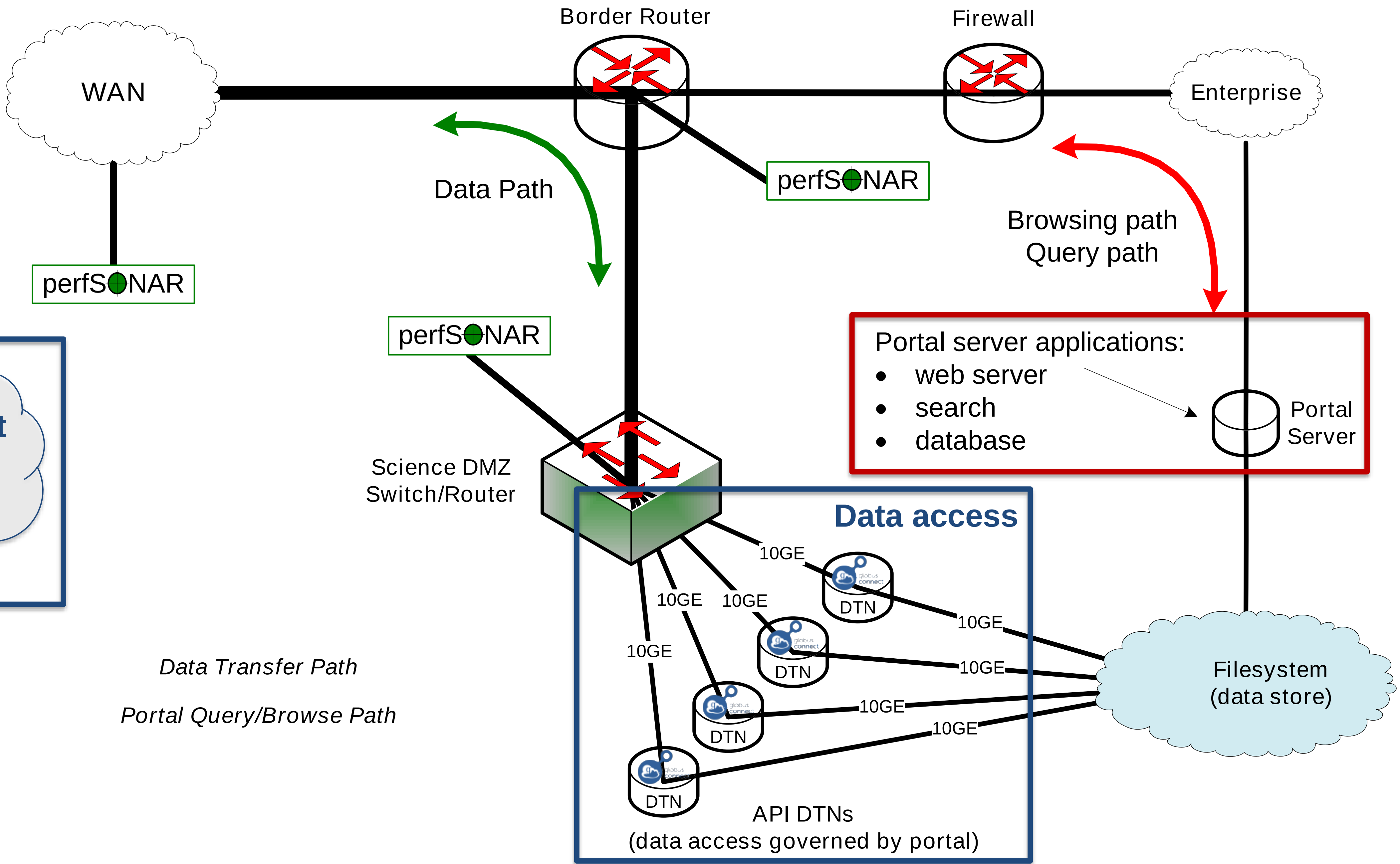
The legacy research data portal



Portal server applications:

- web server
- search
- database
- authentication
- data service

The modern research data portal



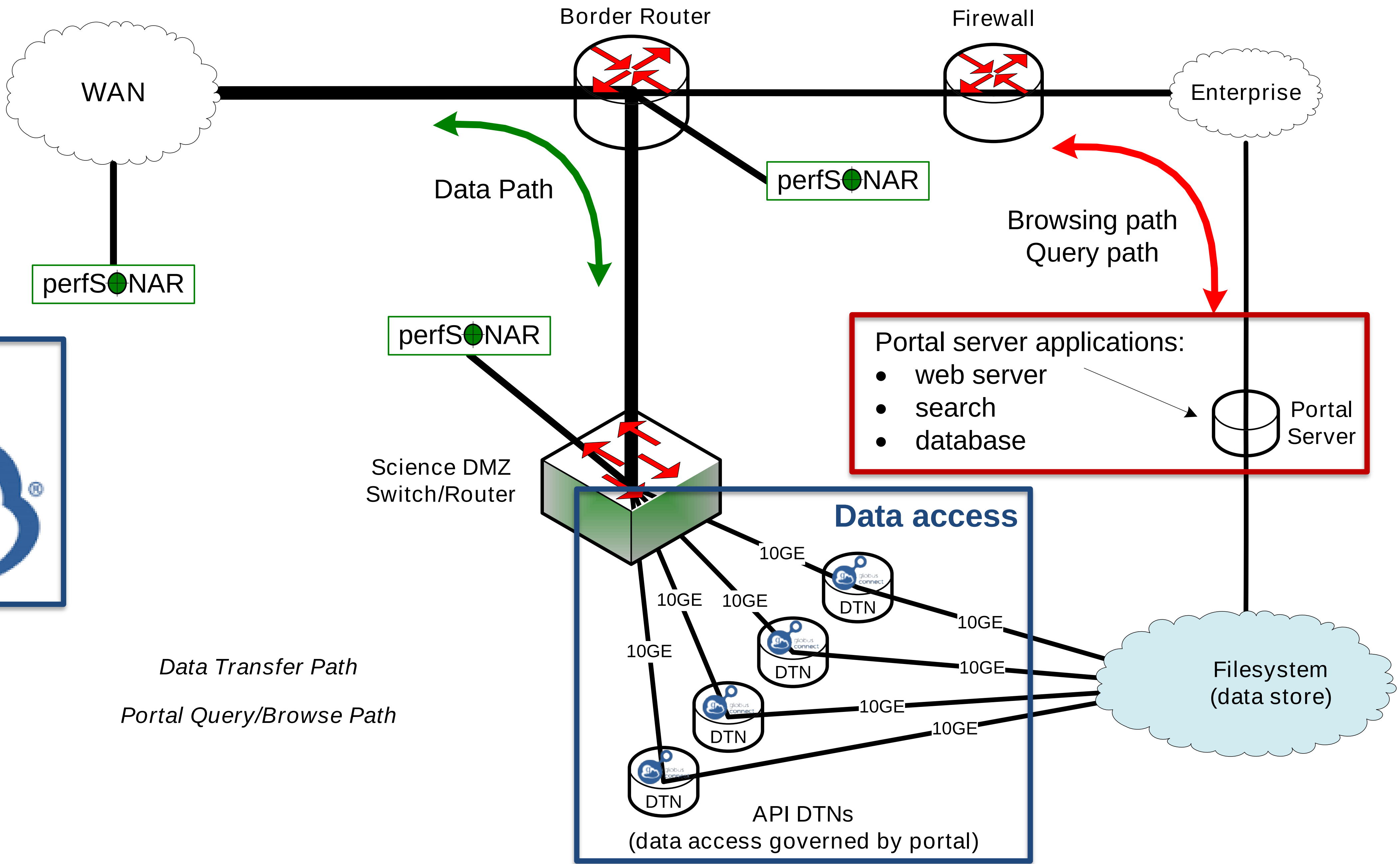
Data Transfer Path

Portal Query/Browse Path

A key message: Outsource all that you can

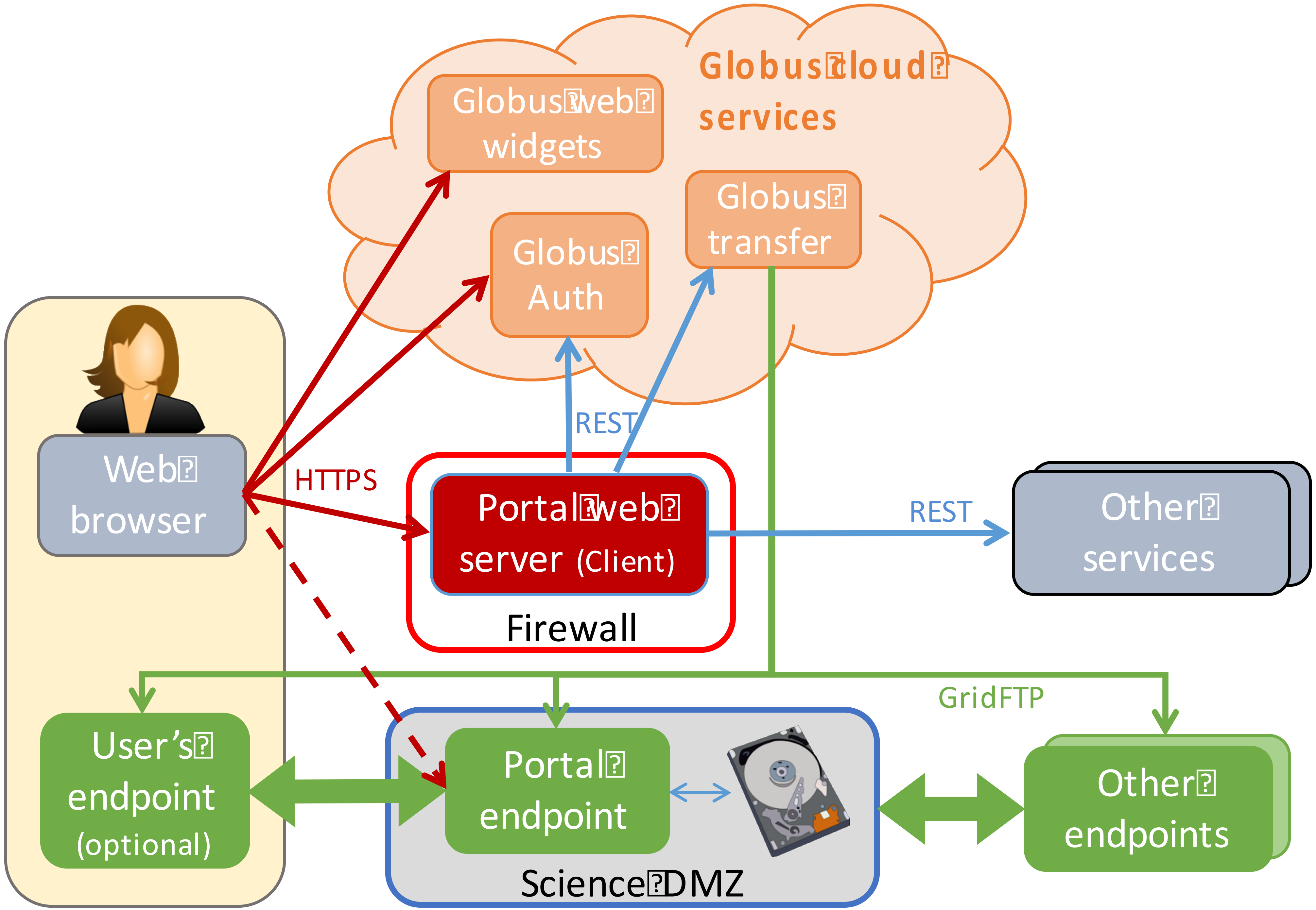
- **Outsource responsibility for <u>determining user identities</u>**
- **Outsource <u>control over who can access</u> different data and services within the portal**
- **Outsource responsibility for <u>managing data uploads and downloads</u> between various locations and storage systems**
- **Leverage <u>standard web user interfaces</u> for common user actions**

The modern research data portal



Data Transfer Path

Portal Query/Browse Path


Globus cloud services
Globus web widgets
Globus Auth
Globus transfer
Web browser
HTTPS
REST
Portal web server (Client)
Firewall
REST
Other services
GridFTP
User's endpoint (optional)
Portal endpoint
Science DMZ
Other endpoints

A simple example of MDRDP logic

User

Authenticate

Identify data

Download data

Portal

1. **Create “shared endpoint”**
2. **Copy data to shared endpoint**
3. **Sets permissions on shared endpoint for user; notify user**
4. **(Eventually) delete shared endpoint**

1. **Create "shared endpoint"**
2. Copy data to shared endpoint
3. Set permissions on shared endpoint for user; notify user

…

4. (Eventually) delete shared endpoint

```
from globus_sdk import TransferClient, TransferData
from globus_sdk import AuthClient
import sys, random, uuid

def rdp(host_id,       # Endpoint for shared endpoint
        source_path,   # Directory to copy data from
        email):        # Email address to share with
    tc = TransferClient()
    ac = AuthClient()
    tc.endpoint_autoactivate(host_id)

    # (1) Create shared endpoint:
    # (a) Create directory to be shared
    share_path = '/~/' + str(uuid.uuid4()) + '/'
    tc.operation_mkdir(host_id, path=share_path)
    # (b) Create shared endpoint on directory
    shared_ep_data = {
      'DATA_TYPE': 'shared_endpoint',
      'host_endpoint': host_id,
      'host_path': share_path,
      'display_name': 'RDP shared endpoint',
      'description': 'RDP shared endpoint'
    }
    r = tc.create_shared_endpoint(shared_ep_data)
    share_id = r['id']
```

Connect to storage system

1. Create “shared endpoint”
2. Copy data to shared endpoint
3. Set permissions on shared endpoint for user; notify user

…

4. (Eventually) delete shared endpoint

```
# (2) Copy data into the shared endpoint
tc.endpoint_autoactivate(share_id)
tdata = TransferData(tc, host_id, share_id,
     label='RDP copy', sync_level='checksum')
tdata.add_item(source_path, '/', recursive=True)
r = tc.submit_transfer(tdata)
tc.task_wait(r['task_id'], timeout=1000,
             polling_interval=10)

# (3) Enable access by user
r = ac.get_identities(usernames=email)
user_id = r['identities'][0]['id']
rule_data = {
  'DATA_TYPE': 'access',
  'principal_type': 'identity', # Grantee is
  'principal': user_id,         #  a user.
  'path': '/',                  # Path is /
  'permissions': 'r',           # Read-only
  'notify_email': email,        # Email invite
  'notify_message':             # Invite msg
      'Requested data are available.'
}
tc.add_endpoint_acl_rule(share_id, rule_data)

# (4) Ultimately, delete the shared endpoint
tc.delete_endpoint(share_id)
```

1. Create “shared endpoint”
2. Copy data to shared endpoint
3. **Set permissions on shared endpoint for user; notify user**

…

4. (Eventually) delete shared endpoint

```
# (2) Copy data into the shared endpoint
tc.endpoint_autoactivate(share_id)
tdata = TransferData(tc, host_id, share_id,
          label='RDP copy', sync_level='checksum')
tdata.add_item(source_path, '/', recursive=True)
r = tc.submit_transfer(tdata)
tc.task_wait(r['task_id'], timeout=1000,
                          polling_interval=10)

# (3) Enable access by user
r = ac.get_identities(usernames=email)
user_id = r['identities'][0]['id']
rule_data = {
    'DATA_TYPE': 'access',
    'principal_type': 'identity', # Grantee is
    'principal': user_id,          #  a user.
    'path': '/',                   # Path is /
    'permissions': 'r',            # Read-only
    'notify_email': email,         # Email invite
    'notify_message':              # Invite msg
            'Requested data are available.'
}
tc.add_endpoint_acl_rule(share_id, rule_data)

# (4) Ultimately, delete the shared endpoint
tc.delete_endpoint(share_id)
```

1. Create "shared endpoint"
2. Copy data to shared endpoint
3. Set permissions on shared endpoint for user; notify user

…

4. (Eventually) delete shared endpoint

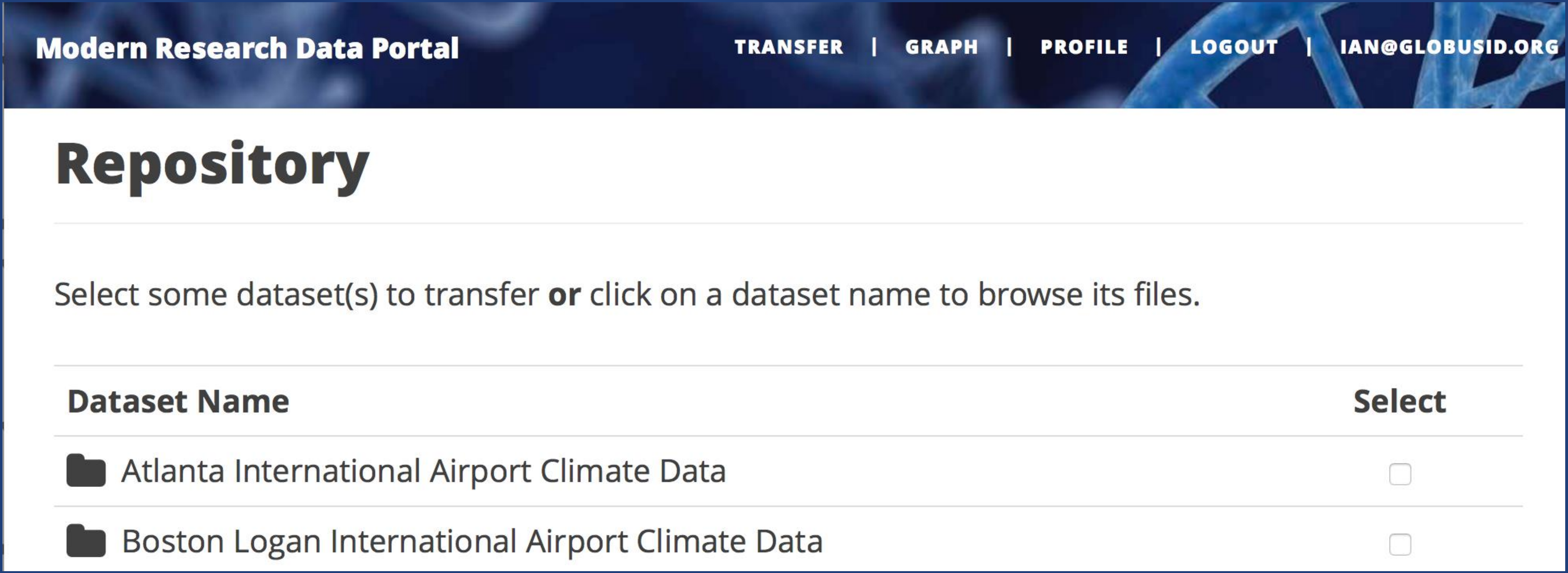
```
# (2) Copy data into the shared endpoint
tc.endpoint_autoactivate(share_id)
tdata = TransferData(tc, host_id, share_id,
        label='RDP copy', sync_level='checksum')
tdata.add_item(source_path, '/', recursive=True)
r = tc.submit_transfer(tdata)
tc.task_wait(r['task_id'], timeout=1000,
             polling_interval=10)

# (3) Enable access by user
r = ac.get_identities(usernames=email)
user_id = r['identities'][0]['id']
rule_data = {
  'DATA_TYPE': 'access',
  'principal_type': 'identity', # Grantee is
  'principal': user_id,          #  a user.
  'path': '/',                   # Path is /
  'permissions': 'r',            # Read-only
  'notify_email': email,         # Email invite
  'notify_message':              # Invite msg
      'Requested data are available.'
}
tc.add_endpoint_acl_rule(share_id, rule_data)

# (4) Ultimately, delete the shared endpoint
tc.delete_endpoint(share_id)
```

An example MRDP

https://docs.globus.org/mrdp


Modern Research Data Portal

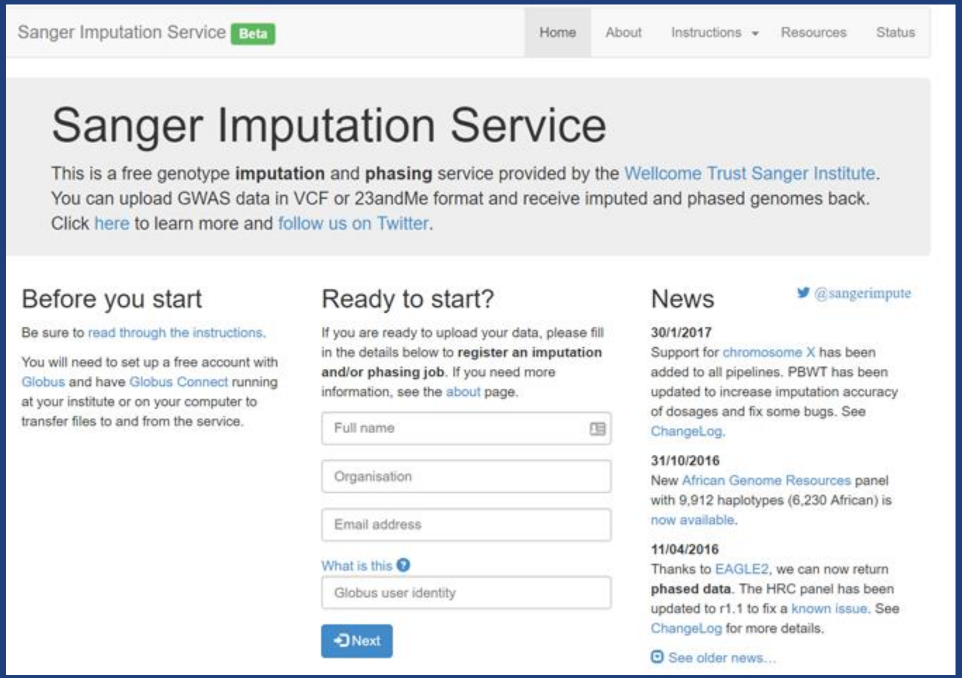
Repository

Select some dataset(s) to transfer **or** click on a dataset name to browse its files.

Dataset Name	Select
Atlanta International Airport Climate Data	☐
Boston Logan International Airport Climate Data	☐

Many variants possible

- Manage access to data at multiple locations
- Manage access to data on cloud
- Upload data for analysis
- Data download from scientific instruments
- Data publication
- Transfer data to computer for analysis



Sanger Imputation Service

This is a free genotype **imputation** and **phasing** service provided by the Wellcome Trust Sanger Institute. You can upload GWAS data in VCF or 23andMe format and receive imputed and phased genomes back. Click here to learn more and follow us on Twitter.

Before you start

Be sure to read through the instructions.

You will need to set up a free account with Globus and have Globus Connect running at your institute or on your computer to transfer files to and from the service.

Ready to start?

If you are ready to upload your data, please fill in the details below to **register an imputation and/or phasing job**. If you need more information, see the about page.

Full name

Organisation

Email address

What is this

Globus user identity

Next

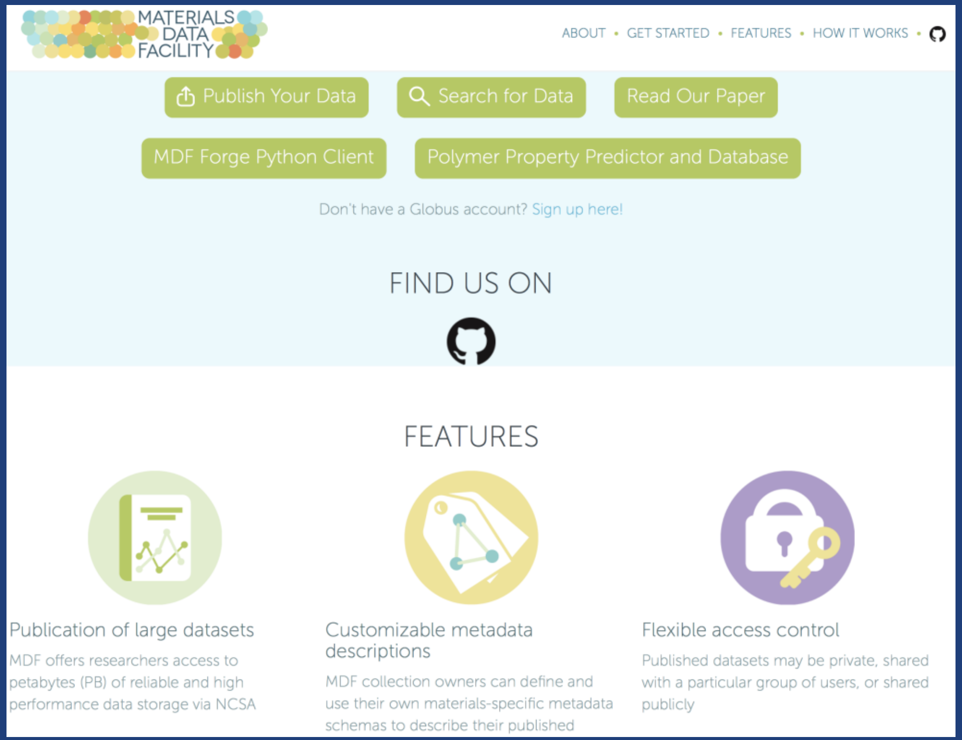
News @sangerimpute

30/1/2017
Support for chromosome X has been added to all pipelines. PBWT has been updated to increase imputation accuracy of dosages and fix some bugs. See ChangeLog.

31/10/2016
New African Genome Resources panel with 9,912 haplotypes (6,230 African) is now available.

11/04/2016
Thanks to EAGLE2, we can now return **phased data**. The HRC panel has been updated to r1.1 to fix a known issue. See ChangeLog for more details.

See older news...



Don't have a Globus account? Sign up here!

FIND US ON

FEATURES

Publication of large datasets

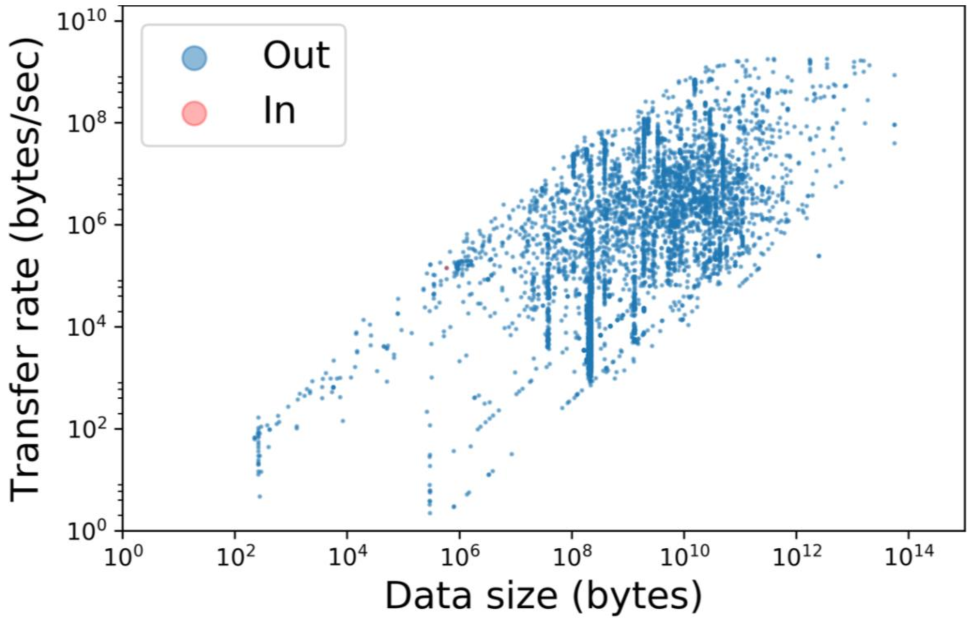
MDF offers researchers access to petabytes (PB) of reliable and high performance data storage via NCSA

Customizable metadata descriptions

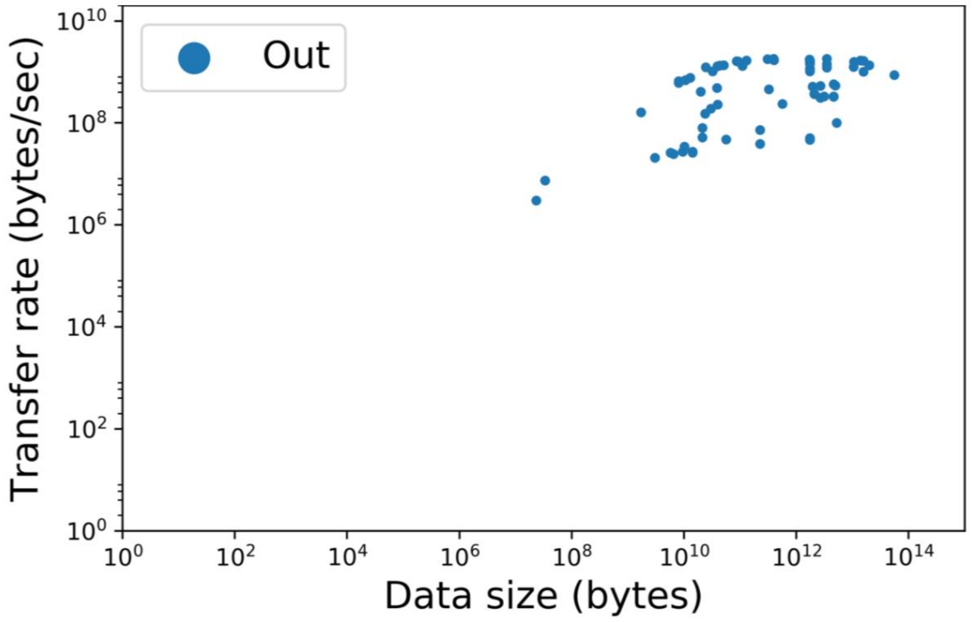
MDF collection owners can define and use their own materials-specific metadata schemas to describe their published

Flexible access control

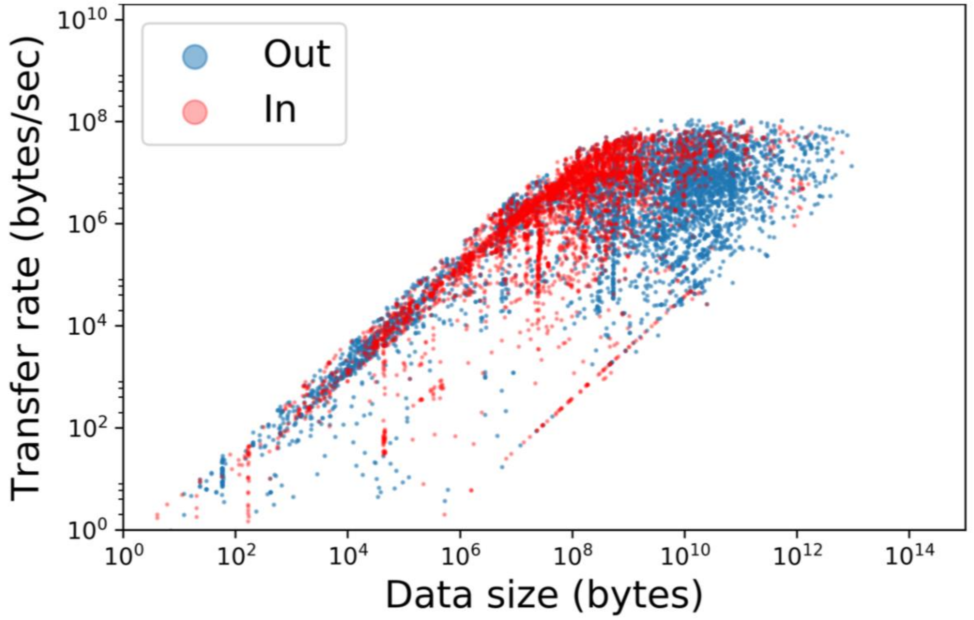
Published datasets may be private, shared with a particular group of users, or shared publicly



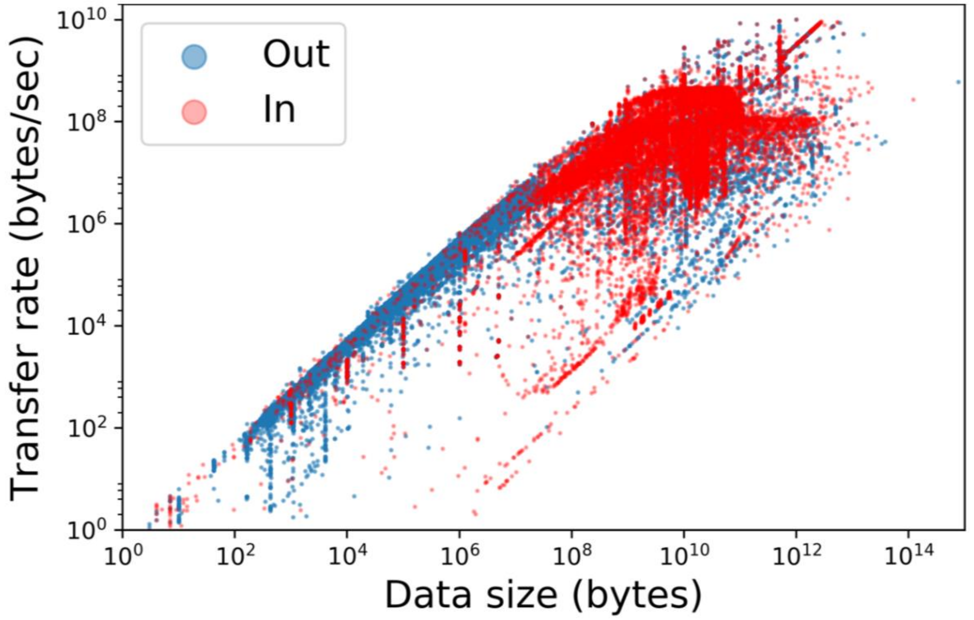
(a) RDA



(b) RDA to NERSC subset



(c) Sanger



(d) Petrel

The Modern Research Data Portal: A Design Pattern for Networked, Data-Intensive Science

https://docs.globus.org/mrdp

The Modern Research Data Portal is a new design pattern for providing secure, scalable, and high performance access to research data.

GitHub Repo

provides code for the simple data portal that you can experiment with online

LEARN MORE

Example Data Portal

allows you to experiment with an example implementation of the design pattern

LEARN MORE

Code Walkthrough

provides a narrative description of the simple data portal code

LEARN MORE

Jupyter Notebook

demonstrates some Globus features described in the technical article

LEARN MORE

A technical article describes this design pattern, reviews representative examples at research laboratories and universities (see below), and uses coding examples to demonstrate how Globus APIs can be used to implement a range of research data portal capabilities.

Chard K, Dart E, Foster I, Shifflett D, Tuecke S, Williams J. (2017) The Modern Research Data Portal: A design pattern for networked, data-intensive science. PeerJ Preprints5:e3194v1 https://doi.org/10.7287/peerj.preprints.3194v1

LEARN MORE

foster@uchicago.edu