



AutoDrug

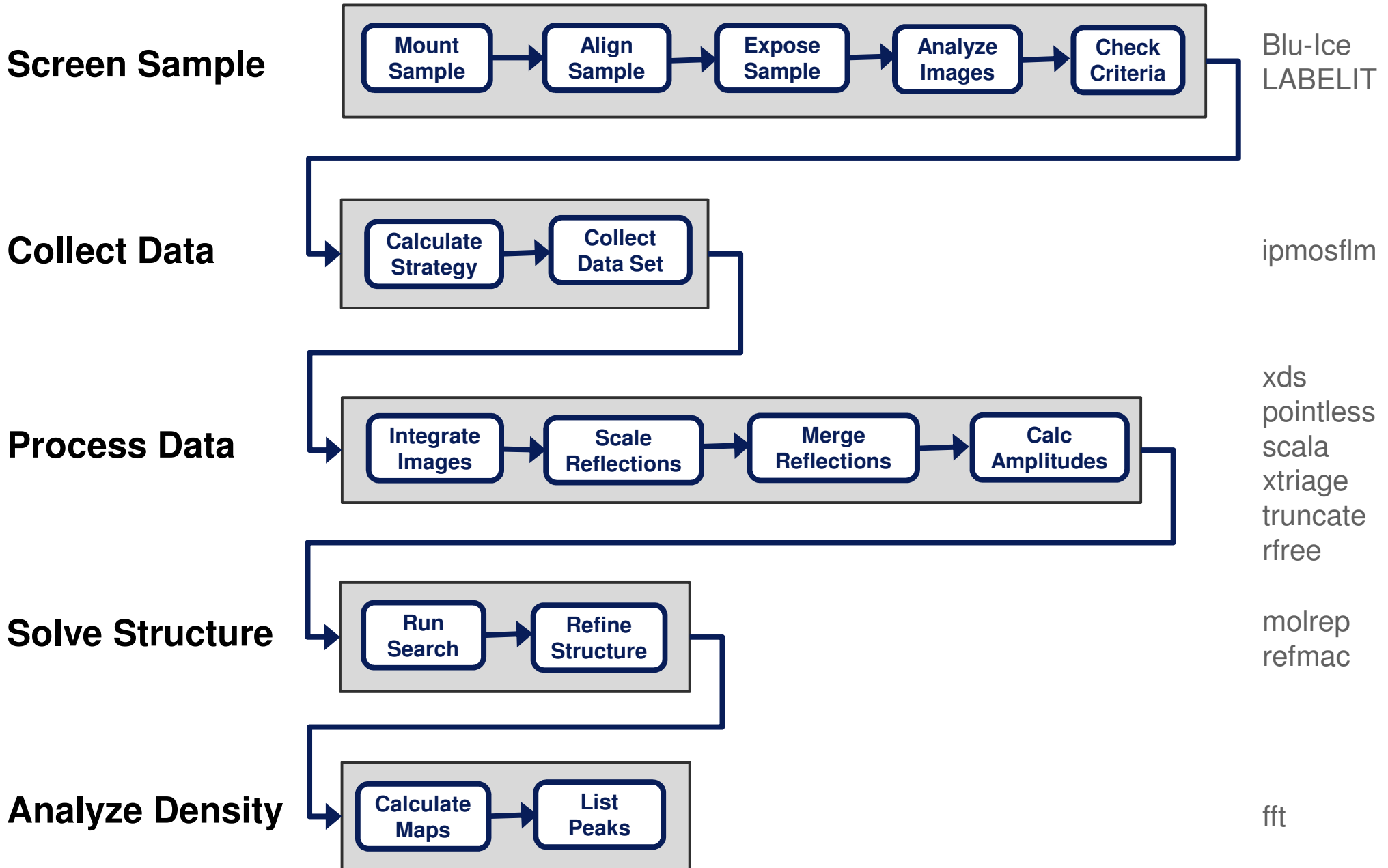
Automated Scientific Workflows for Fragment Based Drug Discovery using RestFlow

Timothy M. McPhillips

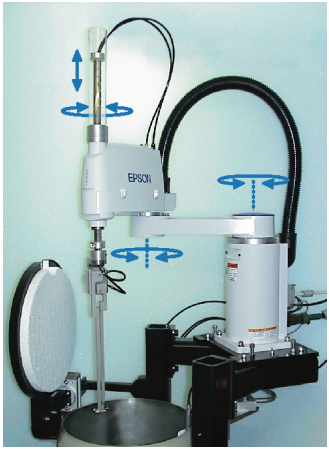
*Stanford Synchrotron Radiation Lightsource
AbsoluteFlow*



Pipeline or Scientific Workflow?



Overview

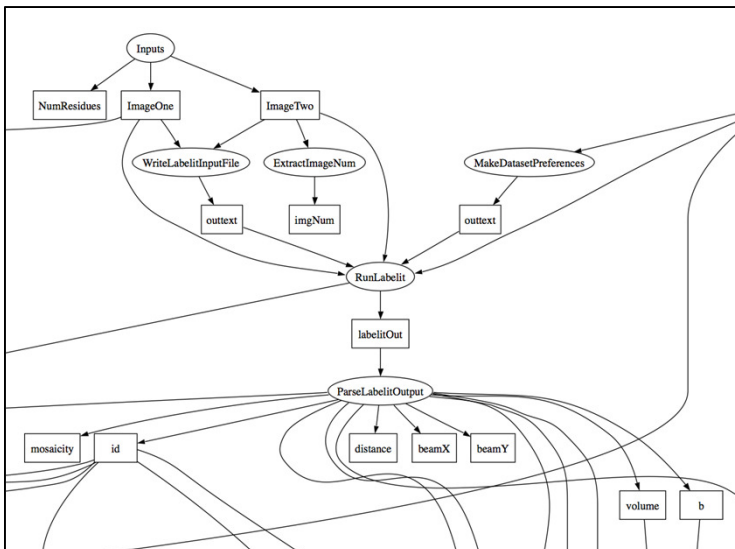
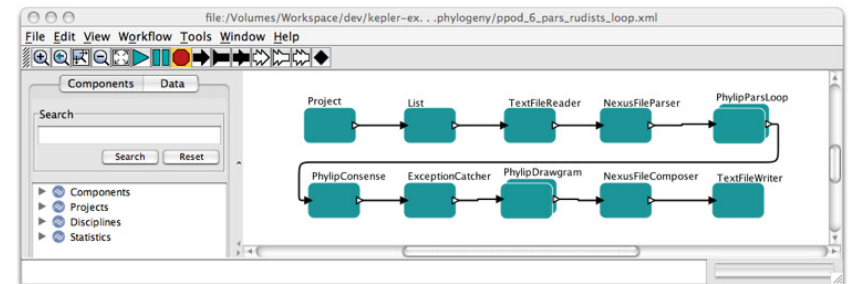


Past automation achievements

- Integrated access to diverse instruments and procedures.
- Limits of automation spanning experiment and data analysis.
- It's not *just* that it's hard.

AutoDrug and scientific workflows

- The origin of RestFlow.
- Why build a new system?



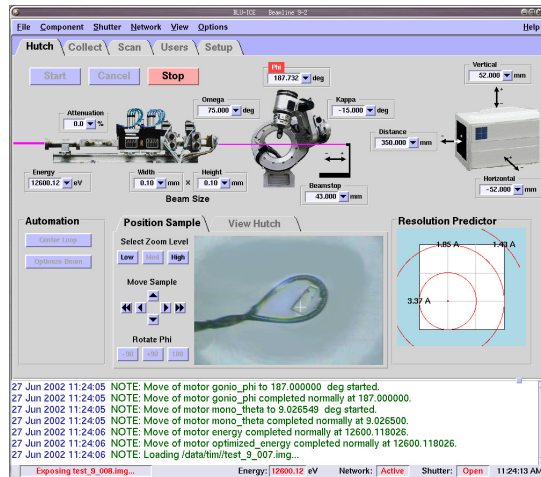
What is RestFlow?

- Master script or Unix pipeline on steroids?
- Complex dataflow graphs, data driven execution, preserved intermediate results, reports.

The future

- Project-scale workflow and data management.

The past: uniform access to resources

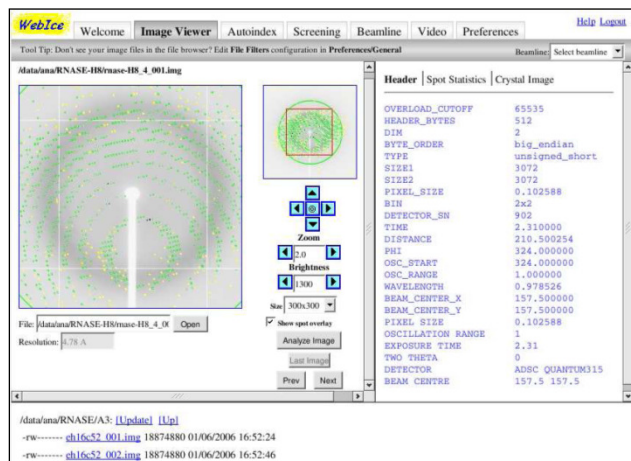
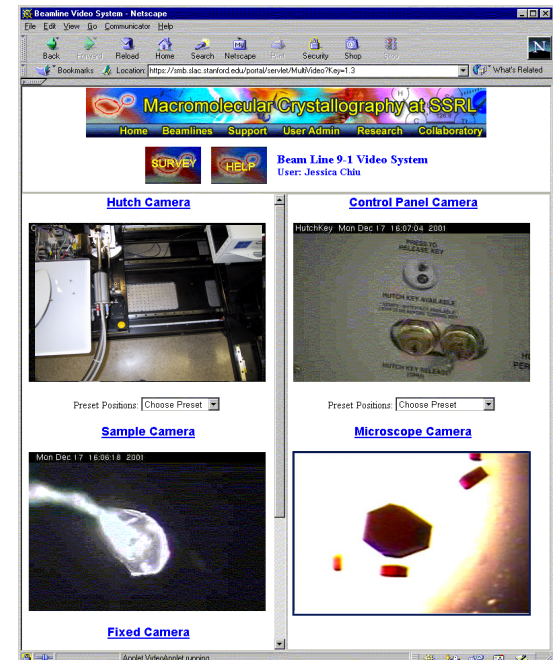


Blu-Ice/DCSS

- Unified access to diverse beam line hardware.
- Single data collection interface for multiple detector types.

Collaboratory Tools

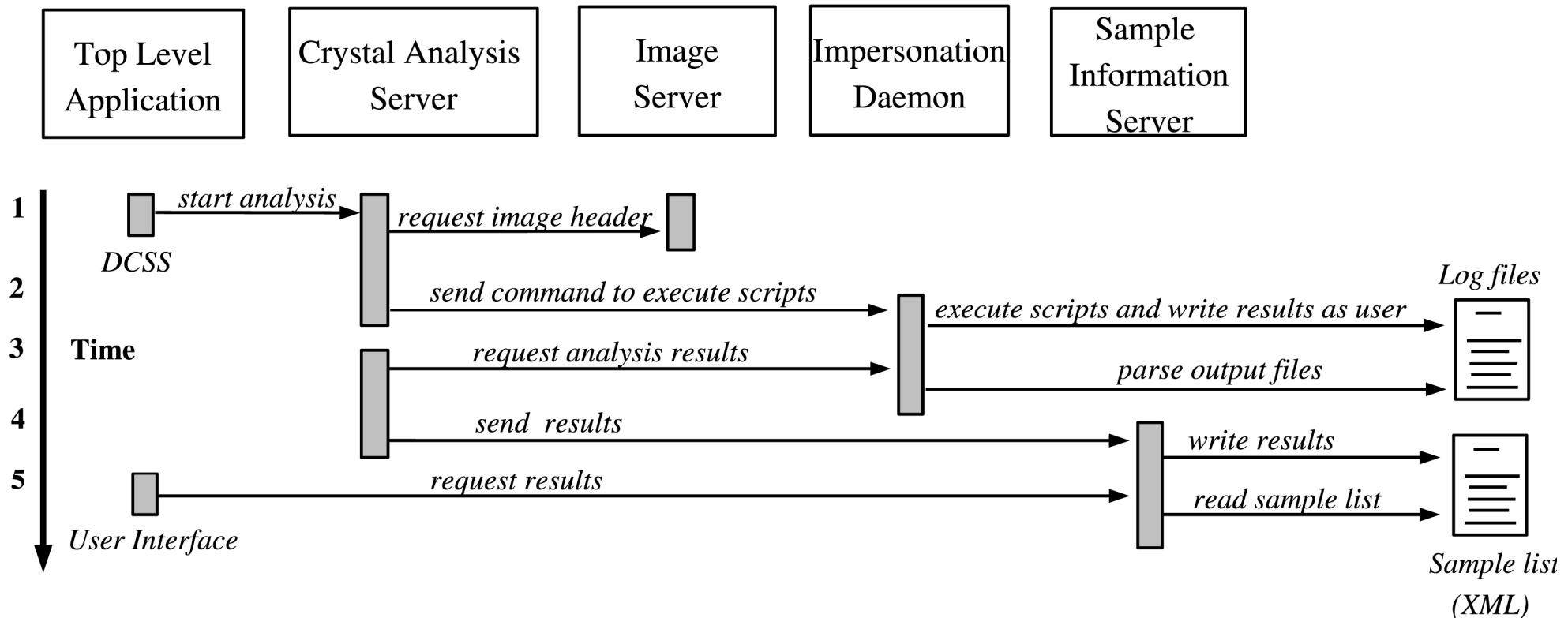
- Equivalent local and remote beam line control.
- Single local and remote data processing environment across beam lines.



Web-Ice

- Unified access to sample screening and scoring, collection strategy, data collection, and data processing.
- Continuous availability during and after beam time.

Automation spanning experiment and analysis is hard



- Client-server approach where scripts run behind a server.
- Challenging to access data with the user's rights and to leave results where only that user can access them.
- Detailed knowledge of many interacting systems required.
- System-level software development (and developers) required.

But that's not the whole story...

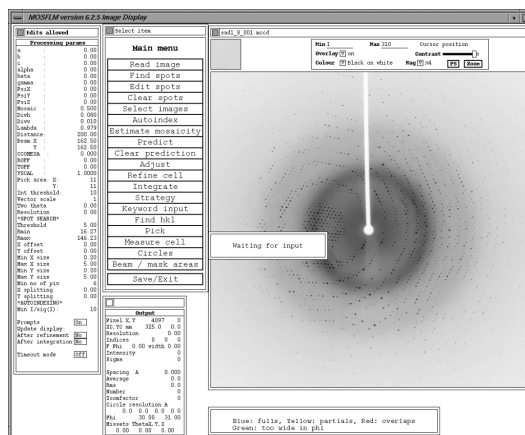
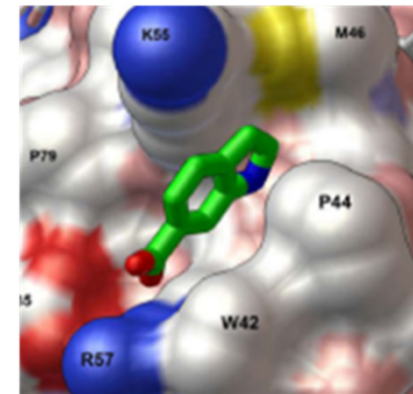


Numerous samples

- Multiple samples of same kind but varying quality.
- Different kinds of samples within a project.
- Samples from *different* projects.
- Varying sensitivity to radiation damage.

Diverse scientific goals

- Determination of new structures.
- High resolution data collection.
- Molecular replacement of related structures.
- Detection and identification of ligands.



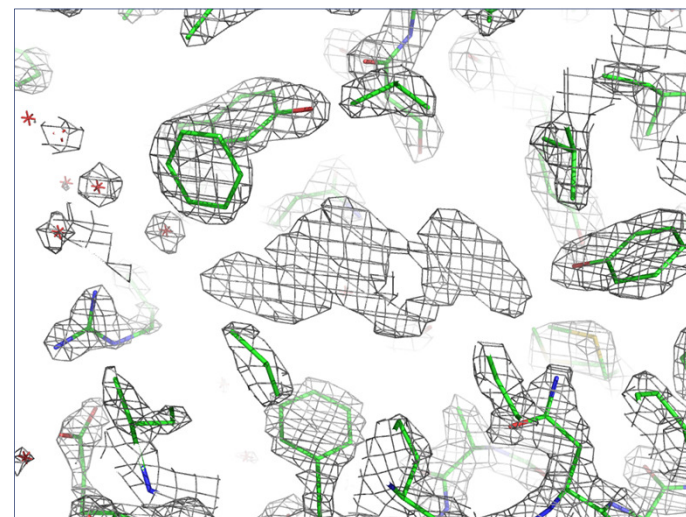
Strategic use of beam time

- Screen all crystals of a particular type first?
- Collect on first crystal of sufficient quality?
- Switch to different crystal when current sample decays?

AutoDrug—the origin of RestFlow

Objectives

- Automate crystallographic stages of fragment-based drug design.
- Screen crystals, collect and reduce data, solve by molecular replacement, identify bound fragments.



AutoDrug

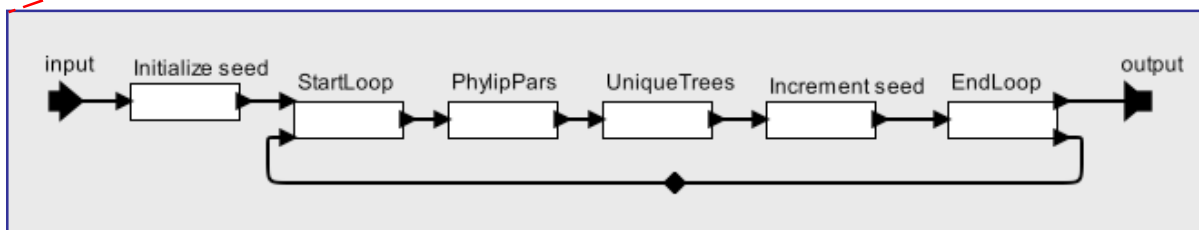
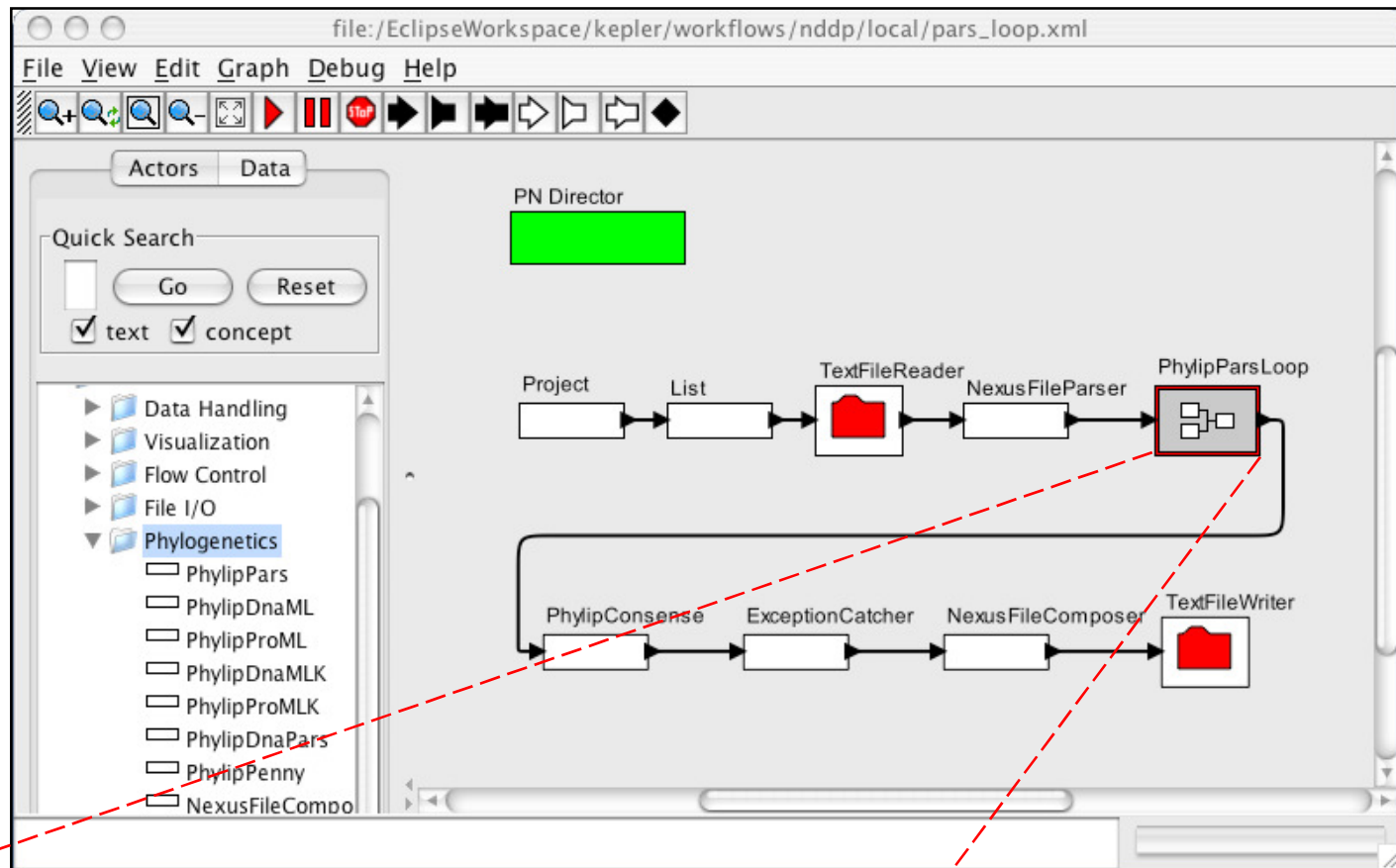
RestFlow

Beamline

Engineering strategy

- Decouple automation of the overall workflow from the development of the automation framework.
- Develop exactly the automation framework we need using tools we understand and can support.
- Remove software developers from the loop?

Scientific workflow in Kepler (2005)

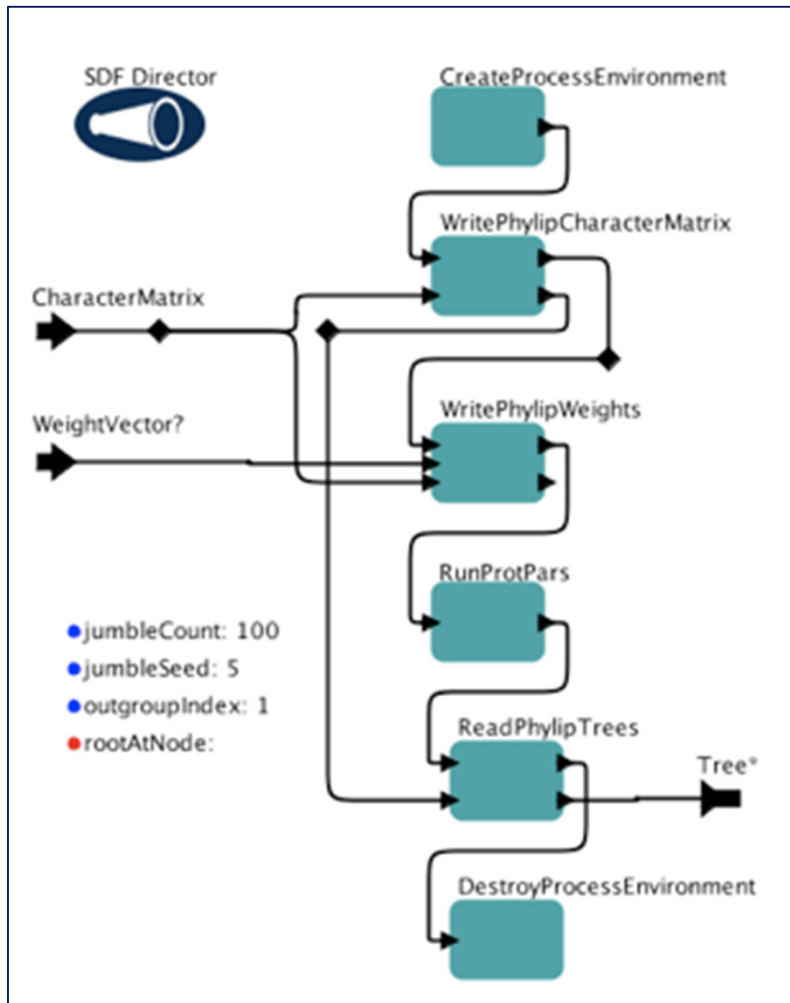


Timothy M. McPhillips and Shawn Bowers (2005). **An Approach for Pipelining Nested Collections in Scientific Workflows.** *SIGMOD Record* 34: 12-17.

Essential functions of a scientific workflow system

1. Automate programs scientists already use.
2. Schedule invocations of programs correctly and efficiently.
3. Manage flow of data to, from, and between these programs.
4. Enable scientists to write or modify their own workflows easily.
5. Make it easy for scientists to predict what a workflow will do when executed.
6. Allow any workflow to be nested as a component in another workflow.
7. Enable users to share, version, and publish their workflows.
- 8. Enable scientists to automate additional programs themselves.**
- 9. Organize intermediate and final data products as desired by users.**

Why build a new system?



Part of a Kepler workflow for inferring phylogenetic trees from protein sequences.

Freely available systems

- *Kepler (Ptolemy II), Taverna, VisTrails, Triana...*
- Graphical programming environments.
- Blocks represent steps in a workflow, and arrows declare paths of data flow.

Limitations

- Little support for organizing intermediate and final results.
- Software developers needed to develop new components.

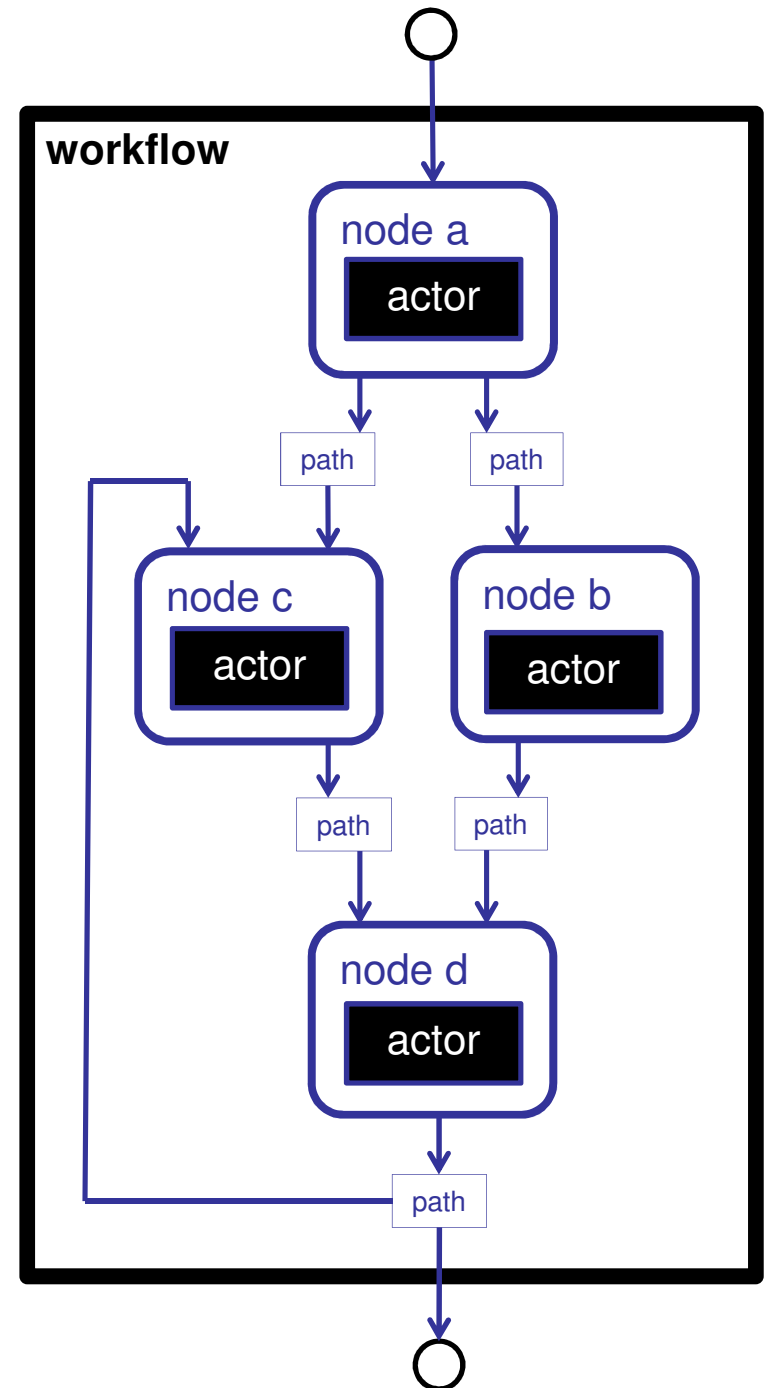
What is RestFlow?

Master script with explicit flow of data?

- Text-based language for running other scripts.
- Actors run inside of *nodes*.
- Arrows between the nodes represent data flow.
- Paths specify where data is stored.

A Unix pipeline on steroids?

- Each actor can run more than once.
- Multiple data streams can flow in and out of each node.
- Intermediate data products can be given unique names and saved.



Actors - simplified scripts

- id: GetImageOscillation

type: PerlActor

properties:

inputs:

imageDir:

imagePrefix:

imageNumber:

imageType:

step: |

construct expected image file name

```
$imageFile = $imageDir . $imagePrefix . "_" . $imageNumber . "." . $imageType;
```

extract oscillation start field from image header

```
$getOscStartCommand = "getImgHeader " . $imageFile . " | grep OSC_START | awk '{print \$2}'";
```

```
$oscStart = ` $getOscStartCommand `;
```

```
chomp($oscStart);
```

extract oscillation range field from image header

```
$getOscRangeCommand = "getImgHeader " . $currentfile . " | grep OSC_RANGE | awk '{print \$2}'";
```

```
$oscRange = ` $getOscRangeCommand `;
```

```
chomp($oscRange);
```

outputs:

oscStart:

oscRange:

- An actor receives its **inputs** in one set of variables and leaves its **outputs** in other variables.
- No command-line parsing or complex result management is needed.
- No file management needed.

A Perl actor

A Bash actor

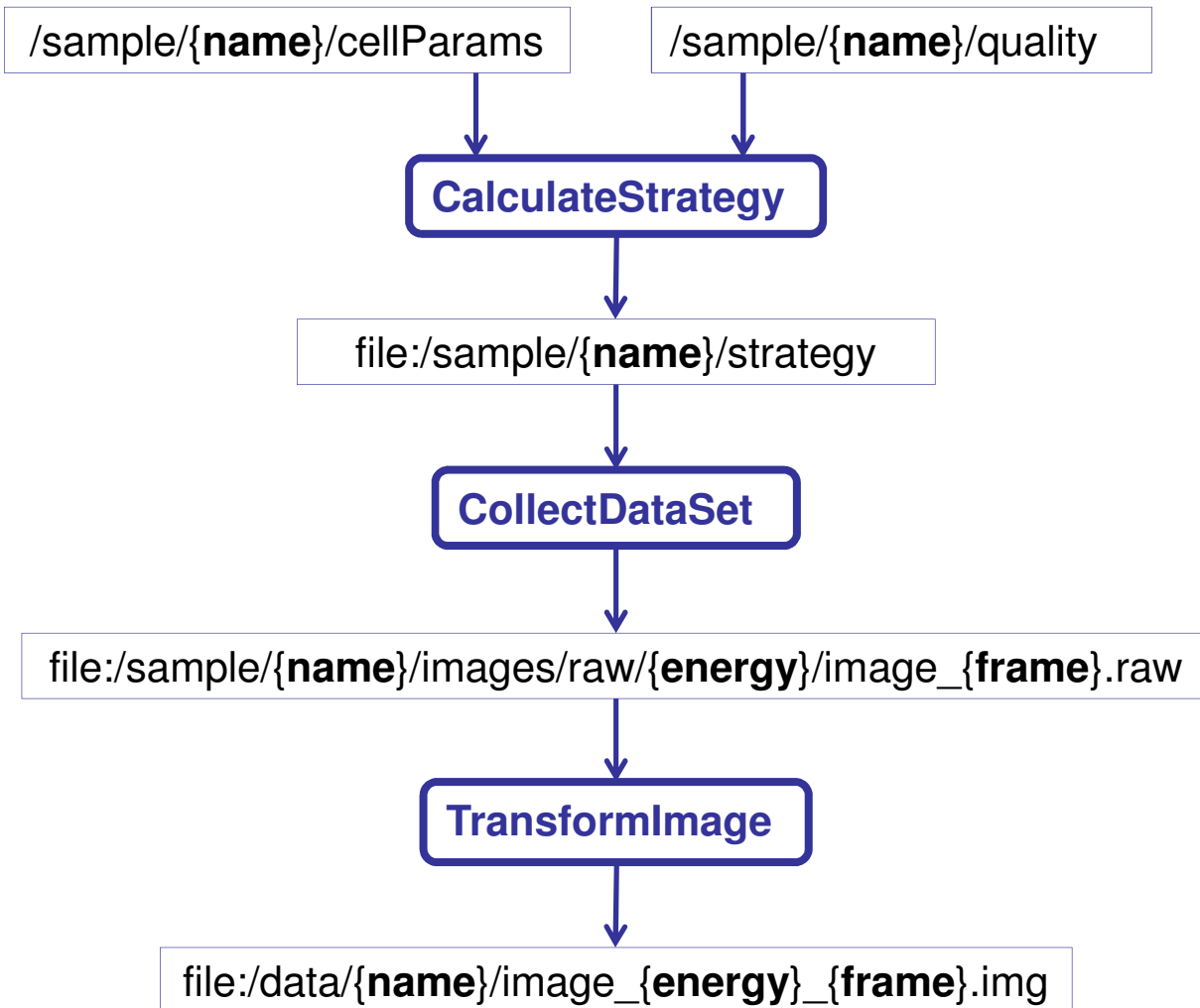
```
- id: LastImgFinder
  type: BashActor
  properties:
    inputs:
      directory:
      imageRoot:
      imagetype:
    step: |
      lastImageLine=`find ${directory} -name ${imageRoot}_*.${imageType} -prune -maxdepth 1 | sort -r`
      lastImageName=`echo ${lastImageLine} | awk '{print $1}'`
    outputs:
      lastImageName:
```

A Blu-Ice (Tcl) actor

```
- id: CenterLoop
  type: BluIceActor
  properties:
    step: |
      set centeringOperation [start_waitable_operation centerLoop]
      wait_for_operation_to_finish $centeringOperation
```

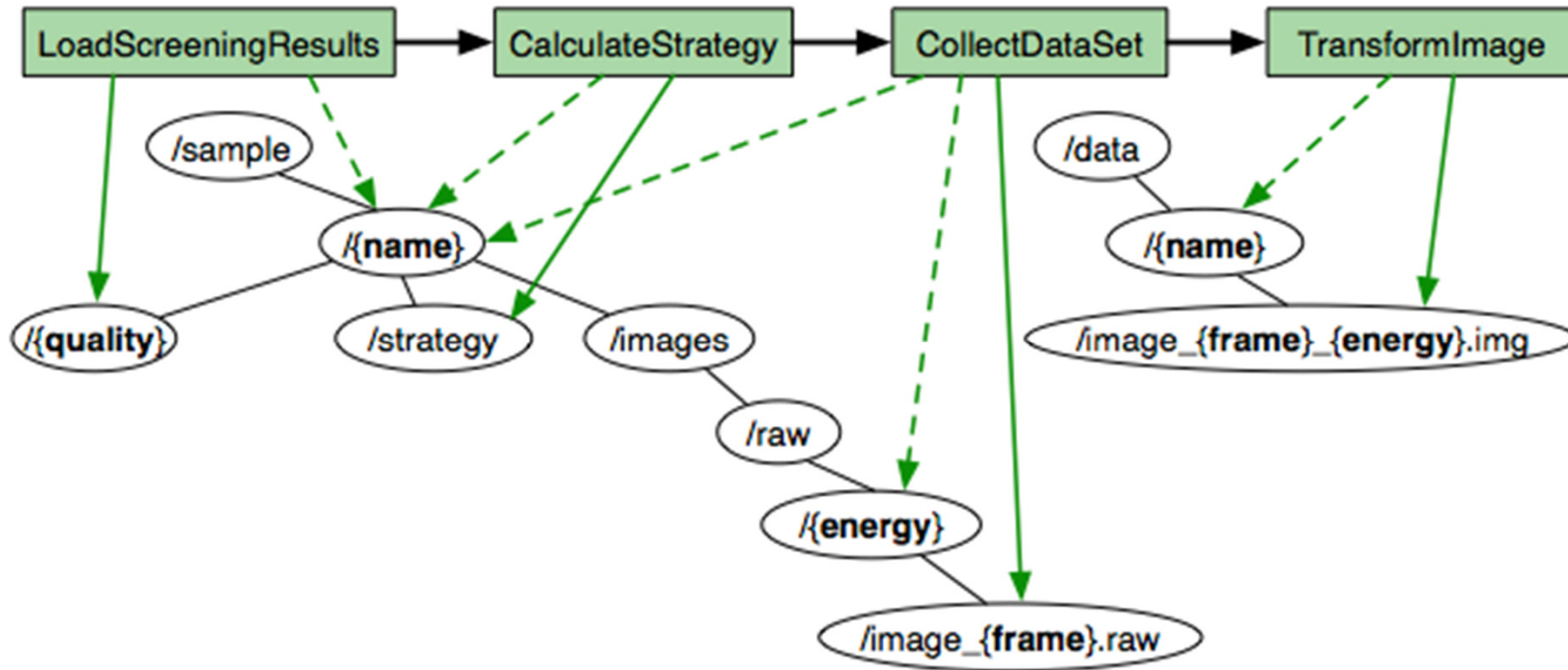
- YAML is the primary format for declaring actors and workflows in RestFlow.
- Like XML but far simpler to edit by hand.
- Workflows also may be constructed dynamically in Java or Python (Jython).

Connecting scripts with path templates



- RestFlow wires actors together using *path templates*.
- Variables in the templates are expanded as the workflow runs.
- Every file gets a unique name.
- Data is organized automatically on the file system.

Automatic organization of workflow outputs



Inflow and outflow expressions...

- Specify the connectivity of the workflow graph via a publish-subscribe metaphor.
- Automatically name and organize output directories and files.

A RestFlow workflow

- id: **CalculateStrategy**

type: Node

properties:

actor: !ref **StrategyCalculator**

inflows:

cell: /sample/{name}/cellParams

quality: /sample/{name}/quality

outflows:

strategy: file:/sample/{name}/strategy

- id: **CollectDataSet**

type: Node

properties:

actor: !ref **DataCollector**

inflows:

runDefinition: /sample/{name}/strategy

outflows:

rawImage: file:/sample/{name}/images/raw/{energy}/image_{frame}.raw

- id: **TransformImage**

type: Node

properties:

actor: !ref **DataTransformer**

inflows:

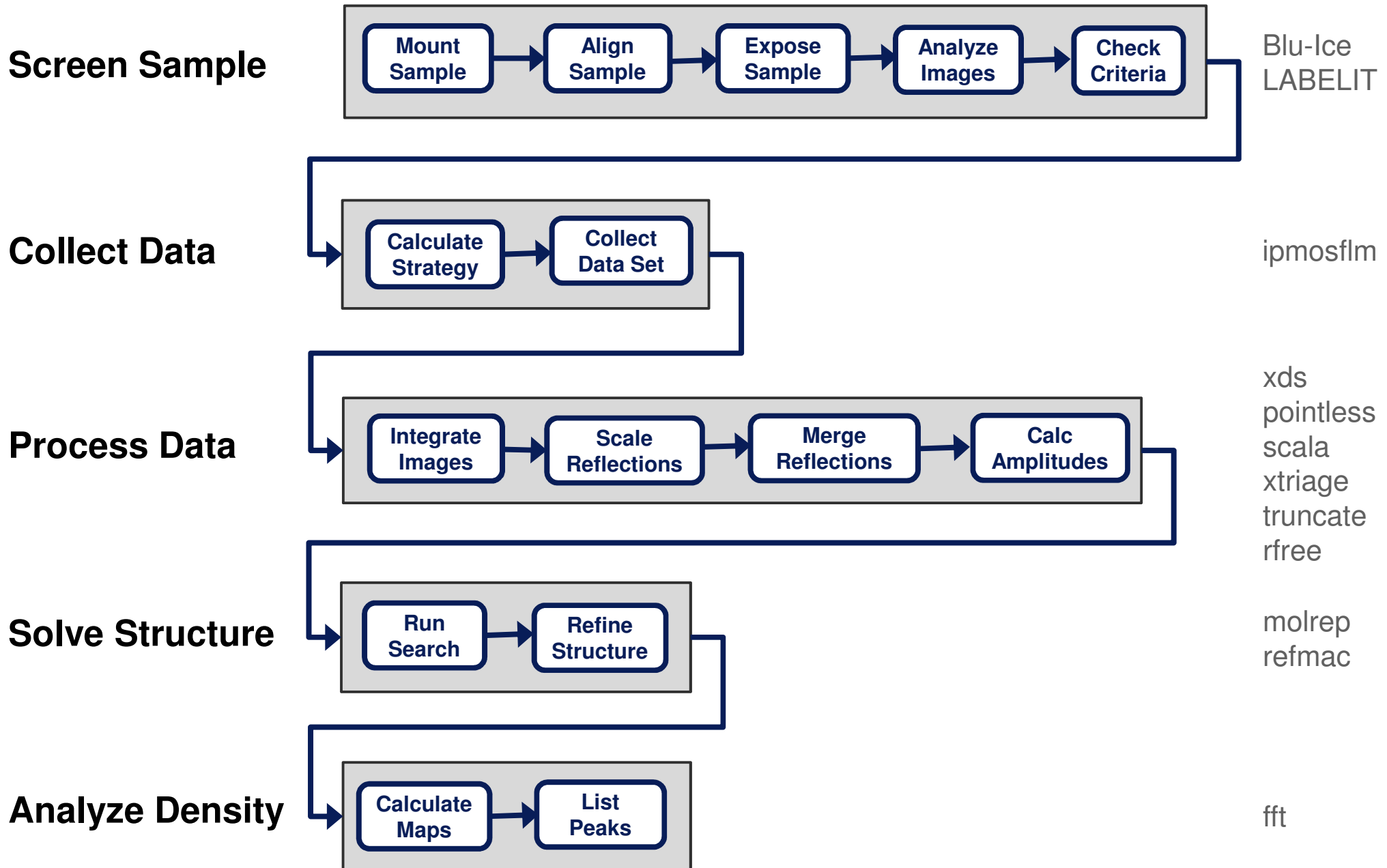
rawImage: /sample/{id}/images/raw/{e}/image_{frameNumber}.raw

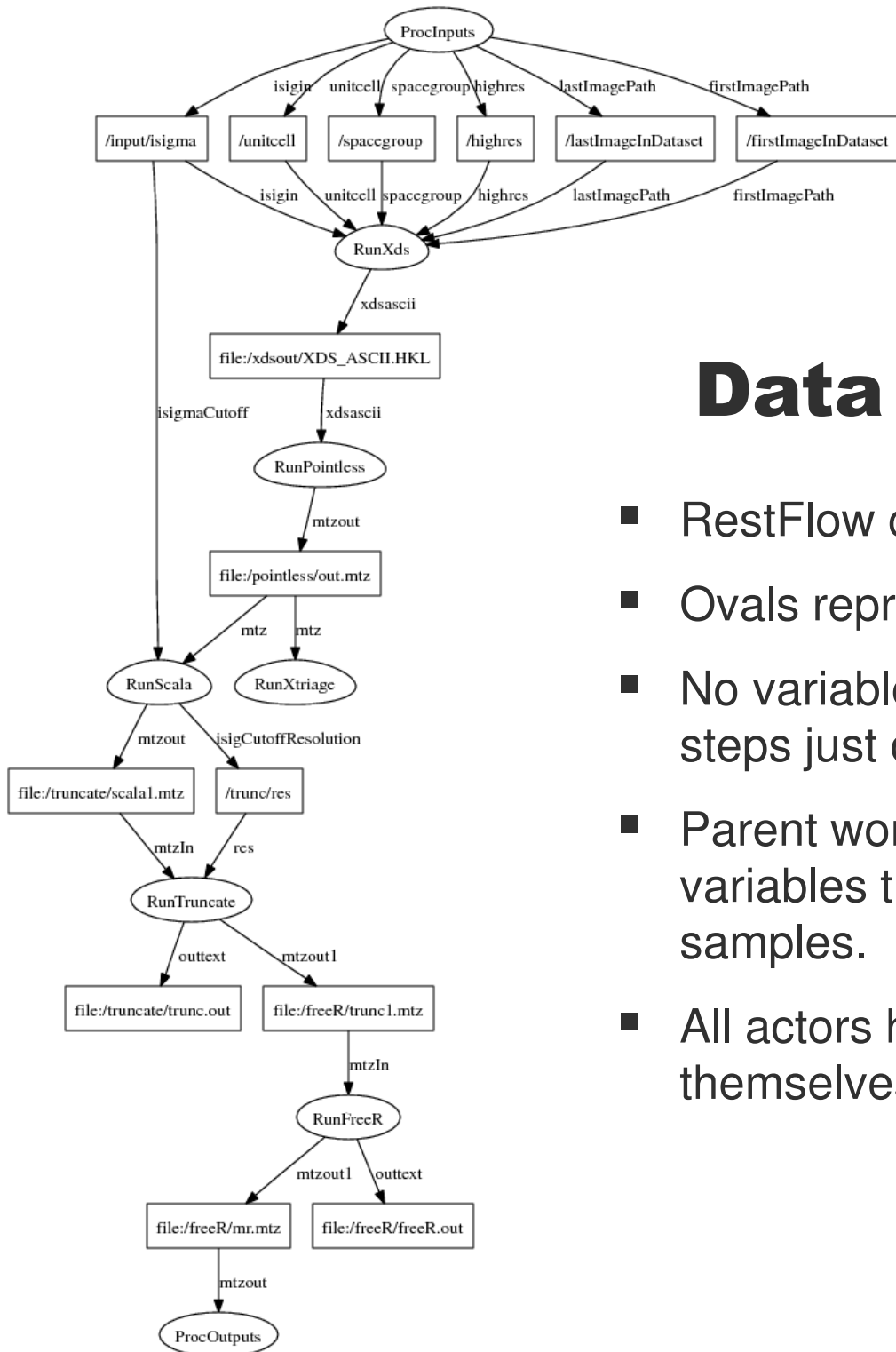
outflows:

correctedImage: file:/data/{id}/image_{e}_{frameNumber}.img

- Text version of previous workflow.
- Each block of text is a node.
- Each node refers to an **actor** to run.
- *Inflows* and *outflows* route data into and out of nodes.

A Generic AutoDrug Workflow



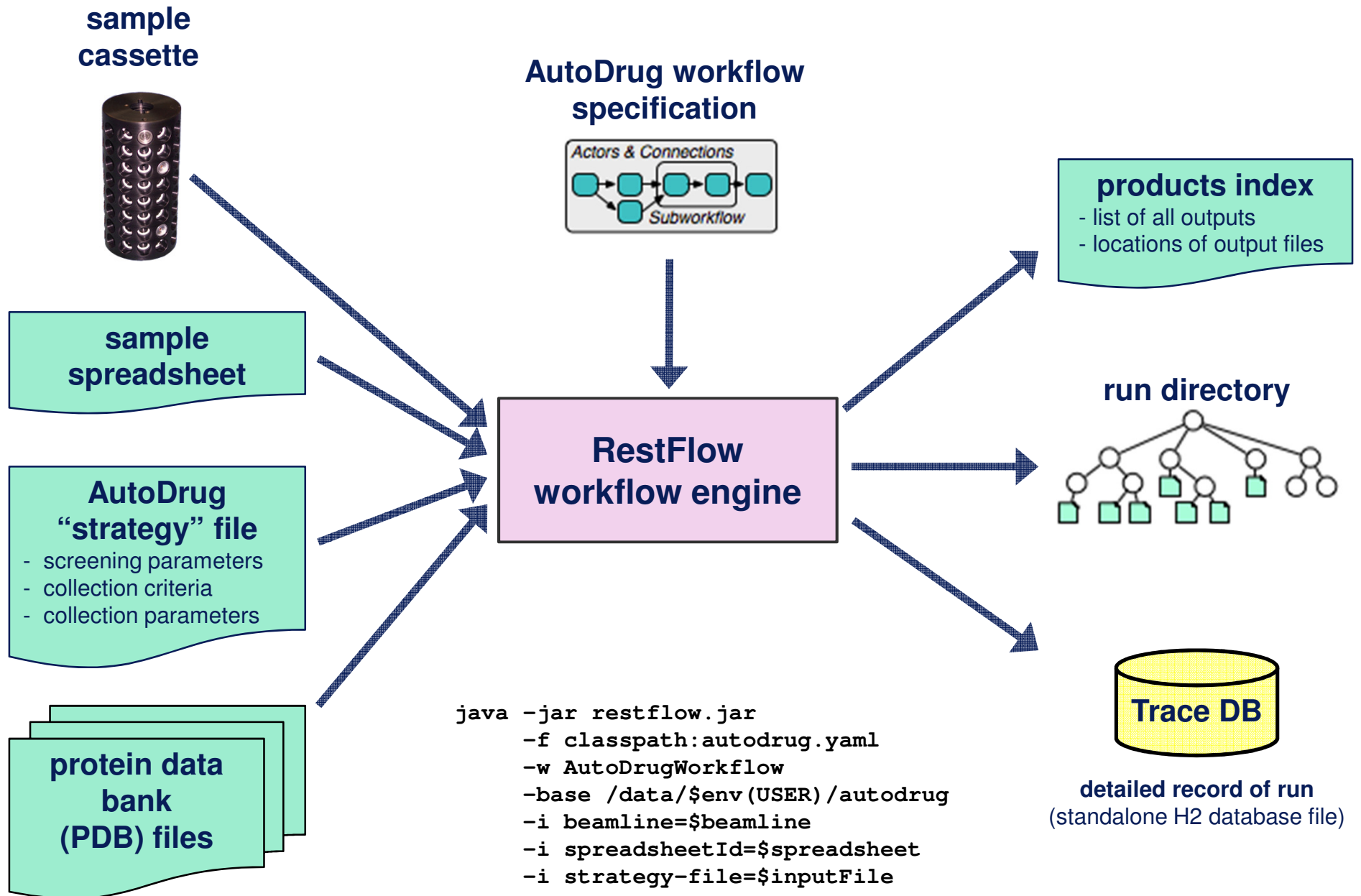


Data processing

- RestFlow can render a workflow with *GraphViz**
- Ovals represent actors, boxes are path templates.
- No variables in these paths because each actor steps just one time per run of subworkflow.
- Parent workflow declares a path *prefix* with variables that distinguish outputs from different samples.
- All actors here actually are subworkflows themselves.

* <http://www.graphviz.org>

Running AutoDrug



Demonstration of AutoDrug

- **Collaborator:** Cocrystal Discovery Inc.
- **Samples:** 96 crystals previously screened and analyzed by CoCD.
- **Groups:** Samples were grouped by drug fragment cocktail, roughly three samples per group.
- **Screening strategy:** Screen all crystals in a group and select single best sample.
- **Collection criteria:** Collect data on best crystal in group only if it meets diffraction quality requirements.

- **Results:**

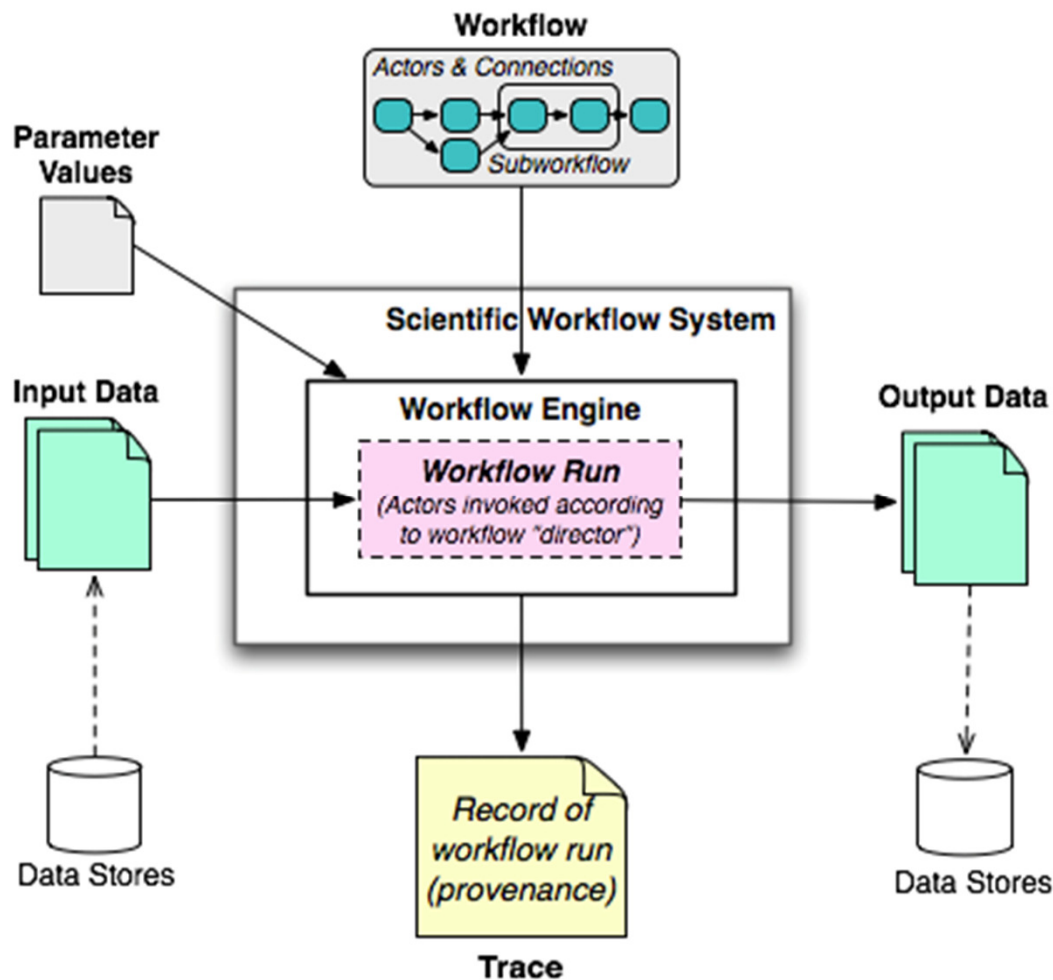
AutoDrug **selected the same 16 crystals** that CoCD's researchers had chosen for data collection.

Seven datasets yielded density above 5σ in the target regions.

Location of **density matched** the results of the **manual experiments**.

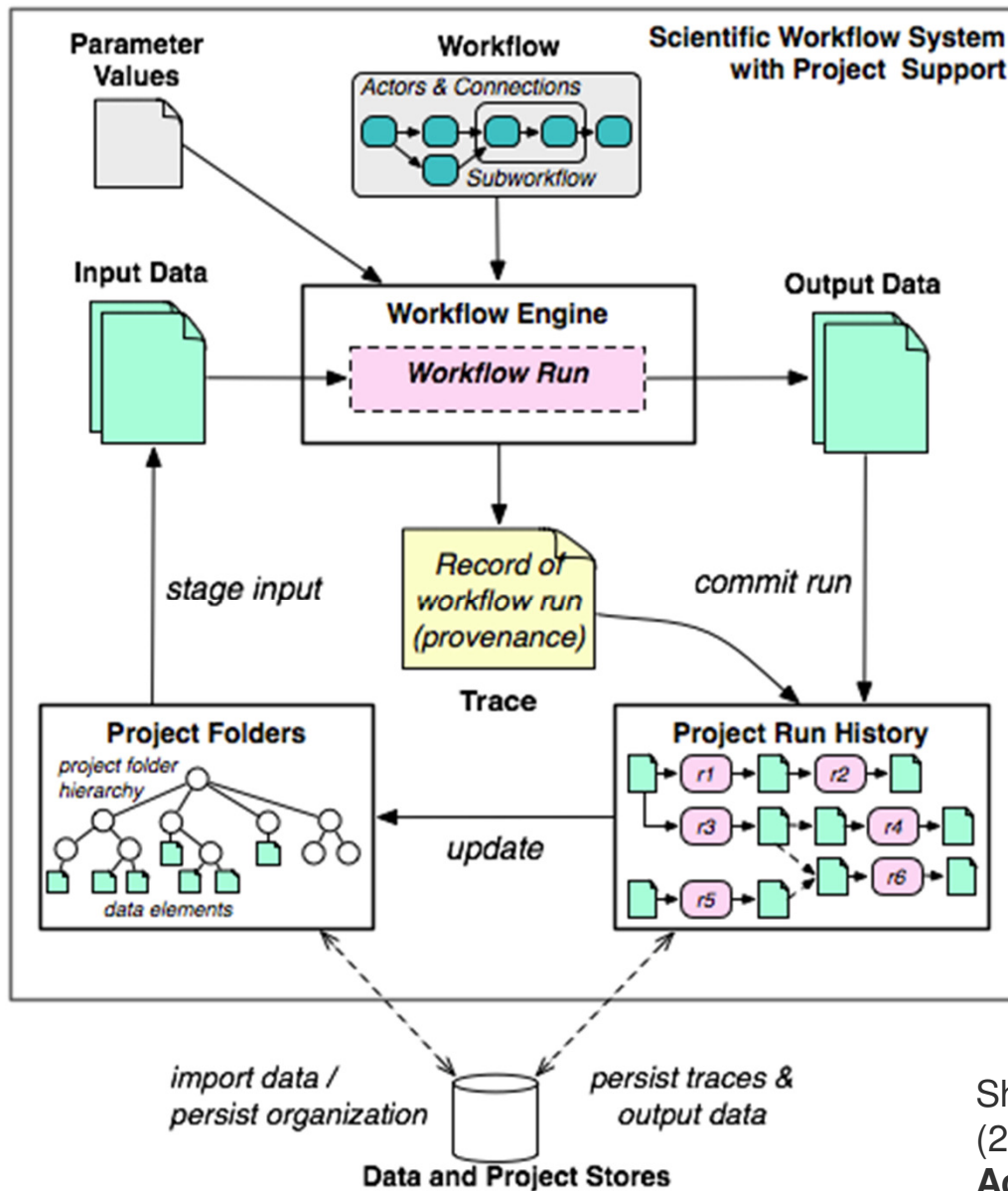
AutoDrug executed the entire process **without manual intervention**.

Future Work



- RestFlow focuses on running one workflow once.
- Input and output data are managed outside the system.
- Records of workflow runs and provenance of results are stored outside system
- Does not manage the flow of data from or through multiple workflow runs.
- **Example scenario:** Solving structures using data collected from multiple crystals on different trips to the light source.

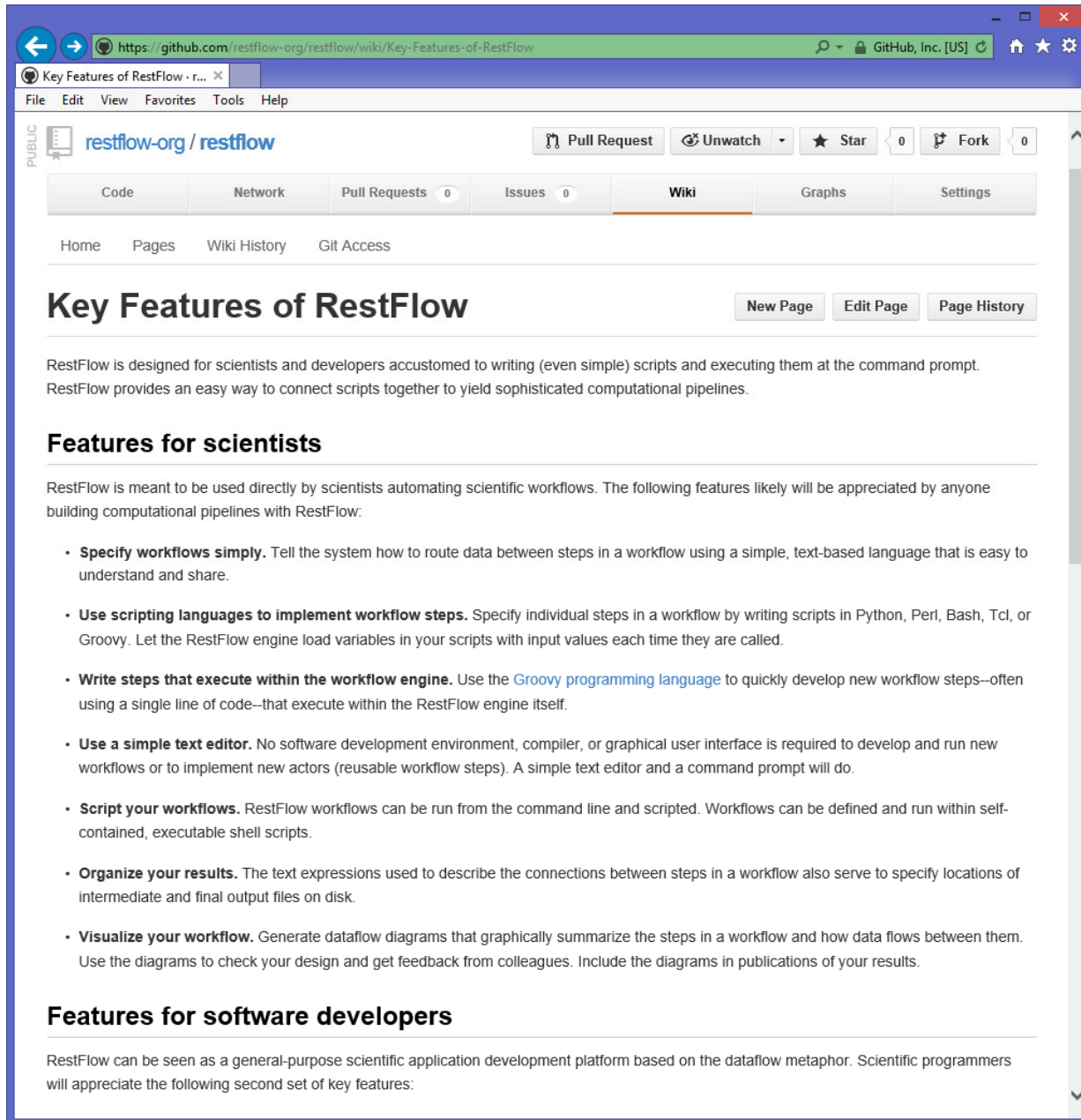
Project-Scale Workflow Management



- Requires expanding the boundary of scientific workflow systems.
- Need to include project data and run histories within the system.
- Published design for such a system in 2007.
- Plan to implement a prototype using cloud technologies this year.

Shawn Bowers, Timothy McPhillips, and Bertram Ludascher (2007). **Project Histories: Managing Data Provenance Across Collection-Oriented Scientific Workflow Runs.** *Lecture Notes in Computer Science 4544*: 122-138.

Status and Availability



The screenshot shows a web browser window displaying the GitHub Wiki page for RestFlow. The page title is "Key Features of RestFlow". The content includes a description of RestFlow as a tool for scientists and developers, followed by a section titled "Features for scientists" which lists several key capabilities: specifying workflows simply, using scripting languages, writing steps that execute within the workflow engine, using a simple text editor, scripting workflows, organizing results, and visualizing the workflow.

Key Features of RestFlow

RestFlow is designed for scientists and developers accustomed to writing (even simple) scripts and executing them at the command prompt. RestFlow provides an easy way to connect scripts together to yield sophisticated computational pipelines.

Features for scientists

RestFlow is meant to be used directly by scientists automating scientific workflows. The following features likely will be appreciated by anyone building computational pipelines with RestFlow:

- **Specify workflows simply.** Tell the system how to route data between steps in a workflow using a simple, text-based language that is easy to understand and share.
- **Use scripting languages to implement workflow steps.** Specify individual steps in a workflow by writing scripts in Python, Perl, Bash, Tcl, or Groovy. Let the RestFlow engine load variables in your scripts with input values each time they are called.
- **Write steps that execute within the workflow engine.** Use the [Groovy programming language](#) to quickly develop new workflow steps—often using a single line of code—that execute within the RestFlow engine itself.
- **Use a simple text editor.** No software development environment, compiler, or graphical user interface is required to develop and run new workflows or to implement new actors (reusable workflow steps). A simple text editor and a command prompt will do.
- **Script your workflows.** RestFlow workflows can be run from the command line and scripted. Workflows can be defined and run within self-contained, executable shell scripts.
- **Organize your results.** The text expressions used to describe the connections between steps in a workflow also serve to specify locations of intermediate and final output files on disk.
- **Visualize your workflow.** Generate dataflow diagrams that graphically summarize the steps in a workflow and how data flows between them. Use the diagrams to check your design and get feedback from colleagues. Include the diagrams in publications of your results.

Features for software developers

RestFlow can be seen as a general-purpose scientific application development platform based on the dataflow metaphor. Scientific programmers will appreciate the following second set of key features:

- RestFlow is licensed for free and unrestricted use in any setting (MIT License).
- Source code is publicly accessible via **GitHub**.
- RestFlow 1.0 release in preparation.
- Find source code, downloads, documentation, tutorials at:

restflow.org



Special thanks to

Cocrystal Discovery, Inc

- Michael D. Feese
- David Bushnell

SSRL Macromolecular Crystallography Group

- Yingsu Tsai
- Scott McPhillips
- Ana Gonzalez
- Michael Soltis

Data and Knowledge System Group (UC Davis)

- Bertram Ludäscher
- Shawn Bowers
- Daniel Zinn

downloads and documentation: <http://www.restflow.org>

scidataflow blog: <http://scidataflow.com>

contact: tmcphillips@absoluteflow.org

NIH NCRR Award P41RR001209

