Materials Data Facility:
A Distributed Model for the Materials Data Community
27 September 2017

Logan Ward[1] (loganw@uchicago.edu)
Ben Blaiszik[1,2] (blaiszik@uchicago.edu),
Ian Foster (foster@uchicago.edu)[1,2], Ryan Chard[2]
Jonathon Gaff[1], Kyle Chard[1], Jim Pruyne[1],
Rachana Ananthakrishnan[1], Steven Tuecke[1]
Michael Ondrejcek[3], Kenton McHenry[3], John Towns[3]
University of Chicago[1], Argonne National Laboratory[2], University of Illinois at Urbana-Champaign[3]

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
NORTHWESTERN UNIVERSITY
ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Argonne NATIONAL LABORATORY

The Materials Data Facility Team

UC/Argonne


Ian Foster (PI)


Ben Blaiszik

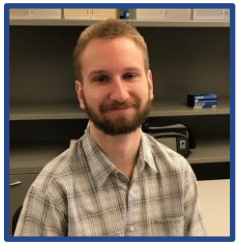

Steve Tuecke


Jim Pruyne


Rachana Ananthakrishnan


Kyle Chard


Logan Ward


Jonathon Gaff


Stephen Rosen


Ryan Chard

Illinois (Urbana-Champaign)


John Towns (PI)

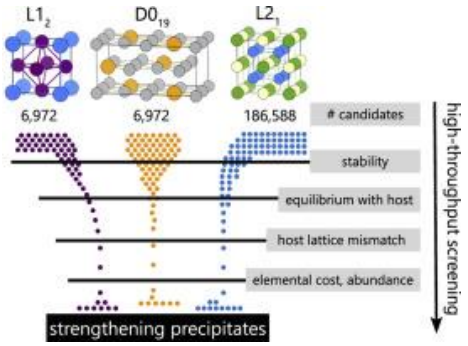

Kenton McHenry


Michal Ondrejcek

Data-Intensive Materials Science

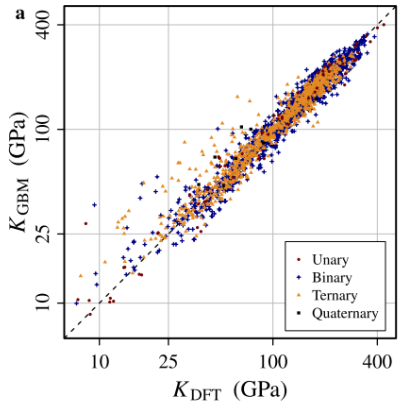
Materials Databases

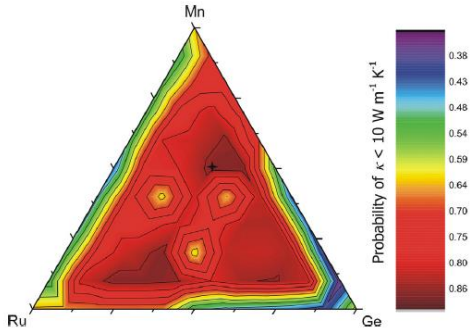
High-Throughput Screening



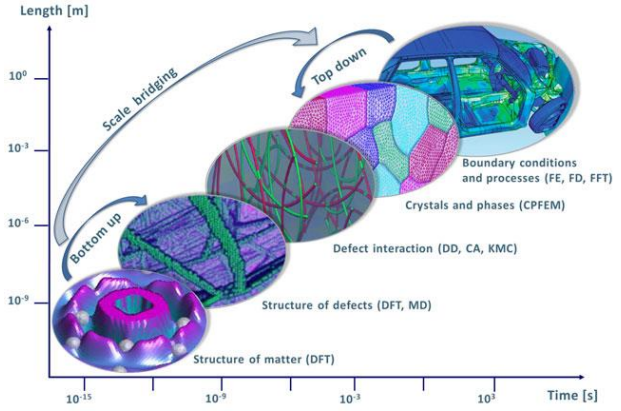
Kirklin *et al*. Acta Mat. (2016)

Machine Learning





Multi-scale Modeling



de Jong *et al*. Sci Rep. (2016)

Sparks *et al*. Scr. Mat. (2015)

https://www.mpg.de/

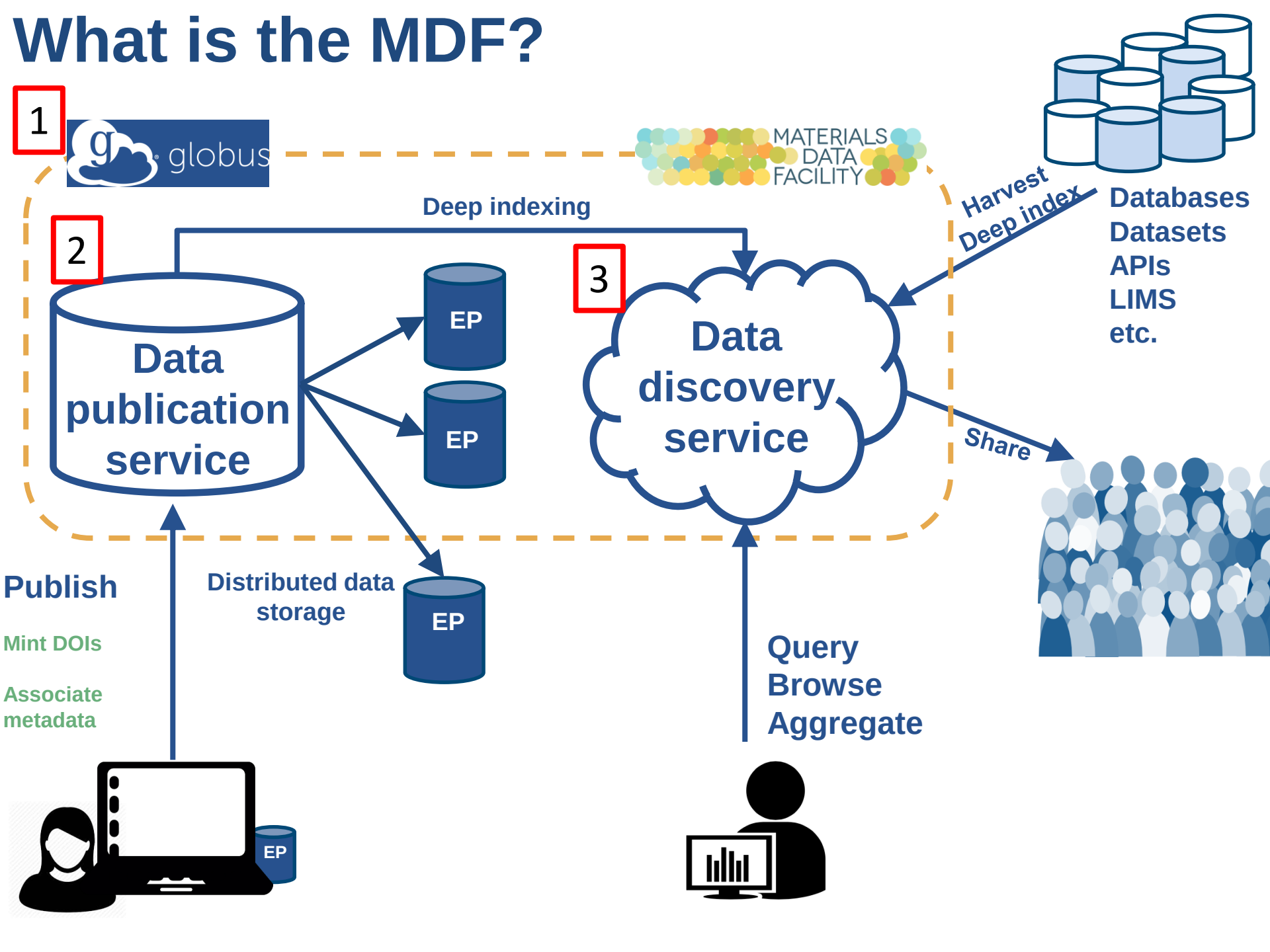
Data-Intensive Materials Science

Science is becoming limited by the ability to handle data

- Where to get it?
- How to selectively share it?
- Where to store it?
- How to know what it is?
- How to build software that uses it?
- How to get others to share theirs?
- How to keep track of provenance?
- ….?

Our goal is to create infrastructure that provides easy answers to these questions

What is the MDF?



GLOBUS

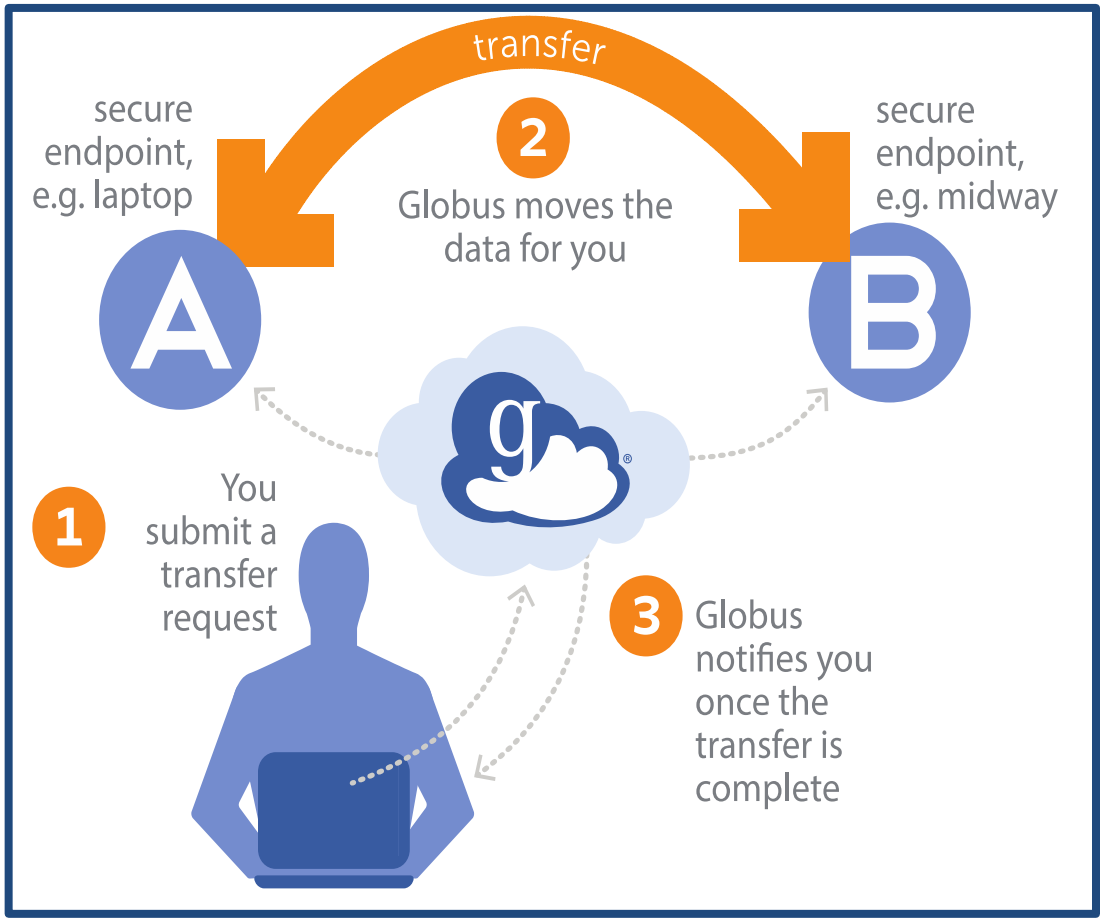
Globus Background

Endpoint

- E.g. laptop or server running a Globus client (e.g. Dropbox client)
- Enables advanced file transfer and sharing
- Currently GridFTP, future GridFTP + HTTP

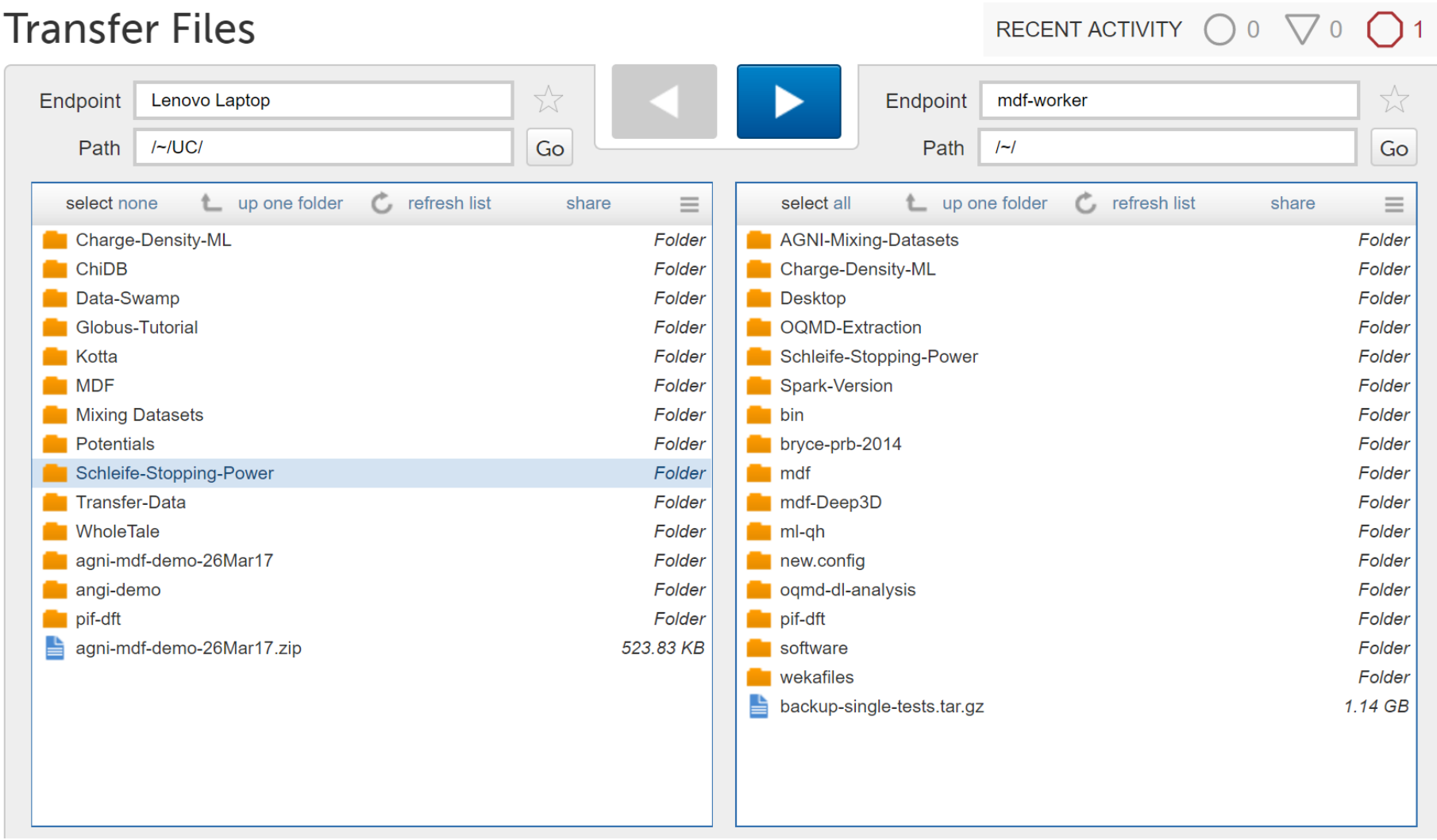
Some Key Features

- REST API for automation and interoperability
- Web UI for convenience
- Optimizes and verifies transfers
- Handles auto-restarts



313,101,618,927 MB TRANSFERRED

Globus Transfer



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

These services form the basis of the Materials Data Facility

Publication

Discovery

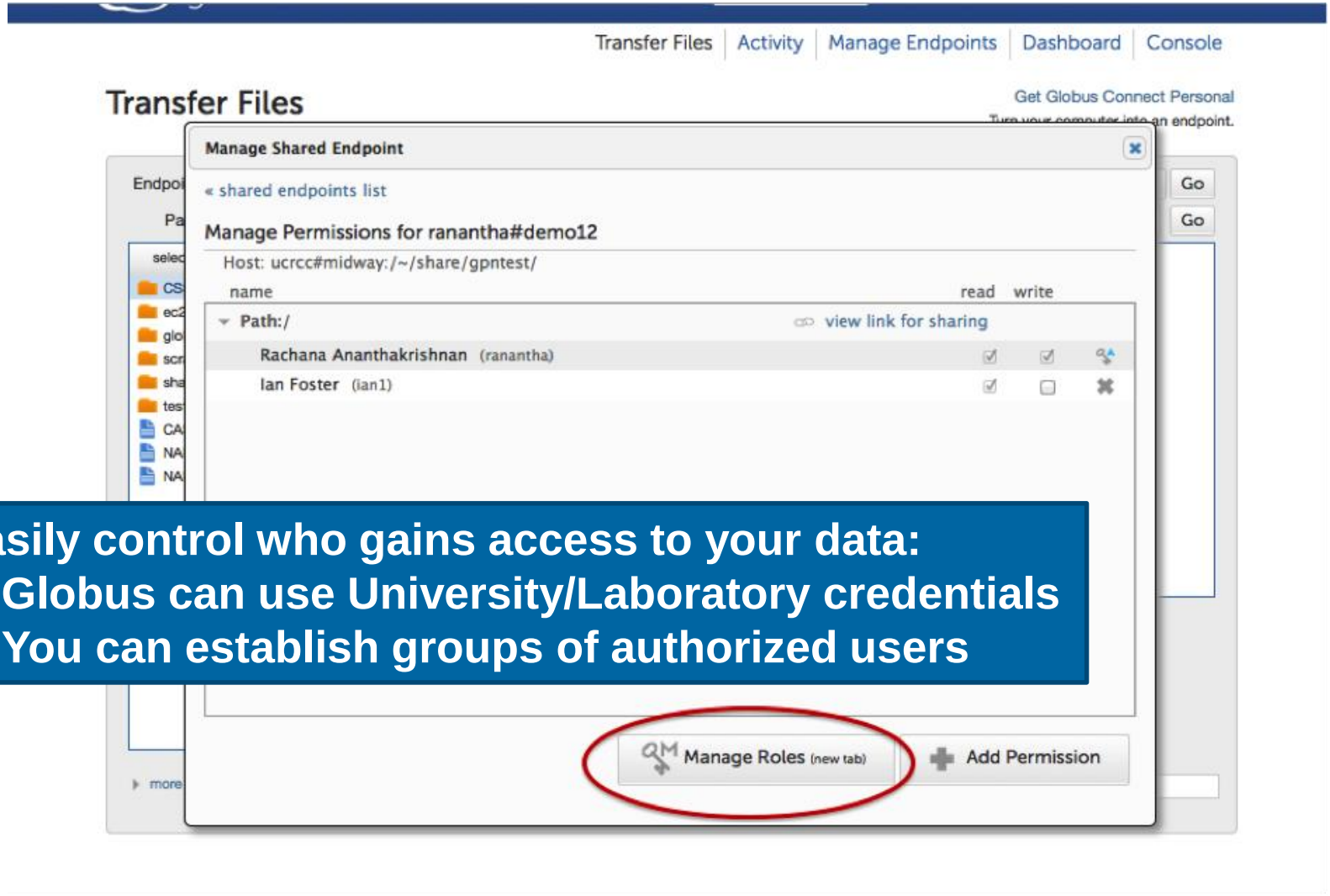
Data transfer

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

Data sharing

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

Data sharing and Globus



Easily control who gains access to your data:

- **Globus can use University/Laboratory credentials**
- **You can establish groups of authorized users**

REST APIs, Clients, and Docs

- 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"])
```

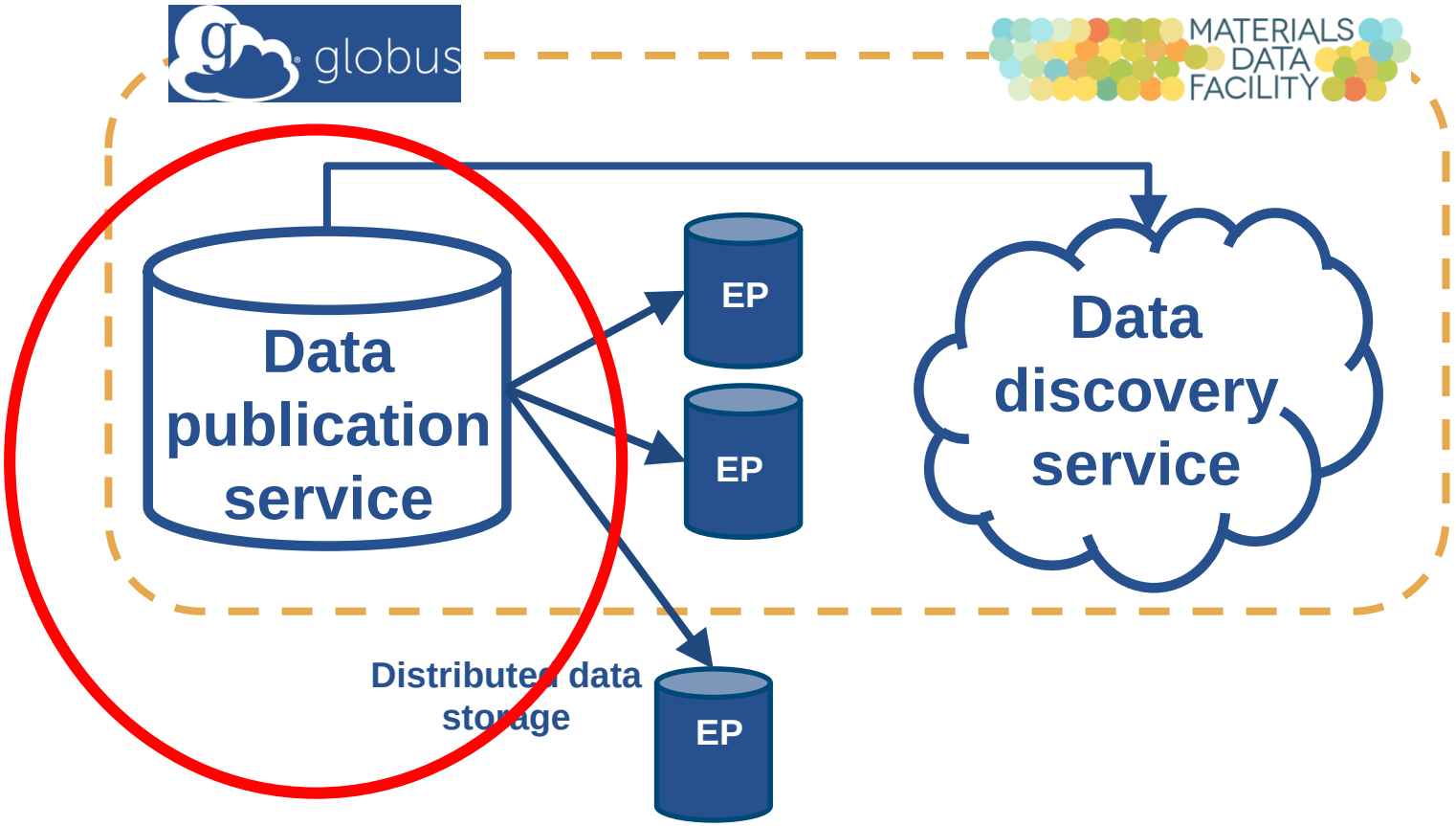
Transfer

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"])
```

DATA PUBLICATION



Materials Data Publication Service

MDF Open Collection home page

Open collection for submission of materials-related datasets

Submit to This Collection

Browse

Issue Date | Author | Title | Subject

Datasets in Collection (sorted by Submit Date in Descending order): 1 to 20 of 25

next >

Issue Date	Title	Author(s)
22-Sep-2017	**Dataset for A New Generation of Effective Core Potentials for Correlated Calculations**	*Bennett, M. Chandler; Melton, Cody A.; Annaberdiyev, Abdulgani; Wang, Guangming; Shulenburger, Luke; Mitas, Lubos*
11-Sep-2017	**Probing the growth and melting pathways of a decagonal quasicrystal in real-time**	*Han, Insung; Xiao, Xianghui; Shahani, Ashwin J.*
6-Sep-2017	**Simulated microstructures of gamma' precipitates in cobalt-based superalloys**	*Jokisaari, Andrea M.; Naghavi, Shahab; Wolverton, Chris; Voorhees, Peter W.; Heinonen, Olle G.*
23-Aug-2017	**Solute transport database in Mg using ab initio and exact diffusion theory**	*Agarwal, Ravi; Trinkle, Dallas R.*
29-Jun-2017	**Characterizing the Unifying Thread in High Temperature Superconductors Using Realistic Simulations**	*Narayan, Awadhesh; Busemeyer, Brian; Wagner, Lucas K.*

Datasets Are Citable

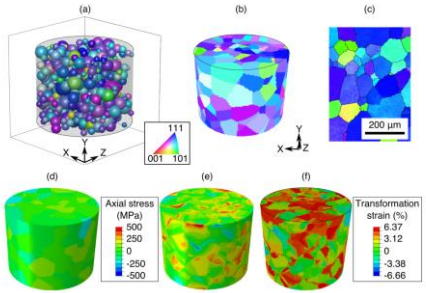
Title 1–20	Cited by	Year
Implications of Grain Size Variation in Magnetic Field Alignment of Block Copolymer Blends Y Rokhlenko, PW Majewski, SR Larson, P Gopalan, KG Yager, CO Osuji American Chemical Society		2017
X-ray Scattering Image Classification Using Deep Learning B Wang, K Yager, D Yu, M Hoai Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, 697-704	1	2017
Dataset of synthetic x-ray scattering images for classification using deep learning KG Yager, J Lhermitte, D Yu, B Wang, Z Guan, J Liu Materials Data Facility	1	2017
Magnetic field alignment of coil-coil diblock copolymers and blends via intrinsic chain anisotropy Y Rokhlenko, P Majewski, S Larson, K Yager, P Gopalan, A Avgeropoulos, ... Bulletin of the American Physical Society 62		2017

Publication via the NCSA

Data Volumes	**15.0 TB** **13.4 TB out**		
Publication	**50** Total datasets		**16** CHiMaD datasets
	94 Authors	**14** Institutions	**>1000** Accesses
Pipeline	**+30** Total datasets		**+14** CHiMaD datasets

Publication Route #1: NCSA Storage

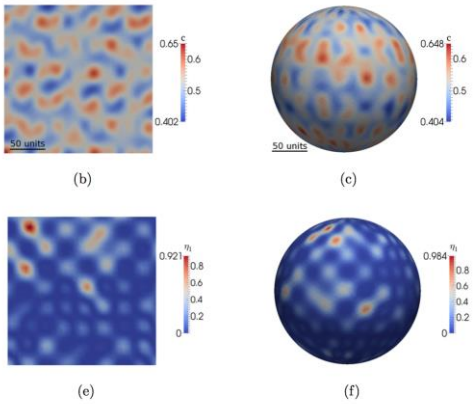
Grain Structure, Grain-averaged Lattice Strains, and Macro-scale Strain Data for Superelastic Nickel-Titanium Shape Memory Alloy Polycrystal Loaded in Tension



- Largest dataset to date (>1.5 TB). Showcases MDF unique capabilities and makes a unique dataset discoverable for code development, analysis, and benchmarking

Paranjape et al.

Phase Field Benchmark I Dataset



Jokisaari et al.

Electron Backscattering and Diffraction Datasets for Ni, Mg, Fe, Si

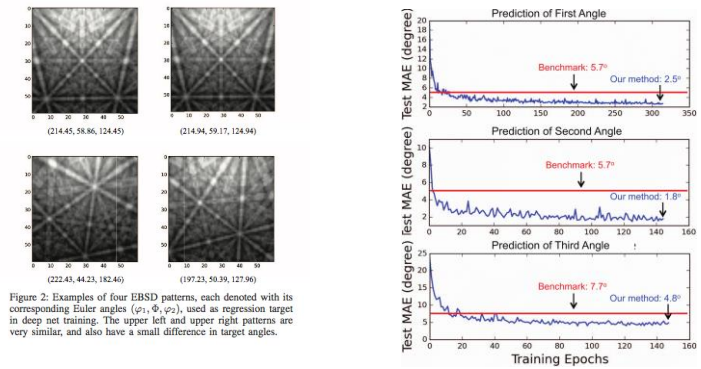
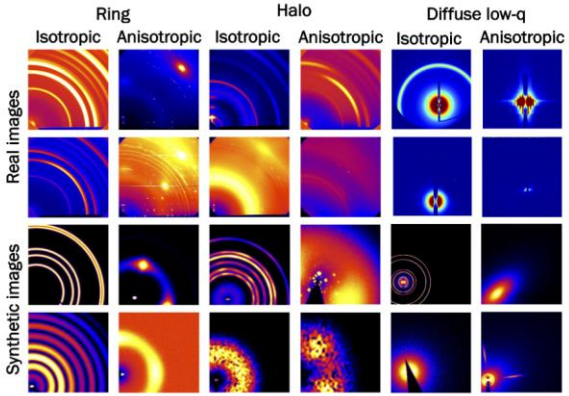


Figure 2: Examples of four EBSD patterns, each denoted with its corresponding Euler angles $(\varphi_1, \Phi, \varphi_2)$, used as regression target in deep net training. The upper left and upper right patterns are very similar, and also have a small difference in target angles.

Marc De Graef et al.

X-ray Scattering Image Classification Using Deep Learning



layer name	output size	kernels
conv1	112×112	7×7, 64, stride 2
		3×3 max pool, stride 2
conv2_x	56×56	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$
conv4_x	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$
conv5_x	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
pooling	1×1	average pooling
fc	1×1	2048×num of attributes

Yager et al. http://dx.doi.org/10.18126/M2Z30Z

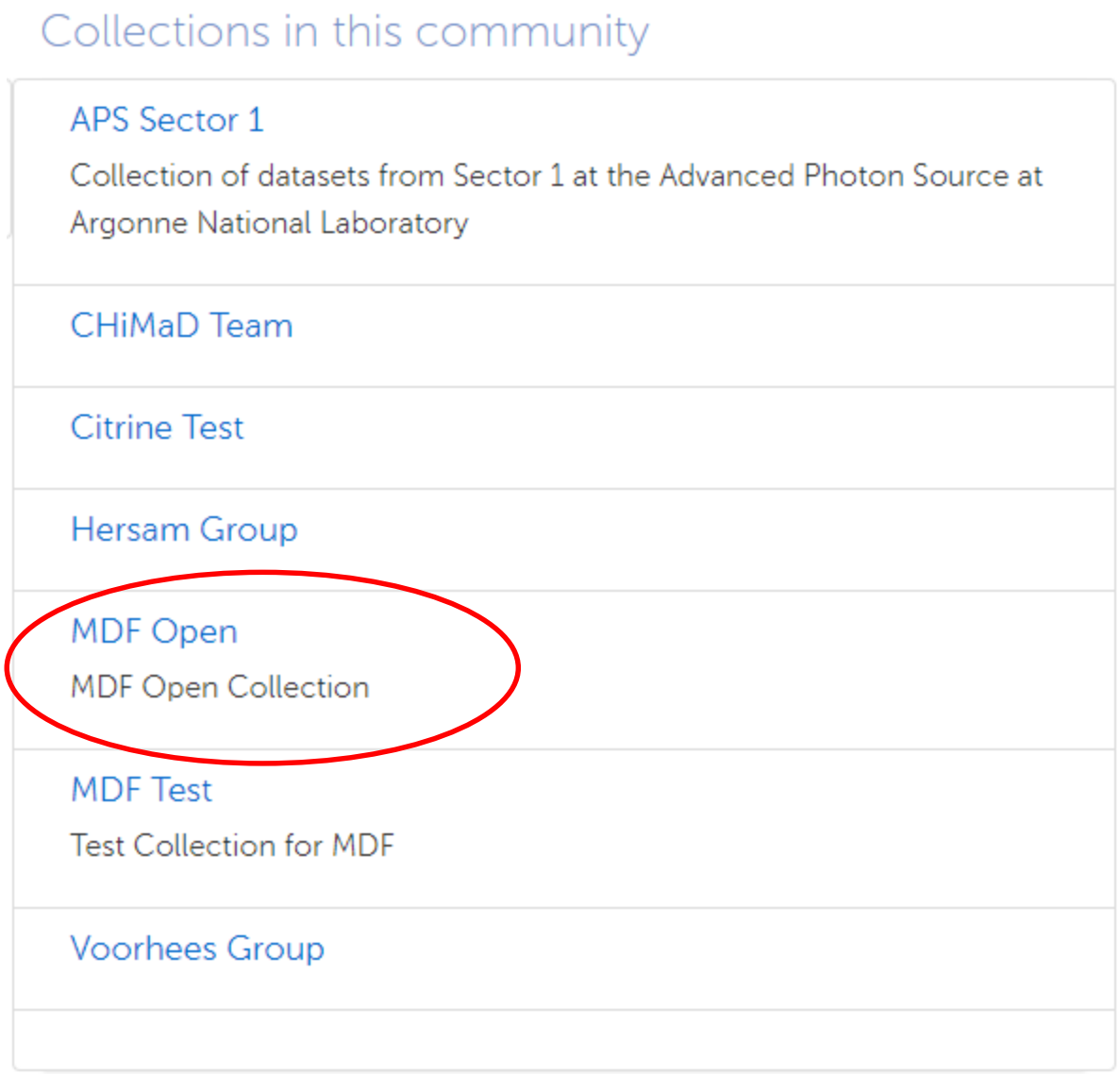
Publish Large Datasets

- Distributed data model leverages Globus production capabilities for file transfer (i.e. dataset assembly), user authentication, and access control groups

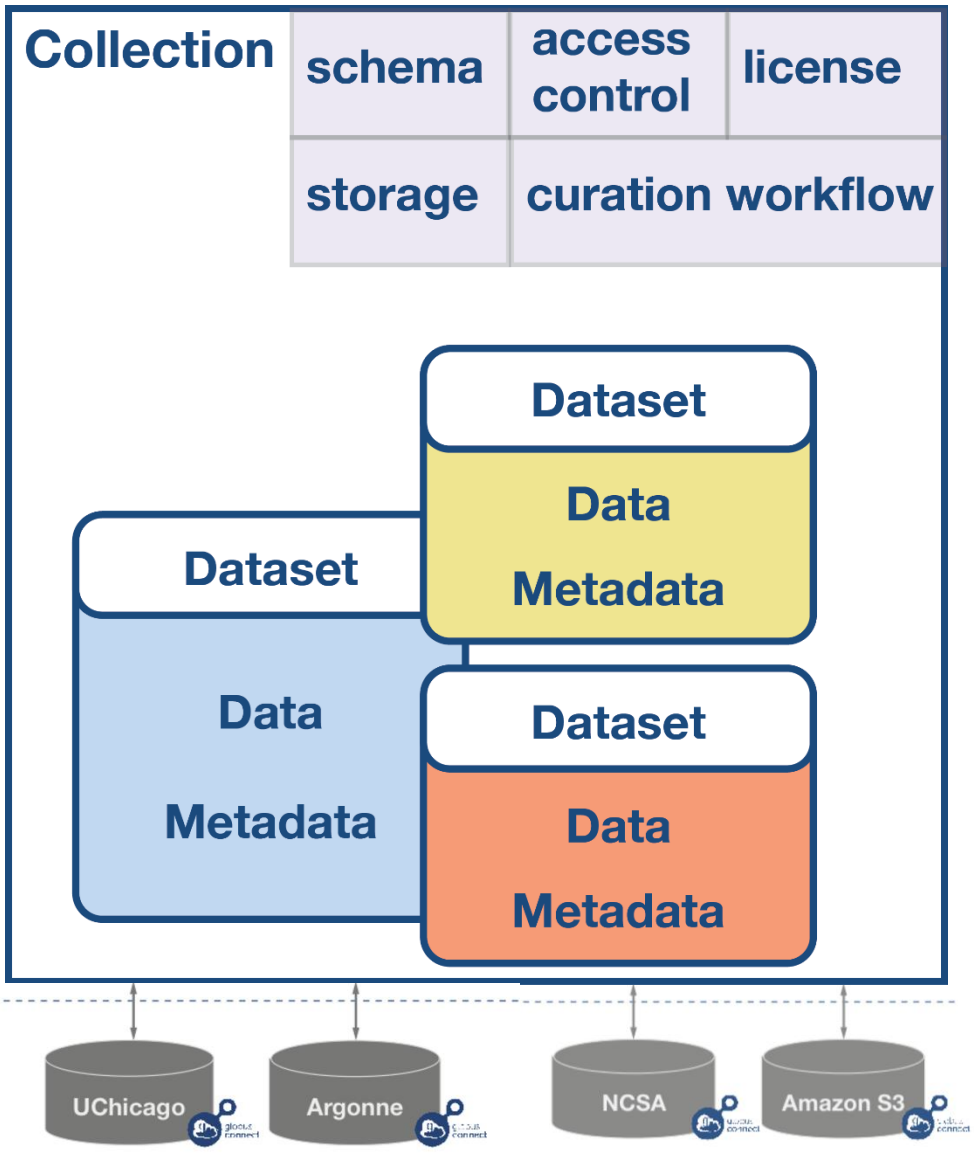
313,101,618,927 MB TRANSFERRED

- 100s of TB of reliable storage @ NCSA, and more storage at Argonne
  - Globus endpoint at ncsa#mdf on Nebula
  - Expandable to many PBs as necessary
  - Automated tape backup for reliability (in progress)

Customization: Collection Model



Customization: Collection Model



- **Collections might be a research group or a research topic...**
- **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 inde

Share Data with Flexible ACLs

- Share data publicly, with a set of users, or keep data private

Leverage Curation Workflows

- Collection administrators can specify the level of curation workflow required for a given collection e.g.
  - No curation
  - Curation of metadata only
  - Curation of metadata and files

Customize Metadata

- **Build a custom metadata schema for your specific research data**
- **Re-use existing metadata schemas**
- **Working in conjunction with NIST researchers to define these schemas**

Future...

- **Can we build a system that allows schema:**
  - **Inheritance**
    - E.g. a schema “polymers” might inherit and expand upon the “base material” of NIST
  - **Versioning**
    - E.g. Understand contextually how to map fields between versions
  - **Dependence**
    - E.g. Allows the ability to build consensus around schemas

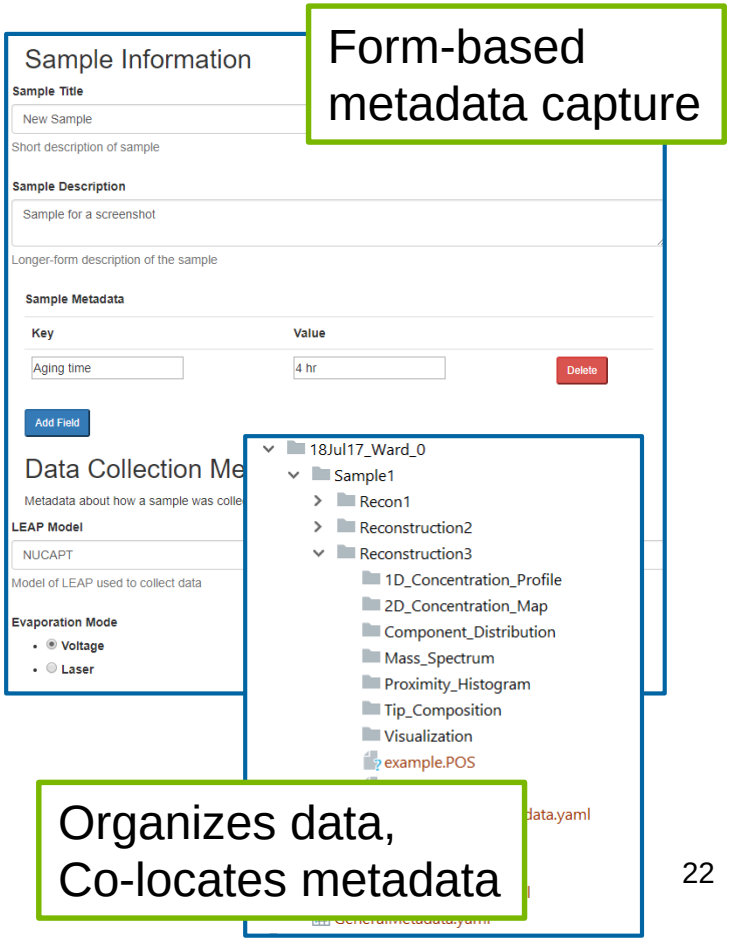
Example: NUCAPT Data Publication

MATERIALS DATA FACILITY

Goal:
- Aid metadata capture
- Simplify data publication

Approach: Lightweight web service
- Form-based metadata capture
- Automatic file management
- “One-click” data publication

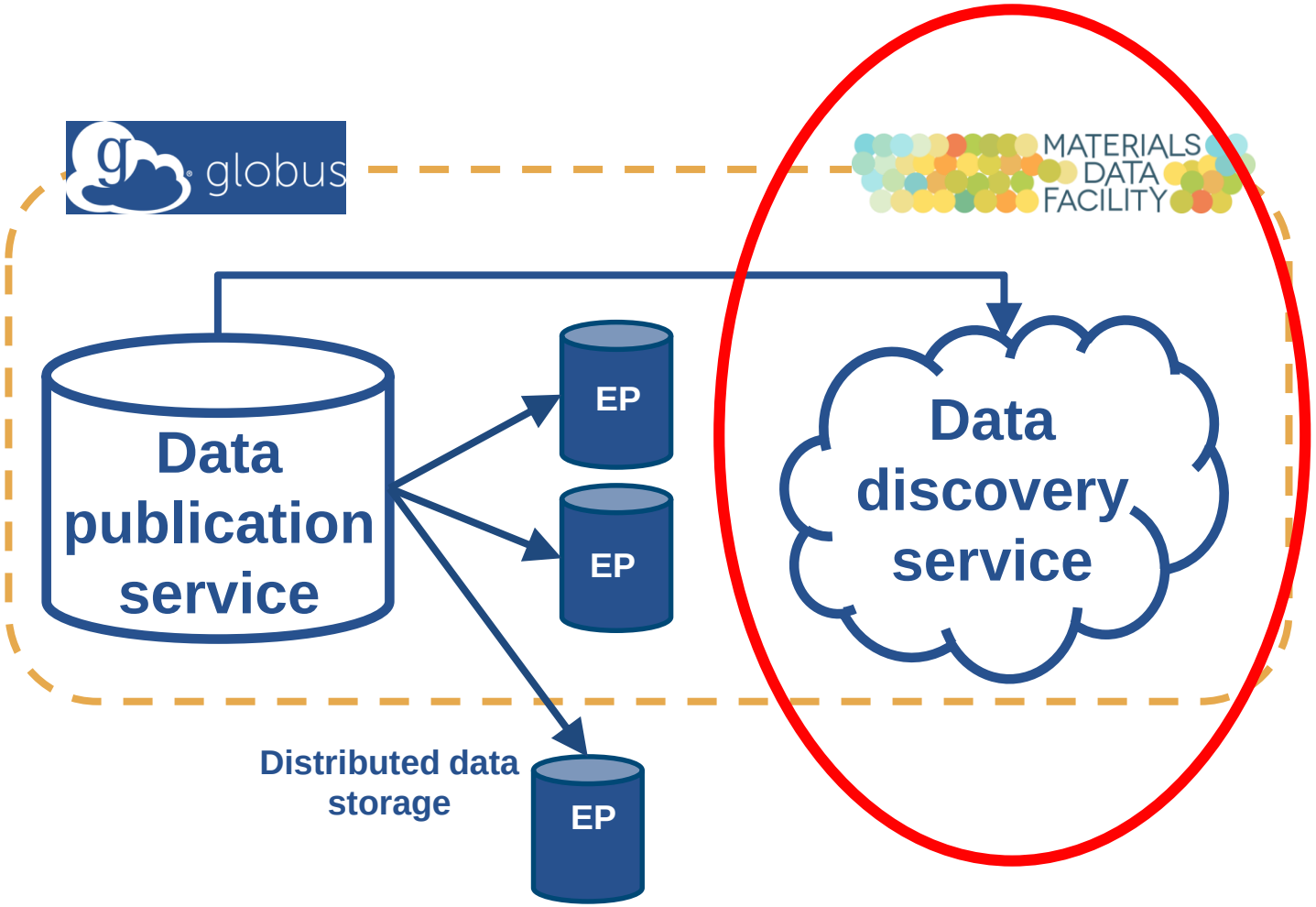
Results:
- Beta version deployed Sept ‘17



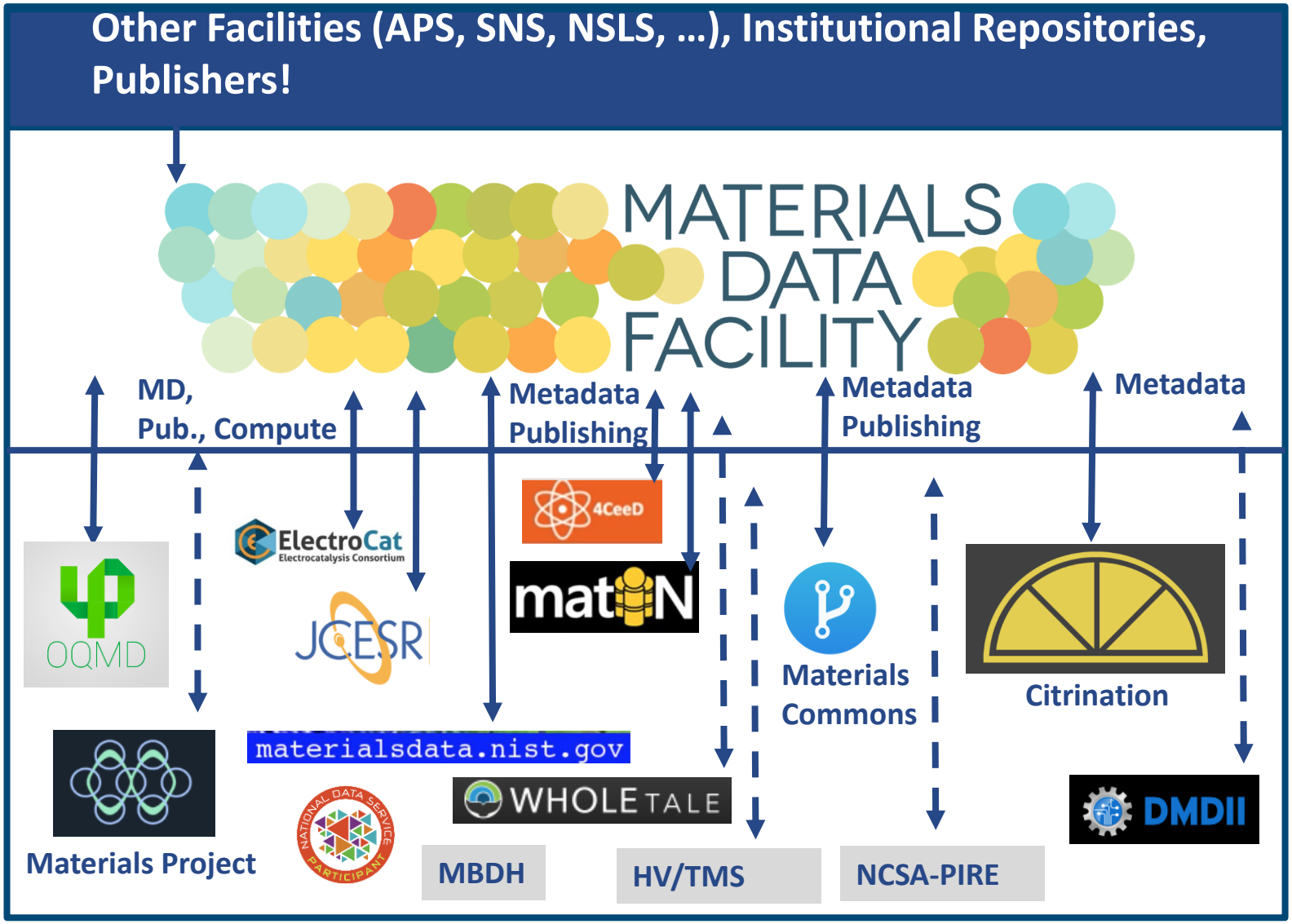
Form-based metadata capture

Organizes data, Co-locates metadata

DATA DISCOVERY [AND USE]

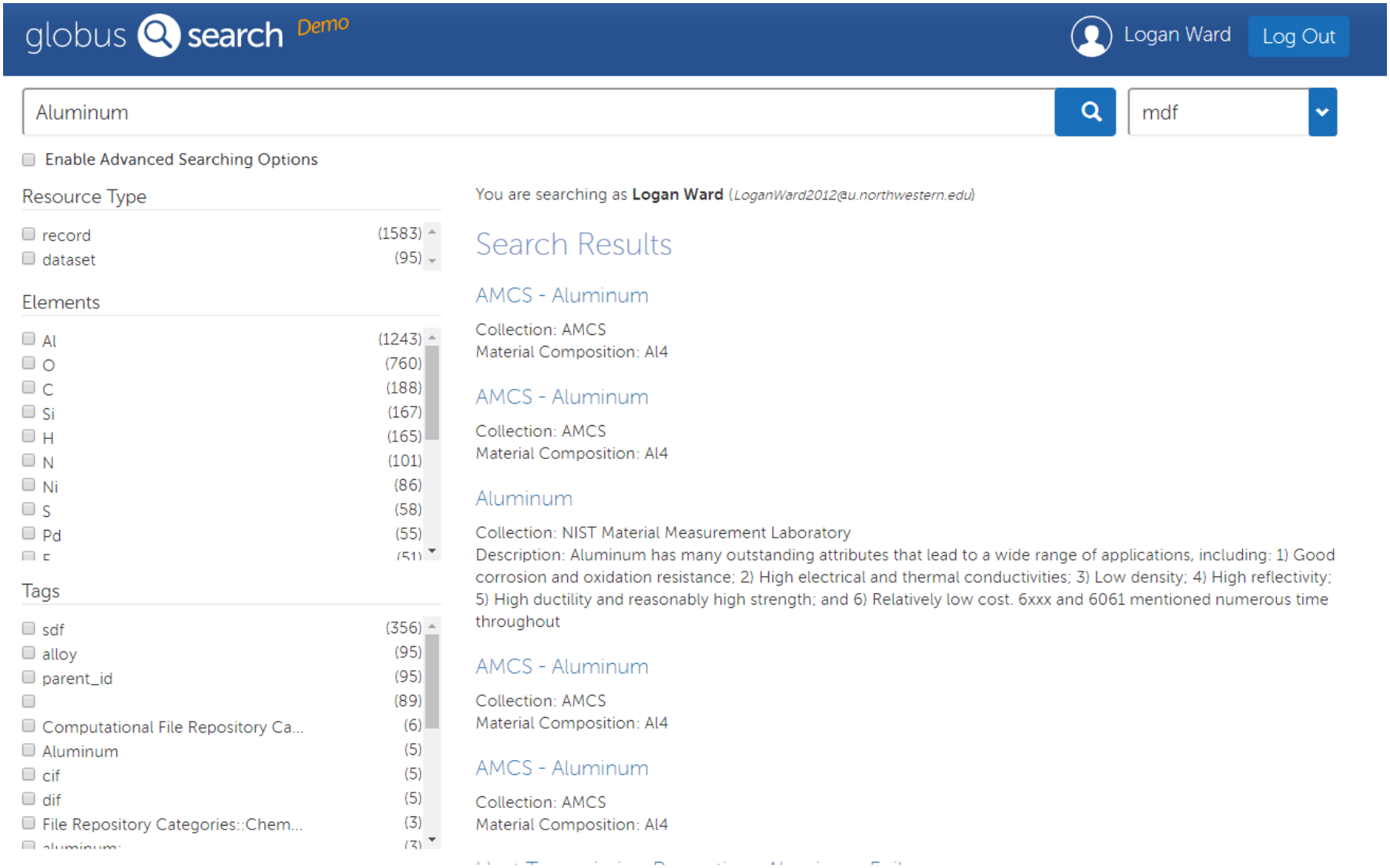


Part 1: Linking with the Data Community



zenodo MICCoM NOMAD The Dataverse Project

Many Databases, Single Search



MDF + NIST Database Tools

Querying Nanomine Data

Example using the Materials Data Facility to query data from NanoMine

In [1]:
```
from mdf_forge.forge import Forge
```

Get All Records

Get all of the records in NanoMine

In [2]:
```
forge = Forge()
```

In [3]:
```
data = forge.search('mdf.source_name=nanomine AND mdf.resource_type=record', advanced=True)
```

In [4]:
```
print('Found %d records in NanoMine'%len(data))
```

```
Found 227 records in NanoMine
```

MDF automates publicizing data and provides a uniform search interface

Get Records with Olefin Matrices

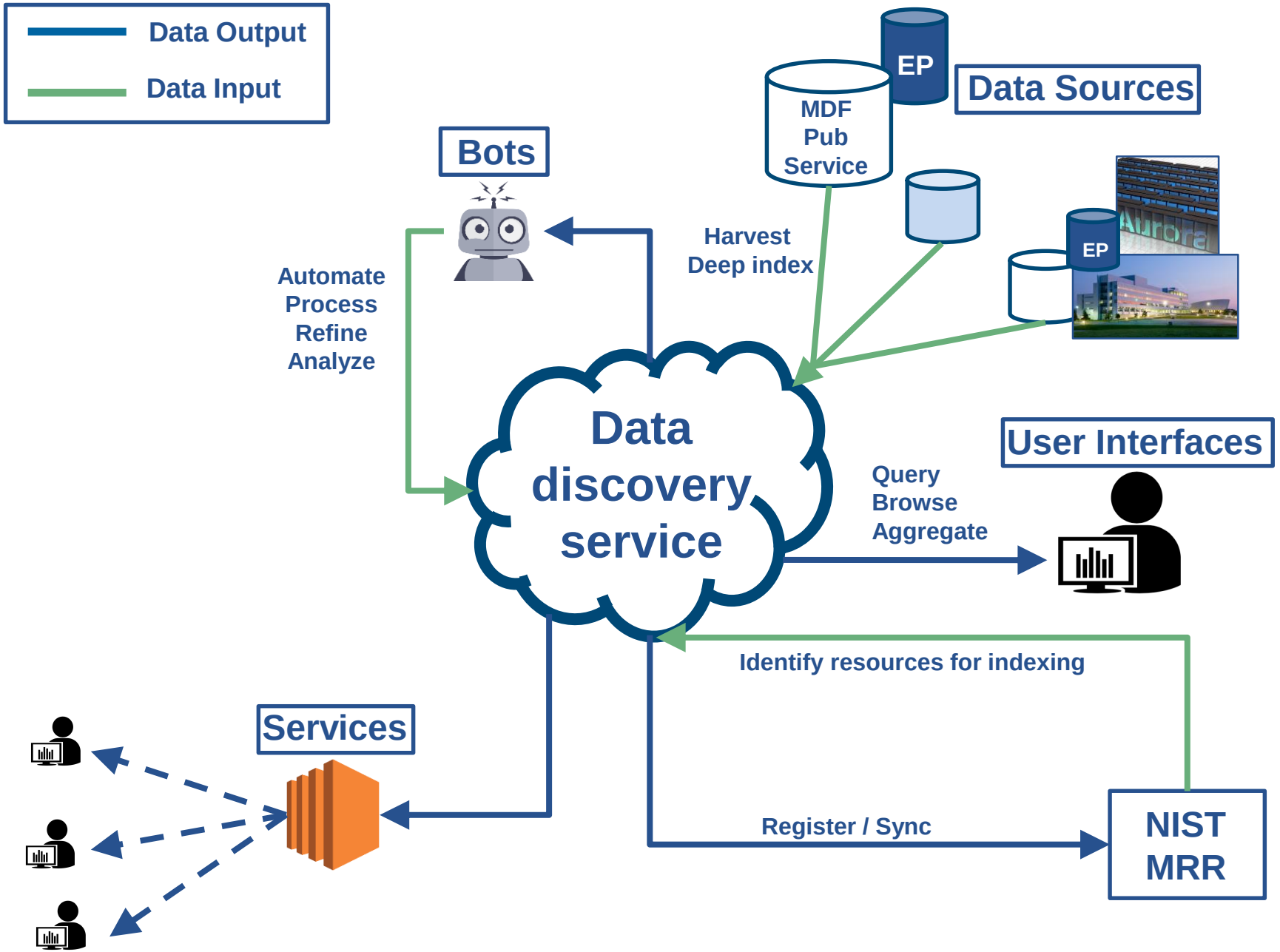
Example of a more-complex query

In [5]:
```
data = forge.search('mdf.source_name=nanomine AND '
                    'content.PolymerNanocomposite.MatrixComponent.ChemicalName=olefin', advanced=True)
```

In [6]:
```
print('Found %d olefin records'%len(data))
```

```
Found 6 olefin records
```

MDF data discovery ecosystem



Registering Data with Other Databases

1. User publishes DFT dataset

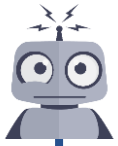
Our datasets are discoverable through many tools

MATERIALS DATA FACILITY

5. Ingest DFT data quality report

2. Bot requests open DFT data periodically
3. Bot accesses data, runs DFT parser to refine data

Jetstream



Alcoa

...

QUESTEK INNOVATIONS LLC

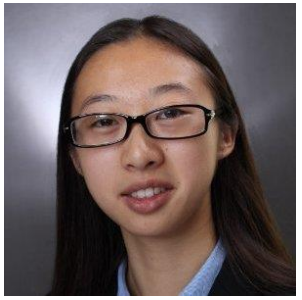
```
{ "category": "system.chemical",
   "chemicalFormula": "MgO2",
    "properties": {
       "units": "eV", "name": "Band gap",
       "scalars": [ { "value": 7.8 } ] } }
```

4. Push metadata to Citrine

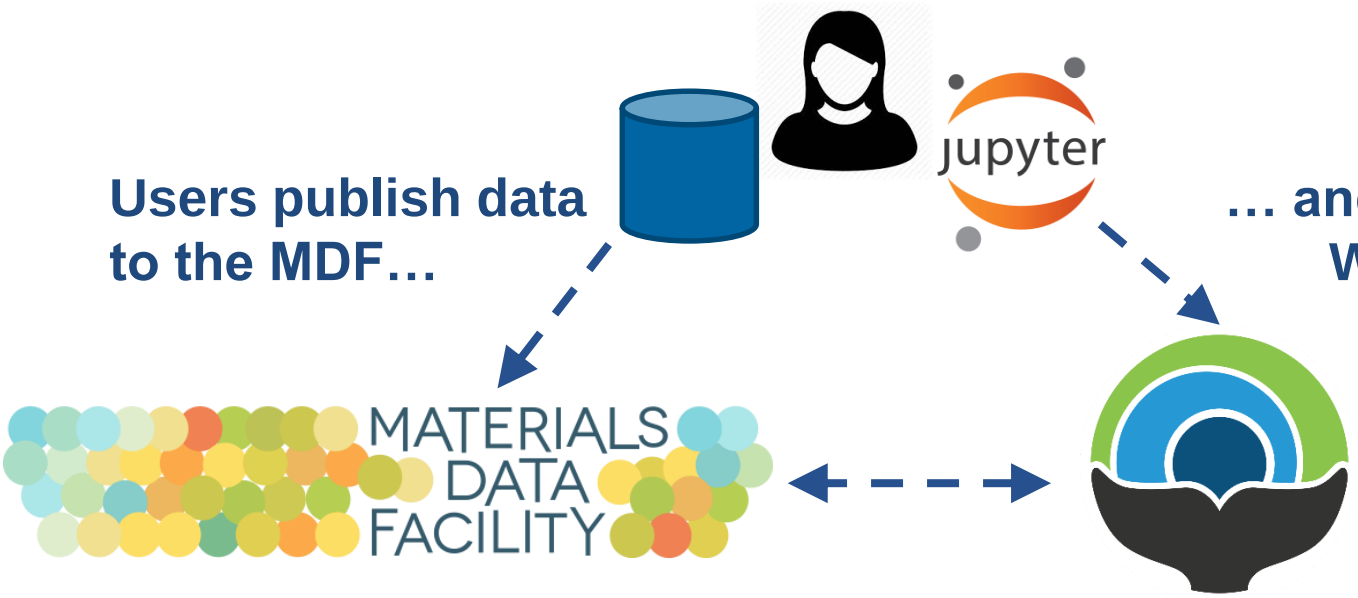
Citrine

Reproducing data-driven MSE with MDF

- Summer Intern (Jiming Chen) reproducing and extending materials and ML papers with the MDF
- Joined our team with the NSF WholeTale project



Jiming Chen (UIUC)



Long-term goals:

- Assemble community-driven resource for ML tools/examples

- Use MDF/WholeTale to create benchmark challenges

Replicating Ward *et al*. 2016

Train a Model to Predict Formation Energy using the MDF

This notebook demonstrates how to create an model to predict the formation energy of crystalline materials using data from the MDF. Specifically, we will use data from the OQMD and train a model using the technique describe in a recent paper by Ward *et al.*

In [1]:
```
%matplotlib inline
from mdf_forge.forge import Forge
from pymatgen import Composition
from pymatgen.core.periodic_table import Element
from matminer.featurizers import composition as cf
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import cross_val_score, cross_val_predict, GridSearchCV
from sklearn import metrics
from matplotlib import pyplot as plt
import numpy as np
import itertools
```

Coming soon: Publishing both data and code with MDF/WholeTale

Setti

In [2]:
```
quic
```

Get OQMD Training Set

Ward *et al.* trained their machine learning models on the formation enthalpies of crystalline compounds form the OQMD. Here, we extract the data using the copy of the OQMD available through the MDF

Download the Data

We first create a `Forge` instance, which simplifies performing search queries against the MDF.

In [3]:
```
forge = Forge()
```

Then, we get all the converged results from the OQMD

In [4]:
```
query_string = 'mdf.source_name:oqmd AND (oqmd.configuration:static OR oqmd.configuration:standard) AND oqmd.converged:True'
if quick_demo:
    query_string += " AND mdf.scroll_id:<10000"
```

APIs, Automation, and Examples

MDF Forge python package (under development)

- Interface to MDF services
- Helper functions for common tasks

Forge

build passing pypi v0.4.0

Forge is the Materials Data Facility Python package to interface and leverage the MDF Data Discovery service. Forge allows users to perform simple queries and facilitiates moving and synthesizing results.

Installation

https://github.com/materials-data-facility/forge

```
pip install mdf_forge
```

For Developers

```
git clone https://github.com/materials-data-facility/forge.git
cd forge
pip install -e .
```

Tools for using these capabilities are available now

```
from mdf_forge.forge import Forge

mdf = Forge()

# free text query
r = mdf.search("materials commons")
```

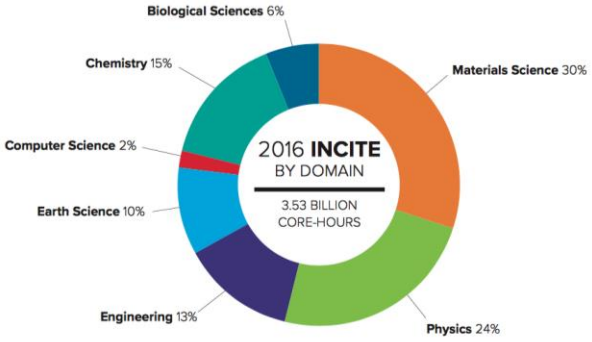
FUTURE DIRECTIONS

Besides what I just showed you

MDF – Argonne Leadership Computing Facility partnership

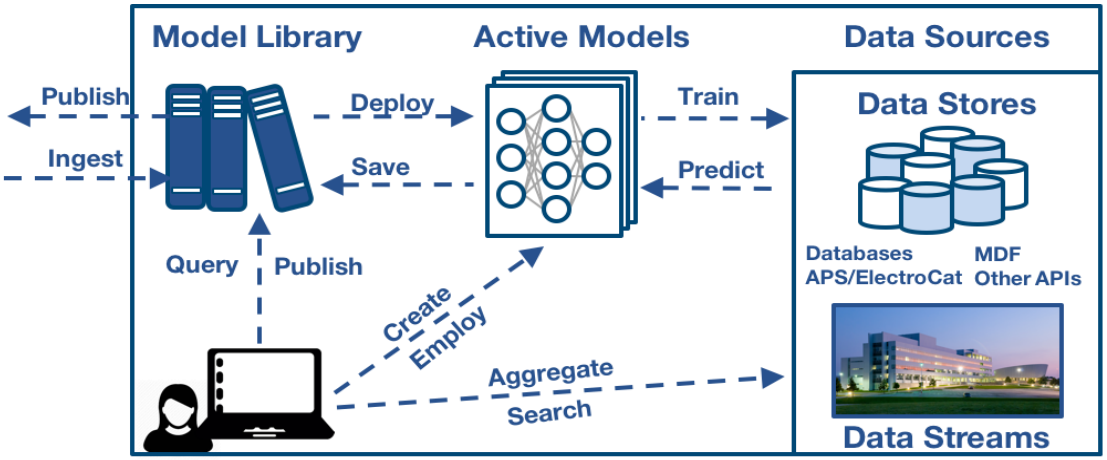
- **MDF will work with materials users to publish and deep index results obtained via ALCF projects (i.e., INCITE, ADSP,)**
  - Direct access to ALCF user base (~1000 /yr)
  - MDF will be incorporated in their existing UX flow (e.g., users can opt into data sharing at the beginning of the project)
  - Covers ~850M core hours of projects in 2017

- **Well-positioned to become the *de facto* standard for data publication and discovery by opening of the A21 exascale system in 2021 Q1**
  - 100x data increases expected



- **Bootstrapping effort has been established to expand capabilities to ALCF users in physics, chemistry, bio, earth sciences**
  - Seed effort to cover a separate team (ex: project management oversight by Blaiszik) to ensure continued focus of the MDF team on materials science related goals
  - Will also be used to investigate co-locating datasets with HPC to allow HPC operations on the data through simple interfaces like Jupyter

DLHub: Advancing Deep Learning Adoption



- **Publish and share models and code linked with full training datasets**
- **Link database with HPC/Cloud computing resources**
- **Provide uniform interface for training, running models**

Summary

Three Major Components of Materials Data Facility

1. Globus
   - High speed data transfer
   - Easy data sharing
2. Data Publication Service
   - Simple data publication, from your own
   - Free data publication via the NCSA
3. Data Discovery Service
   - Single search engine for many materials databases
   - Python API for accessing these databases

Thanks to our sponsors!

U.S. DEPARTMENT OF ENERGY

THE UNIVERSITY OF CHICAGO

Additional Hands-On Demos

1. Using Globus SDK
2. Publishing Data via MDF
3. NUCAPT Data Manager
4. MDF Search API and UI
5. Building ML Model with MDF