JGI JOINT GENOME INSTITUTE
A DOE OFFICE OF SCIENCE USER FACILITY

Distributed Workflow Management using JAWS

Daniela Cassol
dcassol@lbl.gov

JGI JOINT GENOME INSTITUTE
A DOE OFFICE OF SCIENCE USER FACILITY

Office of Biological and Environmental Research

Mission

The mission of the JGI is to provide the global research community with access to the most advanced integrative genome science capabilities in support of the DOE's research mission.

2,243	Primary Users leveraging JGI data generation capabilities in FY23
15,219	Secondary Users that engaged with JGI science gateways
14PB	JGI Data Repository size as of December 2020
2,620	Publications since 2001

NERSC National Energy Research Scientific Computing Center

BioSciences

BERKELEY LAB

U.S. DEPARTMENT OF ENERGY

Office of Science

JGI JOINT GENOME INSTITUTE
A DOE OFFICE OF SCIENCE USER FACILITY

Office of Biological and Environmental Research

Mission

The mission of the JGI is to provide the global research community with access to the most advanced integrative genome science capabilities in support of the DOE's research mission.

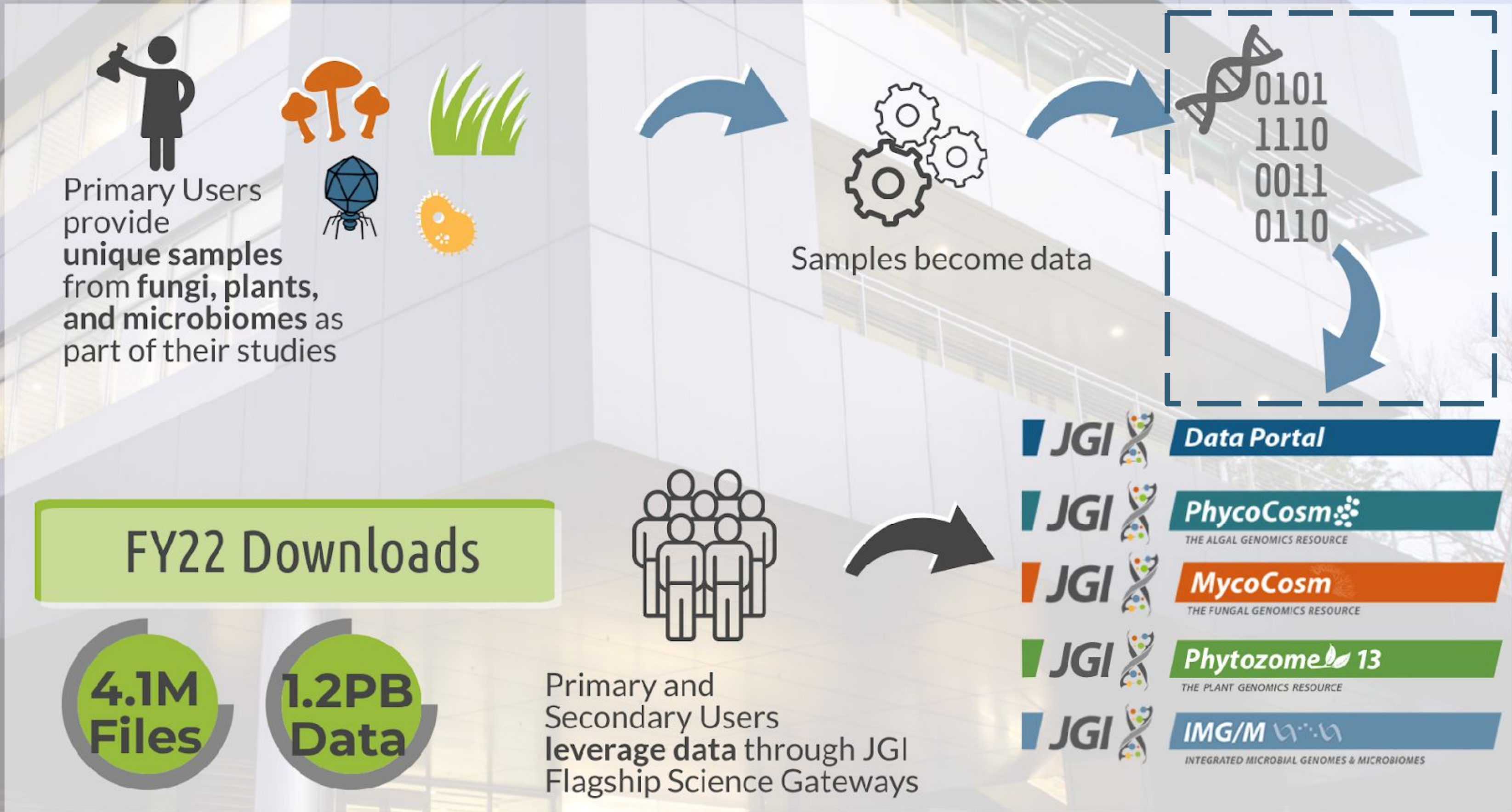
2,243 Primary Users leveraging JGI data generation capabilities in FY23

15,219 Secondary Users that engaged with JGI science gateways

14PB JGI Data Repository size as of December 2020

2,620 Publications since 2001

Data Generation and Reuse


Primary Users provide **unique samples** from **fungi, plants, and microbiomes** as part of their studies

Samples become data

FY22 Downloads

4.1M Files

1.2PB Data

Primary and Secondary Users **leverage data** through JGI Flagship Science Gateways

JGI Data Portal
JGI PhycoCosm
THE ALGAL GENOMICS RESOURCE
JGI MycoCosm
THE FUNGAL GENOMICS RESOURCE
JGI Phytozome 13
THE PLANT GENOMICS RESOURCE
JGI IMG/M
INTEGRATED MICROBIAL GENOMES & MICROBIOMES

NERSC National Energy Research Scientific Computing Center

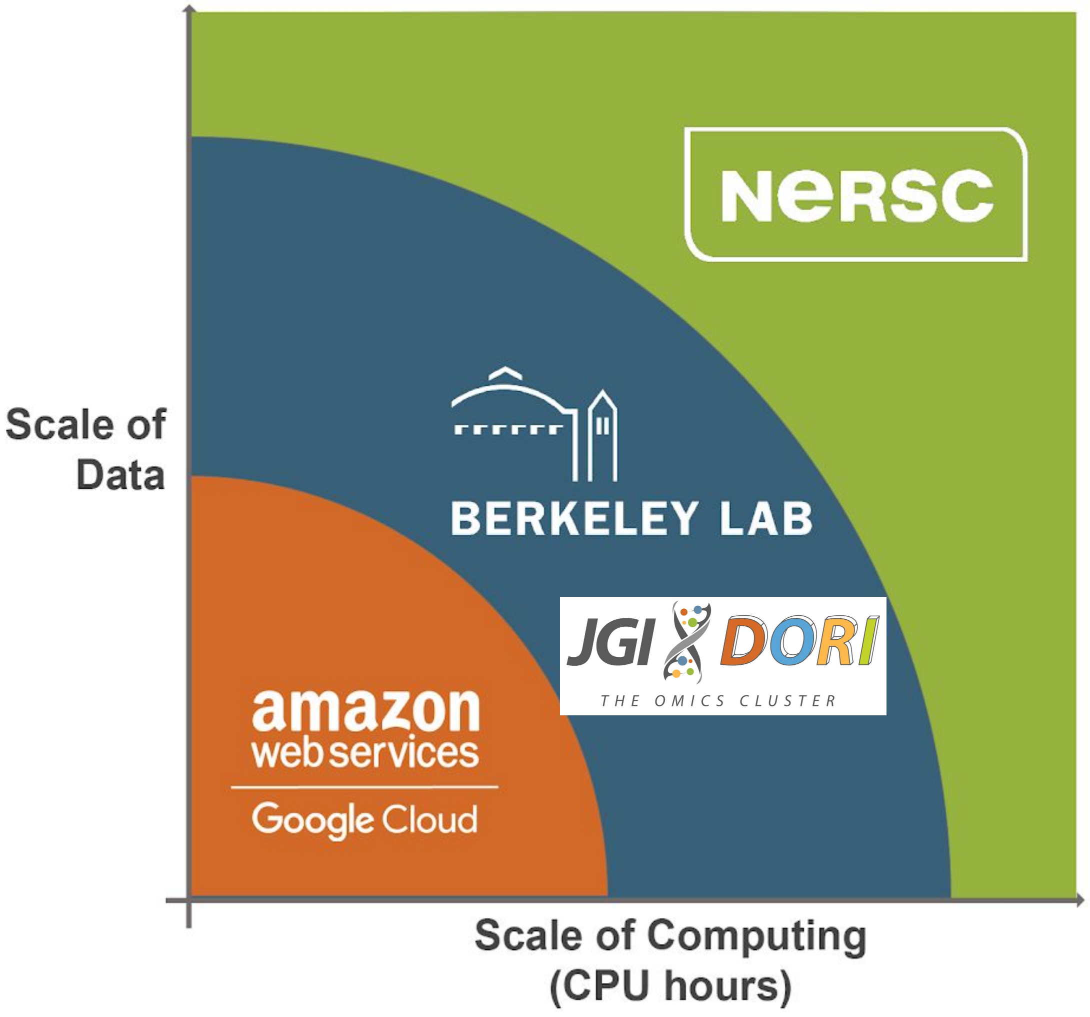
BioSciences

BERKELEY LAB

U.S. DEPARTMENT OF ENERGY | Office of Science

JGI Computing Resources

JGI’s analysis workflow requirements span resources of different scales



JOINT GENOME INSTITUTE

Binning JGI Compute Infrastructure Requirements

🟧 🟦	Prototyping, exploratory analysis, small-scale production work
🟩	Large-scale data or compute needs (>100,000 CPU hours)

Challenges

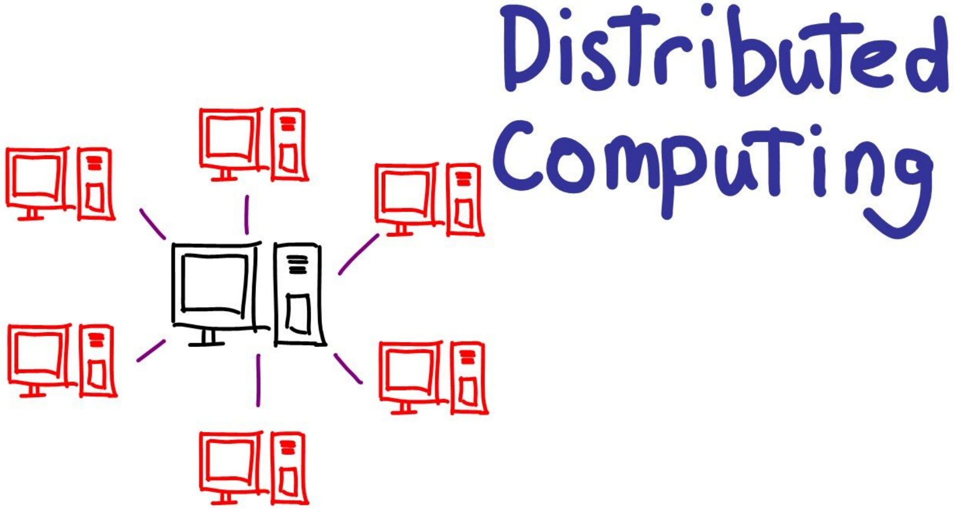
Distributed Computing is Hard

- **Managing multiple user accounts**

- **Different facilities have different policies**
  - Batch schedulers
  - File system availability and data retention
  - Run workflow in *scrontabs* it can be hard

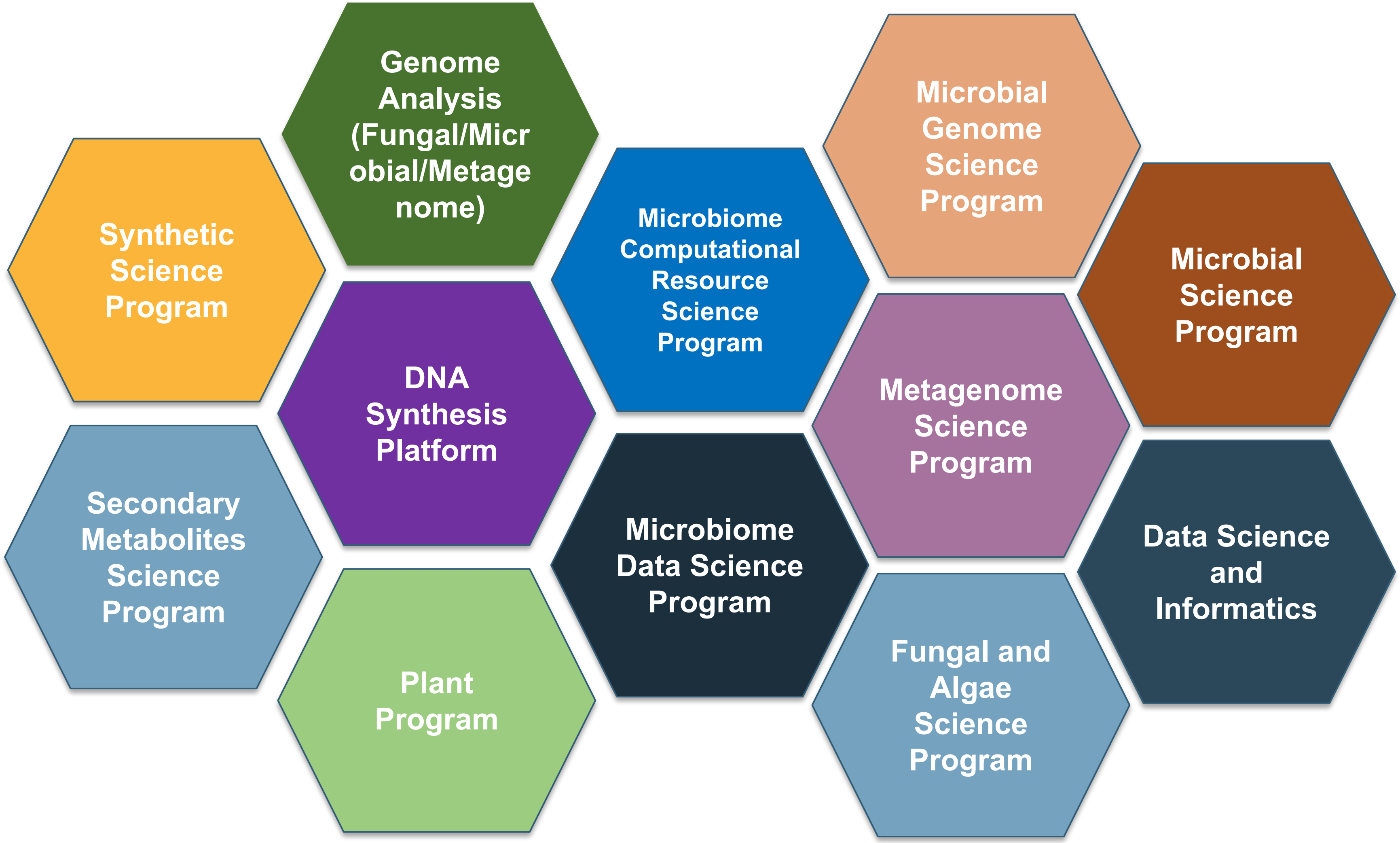
- **Different architectures**
  - CPU vs GPU
  - Local disk vs parallel file systems
  - Memory size and footprint

- **Portability is a lot of work**



Challenges

JGI Internal Workflow Development



Synthetic Science Program

Secondary Metabolites Science Program

Genome Analysis (Fungal/Microbial/Metagenome)

DNA Synthesis Platform

Plant Program

Microbiome Computational Resource Science Program

Microbiome Data Science Program

Microbial Genome Science Program

Metagenome Science Program

Fungal and Algae Science Program

Microbial Science Program

Data Science and Informatics

Challenges

JGI Internal Workflow Development

- **JGI Consists of many internal teams with specialized expertise in particular domains**

JGI artwork

Challenges

JGI Internal Workflow Development

- **JGI Consists of many internal teams with specialized expertise in particular domains**

- **Each team may have their own software development team, potentially spanning decades of development**

- **Teams use a variety of workflow software based on current developers' skills and preferences**

Dilemma: collection of workflows is hard to maintain, often idiosyncratic to specific developer, and may have dependencies that are no longer supported as time goes on (Python 2.x?). Organizationally this is poor use of resources.

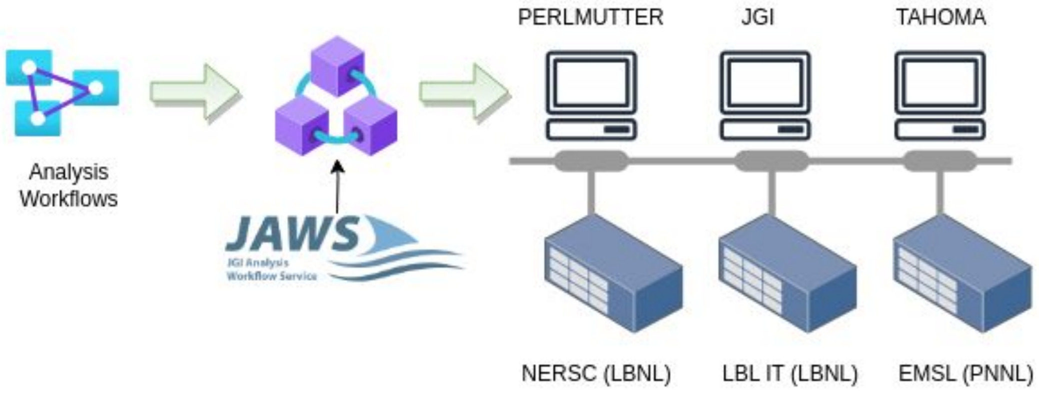
Solution: We need to standardize workflows!

JGI artwork

JGI Analysis Workflow Service - JAWS

JAWS: Unifying Workflows Across JGI Groups

- Developed a workflow manager called **JGI Analysis Workflow Service (JAWS)** to run complex computational workflows. JAWS provides support for distributed computation across multiple HPC sites.
- Provides a **user-friendly common interface** to seamlessly route jobs and data across multiple sites.
- Uses **Cromwell** to execute workflows in a common **Workflow Description Language (WDL**), standardizing the workflow language.
- Improves the **reusability** and **robustness** of bioinformatics workflows in evolving and/or diverse high-performance computing (HPC) and cloud environments.



SIMPLE to write, read, and edit

PORTABLE worflows

REPEATABLE analysis

SHARABLE workflows and resutls

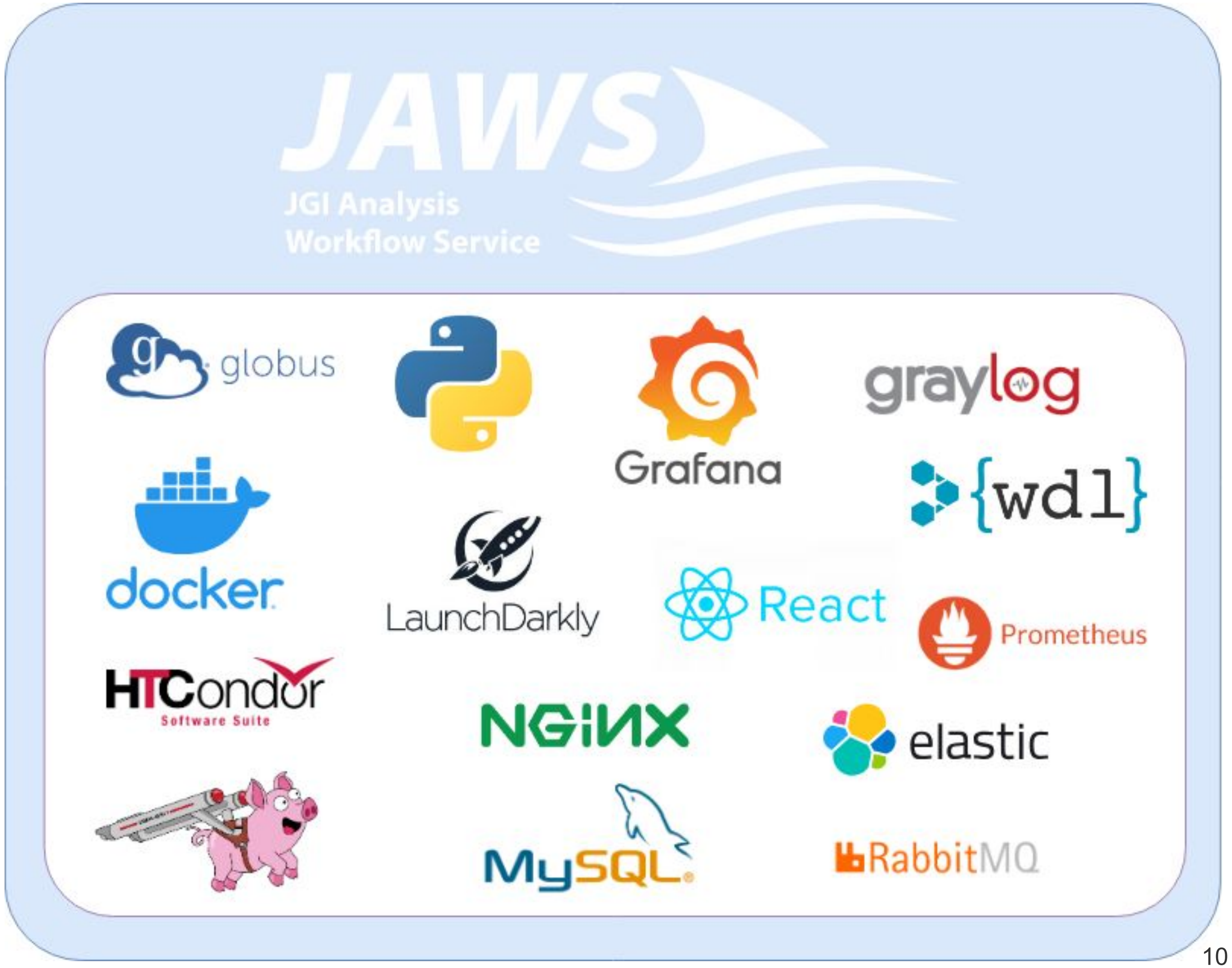
TRACEABLE results

Maintainable code

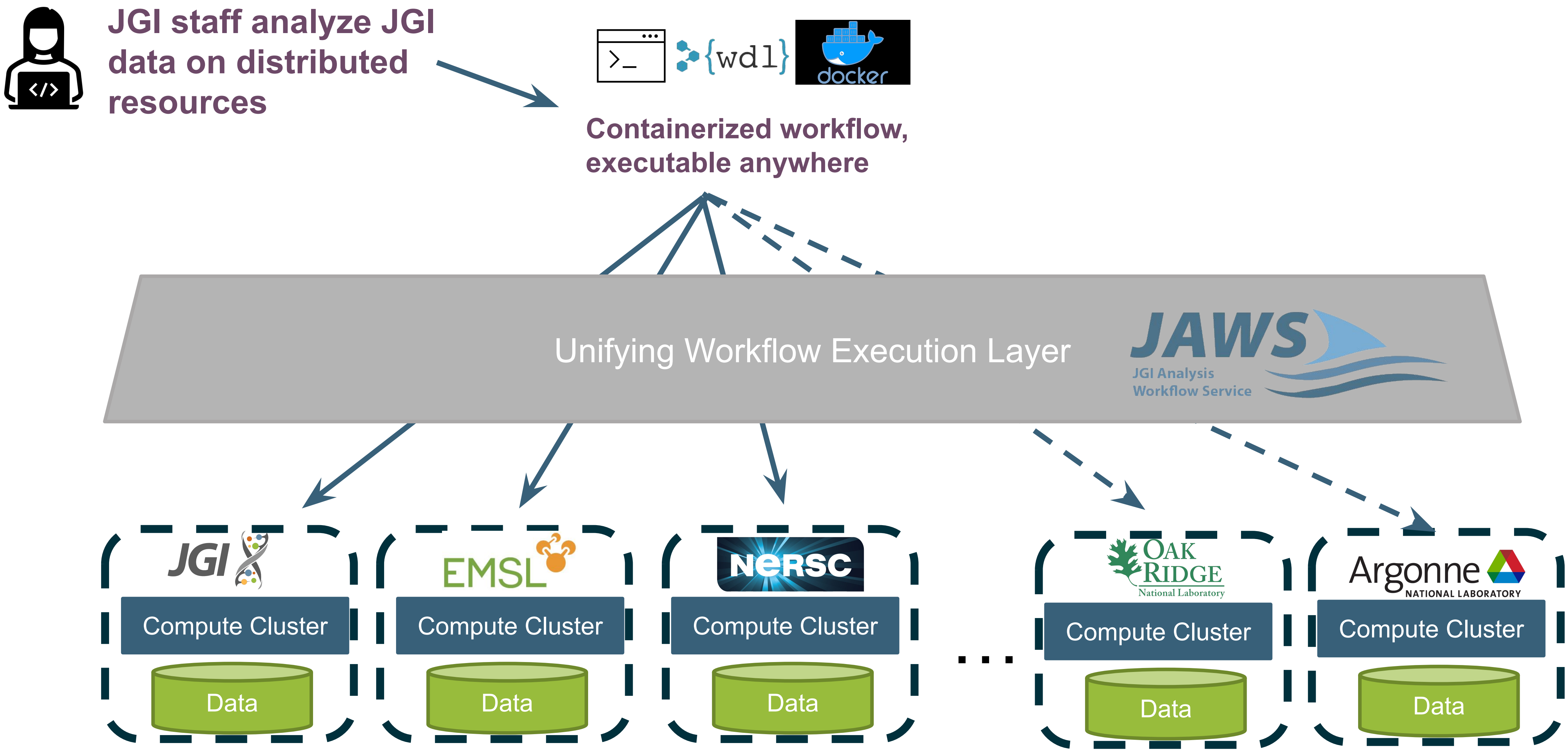
JGI Analysis Workflow Service - JAWS

Standards based, User Friendly, Multisite Distributed

- Built using existing tools that are well supported by community - avoids “NIH”
- Simplifies user access to distributed workflows
- Promotes reuse and collaborative workflow development



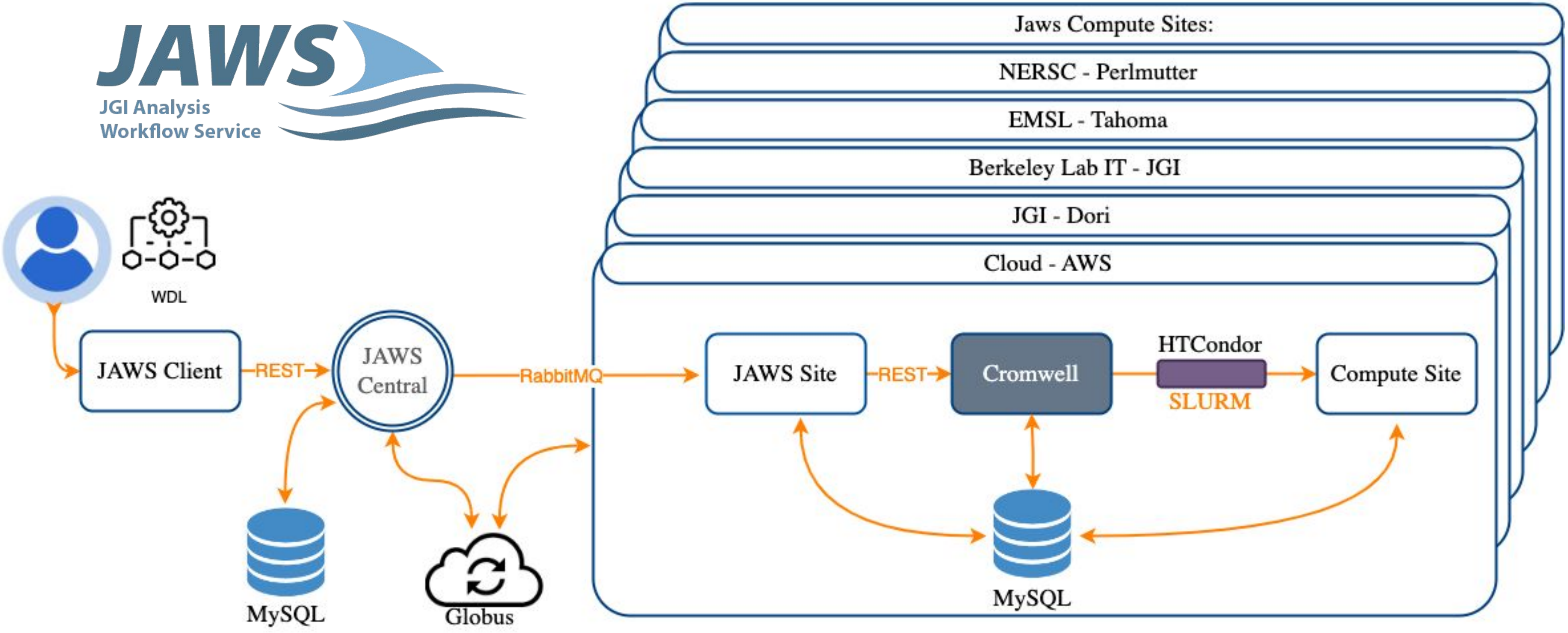
JAWS: Write Once, Run anywhere



JGI staff analyze JGI data on distributed resources

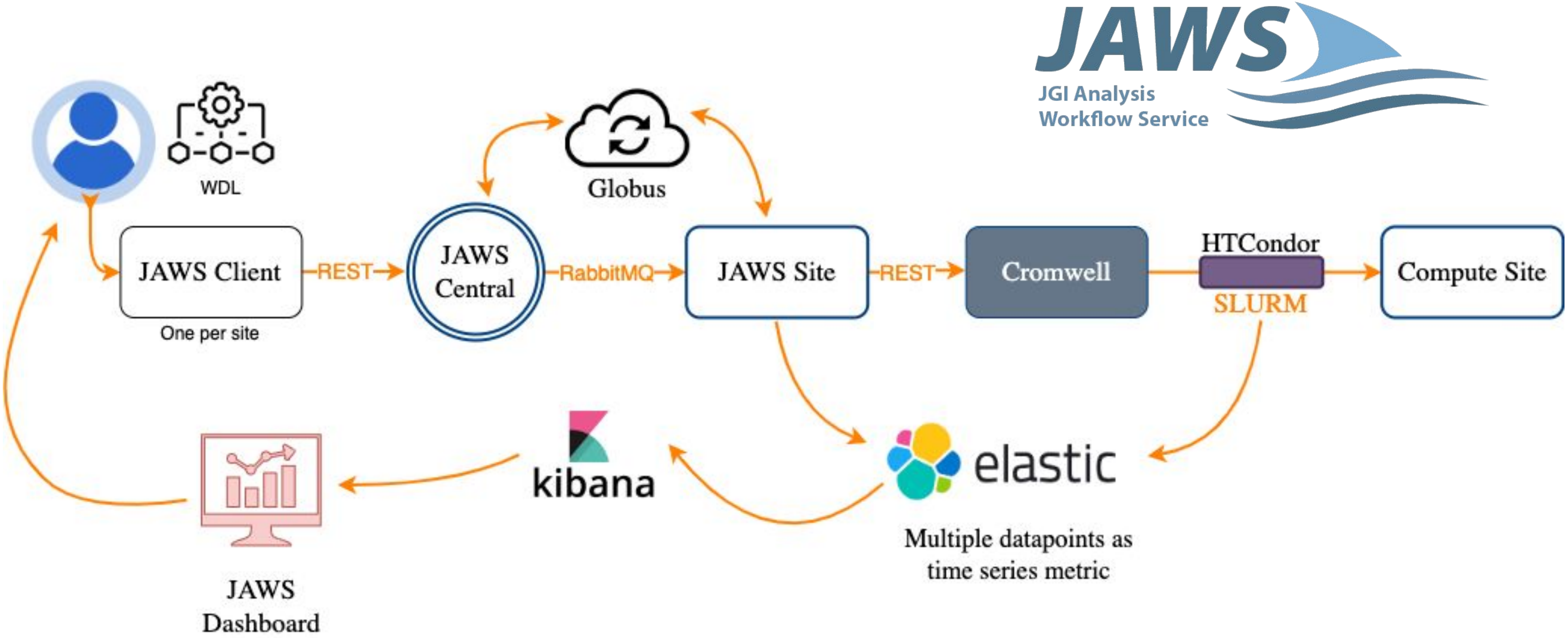
Containerized workflow, executable anywhere

JAWS: High Level Architecture



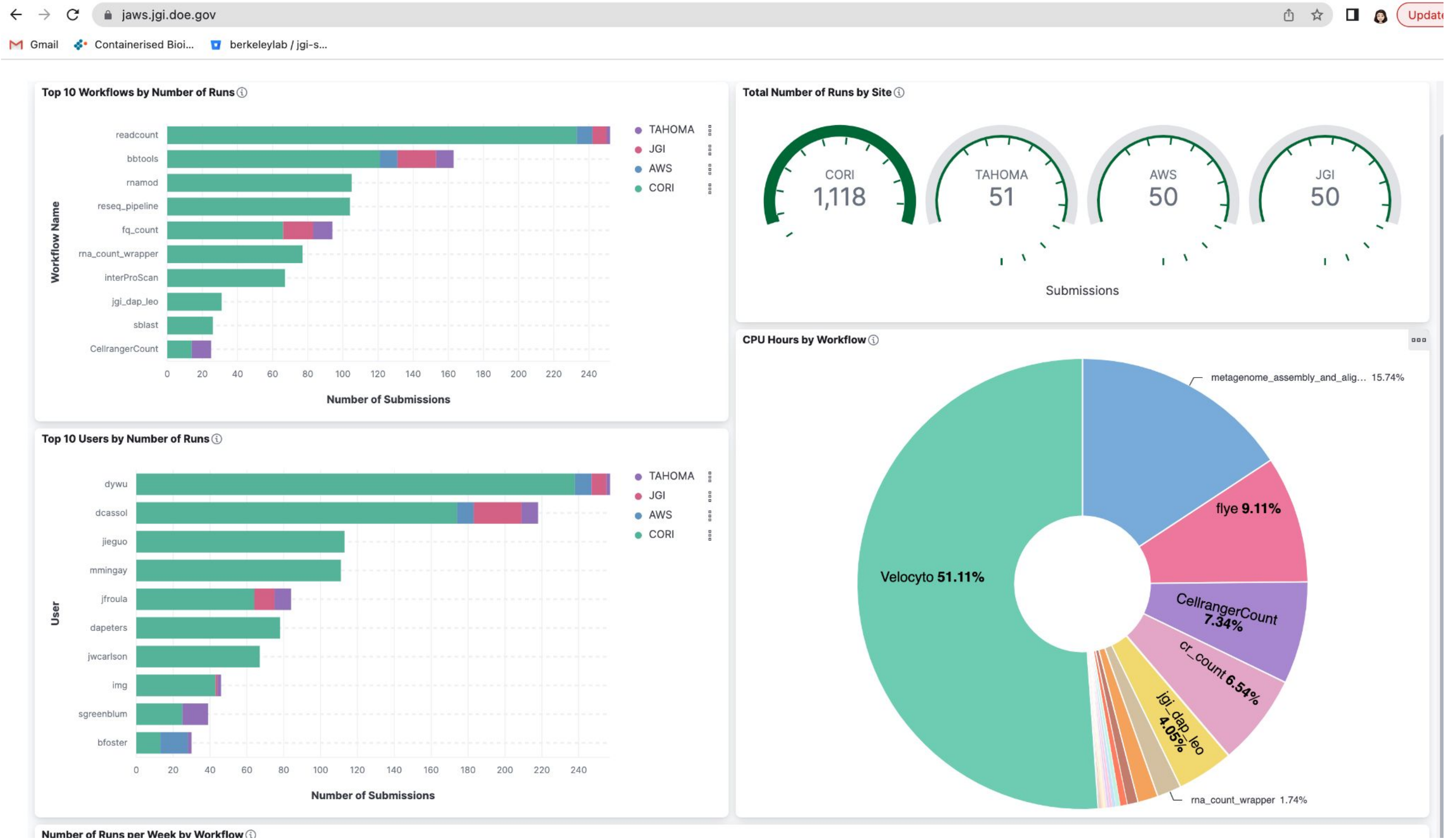
- **JAWS Client**: user-friendly command line interface with rich features
- **JAWS Central**: dispatches jobs to sites and tracks user history
- **JAWS Site**: extends Cromwell with additional features
- **Cromwell**: workflow execution engine
- **HTCondor**: flexible and efficient JAWS backend for SLURM

JAWS: Performance Metrics



Multiple datapoints as time series metric

JAWS Dashboard



Top 10 Workflows by Number of Runs

Total Number of Runs by Site

Top 10 Users by Number of Runs

CPU Hours by Workflow

Number of Runs per Week by Workflow

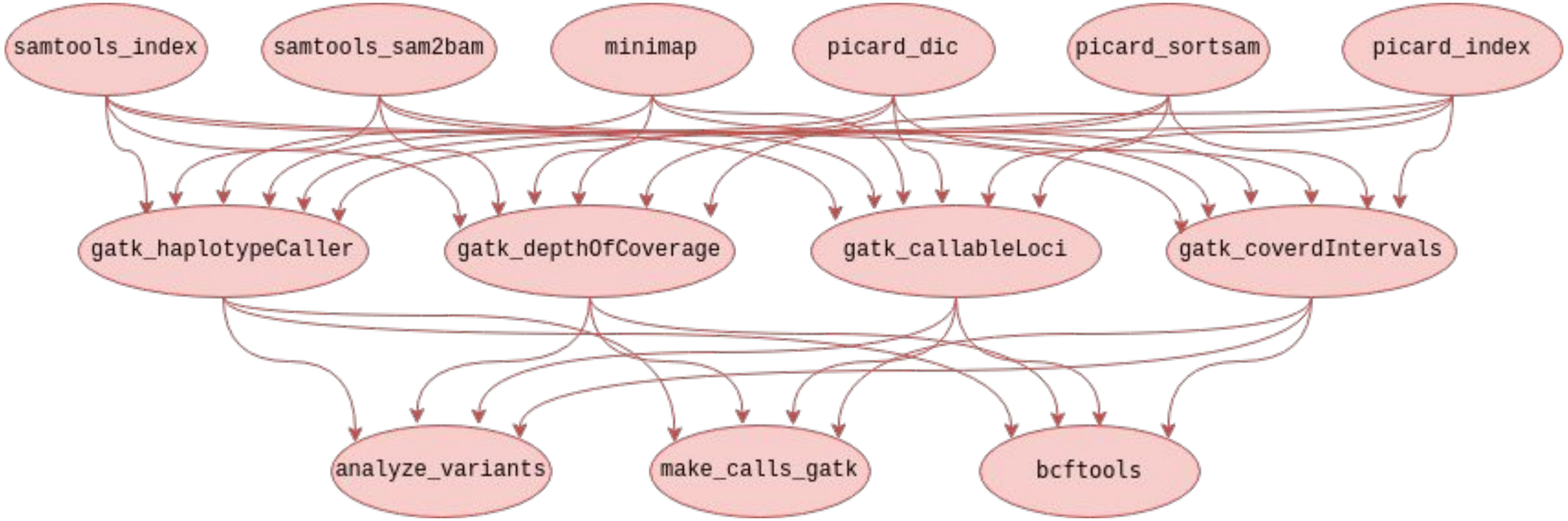
Why WDL? Why Containers?

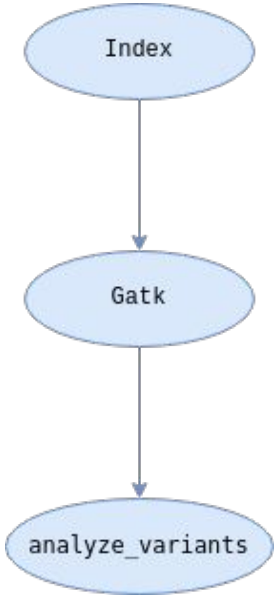
Widely supported, with active communities

- **BioWDL**
  - https://biowdl.github.io/
  - Many complete pipelines, as well as reusable WDL tasks for common CompBio analysis
- **BioContainers**
  - https://biocontainers.pro/
  - 9000+ applications for CompBio analysis
- **Terra.bio**
  - https://terra.bio/
  - Large cloud based workflow service provider that is based on WDL, Docker containers and Cromwell

User case: Migrating from Legacy Workflows to WDL

- Task parallelism involves distributing tasks across independent compute nodes, primarily when no data dependencies exist between tasks
- Example of sub-sub-workflow:





- Execution time: less than one minute/tasks
- I/O filesystem overhead
- > 17,000 tasks

- -71% shards/tasks
- -73% execution –> Reduce I/O filesystem overhead

Migrating from Legacy Workflows to WDL

- **Porting Legacy workflows to WDL - Execution time:**

Workflow	Legacy workflow	Using JAWS/WDL
Generate Reference Database (450M genes)	13 hrs ↓ **53%**	6 hrs (using large memory single node)
Horizontal Transfer (5M new genes)	2 hrs ↓ **35%**	1.3 hrs (single thread)
Phylogenetic Distribution (5M new genes)	6.5 hrs ↓ **80%**	1.3 hrs (using 10 shards)

- **Improving Existing Workflows - Using `/tmp` for some I/O intensive tasks:**

Workflow	% completed before	% completed after
DAP-seq	76%	99%

JAWS Support

JAWS has recently been adding new users + sites

- **Larger user community in general with different needs**
  - Growing user community - no longer “WDL early adopters”
  - Regular JAWS office hours for immediate support
  - Periodic hackathons for intensive training and knowledge transfer
  - Developing a JGI user community around WDL and Containers so that JAWS team is not the sole source of support
  - Personalized paired programming engagements to help WDL newcomers come up to speed and migrate legacy workflows
  - Bi-weekly JAWS User Meetings for community updates and feedback

JAWS Future

Current and Future Work

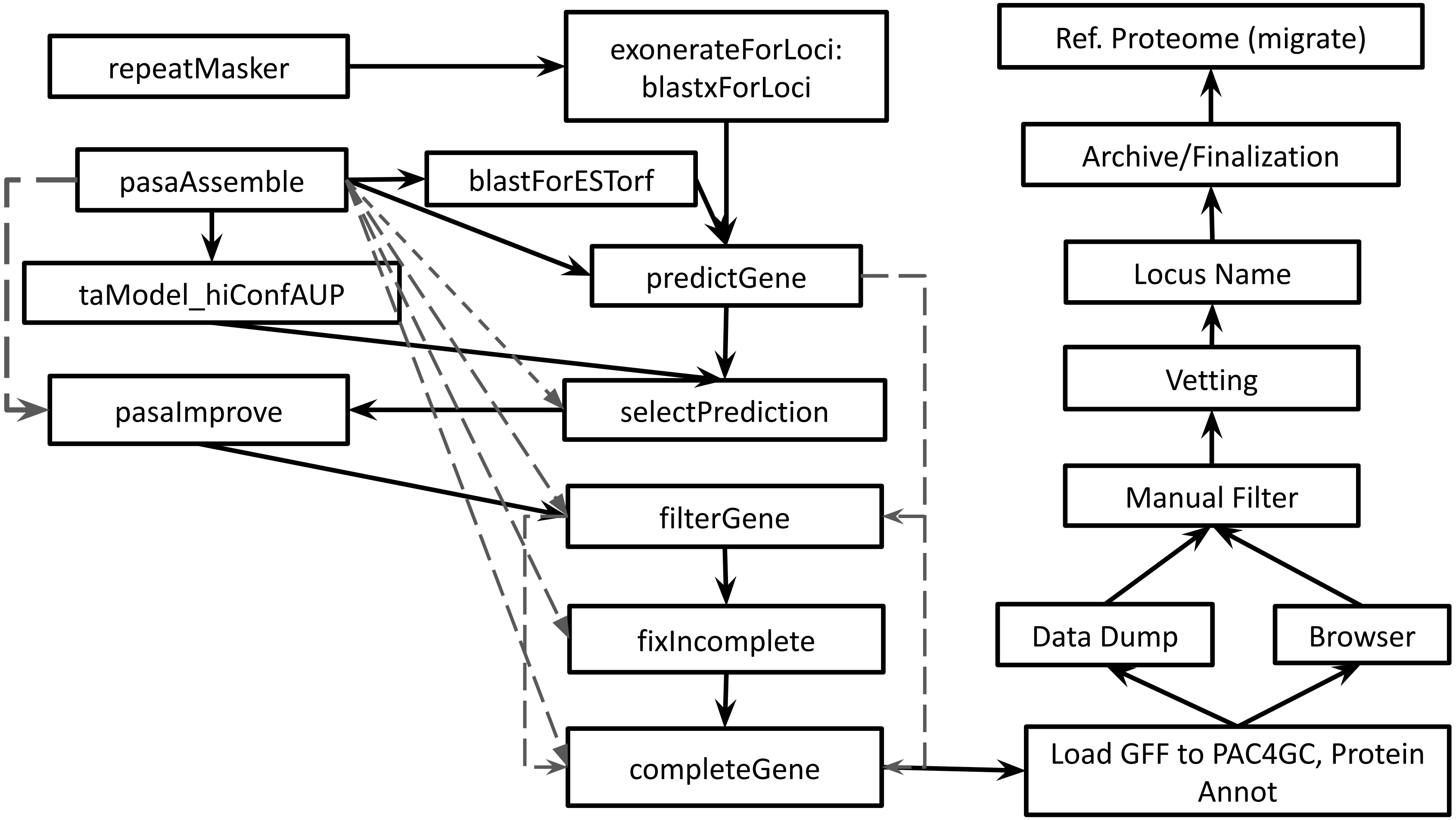
- Redesign of Performance Metrics and Monitoring system
- Deploying JAWS to ORNL and ANL
- NMDC has consolidated their workflow efforts on JAWS
  - Additional requirements for programmatic access (JAWS API)
  - JAWS-NMDC Tahoma Site
- Automatic routing of incoming jobs
- Support for requesting
  - GPU nodes
- Increase resilience by migrating core JAWS services to cloud
- Expand metrics collection and visualization capabilities
- Closer integration with JGI’s JAMO so that data can be directly pulled from/added to JAMO from workflows
- RQC integration

JGI JOINT GENOME INSTITUTE
A DOE OFFICE OF SCIENCE USER FACILITY

Thank you!

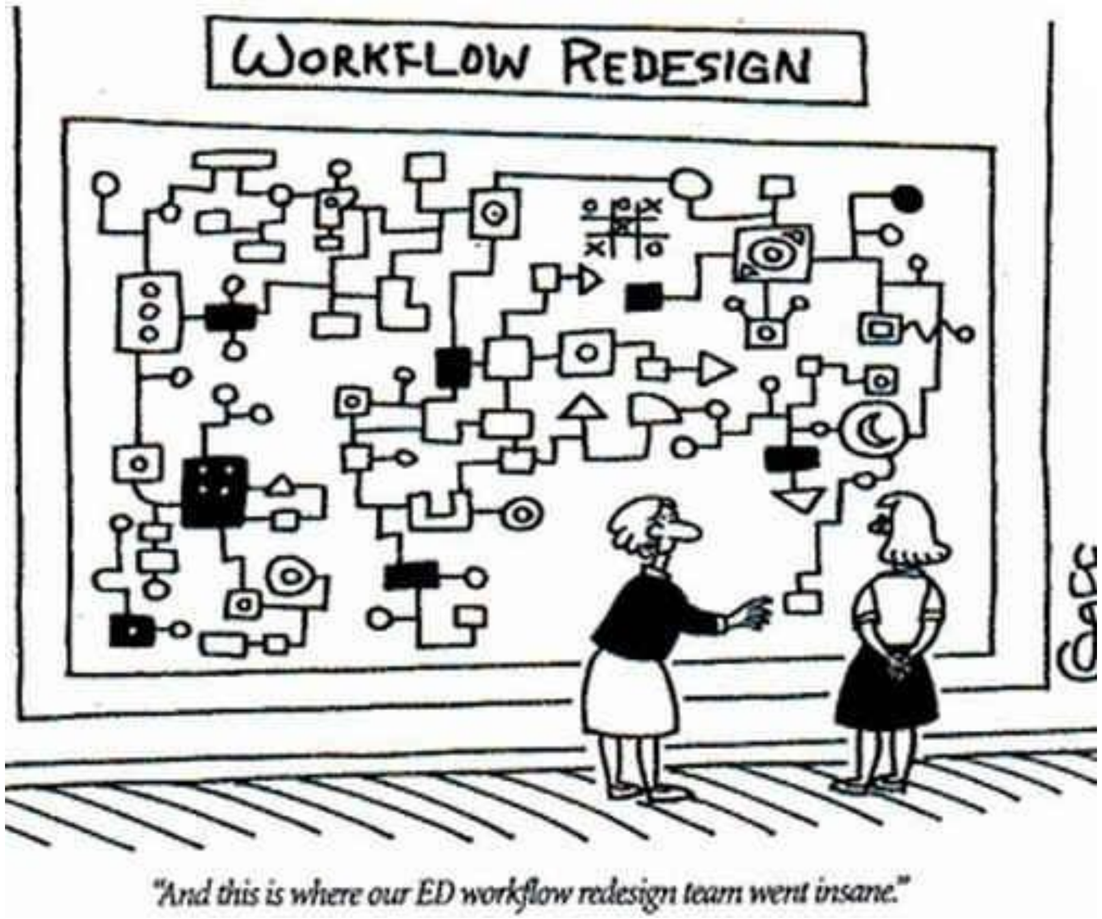
High-level overview of IGC

- **IGC was currently implement in SAPS (Sequence Analysis Pipeline System)**
  - In-house workflow definition language and workflow manager
  - Developed over the past 15 years

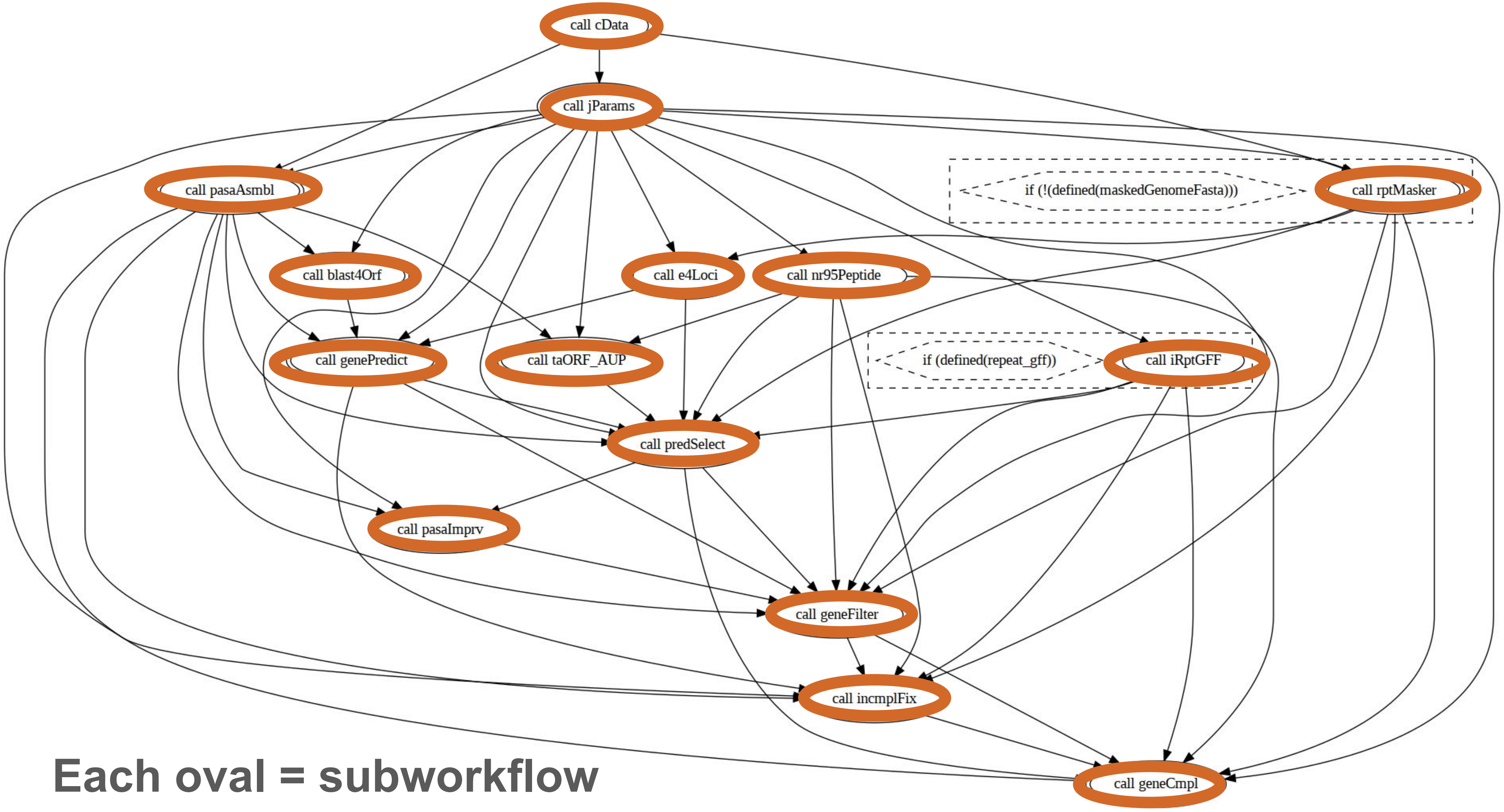


High-level overview of IGC

- **IGC was currently implement in SAPS (Sequence Analysis Pipeline System)**
  - In-house workflow definition language and workflow manager
  - Developed over the past 15 years

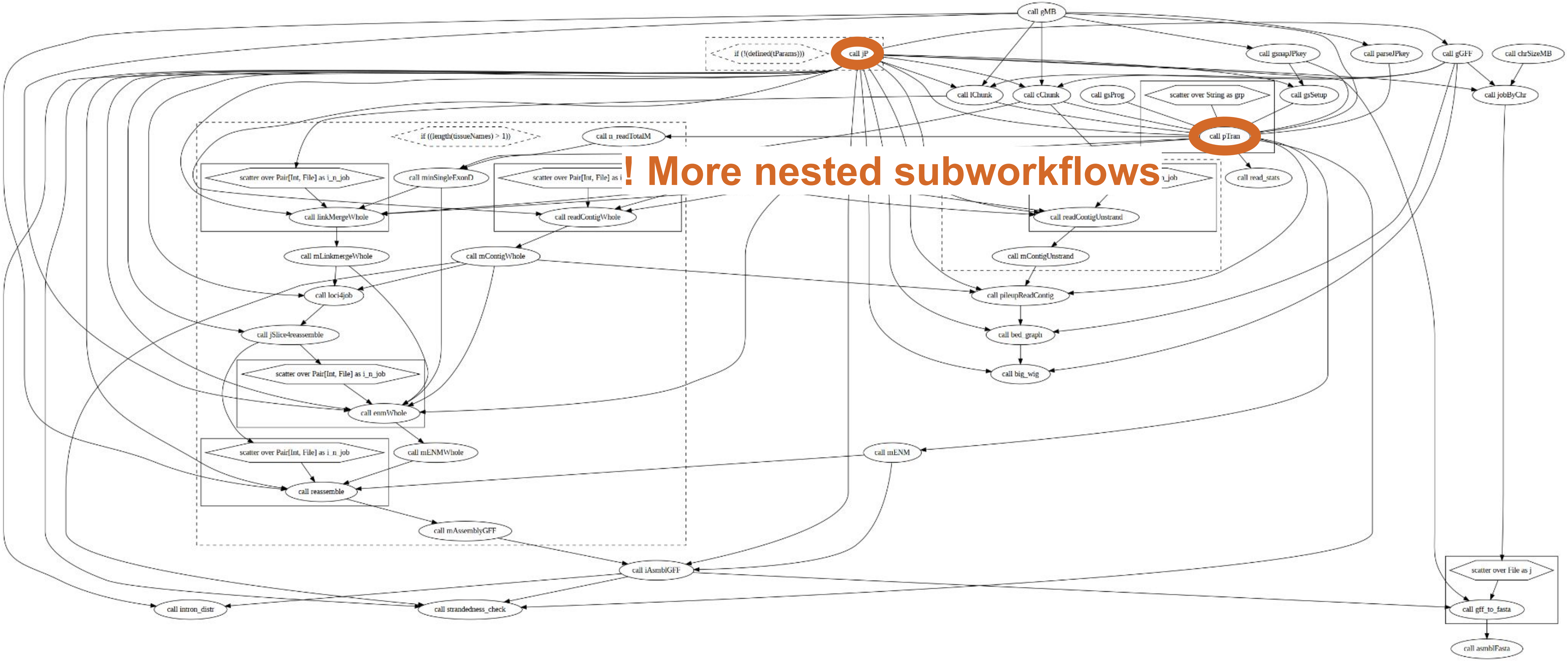

WORKFLOW REDESIGN
"And this is where our ED workflow redesign team went insane."

Workflow Example



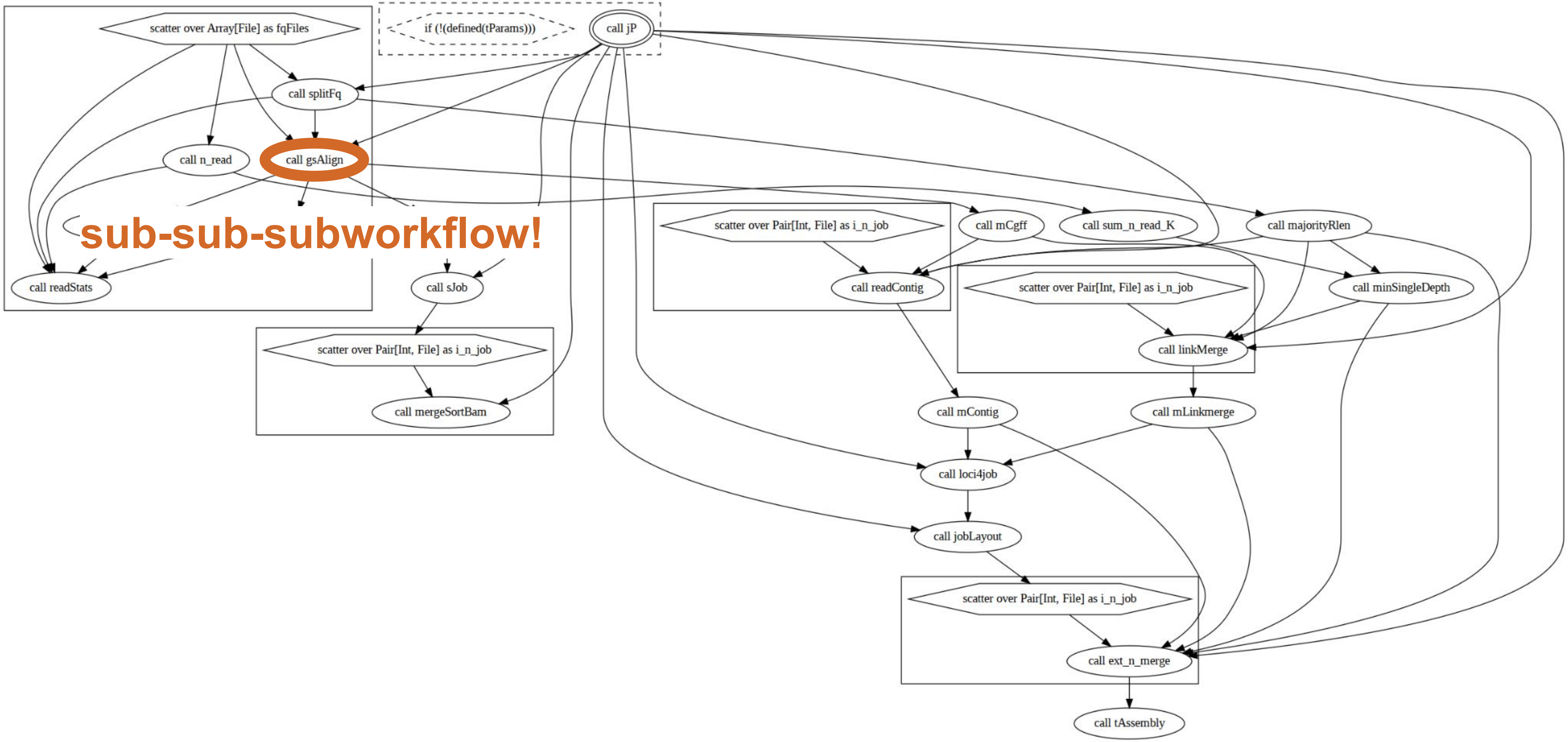
Each oval = subworkflow

A sample subworkflow



! More nested subworkflows

A sub-subworkflow



scatter over Array[File] as fqFiles
if (!(defined(tParams)))
call jP
call splitFq
call n_read
call gsAlign
sub-sub-subworkflow!
call readStats
call sJob
scatter over Pair[Int, File] as i_n_job
call mergeSortBam
scatter over Pair[Int, File] as i_n_job
call readContig
call mCgff
call sum_n_read_K
call majorityRlen
scatter over Pair[Int, File] as i_n_job
call minSingleDepth
call linkMerge
call mContig
call mLinkmerge
call loci4job
call jobLayout
scatter over Pair[Int, File] as i_n_job
call ext_n_merge
call tAssembly