# Cyberinfrastructure for Autonomous Science

Ryan Chard

Data Science and Learning

09 December 2019

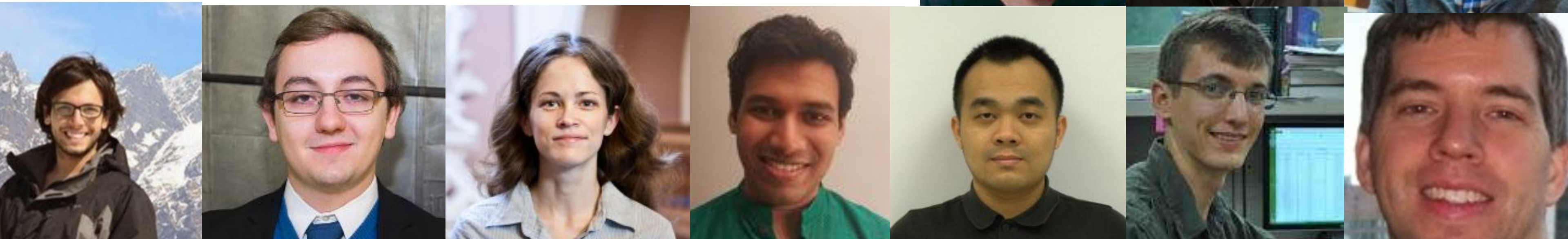
Argonne NATIONAL LABORATORY

globus labs

# Outline

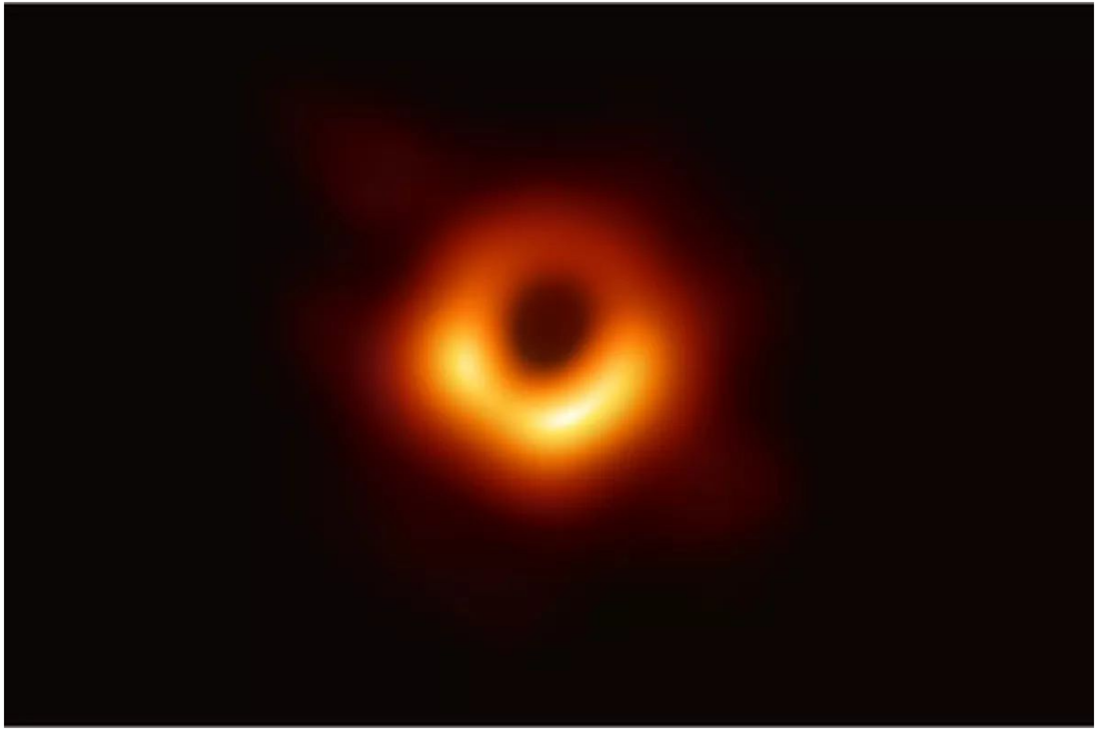


- Scientific data management challenges
- Automation goals
- Cyberinfrastructure for automation
- Simplifying remote execution
- ML in the loop



# Data management challenges as volumes increase





Event Horizon Telescope

- 12 telescopes
- Telescopes generate roughly 900TB per 5 day run
- Data written to ~1000 HDDs
- Transported to MIT & Max Planck via airplane
- Aggregated and analyzed

Global resources, long timescales

Too much data for manual processing

Data loss due to HDD failure

# Data management challenges as volumes increase

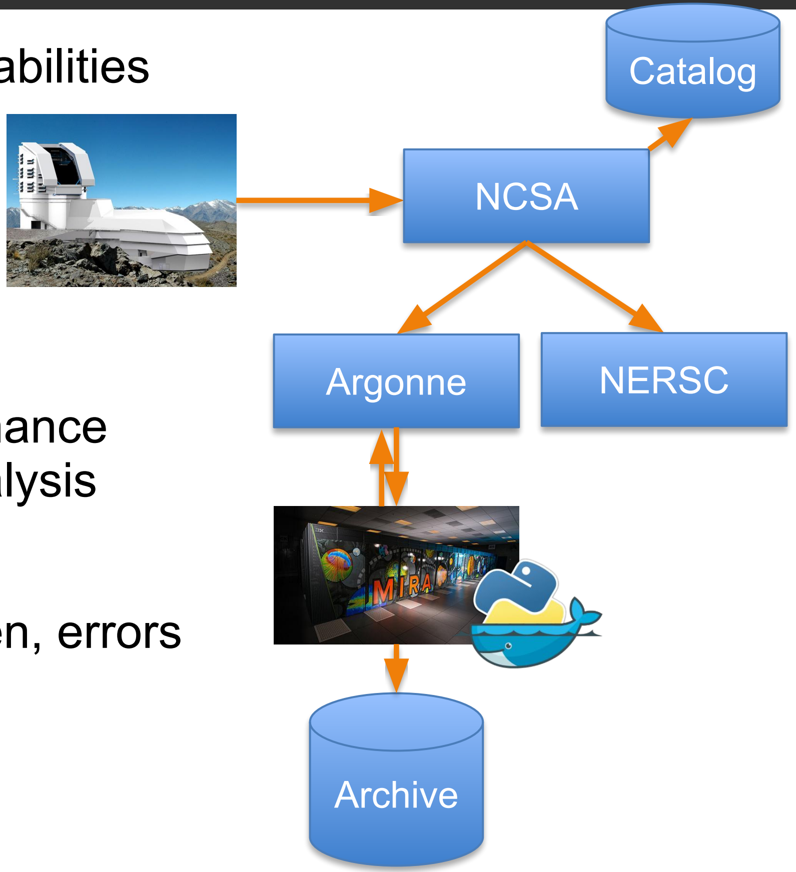


Scientific data are overwhelming finite human capabilities

Scientific results are dependent on

- Data acquired at various locations/times
- Analysis processes executed on distributed resources
- Catalogs of descriptive metadata and provenance
- Dynamic collaborations around data and analysis

Best practices are overlooked, useful data forgotten, errors propagate through pipelines, …




LSST data distribution and analysis pipeline

# Experimental Science



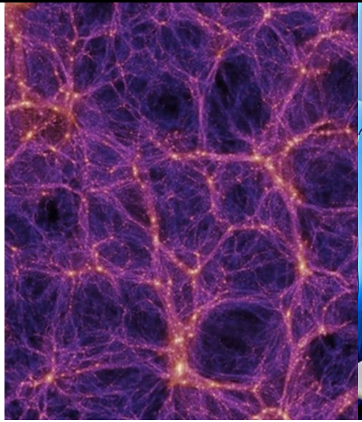
Data issues are particularly evident in large scale experimental science

Researchers are allocated short periods of instrument time

- Must maximize experiment efficiency and output data quality/accuracy
- Analysis requirements exceed local resource capabilities

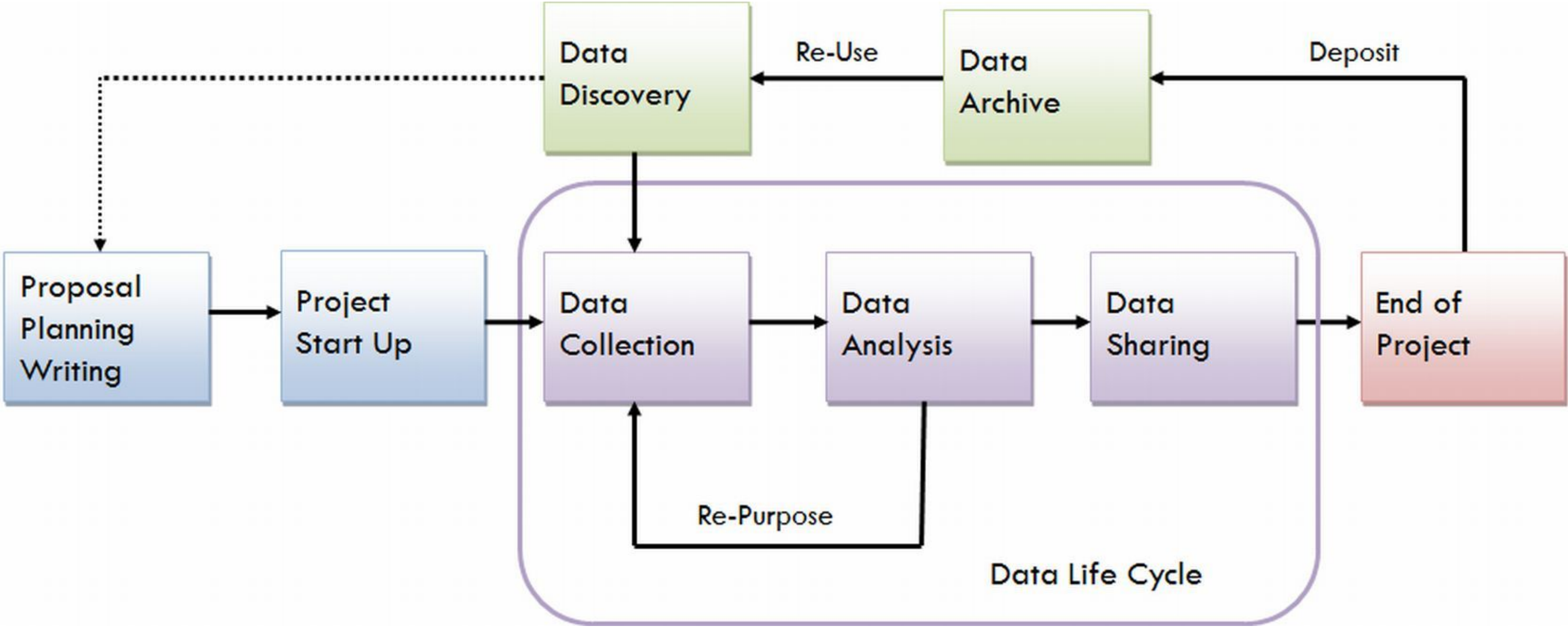
Inefficiencies mean less science is performed

Automating data lifecycle is key

# Scientific Data Lifecycle





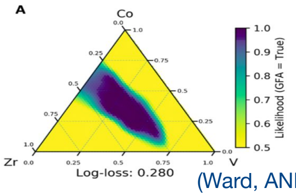

https://data.library.virginia.edu/data-management/lifecycle/

# Automation Use Cases

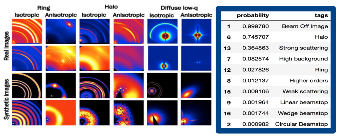
## Best practices


## Publication & archiving

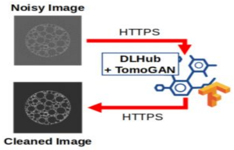

## Model-in-the-Loop




(Ward, ANL/UC)

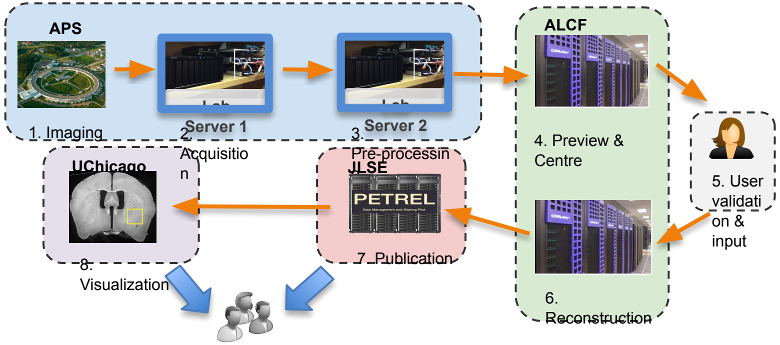



XRD image tagging (Yager, BNL)




TomoGAN (Liu Argonne)

## Analysis

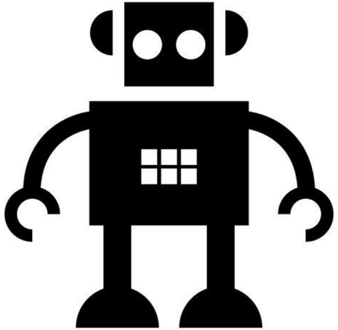


# 1 Orchestrating End-to-End Automations

# Requirements



- Data driven: automation in response to data events
- Accessible to any scientist
- Scalable, secure, and fault tolerant
- Applicable to laptops and supercomputers
- Reliable end-to-end pipelines
- Supports human-in-the-loop timescales



# Globus Automate

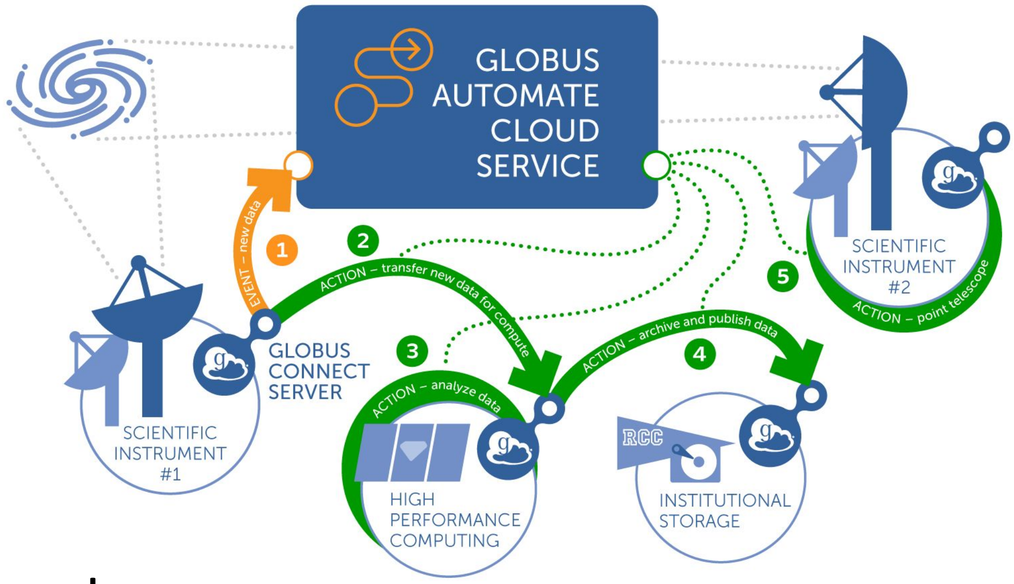


A distributed automation PaaS

Compose pipelines that *step* between Web services

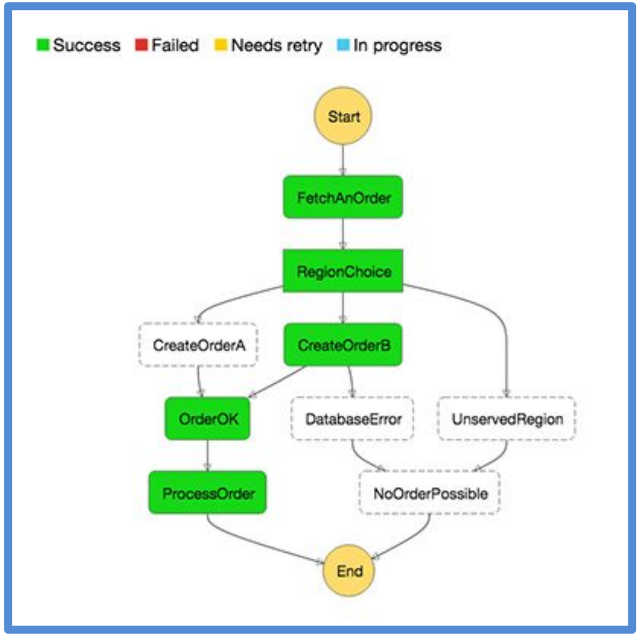
Supports user-defined actions

Integrates Globus Auth at every step



# Globus automation capabilities

- Built on AWS Step Functions
  - Simple JSON-based state machine language
  - Conditions, loops, fault tolerance, etc.
  - Propagates state through the flow
- Standardized API for integrating custom event and action services
  - Actions: synchronous or asynchronous
  - Custom Web forms prompt for user input
- Actions secured with Globus Auth



# Automate Actions

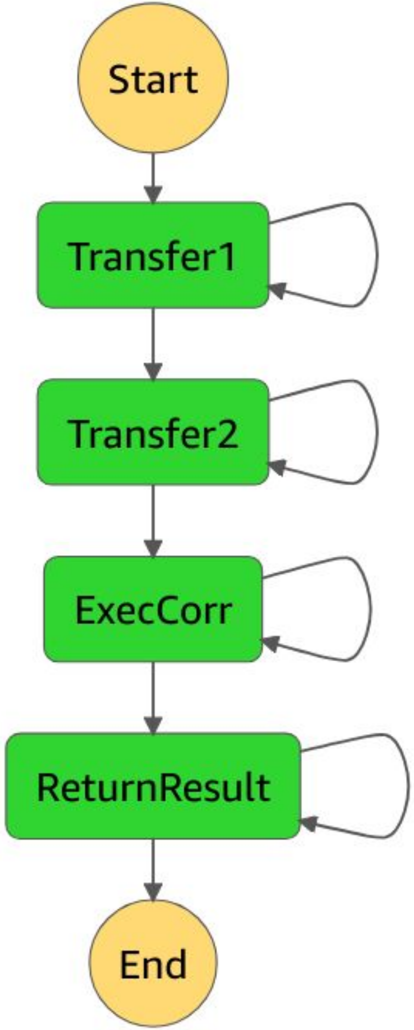
Any Web service can expose the Action API

- /automate/v1/<action>/run, status, cancel, introspect,
- .../status used to enable polling

Framework to create actions

Actions are asynchronous, timeout after 1 year

- Supports human-in-the-loop tasks



# Automate Flows



Definition extends SFN State Machine Language

Definitions are transformed into flows, then into state machines

- Adds auth, async, etc.
- Returns uuid

Accepts JSON blob as input

```
flow_definition = {
  "Comment": "Two step transfer",
  "StartAt": "Transfer1",
  "States": {
    "Transfer1": {
      "Comment": "Initial Transfer from Cam
      "Type": "Action",
      "ActionUrl": "https://actions.automat
      "ActionScope": "https://auth.globus.c
      "InputPath": "$.Transfer1Input",
      "ResultPath": "$.Transfer1Result",
      "Next": "Transfer2"
    },
    "Transfer2": {
      "Comment": "Transfer from DMZ to data
      "Type": "Action",
      "ActionUrl": "https://actions.automat
      "ActionScope": "https://auth.globus.c
      "InputPath": "$.Transfer2Input",
      "ResultPath": "$.Transfer2Result",
      "End": True
    }
  }
}
```

# Running Flows



```
flow_input = {
  "Transfer1Input": {
    "source_endpoint_id": "go#ep1",
    "destination_endpoint_id": "go#ep2",
    "transfer_items": [
      {
        "source_path": "/~/globus_publish.auto",
        "destination_path": "/~/globus_publish_2.auto",
        "recursive": False
      }
    ]
  },
  "Transfer2Input": {
    "source_endpoint_id": "go#ep2",
    "destination_endpoint_id": "go#ep1",
    "transfer_items": [
      {
        "source_path": "/~/globus_publish_2.auto",
        "destination_path": "/~/globus_publish_3.auto",
        "recursive": False
      }
    ]
  }
}
```

Authenticate

Specify input: $.Transfer1Input

fc.run_flow(flow_id, flow_scope, flow_input)

Monitor

# Automate Actions

### Auth
- User login
- Secure service interactions
- App identity and interactions

### Identifier
- Manage namespace
- Mint DOI

### Search
- Catalog
- Faceted search
- Search query

### Transfer
- File operations
- Transfer data
- Set permission

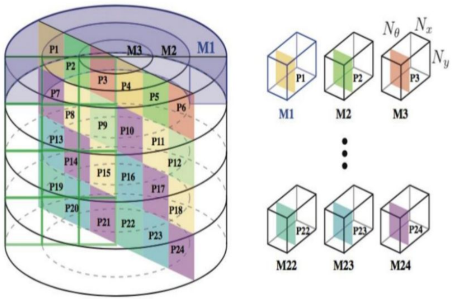
### funcX
- Remote execution
- Secure connections
- Isolated sandboxes

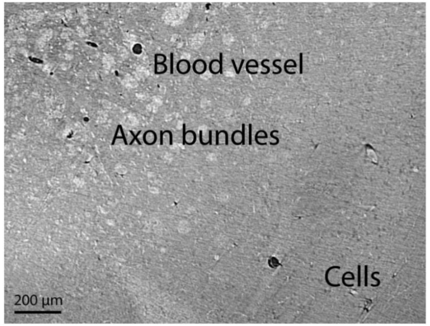
# End-to-end Analysis Pipeline

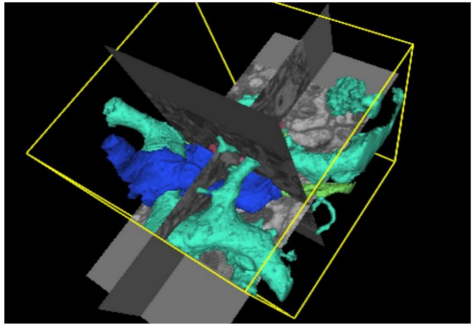


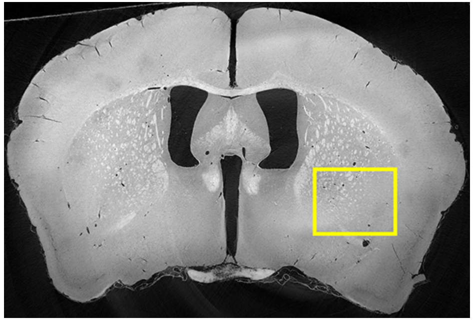
**UChicago’s Kasthuri Lab study brain aging and disease**

- Construct connectomes -- mapping of neuron connections
- Use synchrotron (APS) to rapidly image brains (and other things)
- Beam time once every few months
- Generate segmented datasets/visualizations for the community
- ~20GB/minute for large (cm) unsectioned brains
- Perform semi-standard reconstruction on all data across HPC resources



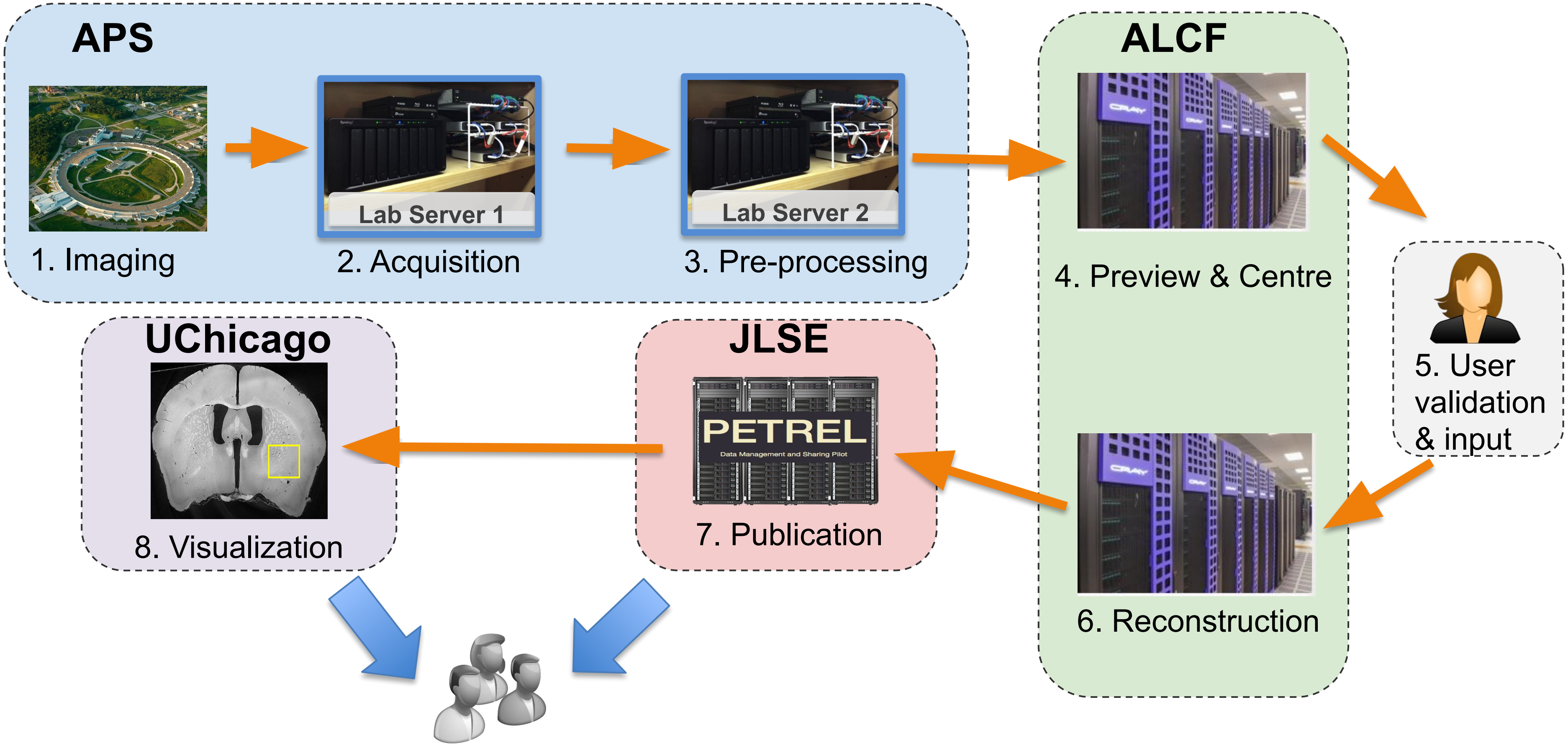





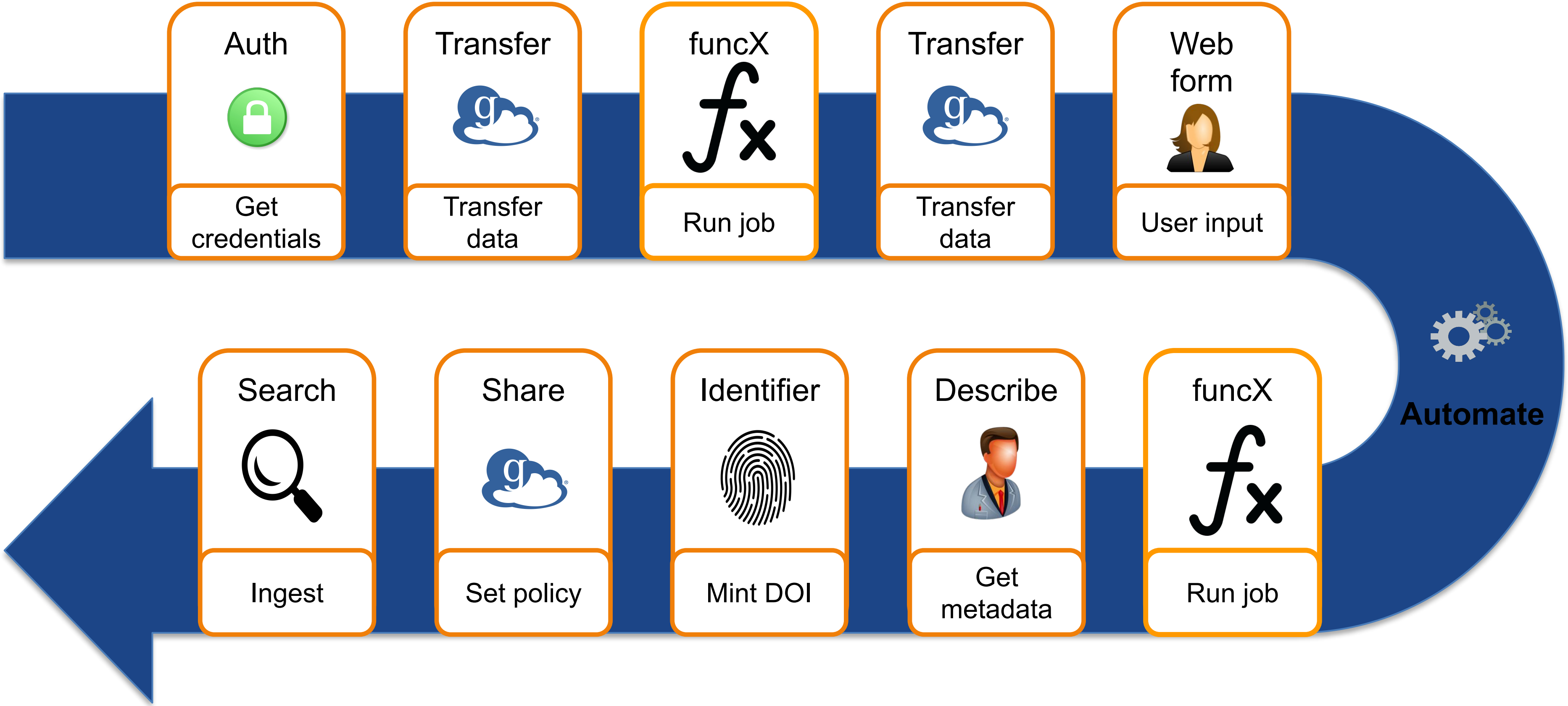
# Neuroanatomy automation





# Neuroanatomy automation





# Experimental Control



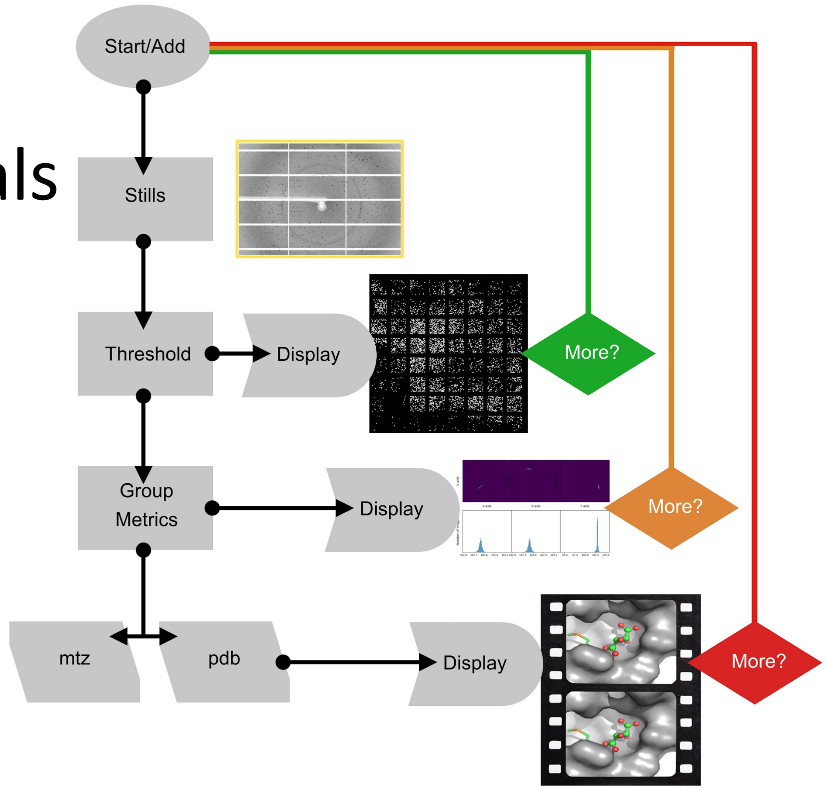
Serial crystallography

- Serially image 26000 crystals

Quality control first 1000

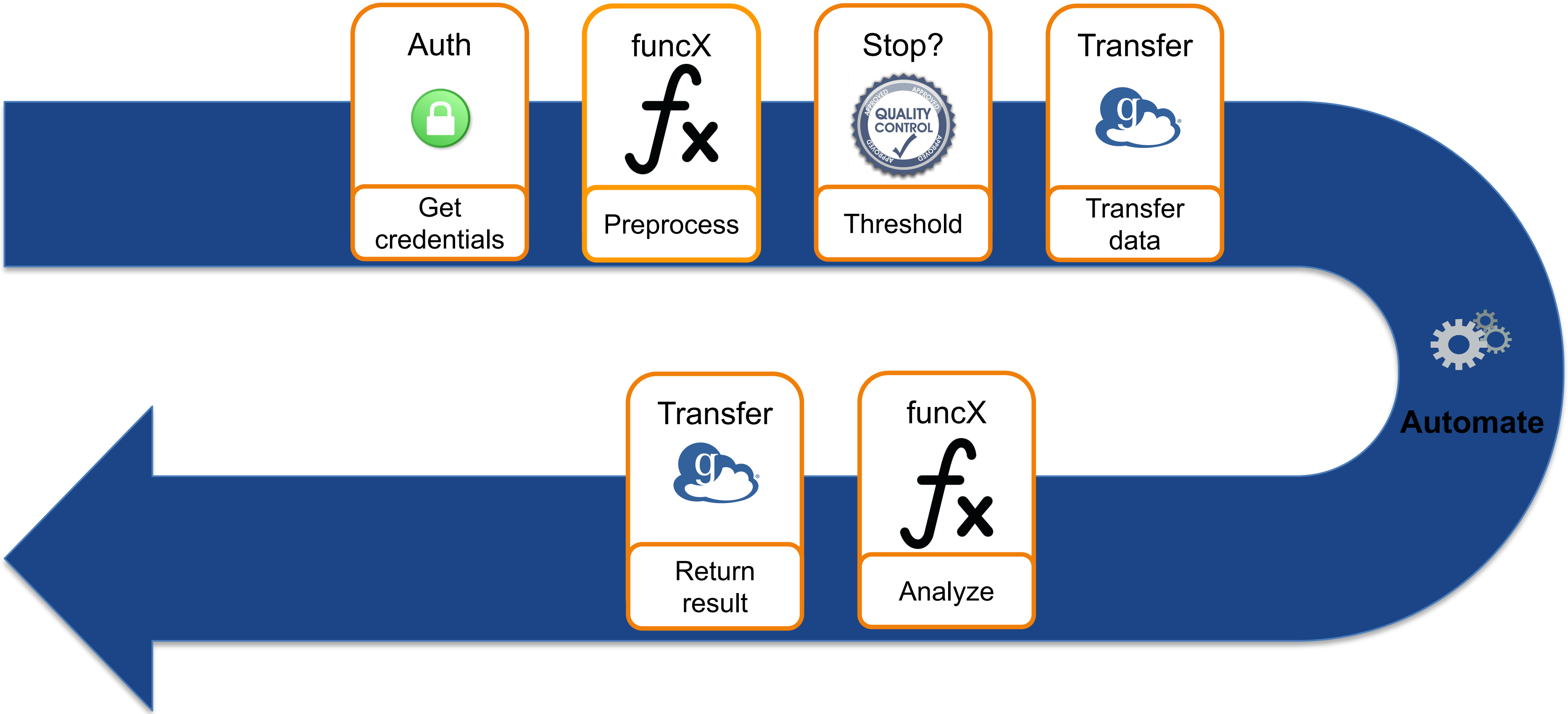
Analyze full 26,000

Return crystal structure to scientist



# SSX Automation





jupyter

# 2 Compute Abstractions: From Laptops to Supercomputers

# Laptops to Supercomputers



Automation tasks need to be applied to laptops, desktops, local clusters, cloud, and supercomputers

Heterogeneous resources, permissions, environments, ...

Need a common way to interact with arbitrary resources and perform tasks

# Serverless computing



Serverless computing is revolutionizing enterprise IT

Function as a Service (FaaS)

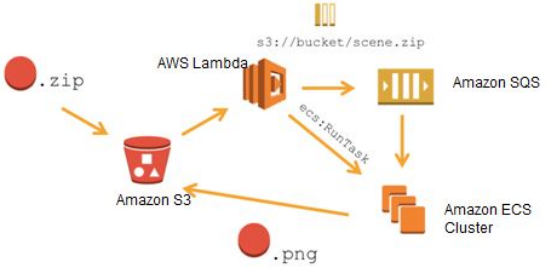
- Pick a runtime (python/JS/R etc.)
- Write function code
- Run at any scale

Low latency, on-demand

Abstracts computing infrastructure

Google Cloud Functions

APACHE OpenWhisk™



# funcX: Serverless Supercomputing

Turn any machine into a function serving platform

Remove barriers to HPC

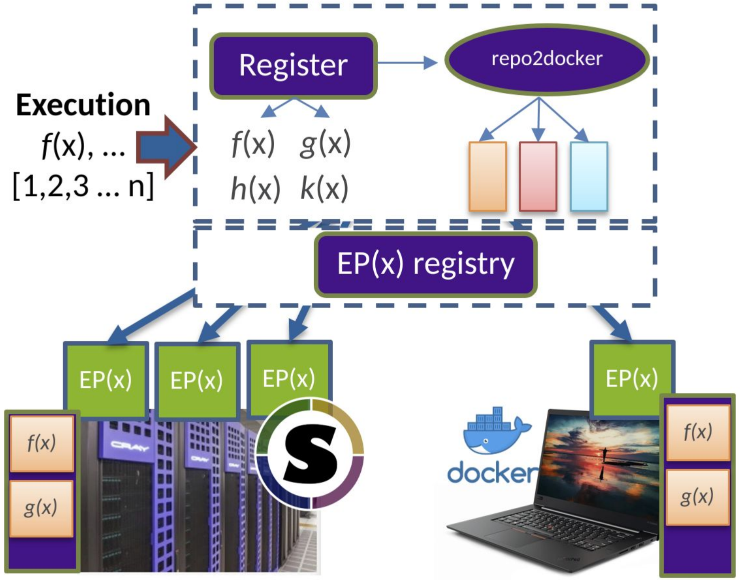
Backfill queues enable elastic HPC

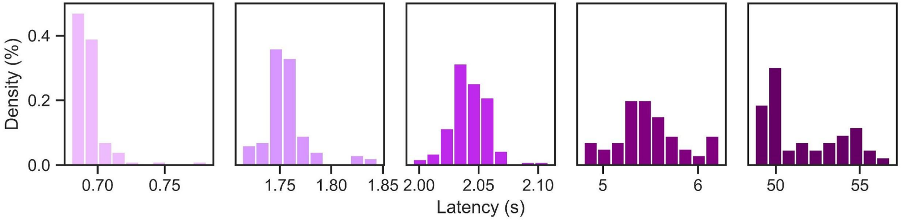
**Endpoints:**

- Lightweight agent with Parsl config
- Abstracts underlying resource

**Functions:**

- Register once, run anywhere
- Container encapsulation
- Globus Auth




(a) tabular file extraction | (b) MNIST digit prediction | (c) DIALS stills process | (d) tomographic preview | (e) correlation spectroscopy

# funcX

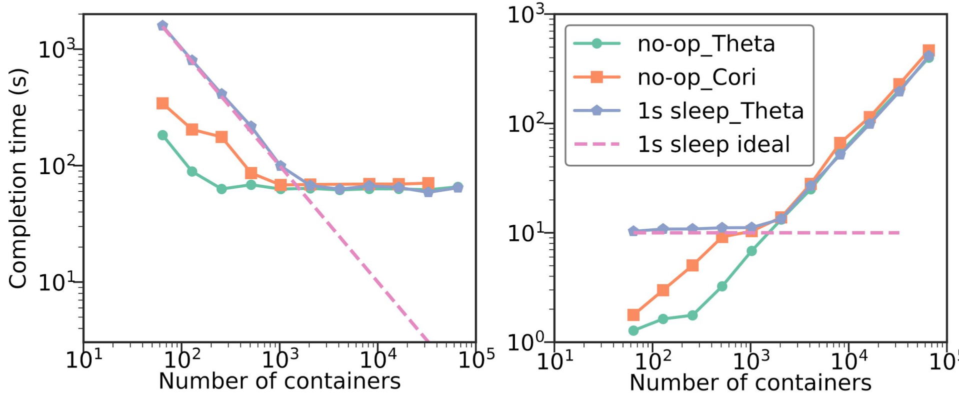
```
config = Config(
    executors=[
        HighThroughputExecutor(
            label="stampede2_htex",
            address=address_by_hostname(),
            provider=SlurmProvider(
                channel=LocalChannel(),
                nodes_per_block=128,
                init_blocks=1,
                partition="skx-normal",
                walltime="12:00:00"
            )
        )
    ]
)
```

Parsl's resource configuration abstracts compute resource

Portable containers encapsulate dependencies

Run >65,000 containers on Theta and Cori

funcX simplifies using HPC



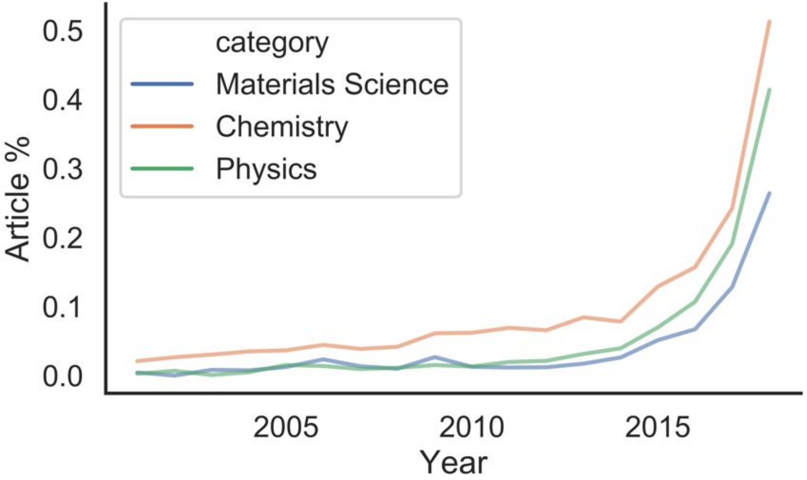

(a) strong scaling

(b) weak scaling

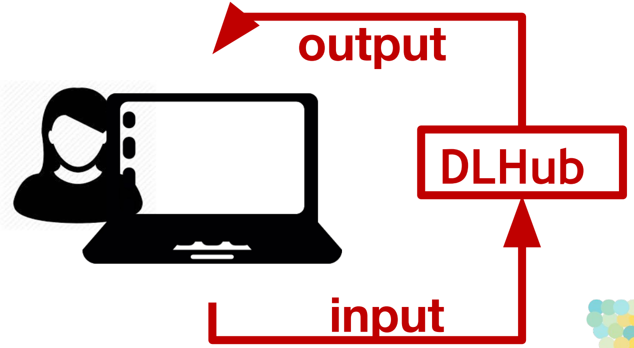
jupyter

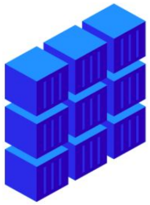
# ML in the Loop

# Scientific ML




- Where are the model and trained weights?
- How do I run the model on my data?
- Should I run the model on my data?
- How do I share my model with the community?
- How can I build on this work?
- How do I cite it?



Model / transform containers

ARGONNE LEADERSHIP COMPUTING FACILITY

parsl

MATERIALS DATA FACILITY

PETREL

# Data and Learning Hub for Science (DLHub)


- Fits into the MDF/Globus ecosystem
- Collect, publish, categorize models from many disciplines (materials science, physics, chemistry, genomics, etc.)
- Serve model inference on-demand via API to simplify sharing, consumption, and access
- Enable new science through reuse, real-time model-in-the-loop integration, and synthesis & ensembling of existing models

# Using DLHub is Easy!

**Python SDK**

```
$ pip install dlhub_sdk
```

**Command Line Interface**

```
$ pip install dlhub_cli
```

1

**Describe**

```
model = KerasModel.create_model("p1b1-example.h5")

model.set_title("CANDLE Pilot 1 - Benchmark 1")
model.set_name("candle_p1b1")
model.set_domains(["genomics","biology","HPC"])
```

2

**Publish**

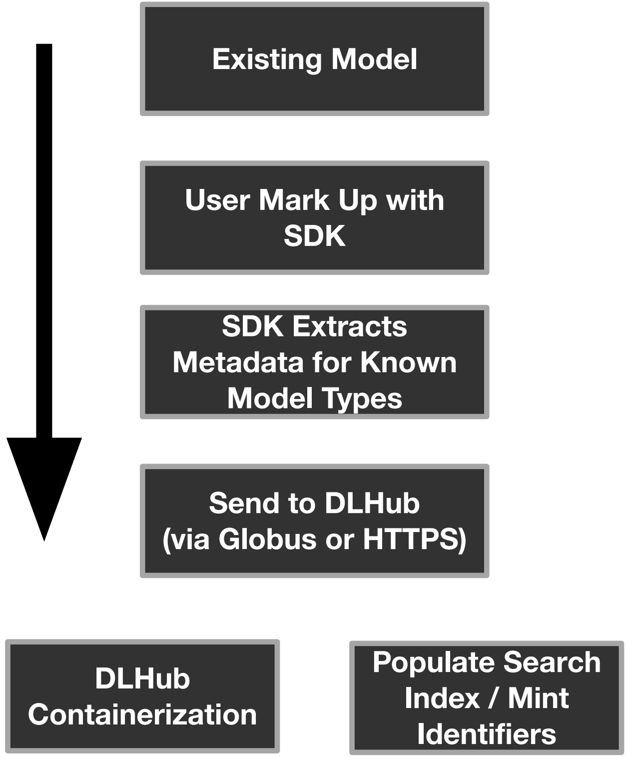
```
from dlhub_sdk.client import DLHubClient

dl = DLHubClient.login()
dl.publish_servable(model)
```

3

**Run**

```
data = np.load(pilot1.npy)
pred = dl.run("blaiszik_uchicago/candle_p1b1", data,
              type="python")
```

# Marking up a Model – Python SDK



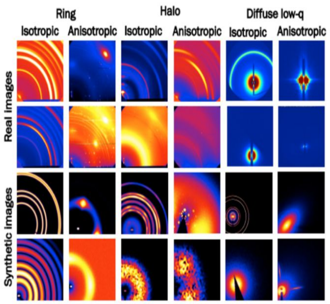

```python
from dlhub_toolbox.models.servables.keras import KerasModel
import pickle as pkl
import json

# Describe the keras model
model = KerasModel('model.hd5', list(map(str, range(10))))

# Describe the model
model.set_title("MNIST Digit Classifier")
model.set_name("mnist_tiny_example")
model.set_domains(["general","digit recognition"])

# Add link to paper describing the dataset
model.add_related_identifier("10.1109/CVPR.2007.383157", "DOI",
                             "IsDescribedBy")

model.set_authors(["Lecunn, Yann", "Cortes, Corinna"])

# Describe the outputs in more detail
model.output['description'] = 'Probabilities of being 0-9'
model.input['description'] = 'Image of a digit'
```

# Model-in-the-Loop

Enable automated use of models in analysis pipelines

Accessible as Automate Action

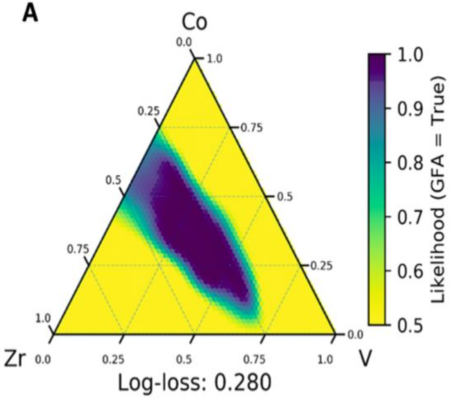
## Tag beamline images




| | probability | tags |
|---|---|---|
| 1 | 0.999780 | Beam Off Image |
| 6 | 0.745707 | Halo |
| 13 | 0.364863 | Strong scattering |
| 7 | 0.082574 | High background |
| 12 | 0.027826 | Ring |
| 8 | 0.012137 | Higher orders |
| 15 | 0.008106 | Weak scattering |
| 9 | 0.001964 | Linear beamstop |
| 16 | 0.001744 | Wedge beamstop |
| 2 | 0.000982 | Circular Beamstop |

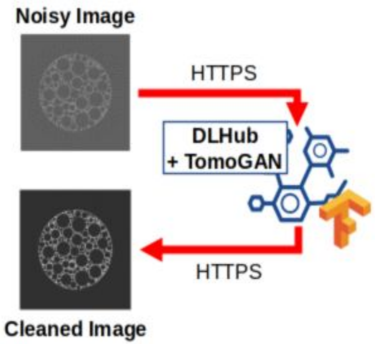
XRD image tagging (Yager, BNL)

## Guide experiments




(Ward, ANL/UC)

## Improve quality




TomoGAN (Liu Argonne)

# Thanks!



Questions?

rchard@anl.gov