High-Throughput Neuroanatomy and Trigger-Action Programming: A Case Study in Research Automation

Ryan Chard, Rafael Vescovi, Ming Du, Hanyu Li, Kyle Chard, Steve Tuecke, Narayanan Kasthuri, Ian Foster

globus labs

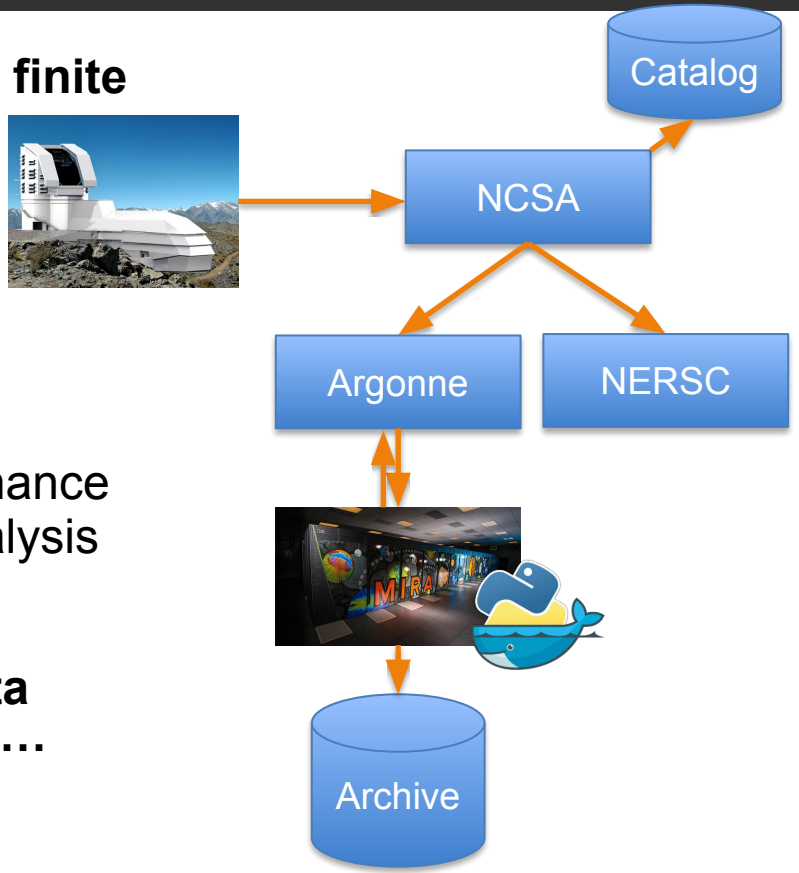
Data management challenges as volumes increase

Data volumes and velocities 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 often overlooked, useful data forgotten, errors propagate through pipelines, …



LSST data distribution and analysis pipeline

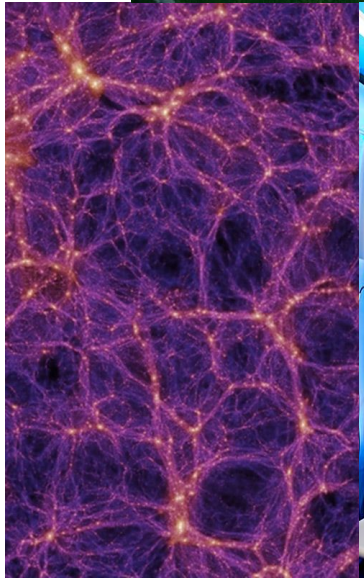
Experimental Science

Data management issues are particularly evident in large scale experimental science

Researchers are allocated short periods of instrument time

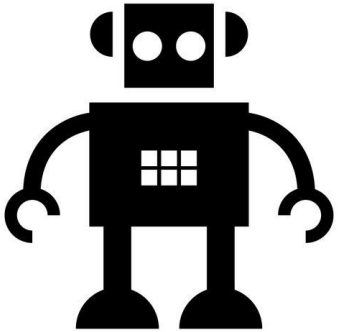
- Must maximize experiment efficiency and output data quality/accuracy

Inefficiencies mean less science is performed and researchers may have to wait months for another chance.



Solution: Automation

Goal: Automate data manipulation tasks from transfer and sharing to acquisition, publication, indexing, analysis, and inference



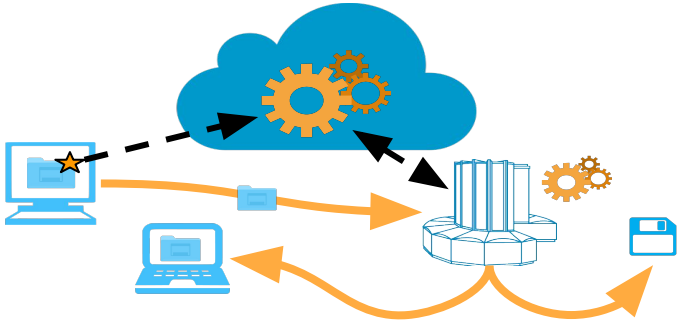
Requirements: A platform that...

- Can automate best practices
- Is data driven -- responds to data events
- Can be applied across arbitrary storage and compute infrastr
- Can be dynamically programmed to respond to new events
- Enable non-expert users to define automations

Approach: Ripple

Ripple: Software Defined Cyberinfrastructure

A Trigger-Action platform for data



- Users define Trigger-Action rules
- Ripple engine processes data triggers and reliably executes actions in response
- Can daisy-chain multiple Trigger-Action rules into complex flows

Ripple

Service

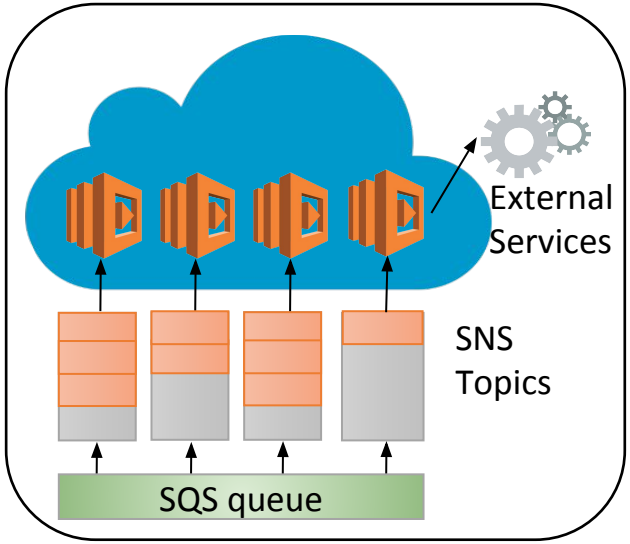
- Process events and orchestrate execution of actions
- Reliably manage event and execution lifecycle
- Perform cloud-based actions (Globus, email, ECS) and remote execution

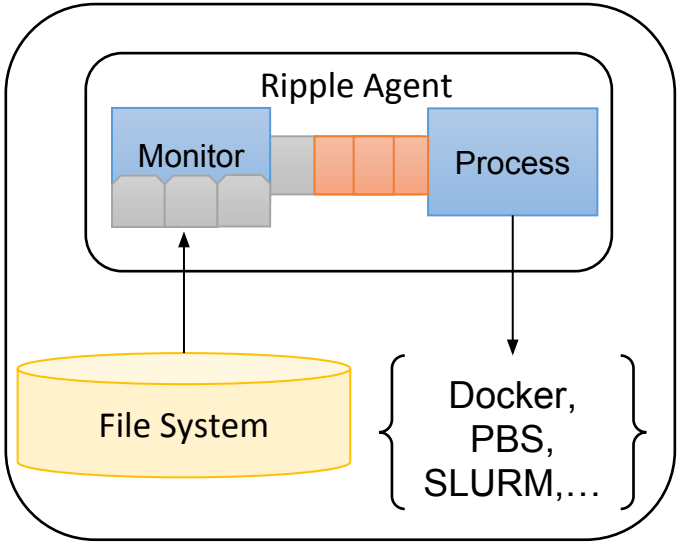
Agents

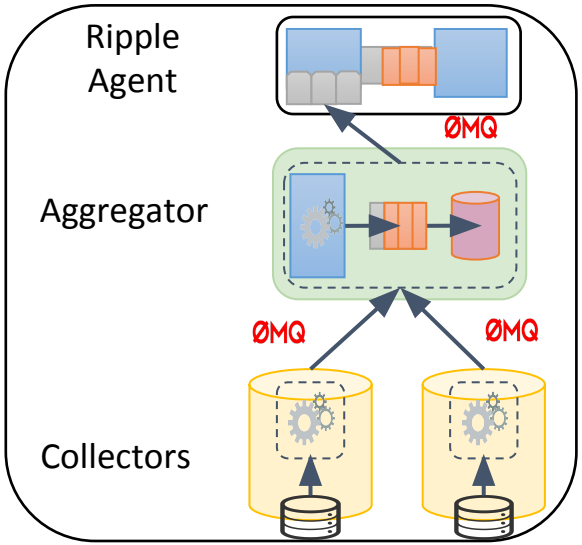
- Deployed locally to monitor file system events
- Detects, evaluates, and reports events of interest to the cloud service
- Performs actions on a user's behalf

Events

- File create, modify, delete, close
- Supports multiple event notification tools (e.g., inotify, kqueue, etc.) and services (e.g. Globus)
- Scalable monitoring solution for large parallel Lustre stores



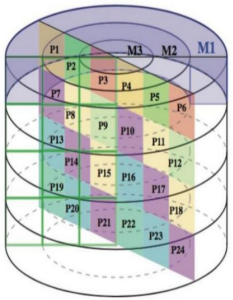


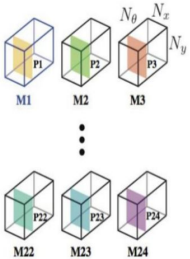


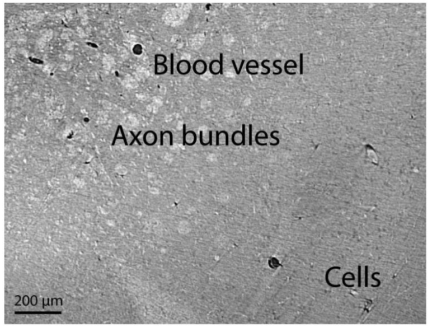
A Neuroanatomy Use Case

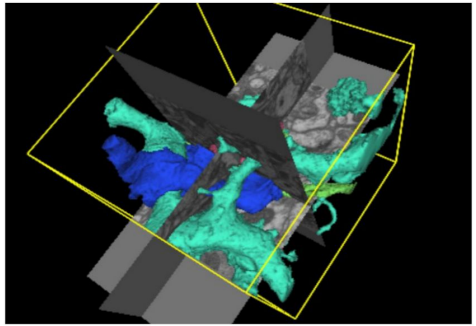
UChicago's Kasthuri Lab study brain aging and disease

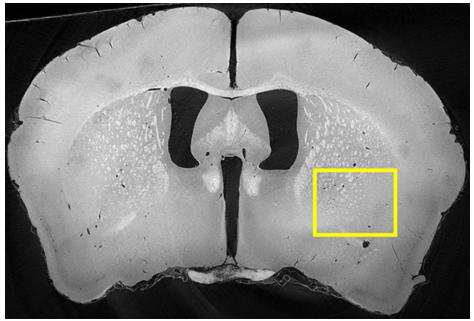
- Construct connectomes -- mapping of neuron connections
- Use synchrotron (APS) to rapidly image brains (and other things)
- Given beam time once every few months
- Generate segmented datasets/visualizations for the community
- ~20GB/minute for large (cm) unsectioned brains
- Perform semi-standard reconstruction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

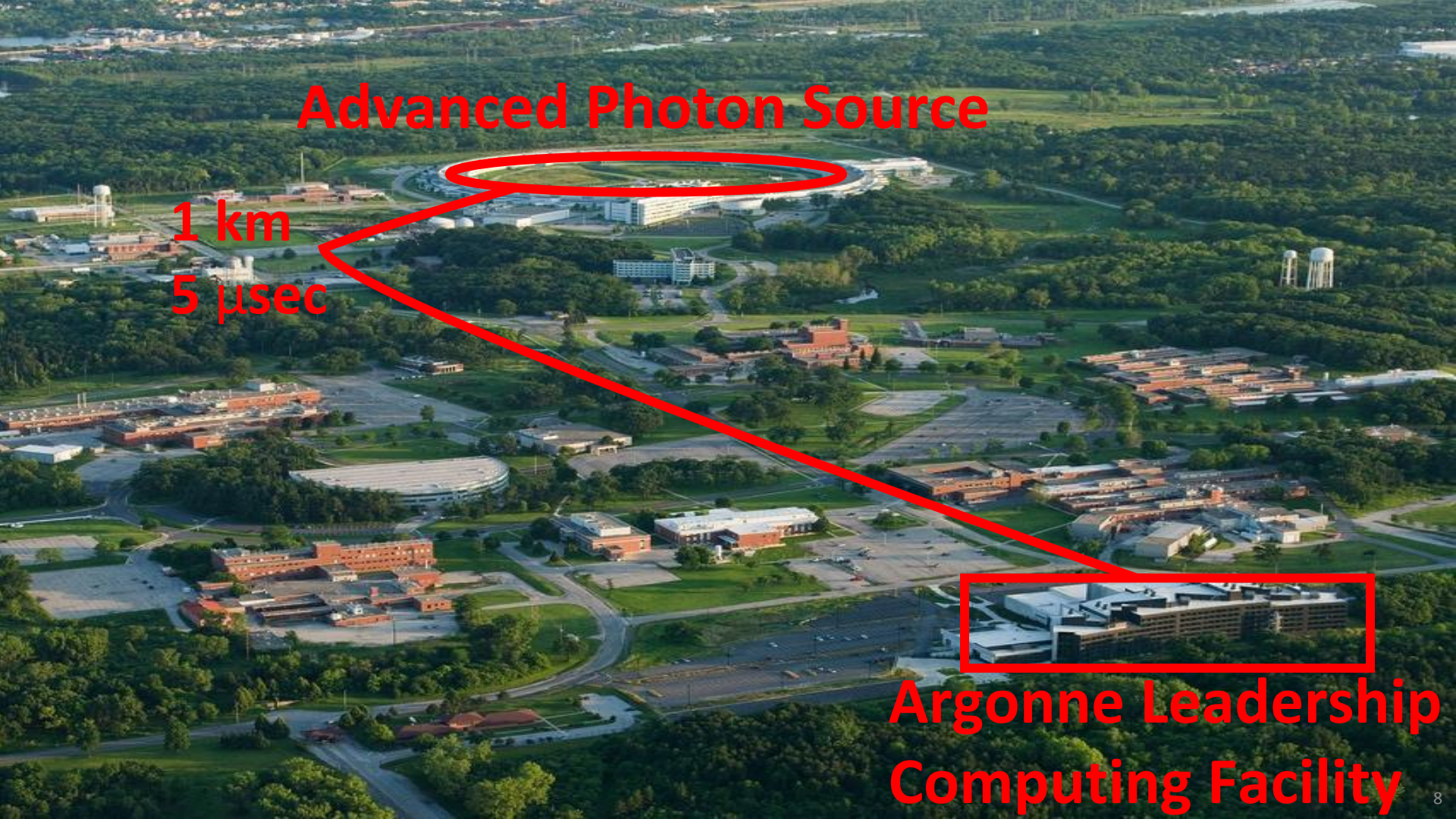




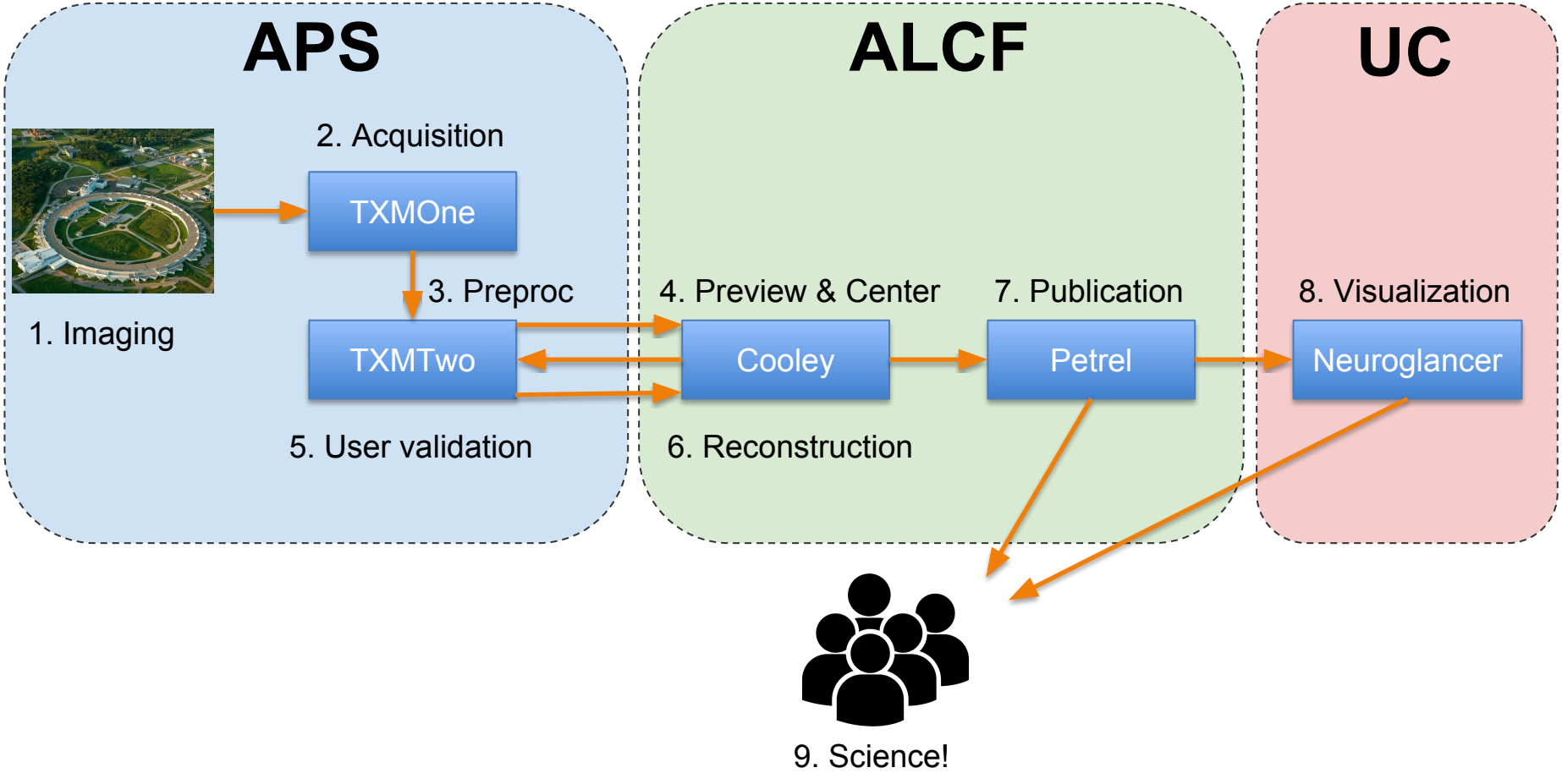







Advanced Photon Source
1 km
5 μsec
Argonne Leadership Computing Facility

Neuroanatomy Reconstruction Pipeline



Neuroanatomy Research with Ripple

Data Source

Collector Storage and Compute
- Capture dataset creation
- Review center position

ALCF Cooley Cluster
- Generate preview and center images
- Reconstruct image
- Extract metadata

ALCF Petrel Archive

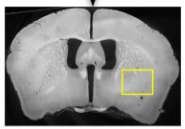
APS beamline 32-ID

1

2

Ingest in Globus Search
Set sharing ACLs

Data publication



Visualize with Neuroglancer

Rules

1
- IF new HDF5 files THEN transfer to Cooley
- IF new center_pos THEN initiate reconstruction

2
- IF transfer complete THEN execute preview and center finding
- IF results THEN return data to APS
- IF reconstruction THEN transfer data to Petrel AND publish dataset

Full Set of Ripple Rules

Step 1: Transfer to ALCF
Trigger: *FileCreated*, `.*.h5`
Action: *GlobusTransfer*, `$FILE`, APS→ALCF

Step 2: Organize HDF5
Trigger: *TransferComplete*, `.h5`, APS→ALCF, User:Brainimaging
Action: *Bash*, `bash automo_organize.sh`

Step 3: Create batch file
Trigger: *FileCreated*, `data.h5`
Action: *Bash*, `echo ... >> $PATH/batch_job.qsub`

Step 4: Submit batch file
Trigger: *FileCreated*, `batch_job.qsub`
Action: *Cobalt*, `qsub $FILE`

Step 5: Return center
Trigger: *FileCreated*, `center_pos.txt`
Action: *GlobusTransfer*, `$PATH`, ALCF→APS

Step 6: Transfer verified center
Trigger: *FileCreated*, `real_center_pos.txt`
Action: *GlobusTransfer*, `$FILE`, APS→ALCF

Step 7: Create batch reconstruction file
Trigger: *TransferComplete*, `real_center_pos.txt`, APS→ALCF
Action: *Bash*, `echo ... >> $PATH/recon_job.qsub`

Step 8: Submit batch reconstruction file
Trigger: *FileCreated*, `recon_job.qsub`
Action: *Cobalt*, `qsub $FILE`

Step 9: Publish results
Trigger: *FileCreated*, `recon_0000.tiff`
Action: *GlobusTransfer*, `$PATH`, ALCF→Petrel

Step 10: Catalog results
Trigger: *TransferComplete*, `recon_0000.tiff`, ALCF→Petrel
Action: *Bash*, `python catalog_data.py $PATH`

TAP Neuroanatomy

It works! http://tomofish.kasthurilab.com

Increased throughput, improved productivity, automated analysis + best practices (publication/replication/ACL/catalog)

Happy scientists!

TAP Neuroanatomy

However...

Reflecting on Ripple Automation

Ripple is designed for tigger-action pairs

- Almost all our use cases rely on multi-step flows

Workflows comprised of cascading rules

- Can be unreliable (will retry, but success may not raise required event)
- Easy to debug with 10s of executions, difficult with 1000s

Existing pipelines don't always map well to TAP

- Force specific outputs to trigger next rule

TAP == easy, correct TAP == hard

- Unforeseen consequences (Ur et al.). Misunderstood file triggers (create vs mod)
- Exaggerated with multi-step flows

Can't easily contribute actions to Ripple

- Difficult to add custom triggers & services e.g., human-in-the-loop

Additional Requirements

What else we need from an automation platform:

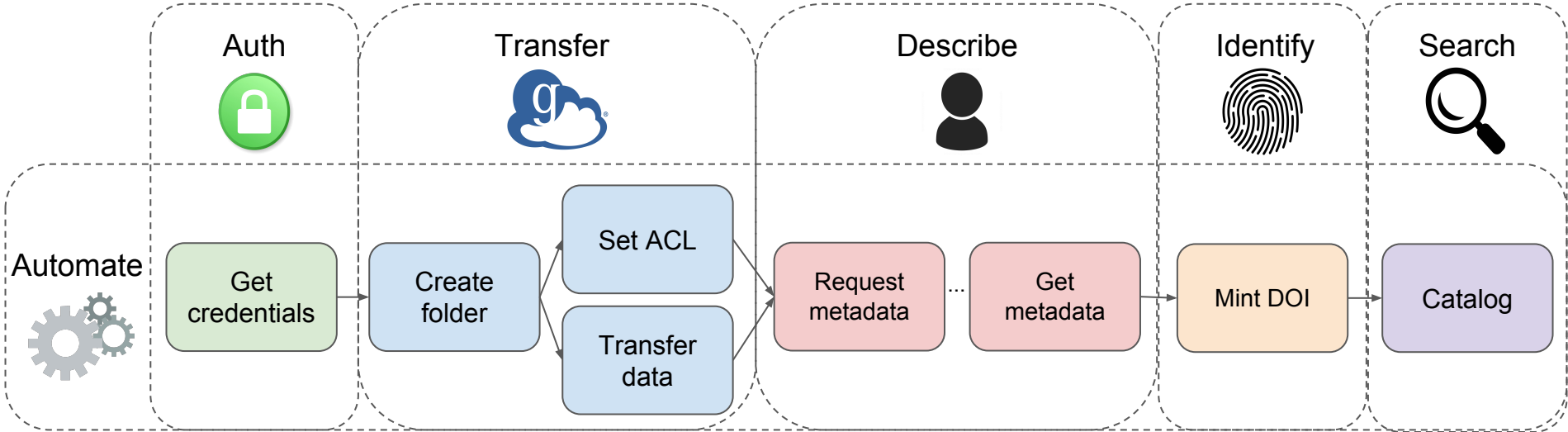
- Flexible, workflow-oriented automation
- Scalable, secure, fault tolerant flows
- Simplified application of TAP automation to common science problems
- Support for asynchronous, human-in-the-loop tasks
- Facilitates user contributed trigger sources & action services

A Vision for Service-Based Automation

Specialized services to perform common tasks

+

A reliable automation platform to link them together



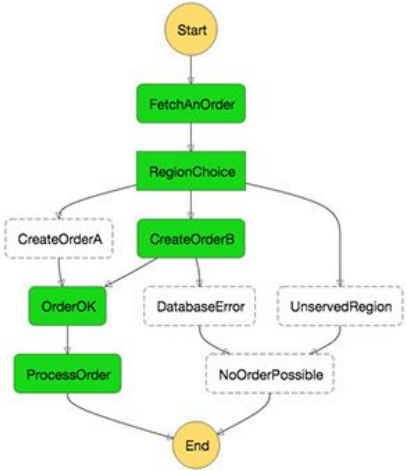
Automate: An Initial Prototype

Cloud service to compose and execute data manipulation flows

Built on AWS Step Functions

- Simple JSON-based state machine language
- Facilitates branching, loops, etc.
- Propagates state through the flow

Standardized API to integrate arbitrary trigger & action services



Secured with Globus Auth

Uses TAP to associate trigger conditions that initiate flows

Automate Prototype: The Service

Users combine action services to create flows by submitting a flow definition and input data JSON documents

- Definition based on state machine language
- Input data is combined with event info on submission

Associate a trigger condition -- event data is passed in when executed

We provide a polling SFN activity that halts a flow until an action_id has completed

Automate Prototype: Actions

Any service can expose the Action API

- /automate/v1/action/run, status, cancel, introspect, …
- .../status used to enable polling
- We give the service an action_id on invocation

When registering an action we make an internal lambda function that calls your service's url

- Makes an ARN for it and maps to a user-friendly name for use in flows

Introspect tells us what input the action accepts -- used during flow creation

The action can then be stepped to in a flow

Automate Prototype: Events

Any service can expose the Events API

- /automate/v1/event/register, poll, introspect, …

Automate polls each event interface and adds responses to a reliable Simple Queue Service queue

- Events processed by lambda functions

Integrates Ripple as an event source

- Can be driven by data events

An Ecosystem for Automation

Initial services include:

Auth

Provide Globus credentials to access services

First step of a flow

Transfer

Thin wrapper around Globus Transfer

Pass it source/dest and data location

Polls until success

Execute

A service around the Parsl library

Enables secure remote execution

Center

Human-in-the-loop

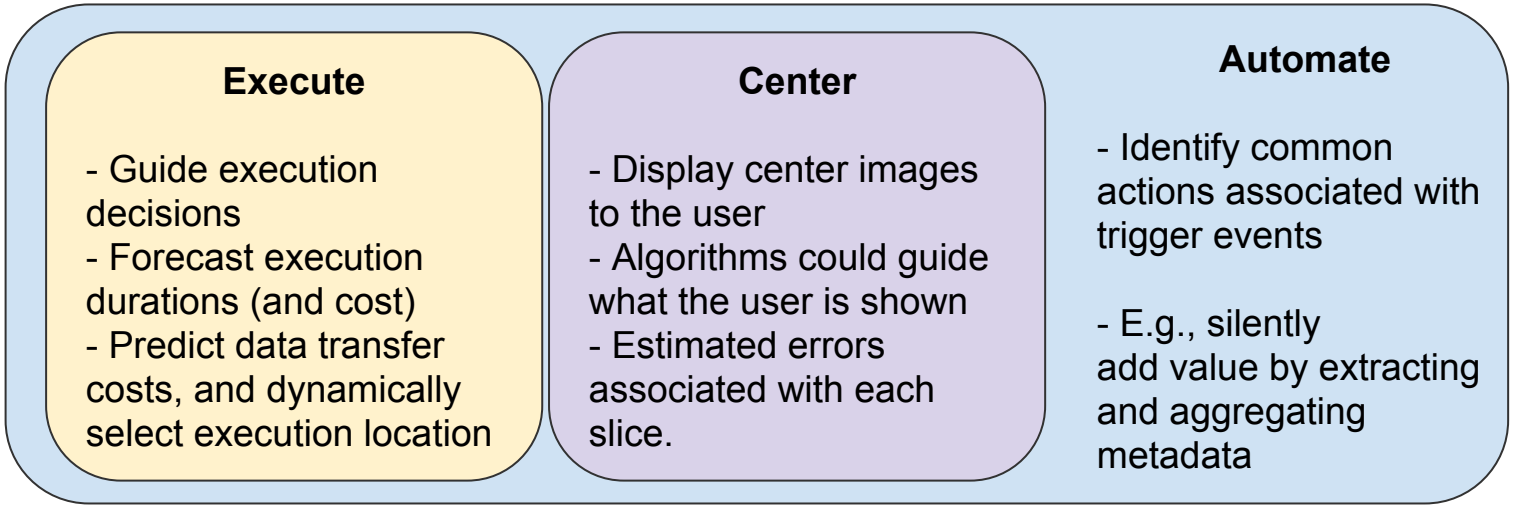
Async task halts flow

Users enter center of rotation value then flow resumes

Autonomous Automation

Automate will provide a fabric for smart, autonomous computing

New opportunities to integrate ML



Next Steps

Test Automate -- beam time last weekend

Add more services

Find more use cases

Integrate autonomous decision making

Thanks!

Questions?