Democratizing Compute at All Scales

aarnet

Australia's Academic and Research Network

Going Beyond Data Management with Globus

Vas Vasiliadis
vas@uchicago.edu

June 7, 2023

THE UNIVERSITY OF CHICAGO

Argonne NATIONAL LABORATORY

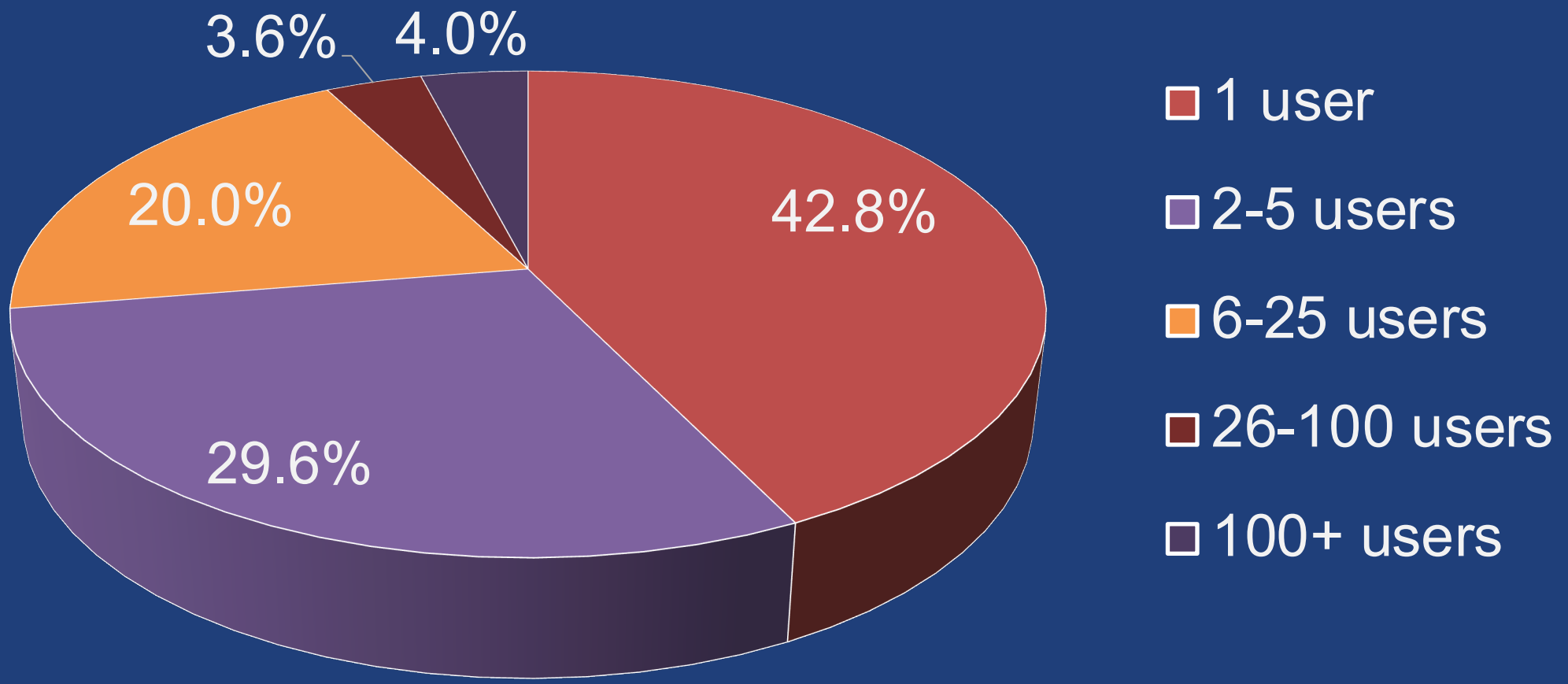
globus

Full Disclosure

All content in this presentation was prepared by a human (borrowing liberally from smarter colleagues).

Any resemblance to content generated by an LLM or other artificially “intelligent” entity is purely coincidental, and likely an artifact of poor software engineering.

A representative picture of advanced computing



% of institutions by number of active users of ACCESS compute resources (12 months ending 4/30/2023)

Data provided by xdmod.access-ci.org

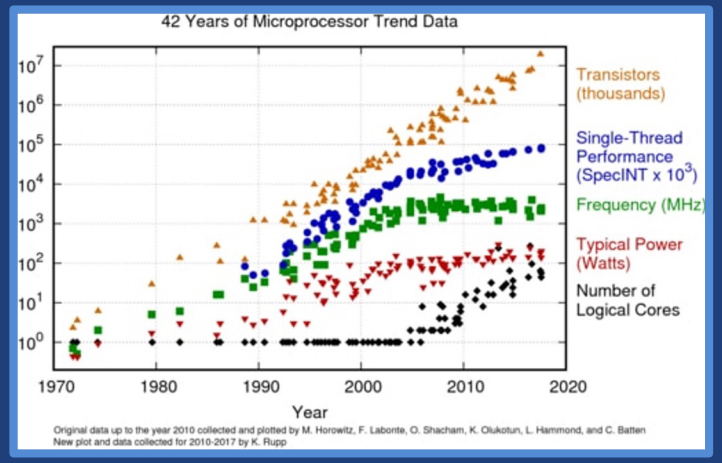
Reasons contributing to this picture

- **Remote computing is notoriously complicated**
  - Authentication
  - Network connections
  - Configuring/managing jobs
  - Interacting with resources (waiting in queues, scaling nodes)
  - Configuring execution environment
  - Getting results back again
- **Researchers need to overcome the same obstacles every time they move to a new resource**

Further confounding factors

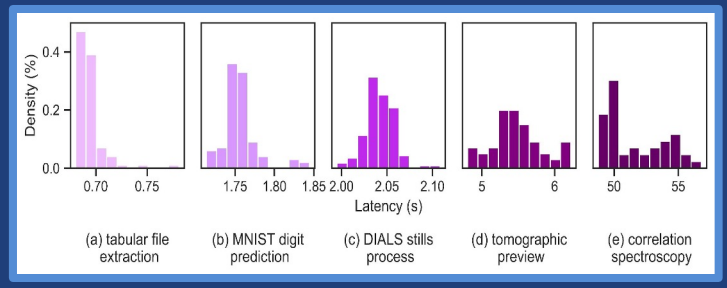
Resources

- Hardware specialization
- Specialization leads to distribution



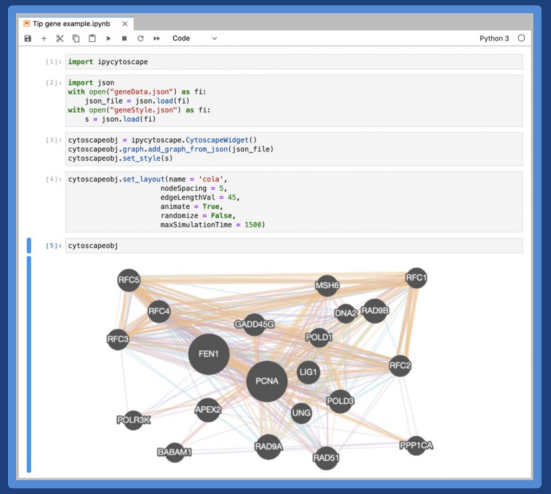
Workloads

- Interactive, real-time workloads
- Machine learning training and inference
- Components may best be executed in different places



Users

- Diverse backgrounds and expertise
- Different user interfaces (e.g., notebooks)



How can we bridge this gap?

Borrow page from data management playbook

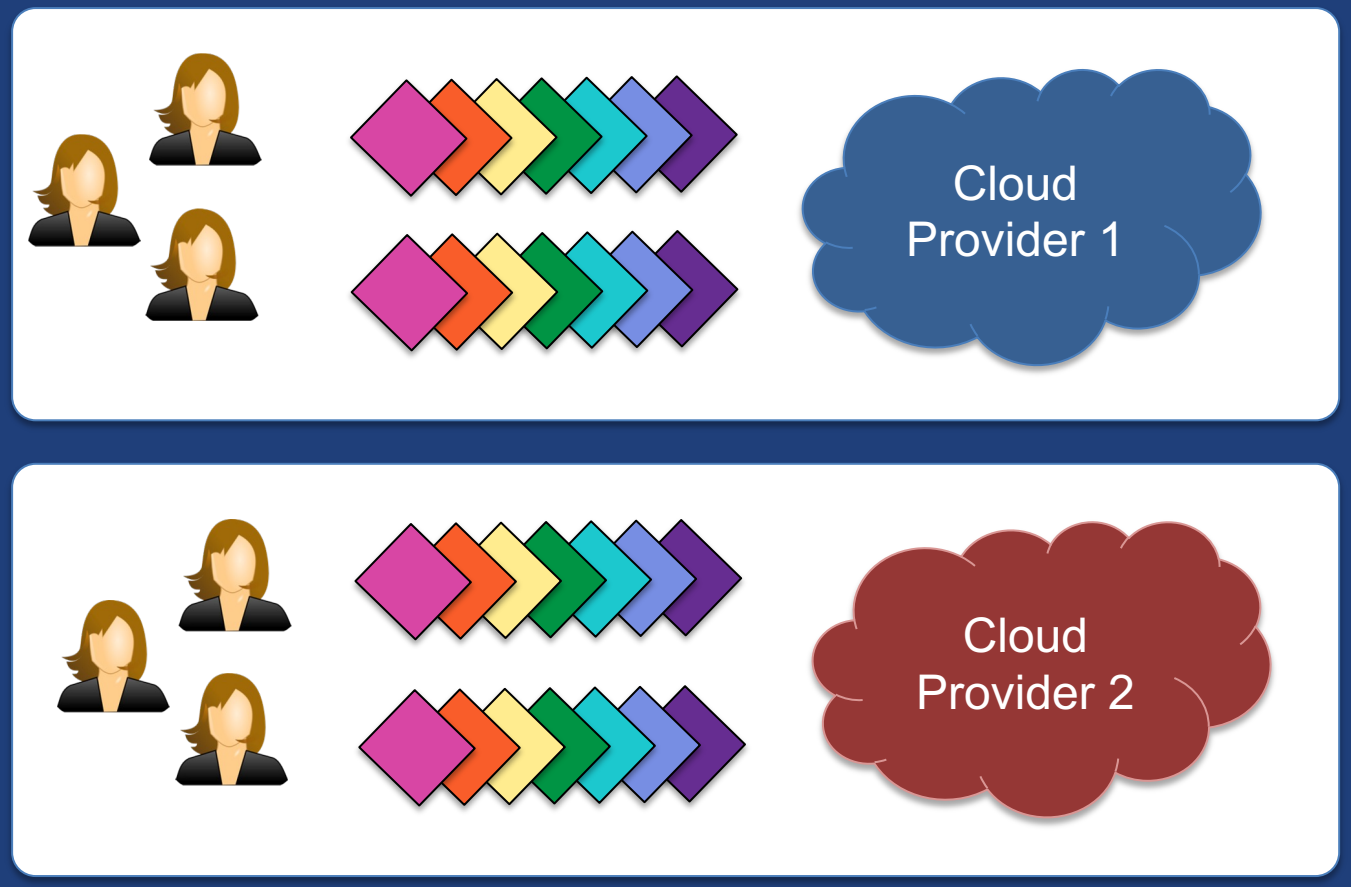
- ➔ **“Fire-and-forget” computation**
- ➔ **Uniform access interface**
- ➔ **Federated access control**
- ➔ **Fast networks making compute resource “local”**

How can we bridge this gap?

Move closer to researchers' environments

- ➔Researchers primarily work in high level languages
- ➔Functions are a natural unit of computation
- ➔FaaS allow researchers to work in a familiar language (e.g., Python) using familiar interfaces (e.g., Jupyter)

FaaS as offered by cloud providers



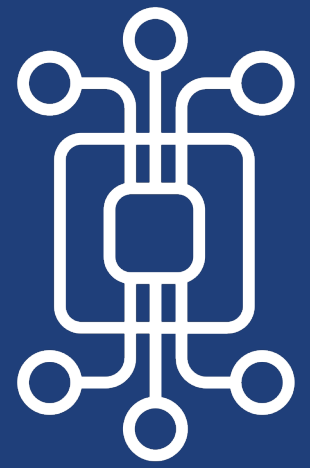
- ***Single provider, single location*** **to submit and manage tasks**
- **Homogenous execution environment**
- **Transparent and elastic execution (of even very small tasks)**
- **Integrated with *cloud provider data management***

FaaS as an interface to the advanced computing ecosystem

Still need…

- **Single interface**
- **Homogenous execution environment**
- **Transparent and elastic execution**
- **Integrated with data management**

Globus Compute helps bridge the gap



Managed, federated Function-as-a-Service for reliably, scaleably and securely executing functions on remote endpoints from laptops to supercomputers

THE UNIVERSITY OF CHICAGO

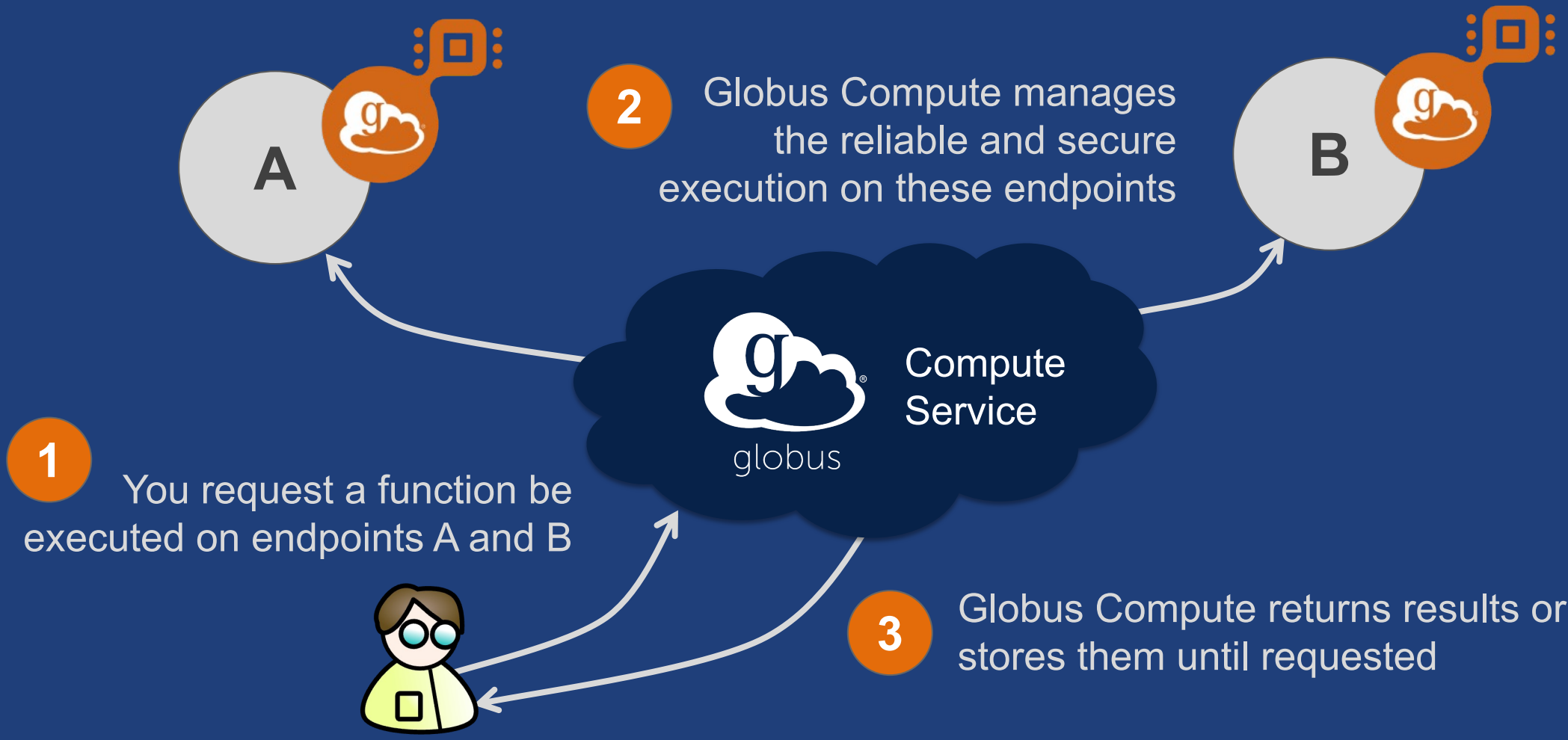
ILLINOIS

Argonne NATIONAL LABORATORY

The **Globus Compute** model

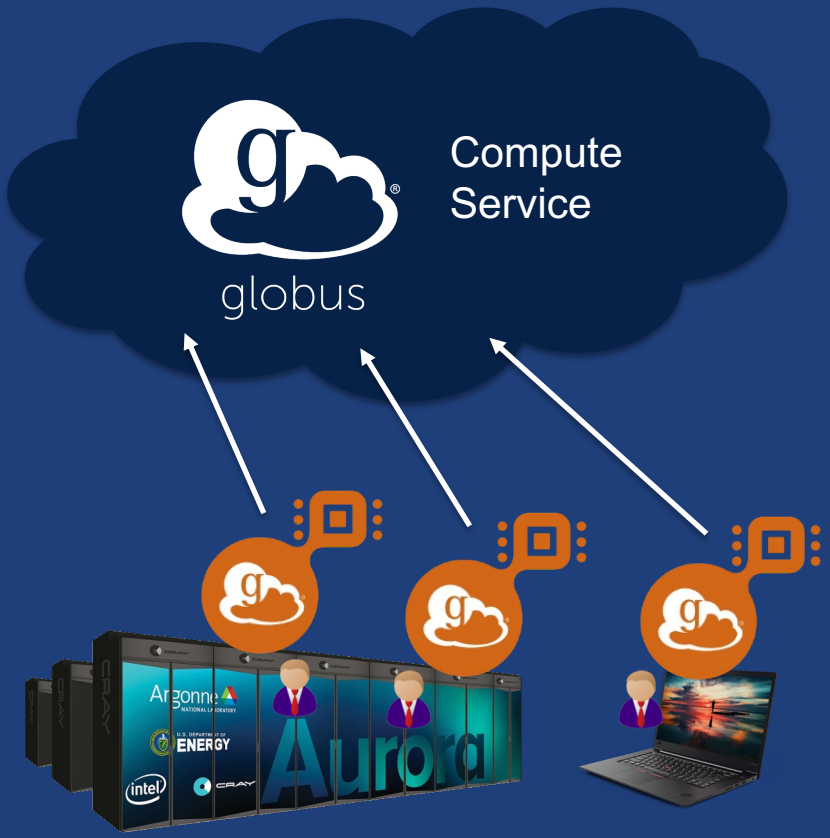
- **Compute service — Highly available cloud-hosted service for managed, fire-and-forget function execution**
- **Compute endpoint — Abstracts access to compute resources, from edge device to supercomputer**
- **Compute SDK — Python interface for interacting with the service, suing the familiar (to many) Globus look and feel**
- **Security — Leverages Globus Auth; Compute endpoints are resource servers, authN and access via Oauth tokens**

User interaction with Globus Compute



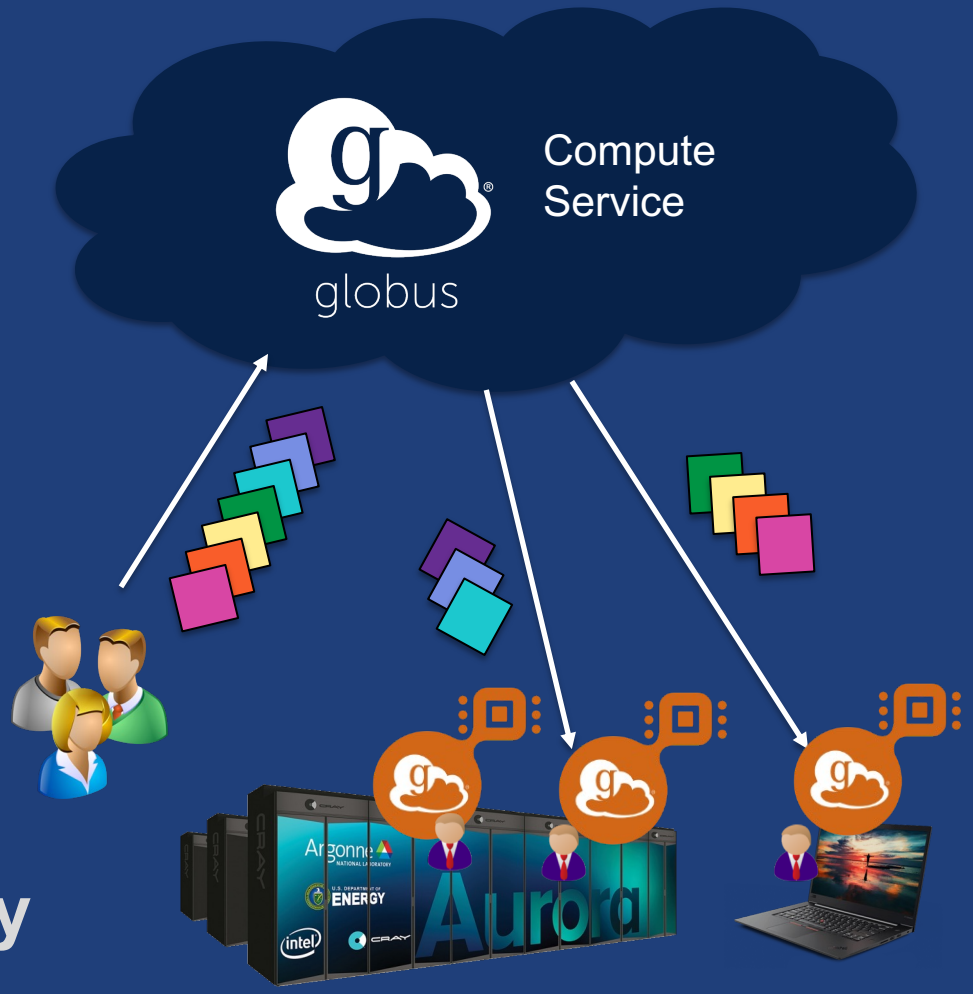
Globus Compute transforms any computing resource into a function serving endpoint

- **Python `pip` installable agent**
- **Elastic resource provisioning from local, cluster, or cloud system (via Parsl)**
- **Parallel execution using local fork or via common schedulers**
  - Slurm, PBS, LSF, Cobalt, K8s



Executing functions with Globus Compute

- **Users invoke functions as tasks**
  - Register Python function
  - Pass input arguments
  - Select endpoint(s)
- **Service stores tasks in the cloud**
- **Endpoints fetch waiting tasks (when online), run tasks, and return results**
- **Results stored in the cloud and on Globus storage endpoints**
- **Users retrieve results asynchronously**



Common Use Cases

Use case 1: Fire-and-forget execution

Executing a bag of tasks, e.g., running simulations with different parameters, executing ML inferences, on one or more remote computers directly from your environment, e.g., Jupyter on your laptop

- **Fire-and-forget execution managed by Globus Compute; tasks/results cached until endpoint/client are online**
- **Portability across different systems—optionally making use of specialized hardware**
- **Elastic scaling to provision resources as needed (from HPC and cloud systems)**

ML-based drug screening

CHEMICAL LIBRARY DATABASE

4B known molecules

Enamine

DRUGBANK

BindingDB GDB

eMolecules

cureFFI MOSES

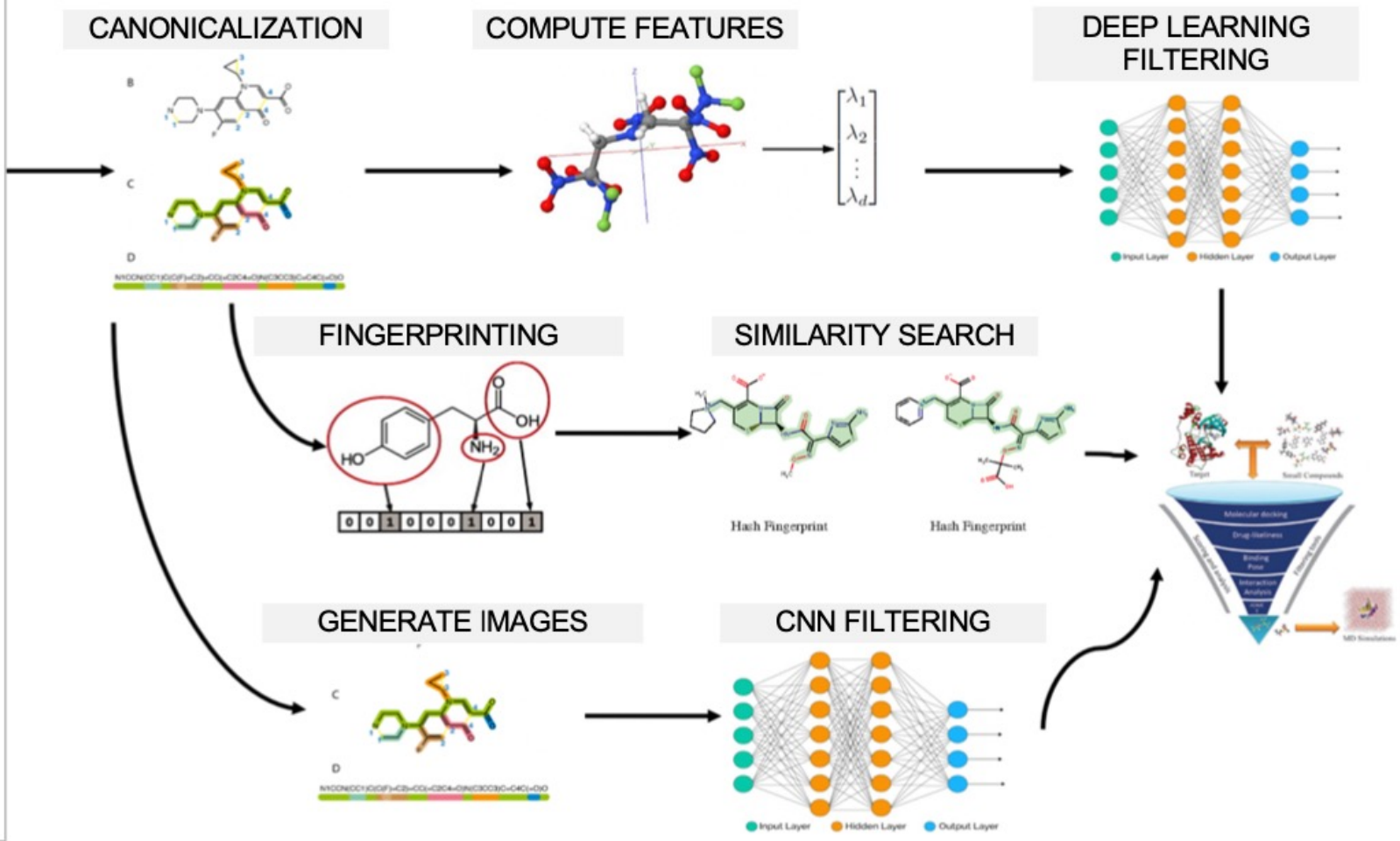
ZINC15

LINCS

SureChEMBL

PubChem

AND MORE

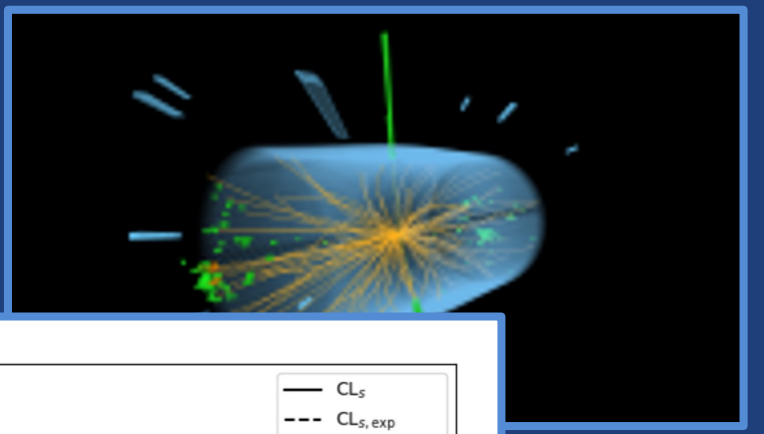


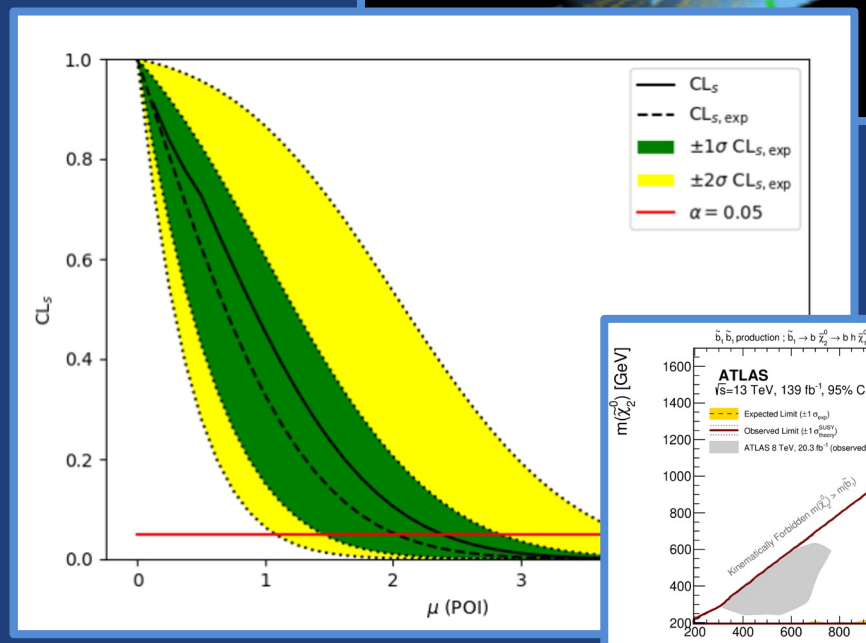
COMPUTING RESOURCES

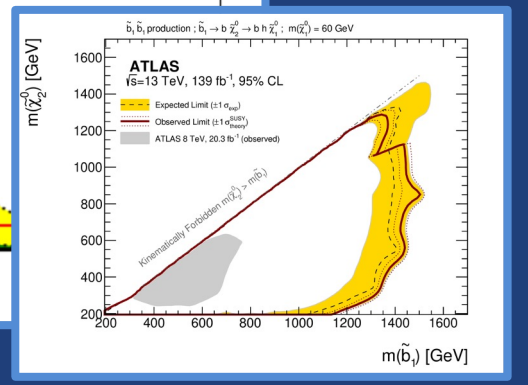
National Virtual Biotechnology Laboratory, arXiv:2006.02431

Distributed statistical inference with pyhf

- Large Hadron Collider statistical inference to extract physics information
- Tools traditionally implemented in C++; difficult for new users to setup/run
- pyhf: pure-Python library with automatic differentiation and hardware acceleration
- Hypothesis fitting is a pleasingly parallel problem with short duration tasks
- Researchers want to opportunistically leverage institution, HPC, OSG resources







Feickert, Heinrich, Stark, Galewsky - arxiv.org/abs/2103.02182

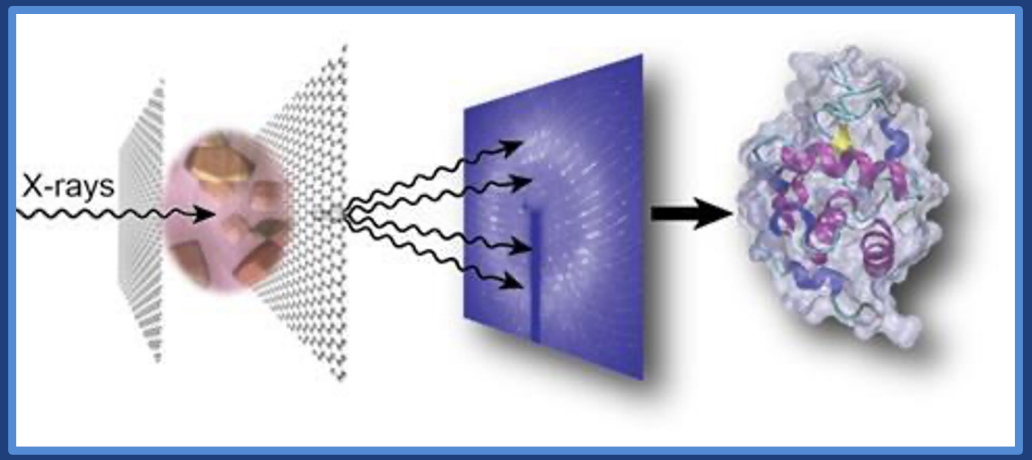
Use case 2: Automated analysis of data

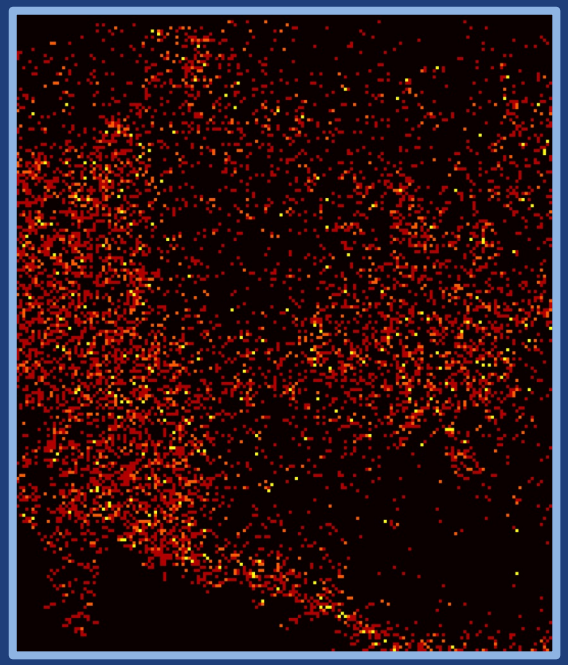
Constructing and running automated analysis pipelines with data processing steps that need to be executed in different locations

- **Automatically process data as they are acquired (event- and workflow-based)**
- **Integrate with data movement and other actions—human and machine**
- **Execute functions across the computing continuum e.g., near instrument, in data center, on specialized hardware**

Enabling serial crystallography at scale

- **Serially image chips with 000's of embedded crystals**
- **Quality control first 1,000 to report failures**
- **Analyze batches of images as they are collected**
- **Report statistics and images during experiment**
- **Return structure to scientist**

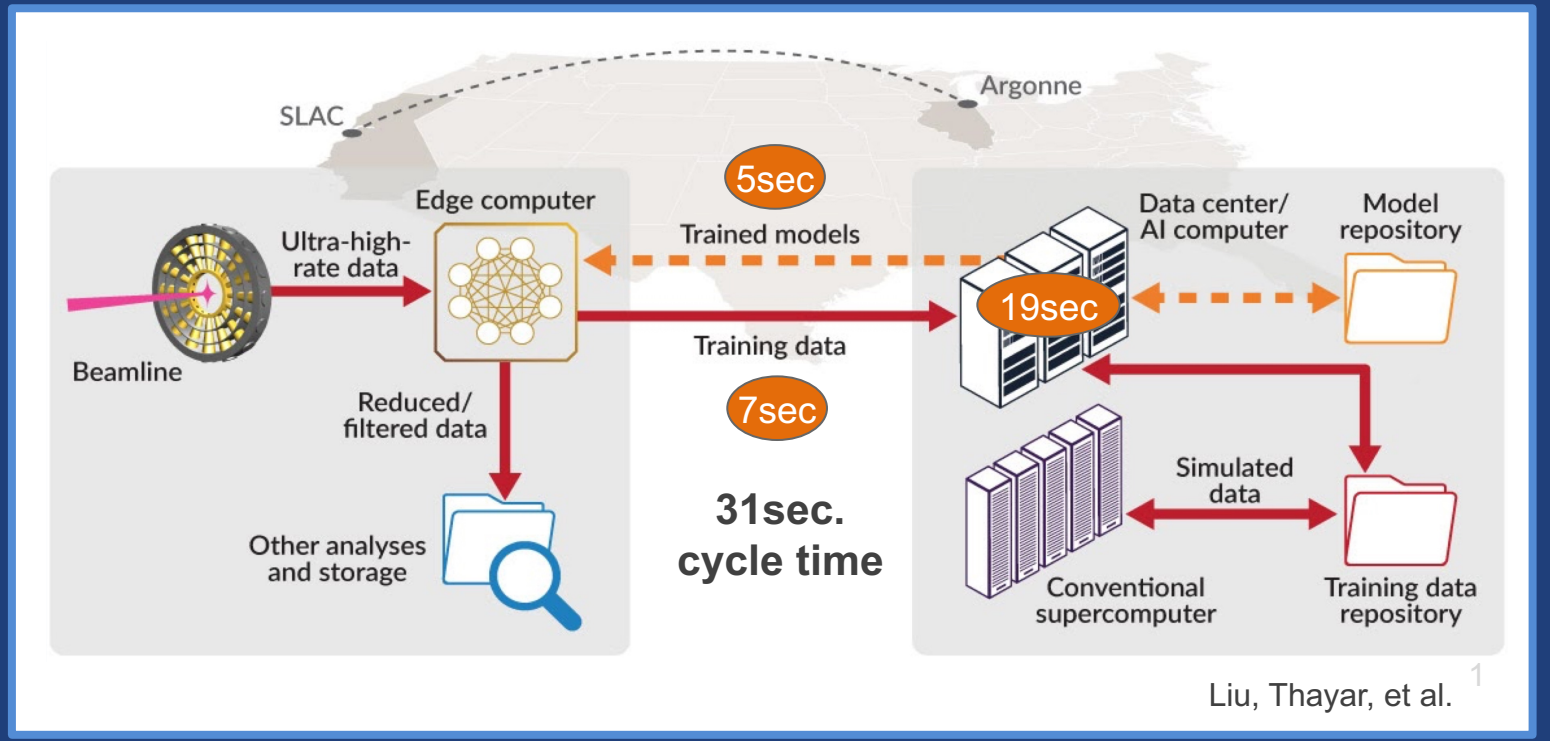




Darren Sherrell, Gyorgy Babnigg, Andrzej Joachimiak

Remote training of deep neural networks

- **DNN at the edge for fast processing, filtering, QC**
- **Requires tight coupling with simulation and training with real-time data**
- **Near real-time steering of experiment towards points of interest**

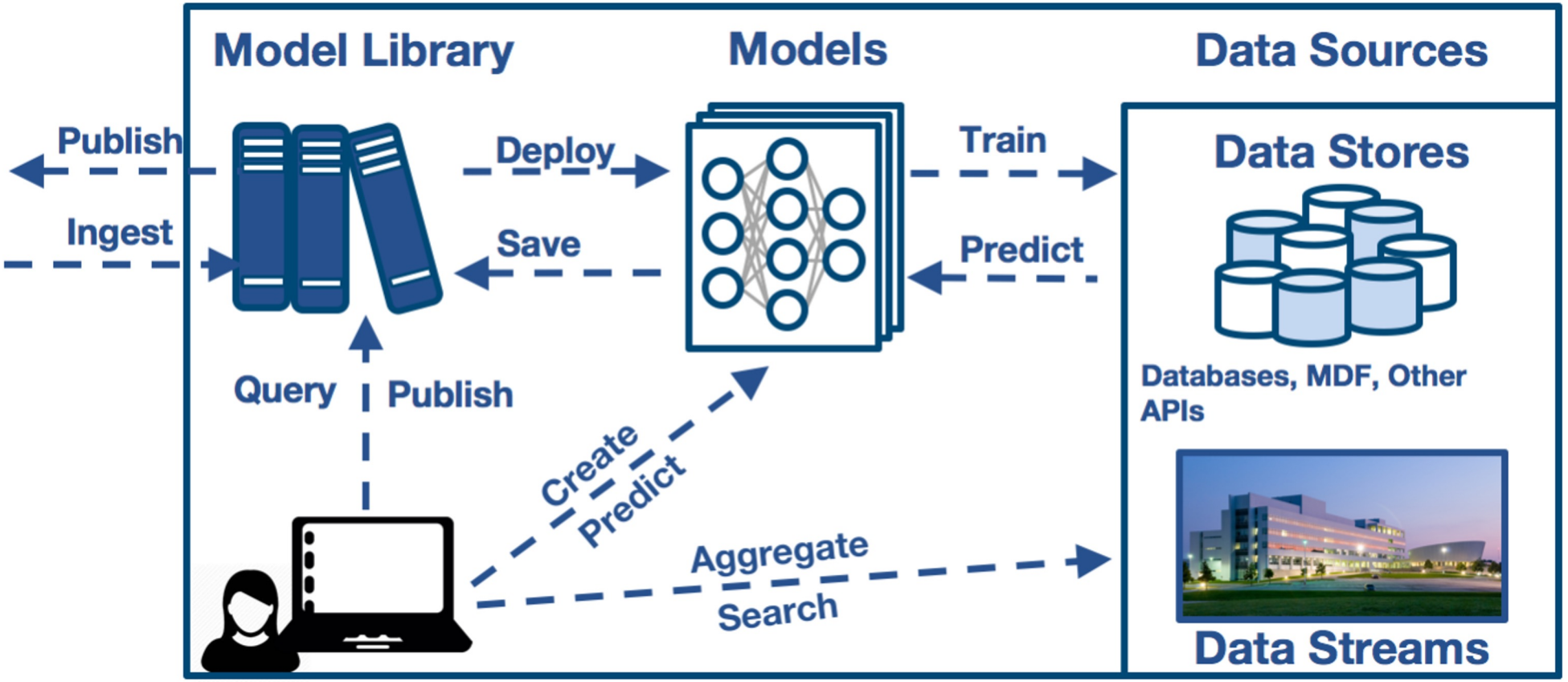


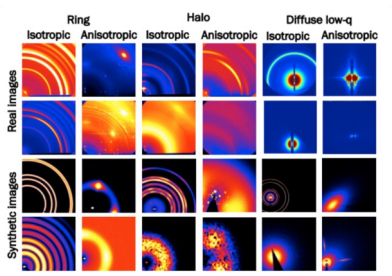
Use case 3: Globus Compute as a platform

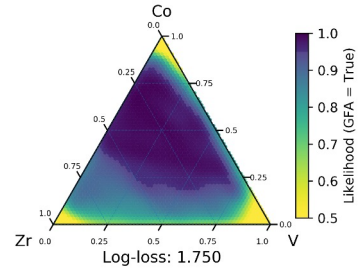
Building new applications and services that seamlessly execute application components or user workloads on remote resources

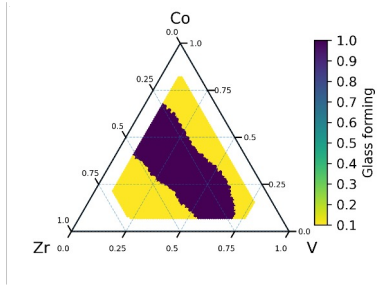
- **Robust, secure, and scalable platform for managing parallel and distributed execution across a federated ecosystem of computing endpoints**
- **Simple cloud-based API and Python SDK for integration**

The Data and Learning Hub for Science (DLHub)





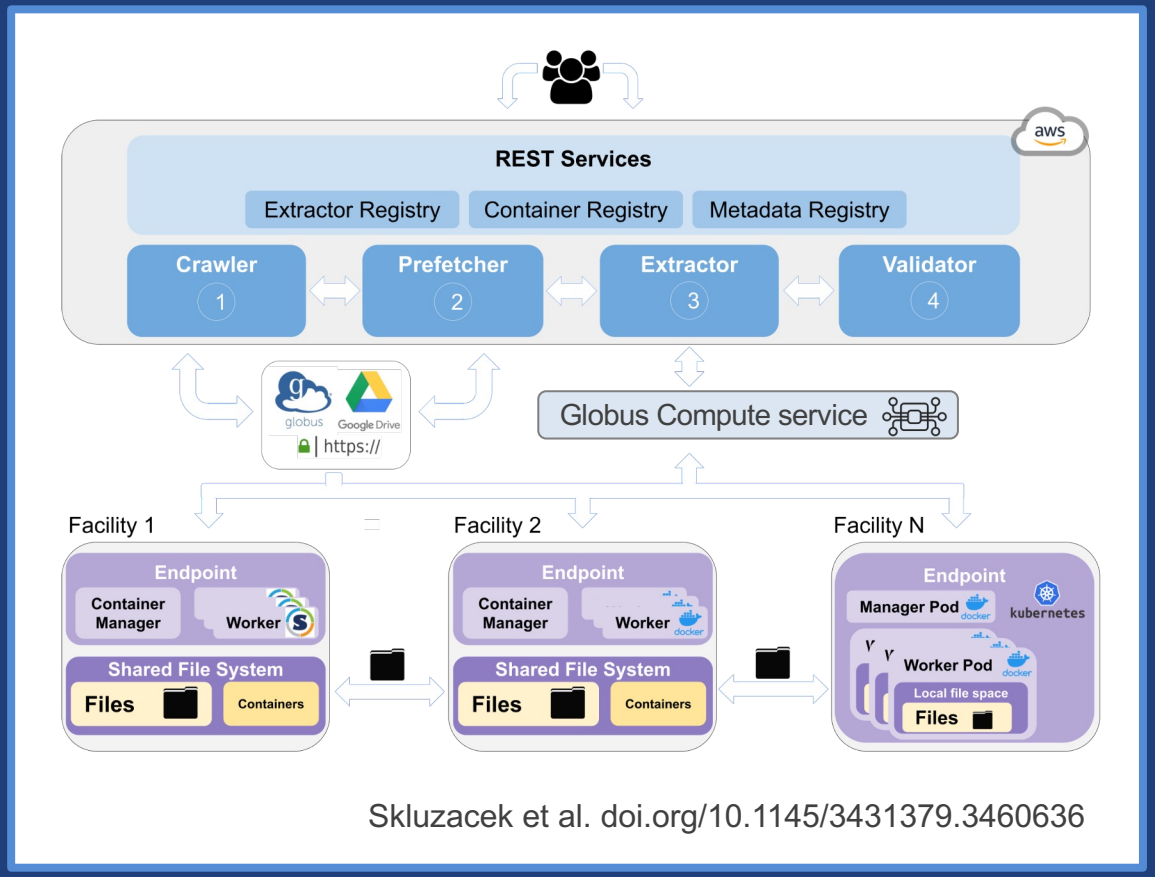




Feickert et al., arXiv:2103.02182

Xtract: Bulk metadata extraction

- **Automated, scalable system for bulk metadata extraction from large, distributed research data repositories**
- **Orchestrates application of metadata extractors to groups of files**
- **Uses Globus Compute to dispatch extractors to data**



Skluzacek et al. doi.org/10.1145/3431379.3460636

Globus Compute as a research computing platform

1. Compute

Researcher needs to run a computation on a remote PC, cloud, supercomputer

2. Specialize

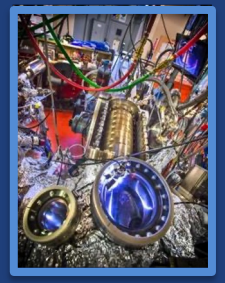
Researcher needs to move it to a new system or architecture to improve performance

Compute Facility

3. Share

Collaborator wants to run their colleague's computation on another system closer to their data

Instrument



4. Community Access

Collaborators want to share access to a single allocation to run compute tasks

5. Build

Gateway and application developers want to add remote computation to their code

Seems too easy? Let's take a look…

- **Run function on my laptop …test, debug**
- **Scale (a bit) on a cloud VM (or K8s cluster)**
- **Scale (more) on my campus cluster**
- **…run complete model on DOE supercomputer :--(**

Globus Compute current state

- **Service is running in production**
  - Tried and tested by 10MMs of invocations on 100Ks endpoints
- **Globus Computer Agent supports single user**
  - Akin to Globus Connect Personal
- **Endpoints visible via Globus web app**
- **Function registration and invocation via SDK**

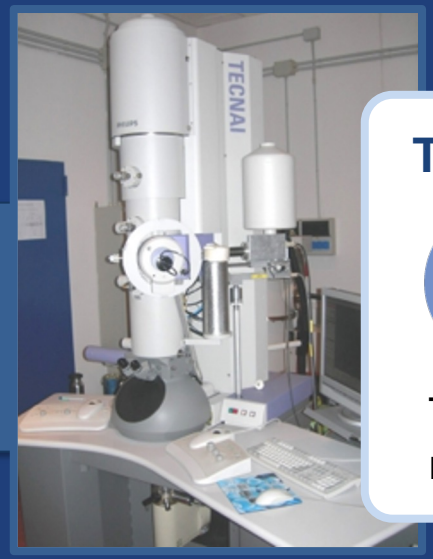
Compute service will evolve rapidly

- **Multi-user compute endpoints**
  - Akin to Globus Connect Server
- **Native integration with transfer for stage in and stage out of data for compute tasks**
  - Can be done today via Globus Flows
- **Expanding compute service interfaces in the webapp for administrators and users**

With great power comes great capability

Automating cryoEM flows

Globus Flows



Transfer

Transfer raw files

Compute

Launch analysis job

Carbon!

Correct, classify, …

cryoSPARC

RELION

cryoDRGN

Compute

Extract metadata

Search

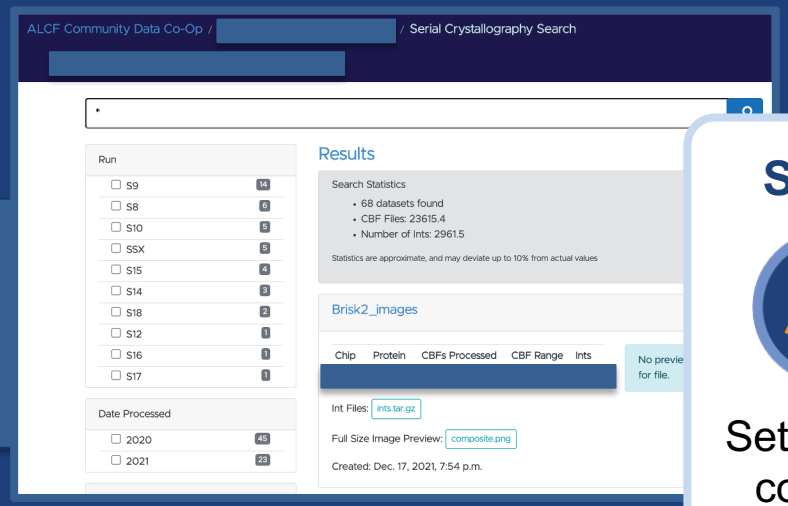
Ingest to index

Transfer

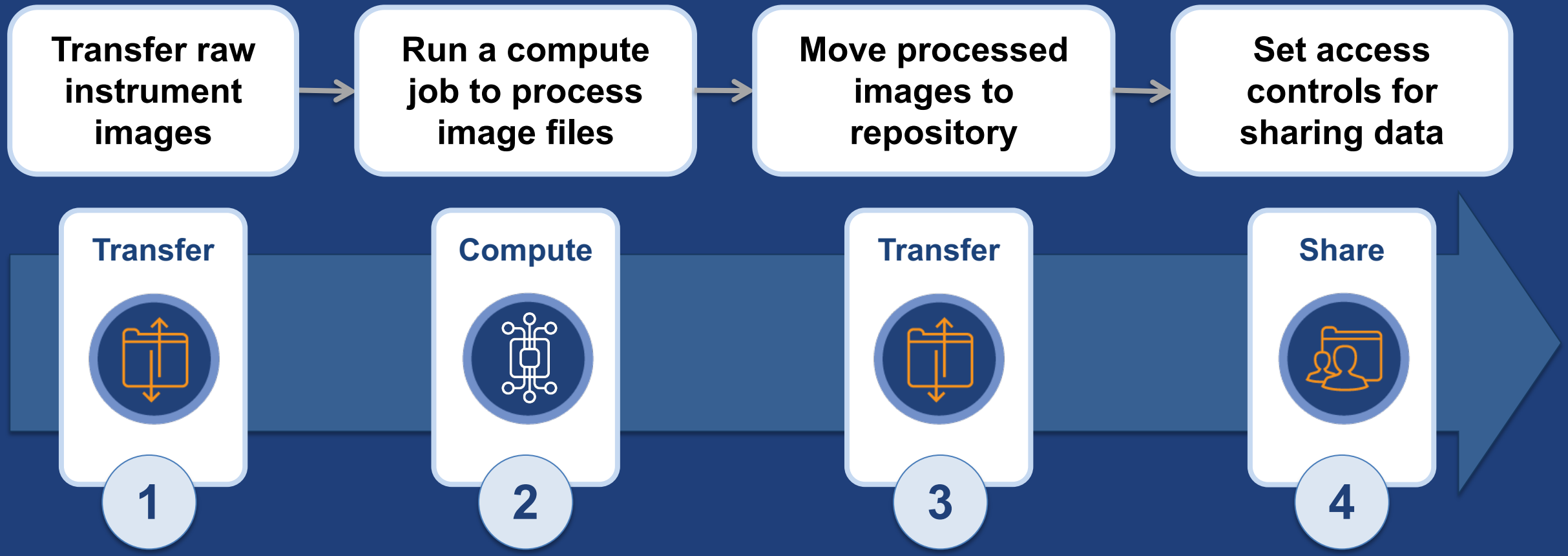
Move final files to repo

Share

Set access controls

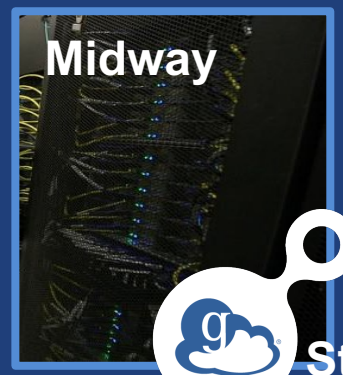


Common instrument flow with computation



Our “instrument” setup

Sharing Resource



access result files

Storage Endpoint

Computing Resource

EC2 Instance

Compute Endpoint

```
def process_images(input_path=None, result_path=None):

    import os
    import glob
    from PIL import Image

    files = (file for file in glob.glob(os.path.join(input_path,'*')) \
        if os.path.isfile(os.path.join(input_path, file)))

    if not os.path.exists(result_path):
        os.makedirs(result_path)

    for file in files:
        image = Image.open(file)

        # Generate thumbnail
        image.thumbnail((200, 200))

        # Save thumbnail image
        image.save(f"{result_path}/t
```

Registered Compute Function

Compute Service

globus

invoke image processing function

2

set permissions

4

transfer result files

3

Storage Endpoint

Instrument

EC2 Instance

Monitor script

0

trigger flow run

transfer raw files

1

Transfer Service

globus

transfer control

Storage Endpoint

Thank you, funders...

U.S. DEPARTMENT OF
ENERGY

THE UNIVERSITY OF
CHICAGO

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

Argonne
NATIONAL LABORATORY

powered by
amazon
web services