Materials Data Facility - Data Services to Advance Materials Science Research

Ben Blaiszik (blaiszik@uchicago.edu),
Kyle Chard, Jim Pruyne, Rachana Ananthakrishnan
Michael Ondrejcek, Kenton McHenry
PIs: Ian Foster (foster@uchicago.edu), Steven Tuecke, John Towns

materialsdatafacility.org
globus.org

Materials Genome Initiative

NIST National Institute of Standards and Technology U.S. Department of Commerce

CHiMaD Center for Hierarchical Materials Design

THE UNIVERSITY OF CHICAGO

ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Argonne NATIONAL LABORATORY

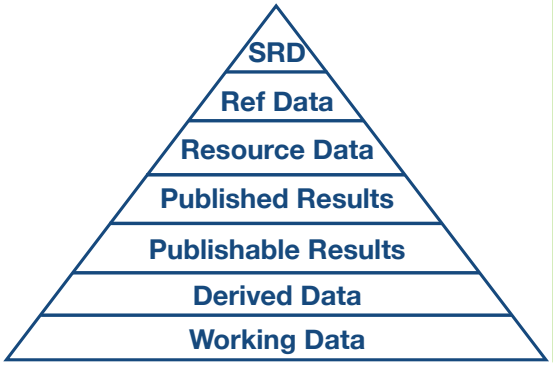
MATERIALS DATA FACILITY

Publication

- Identify datasets with persistent identifiers (e.g., DOI or Handle)
- Describe datasets with appropriate metadata and provenance
- Verify dataset contents over time
- Handle big (and small) data: We have already ingested datasets with > 1.5M files and > 1TB in size

REST APIs

Materialsdatafacility.org



Discovery

- Search, query, and access datasets in modern ways
- Automatically index flexible metadata and harvest file contents
- Provide simple user interfaces (i.e., after Google and Amazon)

- Opened to external users in Mar. 2016
  ~ 7 TB of data published, ~1M records indexed
- 100s of TB available now to support open data publication – contact us now!

NIST National Institute of Standards and Technology U.S. Department of Commerce

CHiMaD Center for Hierarchical Materials Design

THE UNIVERSITY OF CHICAGO

ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Argonne NATIONAL LABORATORY

Globus Platform-as-a-Service (PaaS)

Identity management

- create and manage a unique identity linked to external identities for authentication

User groups

- Manage user group creation and administration flows
- Share data with user groups

Publication

Discovery

Data transfer

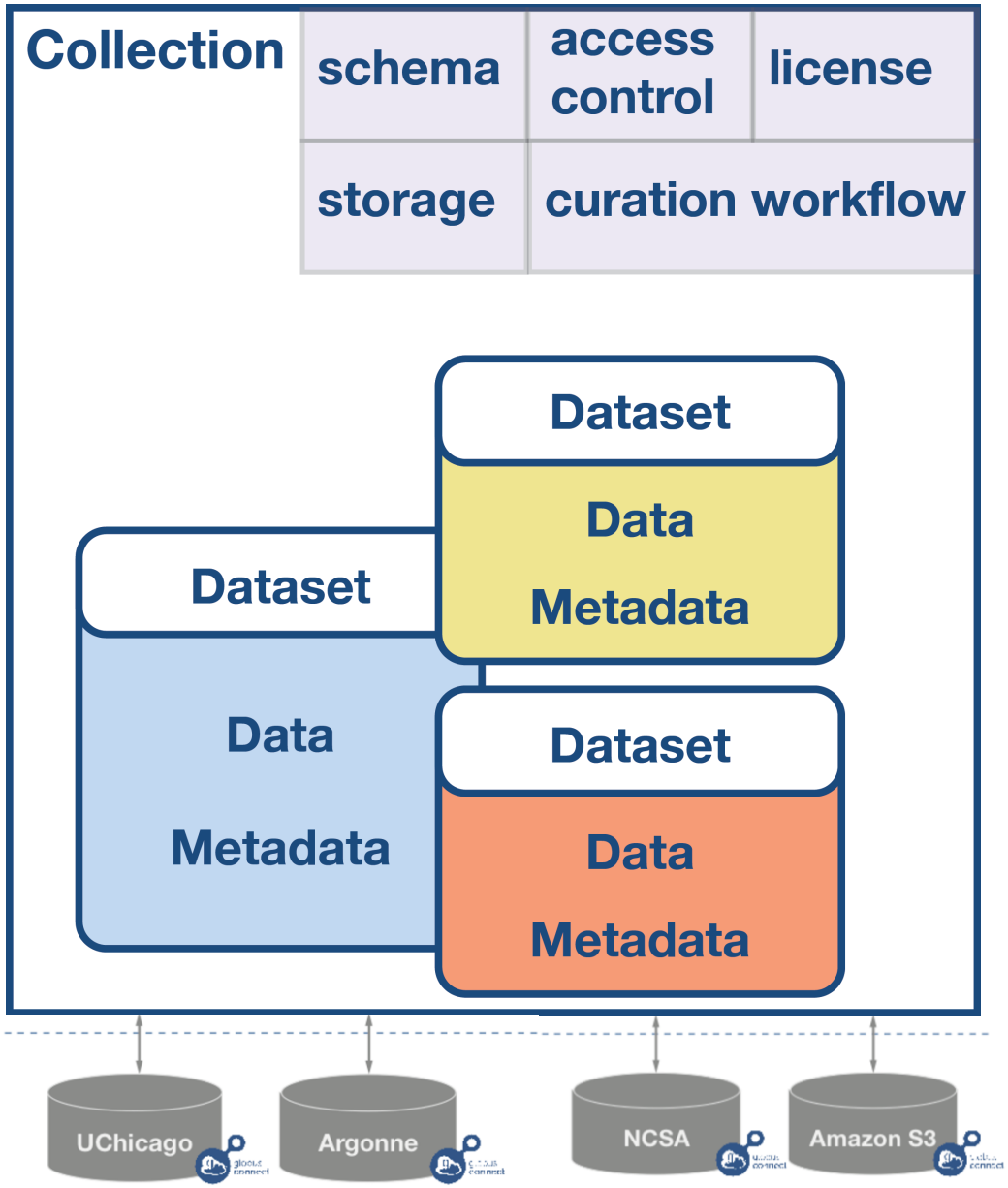
- High-performance data transfer from a web browser
- Optimize transfer settings and verify transfer integrity
- Add your laptop to the endpoint network with Globus Connect Personal

Data sharing

- Share directly from your storage device (laptop or cluster)
- File and directory-level ACLs

Data Publication Service

Publication Data Model



- **Collections have specified**
  - Mapping to storage endpoint
    - Currently handled as automatically created shared endpoints
  - Metadata schemas
  - Access control policies
  - Licenses
  - Curation workflows
- **Collections contain**
  - Datasets
    - Data
    - Metadata
- **Metadata Persistence**
  - Metadata log file with dataset
  - Metadata replicated in search index

Create Dataset

1

Create dataset in web UI or via API

2

Assemble data files and metadata

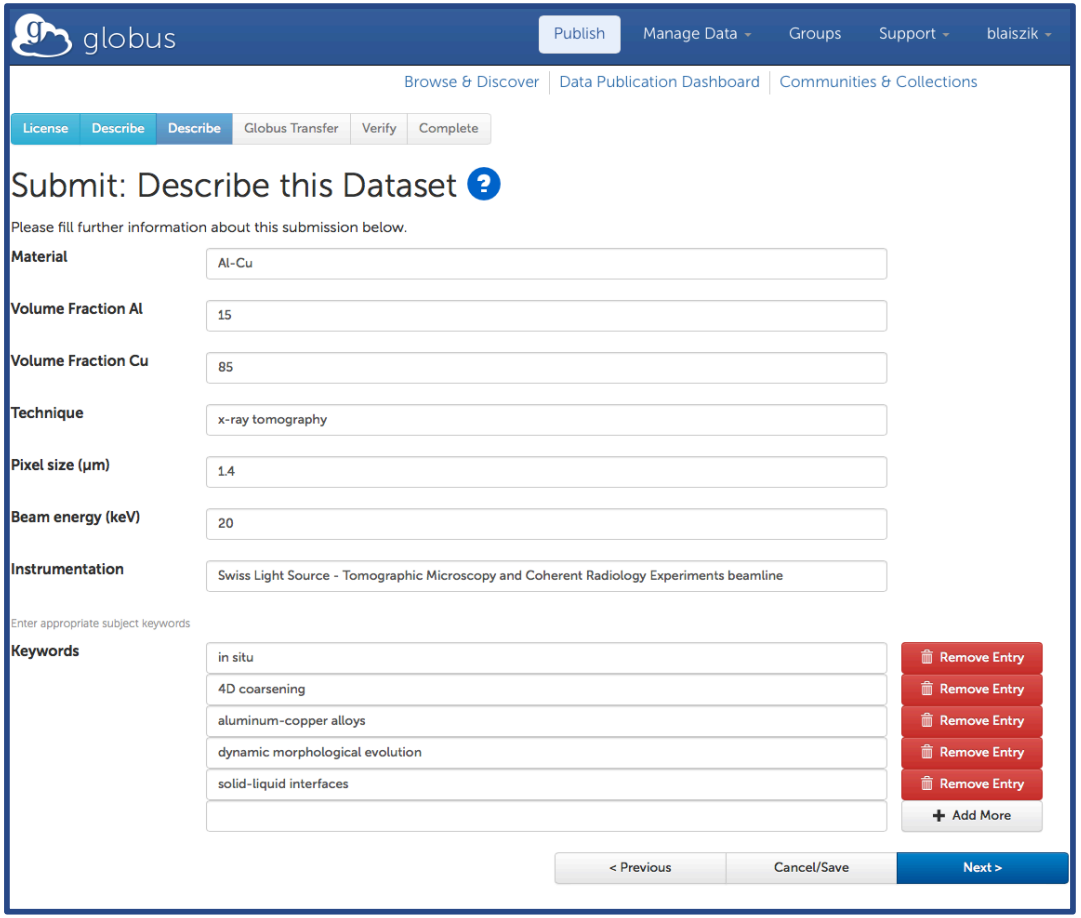
3

Curate dataset (optional)

4

Mint dataset identifiers

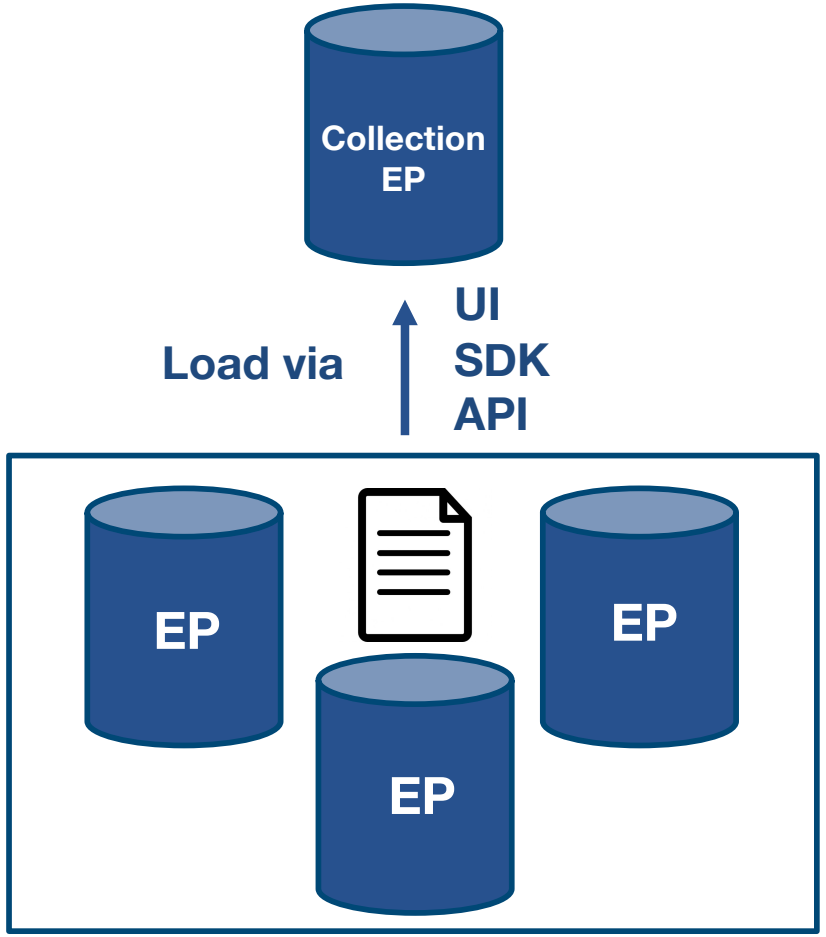
- Customizable metadata can be added via auto-generated forms or REST API [coming soon] (schema is validated)
- Creates a directory on the collection-specified endpoint, handles ACLs



Assemble Dataset

1. **Create dataset in web UI or via API**
2. **Assemble data files and metadata**
3. **Curate dataset (optional)**
4. **Mint dataset identifiers**

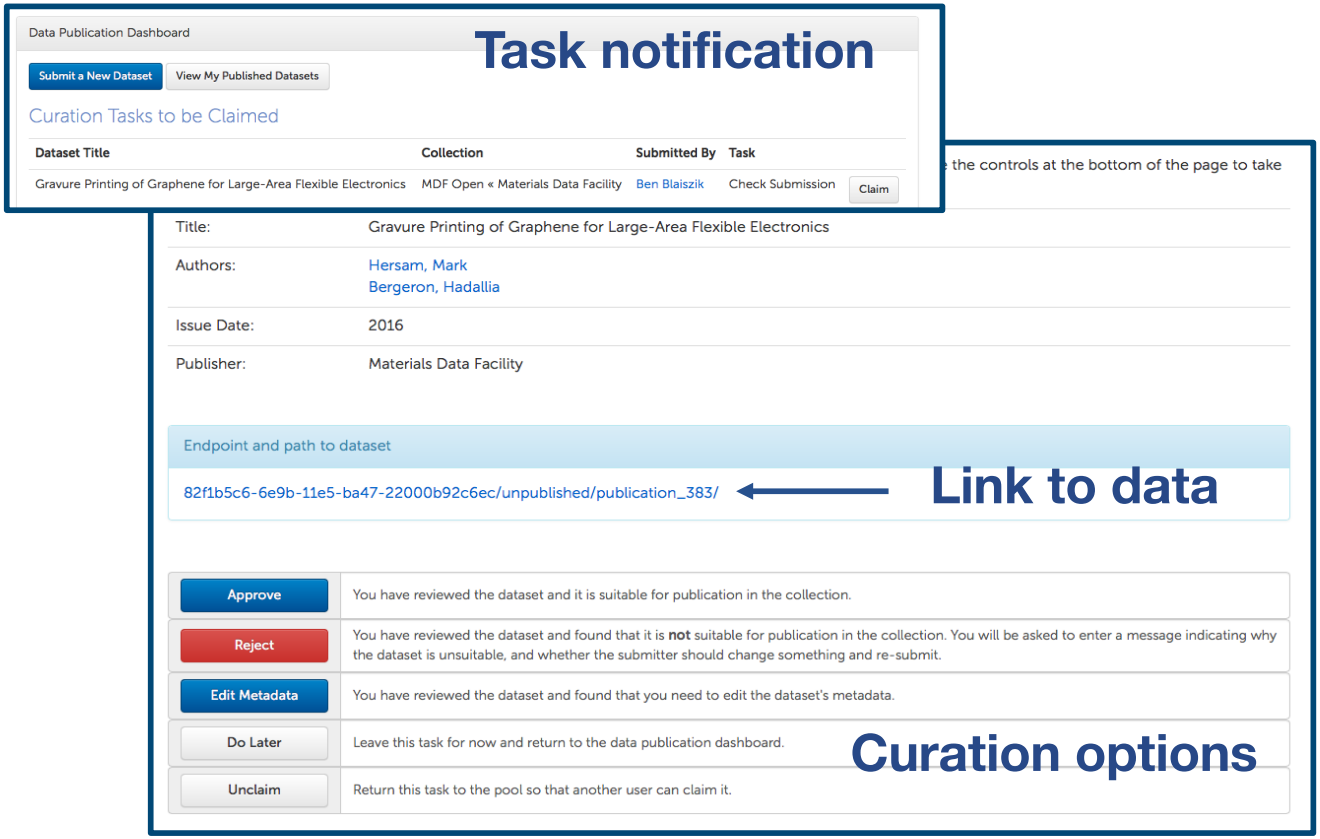
- Data can be assembled from multiple endpoints asynchronously



Curate Dataset (optional)

1. **Create dataset in web UI or via API**
2. **Assemble data files and metadata**
3. **Curate dataset (optional)**
4. **Mint dataset identifiers**

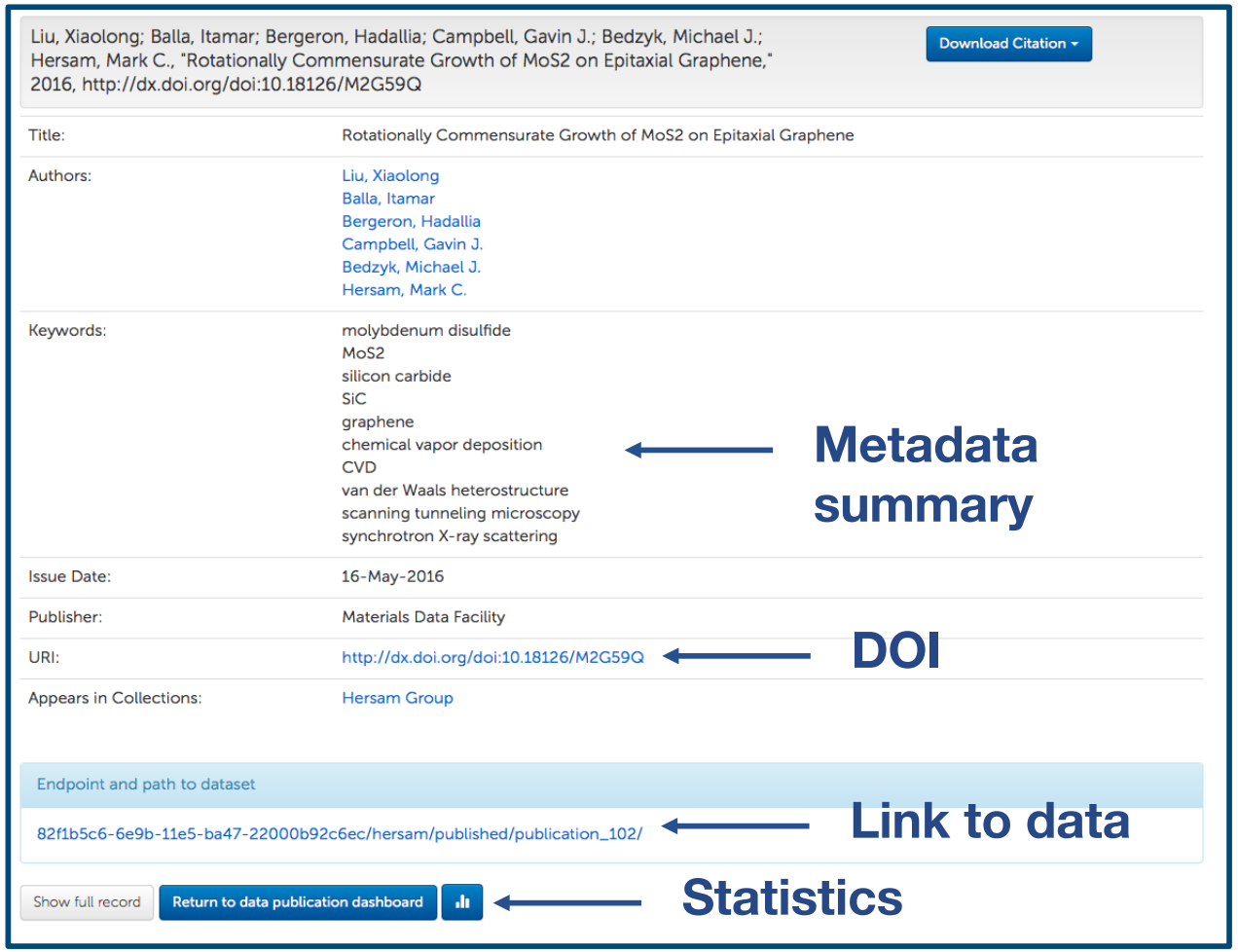
- If a collection specifies that curation is required, a notification is sent to the curator of the collection where the dataset is deposited
- Curator approves or sends the dataset back to the submitting user



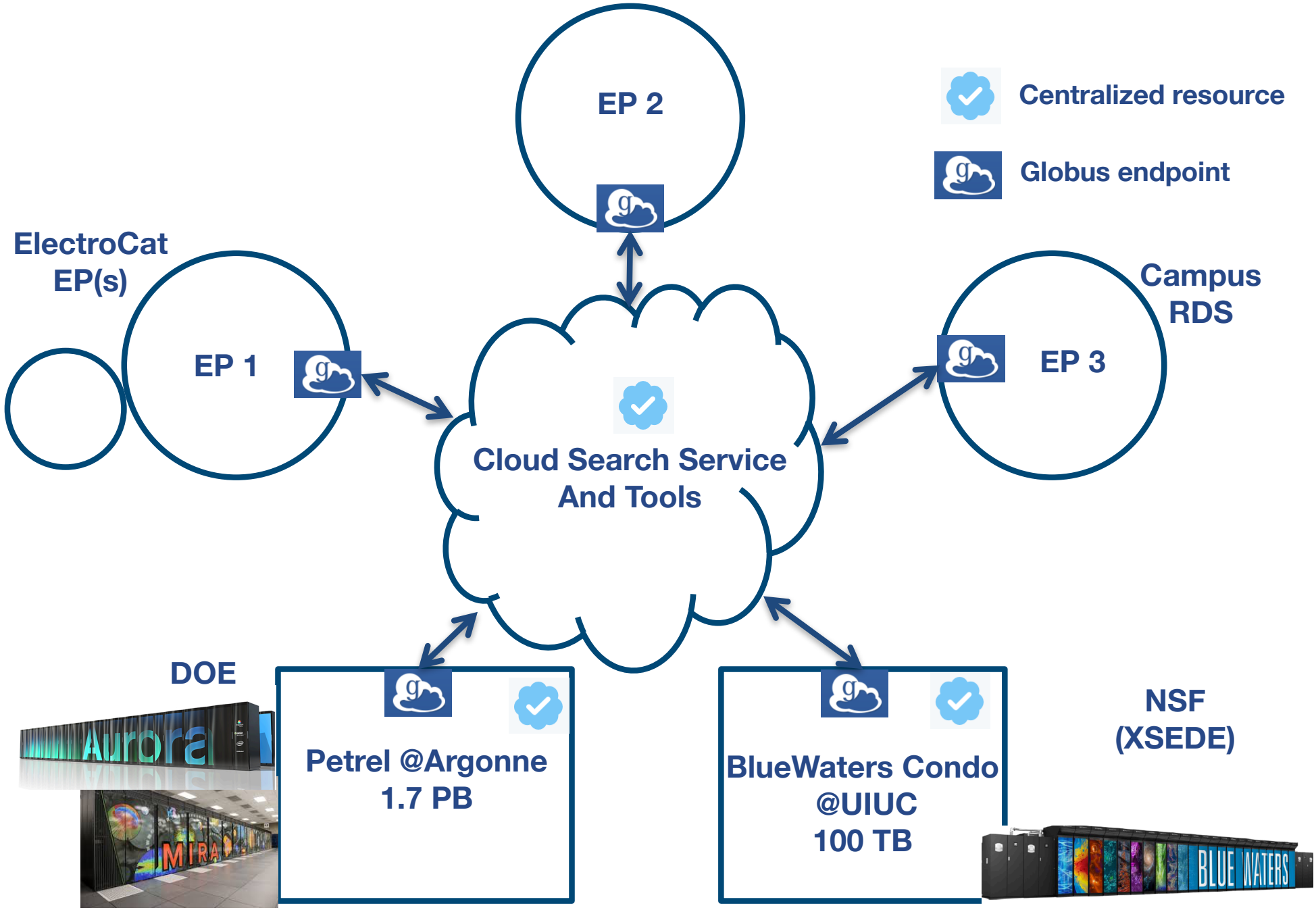
Mint Dataset Identifiers

1. **Create dataset in web UI or via API**
2. **Assemble data files and metadata**
3. **Curate dataset (optional)**
4. **Mint dataset identifiers**

- Based on collection specifications, an identifier is minted (e.g., DOI, or Handle)
- A landing page with metadata summary and data link is also created for the DOI

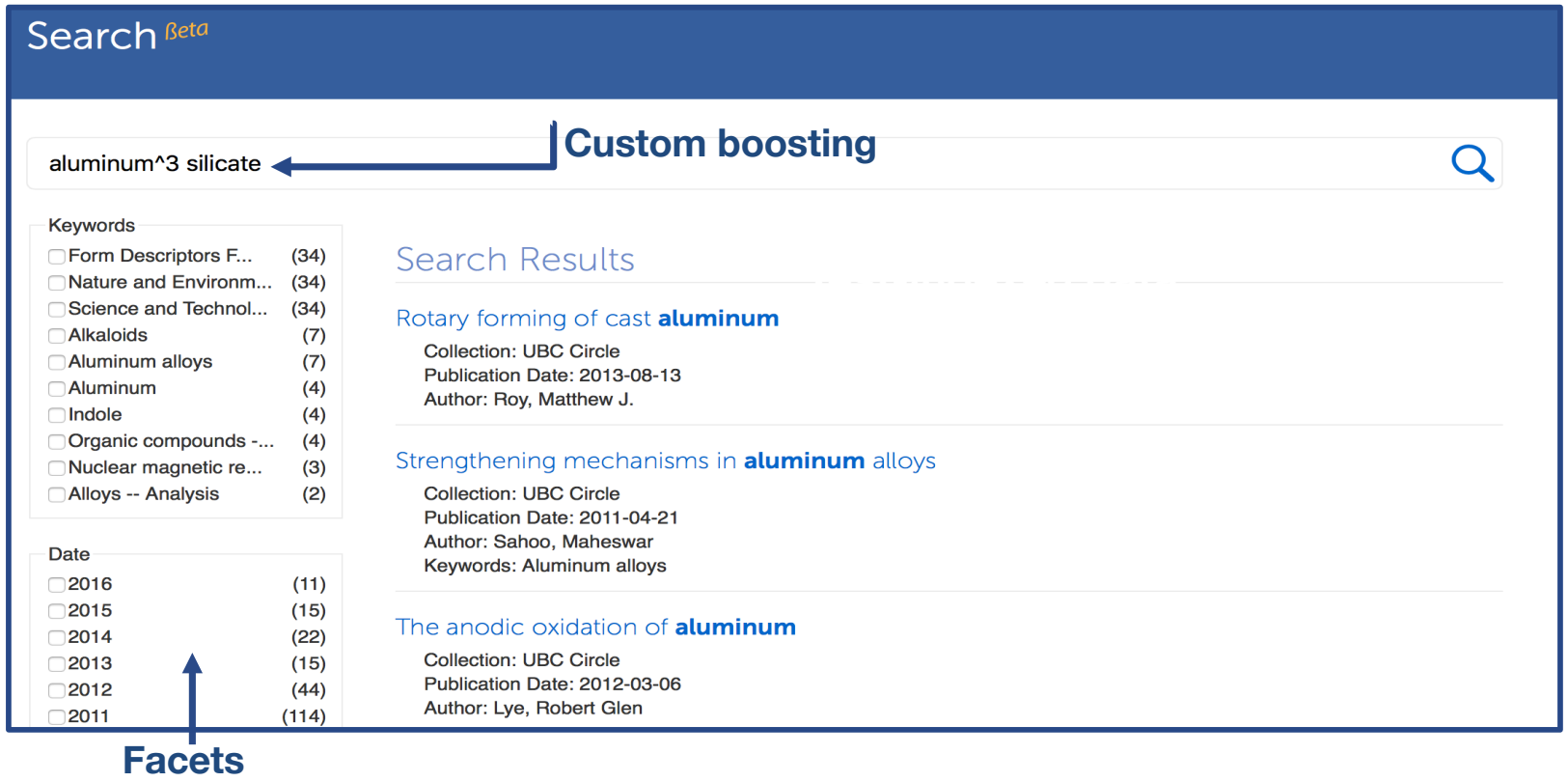


Distributed Publication Model



Data Discovery Service

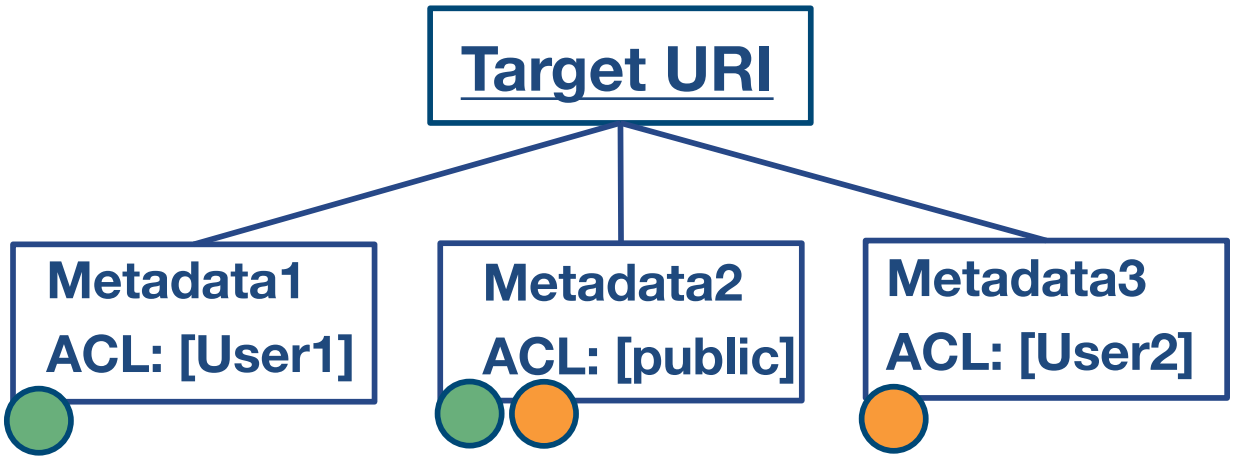
Data Search Service



- SaaS search service
- Indexing collections of data (many topics) and deep indexing specific datasets (e.g., atomistic modeling since there is a lot of data available)
- Will also be capable of indexing endpoint contents
- REST APIs and emerging python client
- Custom faceting and search boosting
- Partial matches, fuzzy matches, range queries
- Multi-scale search (current research topic)

Data Search Model

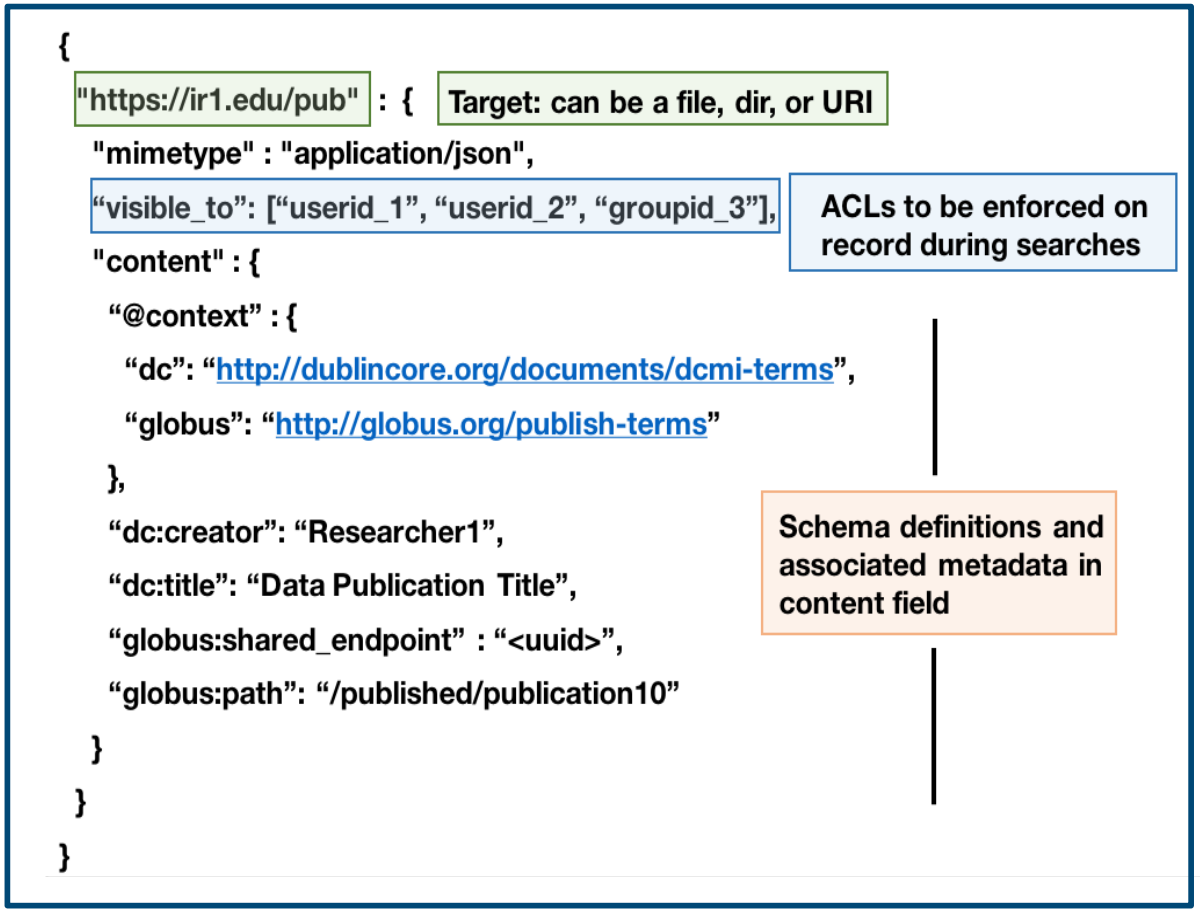
Multiple sets of access-controlled metadata may be attached to a target



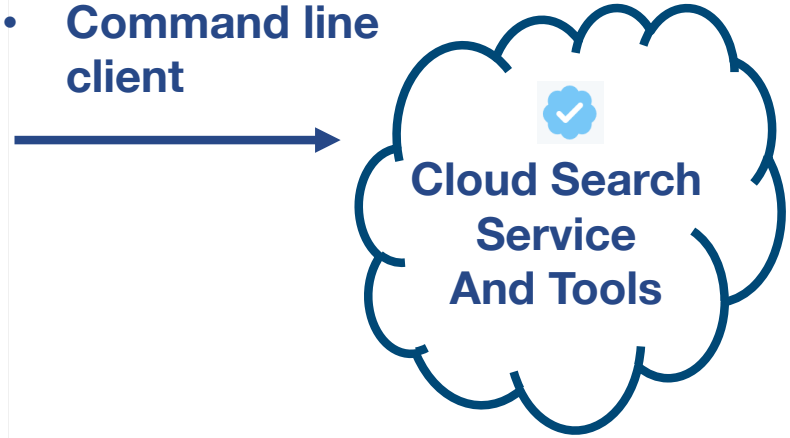
User1:
Sees union of
Metadata1, Metadata2

User2:
Sees union of
Metadata2, Metadata3

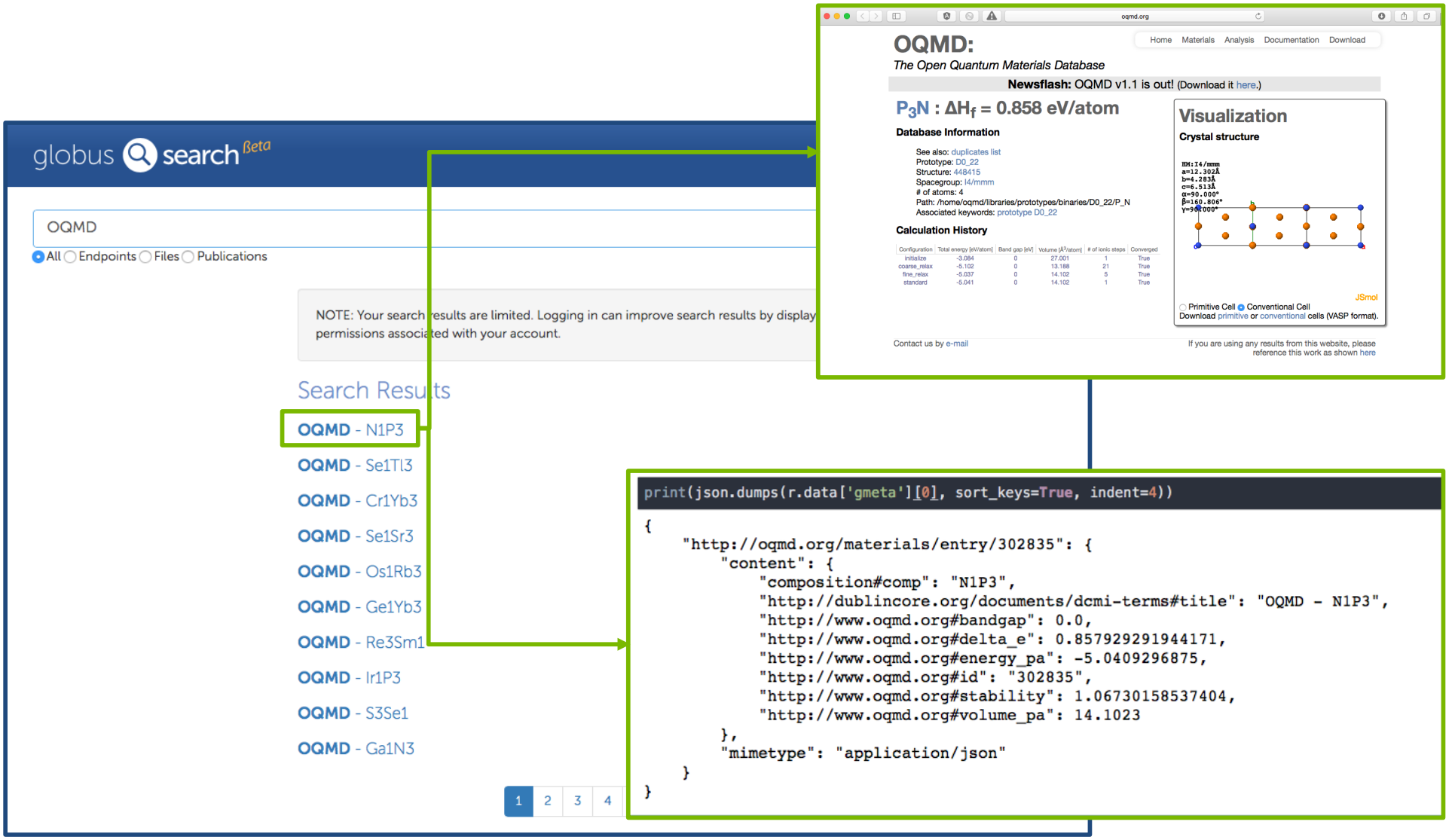
Data Search Document Format



- REST API
- Python
- Command line client



Data Search Example - OQMD



Search Result Aggregation…

- REST API, command-line client, and python client
- Indexed ~10 materials-related sources (OQMD, HOPV, JANAF-NIST, Khazana, Nanomine, Ab Initio Solute-Solvent Diffusion Database, etc.) ~1M records

Authenticate Search Client

Uses a valid authentication token if it exists, redirects to web if not

```
client = globus_auth.login("https://datasearch.api.demo.globus.org/")
```

Search for Data and Aggregate into a DataFrame

```
params = {
        'q': "OQMD",
        'highlight': True,
        'resource_type': None,
        'count': 20,
        'from': 0,
        'stats': True,
        'facets': None,
        'filters': None,
    }

r = client.search(**params)
print("Found %d results"%(r.data['gstats']['total']))
```

```
Found 264251 results
```

Aggregate a Larger Query

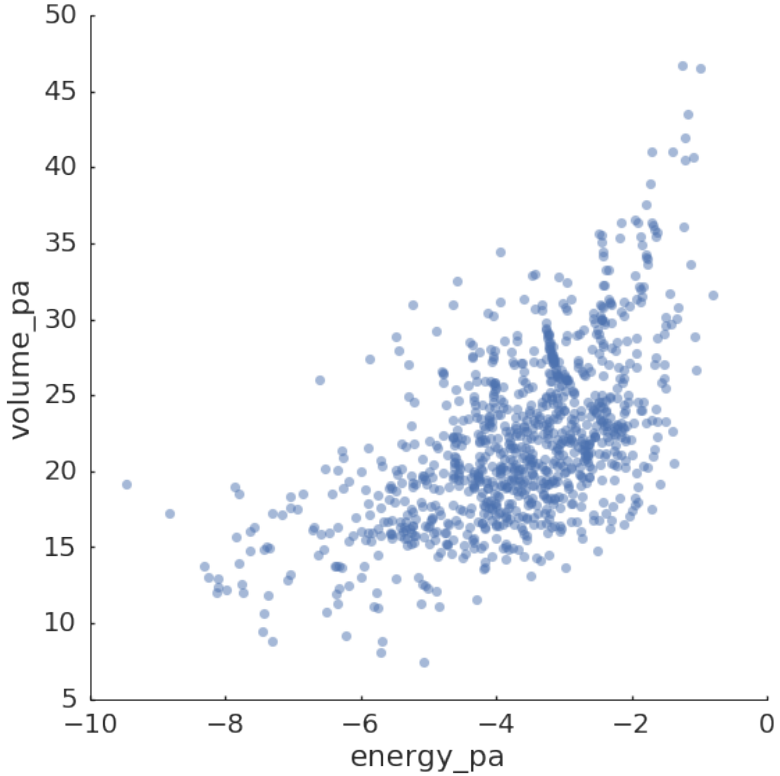
```
n_records = 1000
offset = 0
vectors = []

for i in range(0,5):
    params = { 'q': "OQMD",
               'count': n_records,
               'from': i*n_records}

    r = client.search(**params)
    result_iterator = r.data['gmeta']
    for result in result_iterator:
        vectors.append(get_vectors(result))

df = pd.DataFrame(vectors, columns=["composition", "delta_e",
                                    "energy_pa", "volume_pa",
                                    "magmom_pa","stability"])
```

Search Result Aggregation…

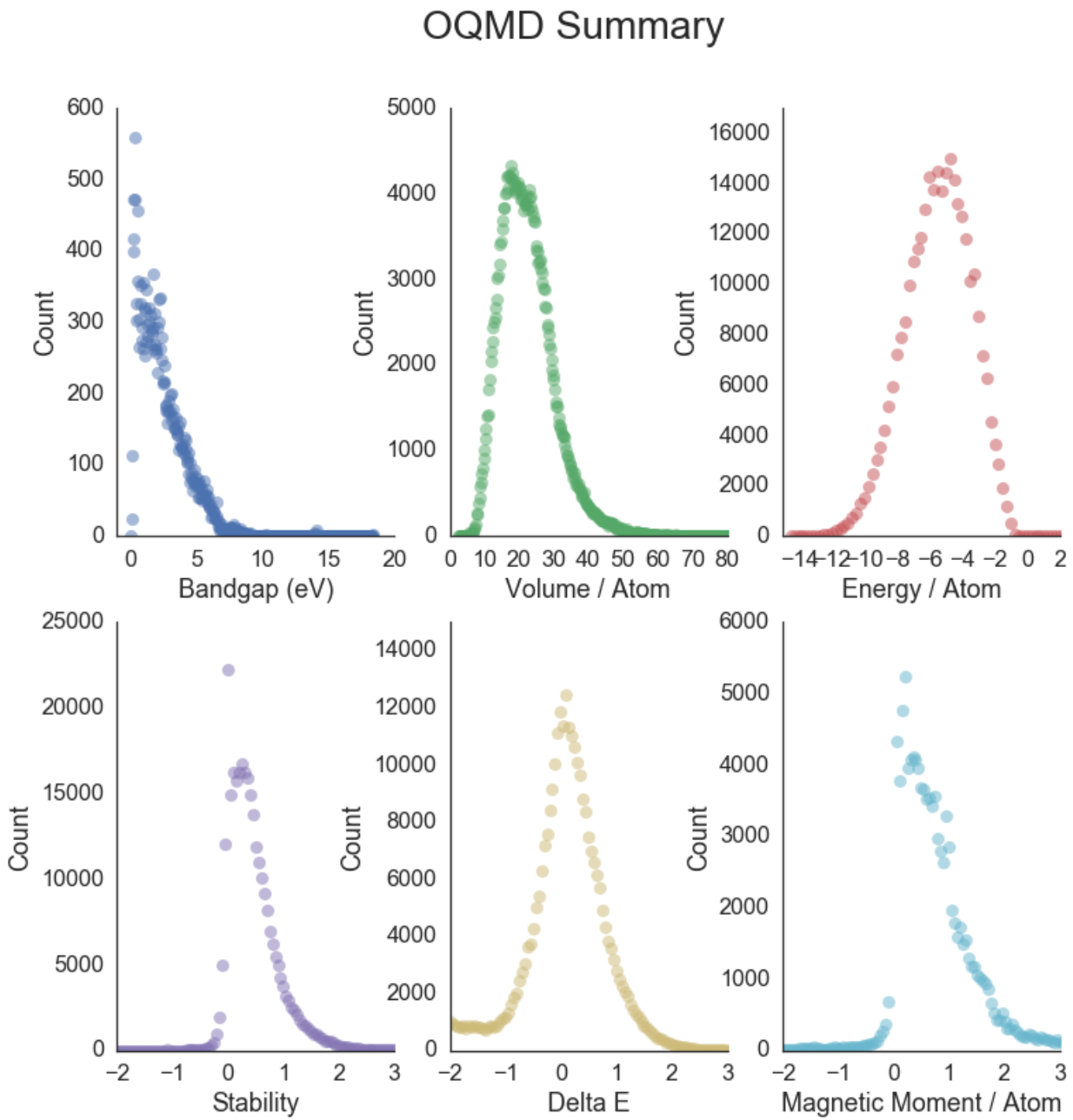
- REST API, command-line client, and python client
- Indexed ~10 materials-related sources (OQMD, HOPV, JANAF-NIST, Khazana, Nanomine, Ab Initio Solute-Solvent Diffusion Database, etc.) ~1M records

Visualize the Results

```
pal = sns.color_palette("muted")

x_col = "energy_pa"
y_col = "volume_pa"

# Make plot
g = sns.lmplot(x=x_col, y=y_col, data=df, fit_reg=False,
               palette=pal, size=8, scatter_kws={"s": 50,
                                                 `"alpha": 0.5})
```



Dataset Summary Aggregation



CHiMaD Nanomine Example

Perform Nanomine Query

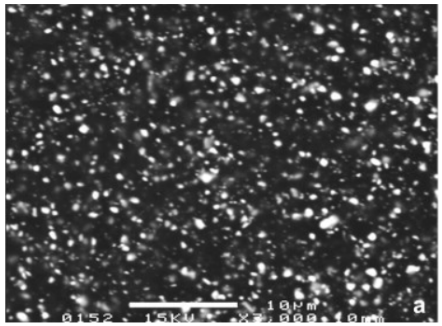
```
params = {
        'q': "nanomine",
        'count': 10,
        'from': 0
}

r = client.search(**params)
res = pop_top_gmeta(r)
head = res[0]['content'][0]['_#content']['_#PolymerNanocomposite']
```

```
head.keys()
```

```
dict_keys(['_#PROCESSING', '_#ID', '_#PROPERTIES', '_#MATERIALS', '_#CHARACTERIZATION', '_
#DATA_SOURCE', '_#MICROSTRUCTURE'])
```

Get Microstructure Image

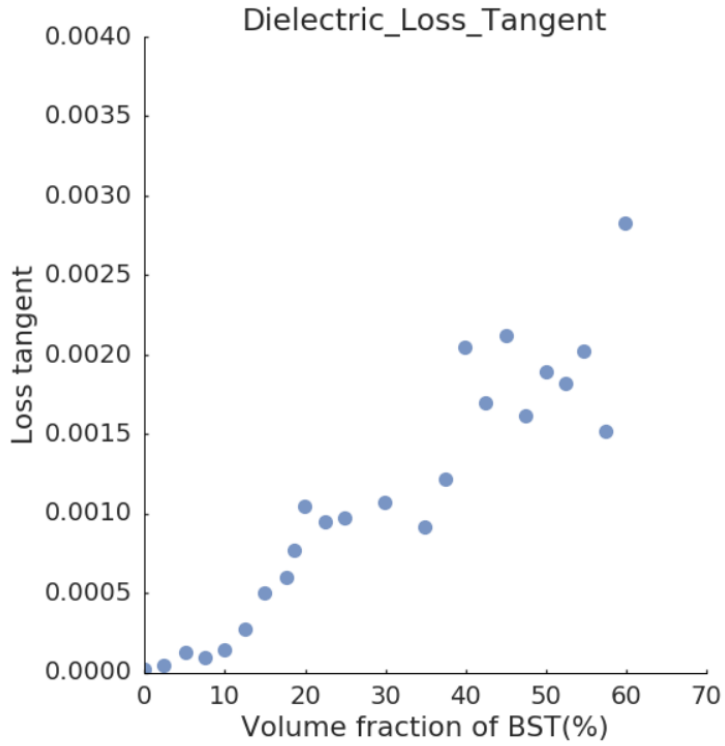
```
microstructure = head['_#MICROSTRUCTURE']
Image(url=microstructure['_#ImageFile']['_#File'])
```



Example Plot of Dielectric Loss Tangent

```
pal = sns.color_palette("muted")
g = sns.lmplot(x=headers[0], y=headers[1], data=df_nm,
               fit_reg=False, palette=pal, size=8,
               scatter_kws={"s": 125, "alpha": 0.75})
g.ax.set_title('%s'%(_property))
g.ax.set_ylim(0.0,0.004)
g.ax.set_xlim(0.0,70)
```

```
(0.0, 70)
```



Where Can I get More Information?

JOM
August 2016, Volume 68, Issue 8, pp 2045–2052

MATERIALS DATA FACILITY

The Materials Data Facility: Data Services to Advance Materials Science Research

B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, I. Foster

Article
First Online: 06 July 2016
DOI: 10.1007/s11837-016-2001-3

Cite this article as:
Blaiszik, B., Chard, K., Pruyne, J. et al. JOM (2016) 68: 2045. doi:10.1007/s11837-016-2001-3

http://dx.doi.org/10.1007/s11837-016-2001-3

Links:

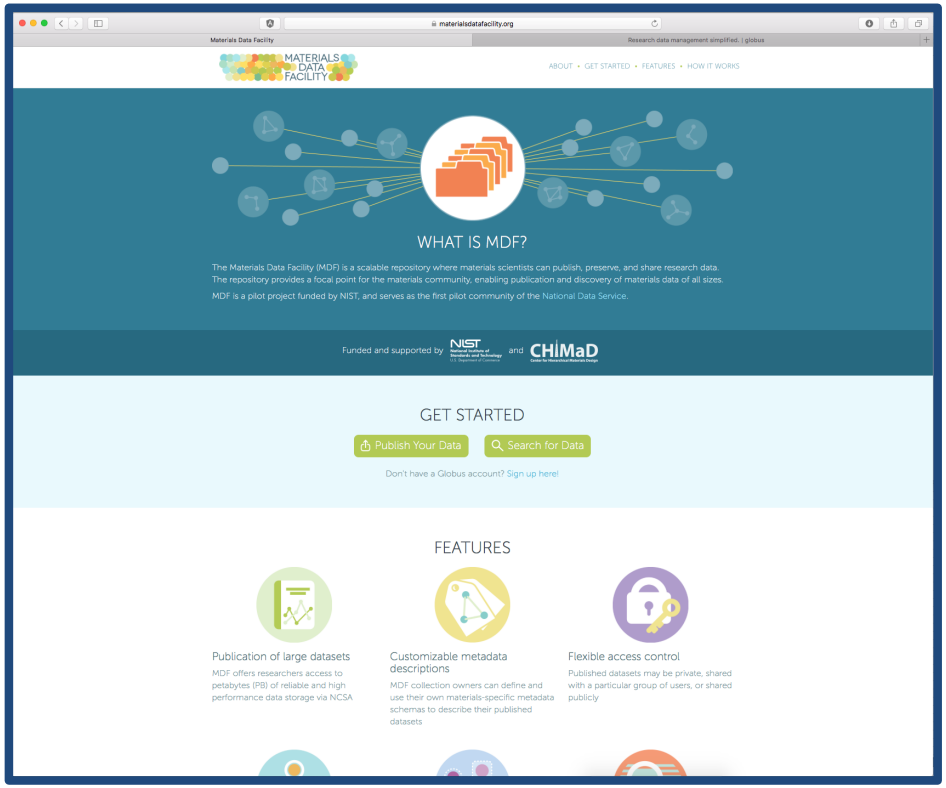
materialsdatafacility.org

globus.org

Contact:

blaiszik@uchicago.edu

foster@uchicago.edu



Contributors (unordered)

UC/Argonne


Ian Foster (PI)


Ben Blaiszik

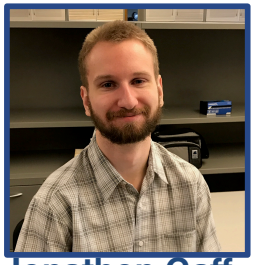

Steve Tuecke (PI)


Jim Pruyne


Rachana


Kyle Chard


Logan Ward


Jonathon Gaff

Stephen Rosen

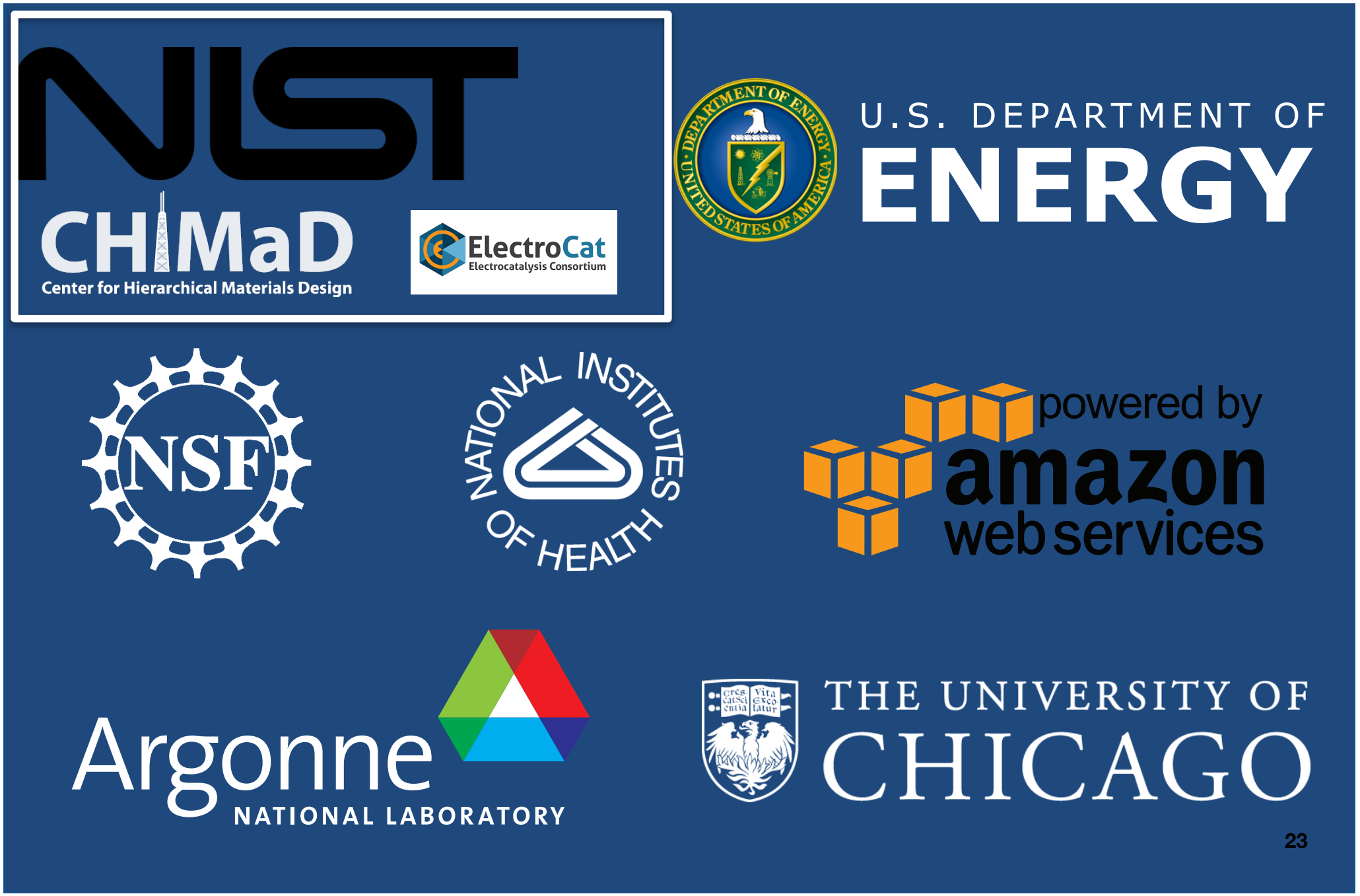
Illinois (Urbana-Champaign)


John Towns (PI)


Kenton McHenry

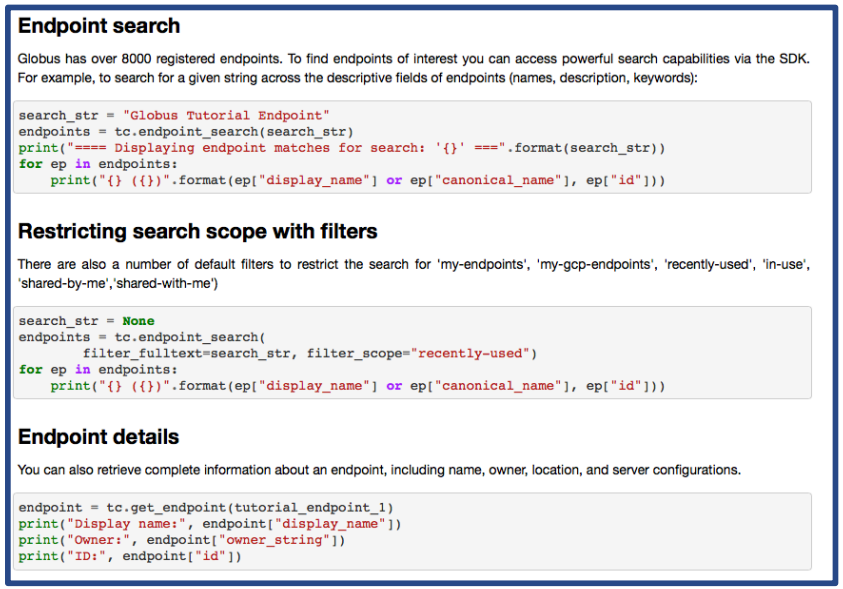

Michal Ondrejcek

Thanks to Our Sponsors!



REST APIs, Clients, and Docs

- Globus Docs
  - https://docs.globus.org
- New Python SDK available
  - https://github.com/globusonline/globus-sdk-python
- Jupyter Notebook Examples
  - https://github.com/globus/globus-jupyter-notebooks
- Sample Data Portal
  - https://github.com/globus/globus-sample-data-portal
- (alpha) MDF Data Publication Service API



Endpoint search

Globus has over 8000 registered endpoints. To find endpoints of interest you can access powerful search capabilities via the SDK. For example, to search for a given string across the descriptive fields of endpoints (names, description, keywords):

```
search_str = "Globus Tutorial Endpoint"
endpoints = tc.endpoint_search(search_str)
print("==== Displaying endpoint matches for search: '{}' ===".format(search_str))
for ep in endpoints:
    print("{} ({})".format(ep["display_name"] or ep["canonical_name"], ep["id"]))
```

Restricting search scope with filters

There are also a number of default filters to restrict the search for 'my-endpoints', 'my-gcp-endpoints', 'recently-used', 'in-use', 'shared-by-me','shared-with-me')

```
search_str = None
endpoints = tc.endpoint_search(
        filter_fulltext=search_str, filter_scope="recently-used")
for ep in endpoints:
    print("{} ({})".format(ep["display_name"] or ep["canonical_name"], ep["id"]))
```

Endpoint details

You can also retrieve complete information about an endpoint, including name, owner, location, and server configurations.

```
endpoint = tc.get_endpoint(tutorial_endpoint_1)
print("Display name:", endpoint["display_name"])
print("Owner:", endpoint["owner_string"])
print("ID:", endpoint["id"])
```

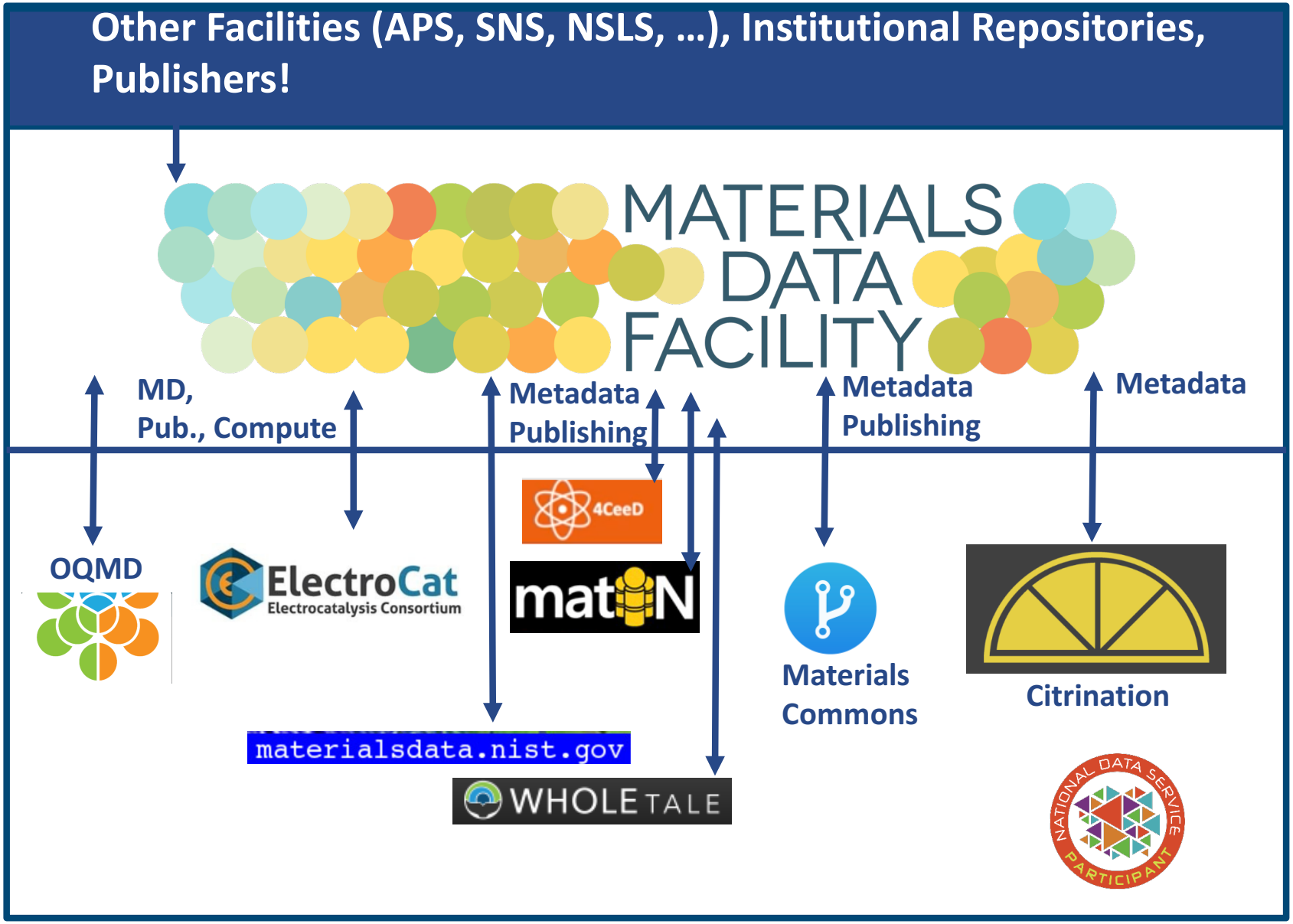
Transfer

Creating a transfer is a two stage process. First you must create a description of the data you want to transfer (which also creates a unique submission_id), and then you can submit the request to Globus to transfer that data.

If the submit_transfer fails, you can safely resubmit the same transfer_data again. The submission_id will ensure that this transfer request will be submitted once and only once.

```
# help(tc.submit_transfer)
source_endpoint_id = tutorial_endpoint_1
source_path = "/share/godata/"

dest_endpoint_id = tutorial_endpoint_2
dest_path = "/~/"

label = "My tutorial transfer"

# TransferData() automatically gets a submission_id for once-and-only-once submission
tdata = globus_sdk.TransferData(tc, source_endpoint_id,
                                dest_endpoint_id,
                                label=label)

## Recursively transfer source path contents
tdata.add_item(source_path, dest_path, recursive=True)

## Alternatively, transfer a specific file
# tdata.add_item("/source/path/file.txt",
#                "/dest/path/file.txt"))

# Ensure endpoints are activated
tc.endpoint_autoactivate(source_endpoint_id)
tc.endpoint_autoactivate(dest_endpoint_id)

submit_result = tc.submit_transfer(tdata)
print("Task ID:", submit_result["task_id"])
```

Integration with the Community is Key



zenodo MICCoM NOMAD The Dataverse Project