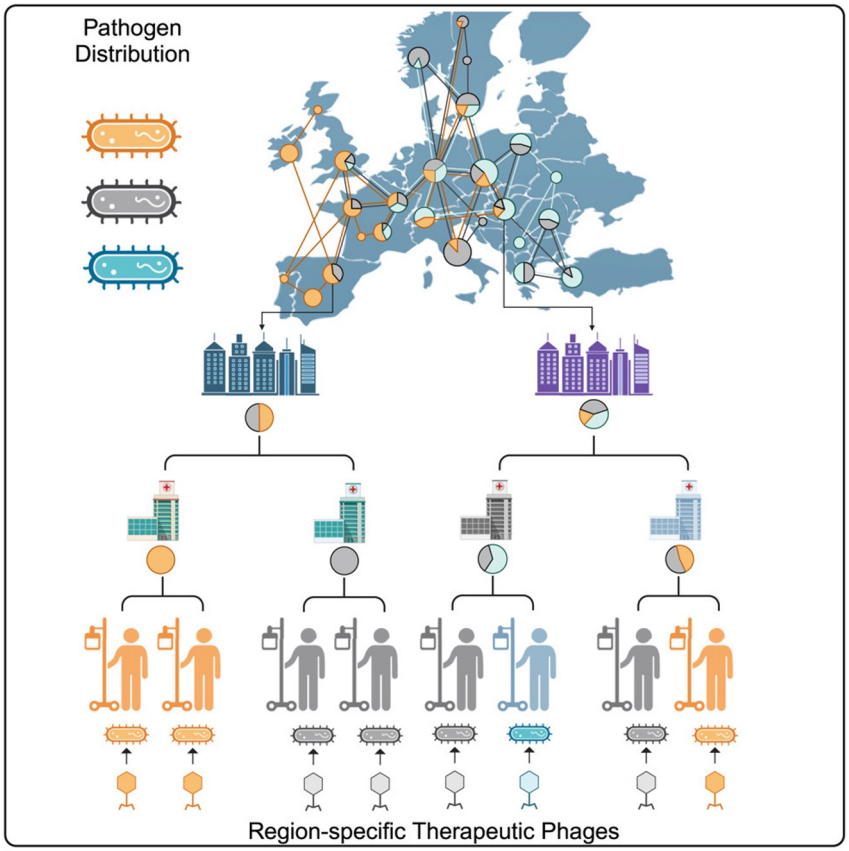
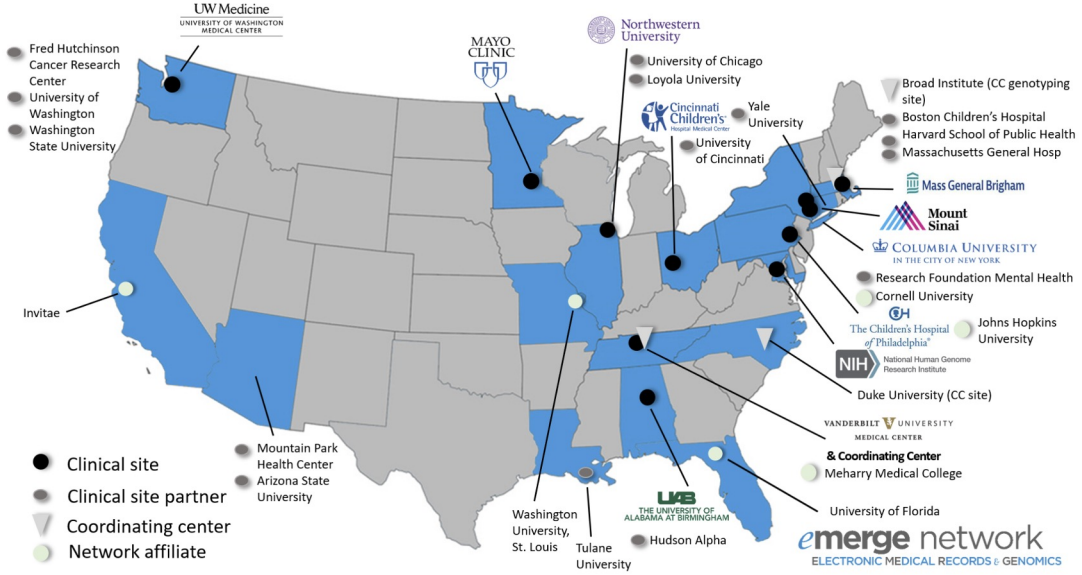
SASW, Secure and Automated Scientific Workflow

Susmit Shannigrahi, Manas Das, Lixia Zhang

Scientific Workflows are Increasingly Collaborative and Distributed

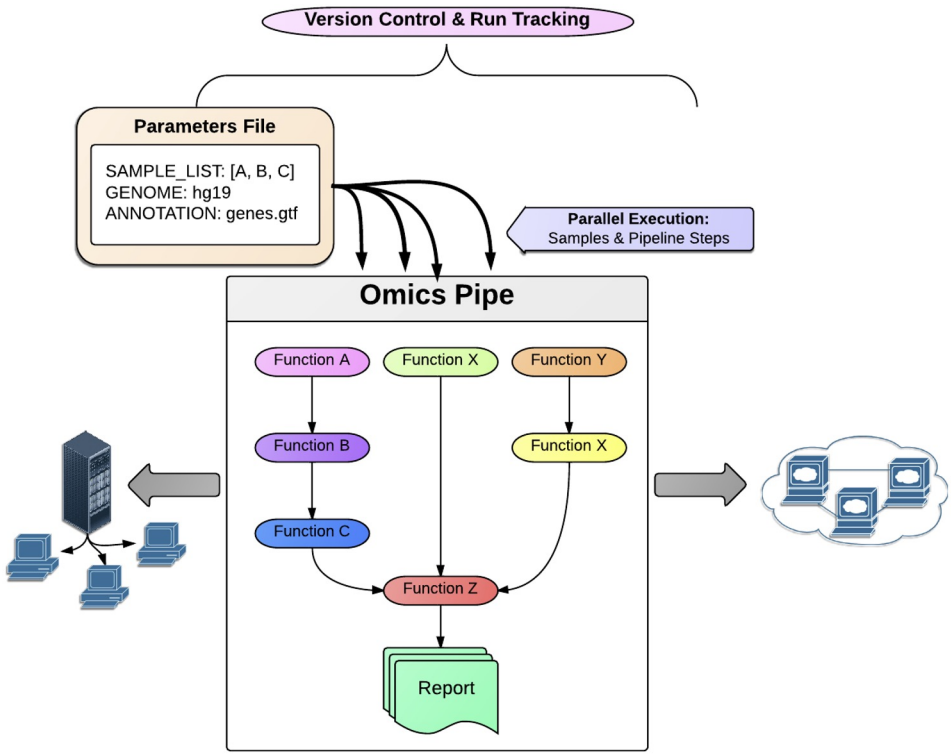


https://www.cell.com/cell/fulltext/S0092-8674%2824%2901027-4



https://emerge-network.org/emerge-sites/

A Sample multi-omics genomics workflow



https://www.biorxiv.org/content/biorxiv/early/2014/08/23/008383.full.pdf

A Distributed Approach Comes with Interesting Challenges

Geographically distributed computation placement

Matching computations with resource availability

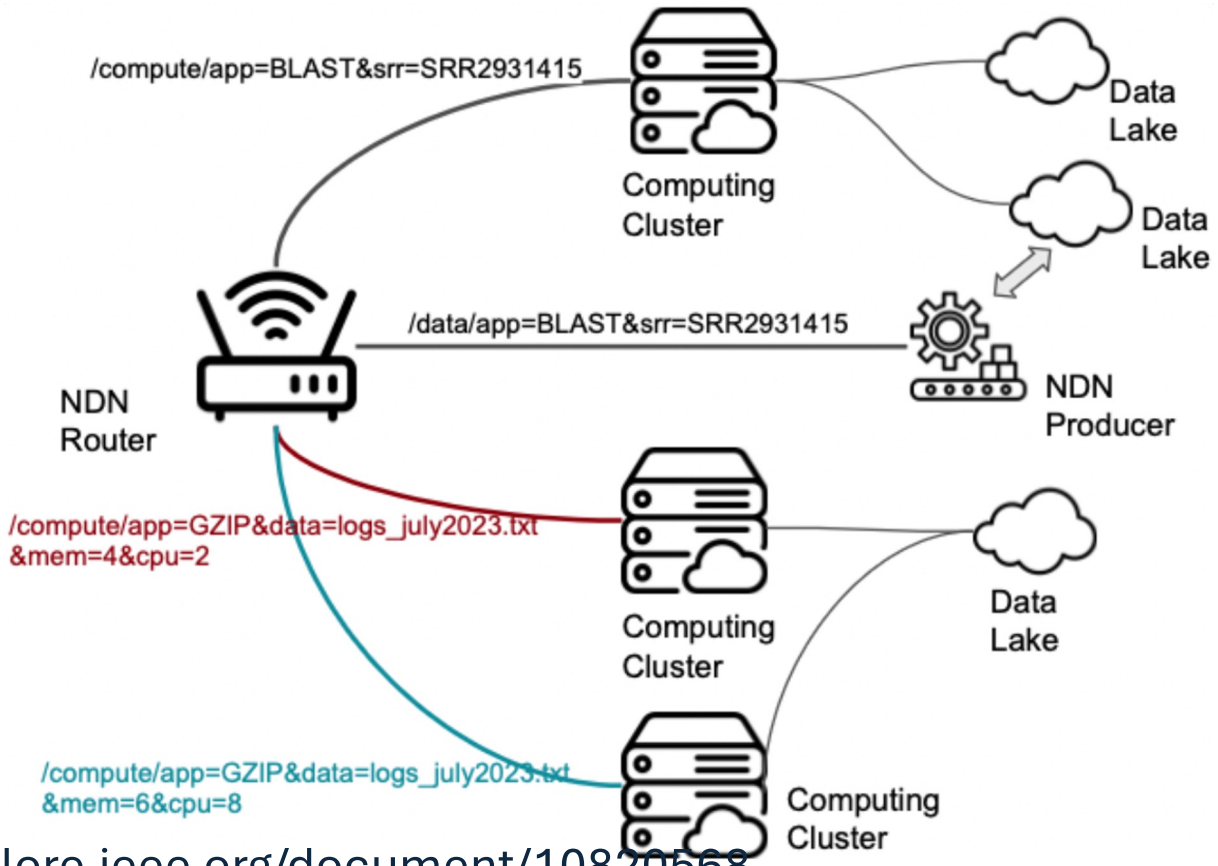
Local policy and security

Complex coordination across sites

Perfect for decentralization

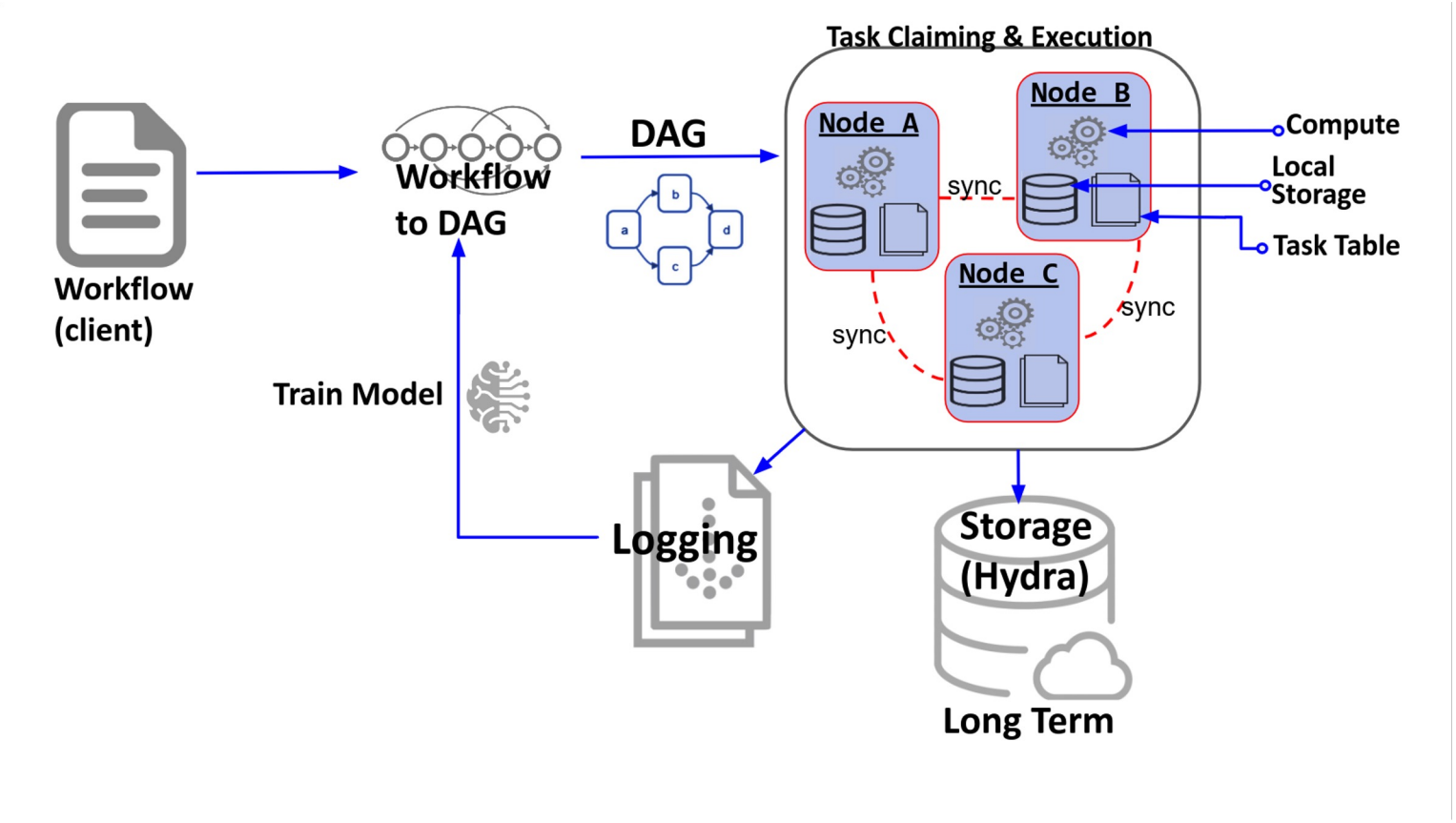
- Traditional workflow systems rely on centralized controllers
- Centralized bottlenecks restrict scalability, fault tolerance, and flexibility
- Multi-institutional workflows require autonomy
- Challenge: Enable coordination *without* centralized orchestration

Prior Work: Semantic Naming for Computation Tasks



https://ieeexplore.ieee.org/document/10820568

System overview - Workflow to DAG

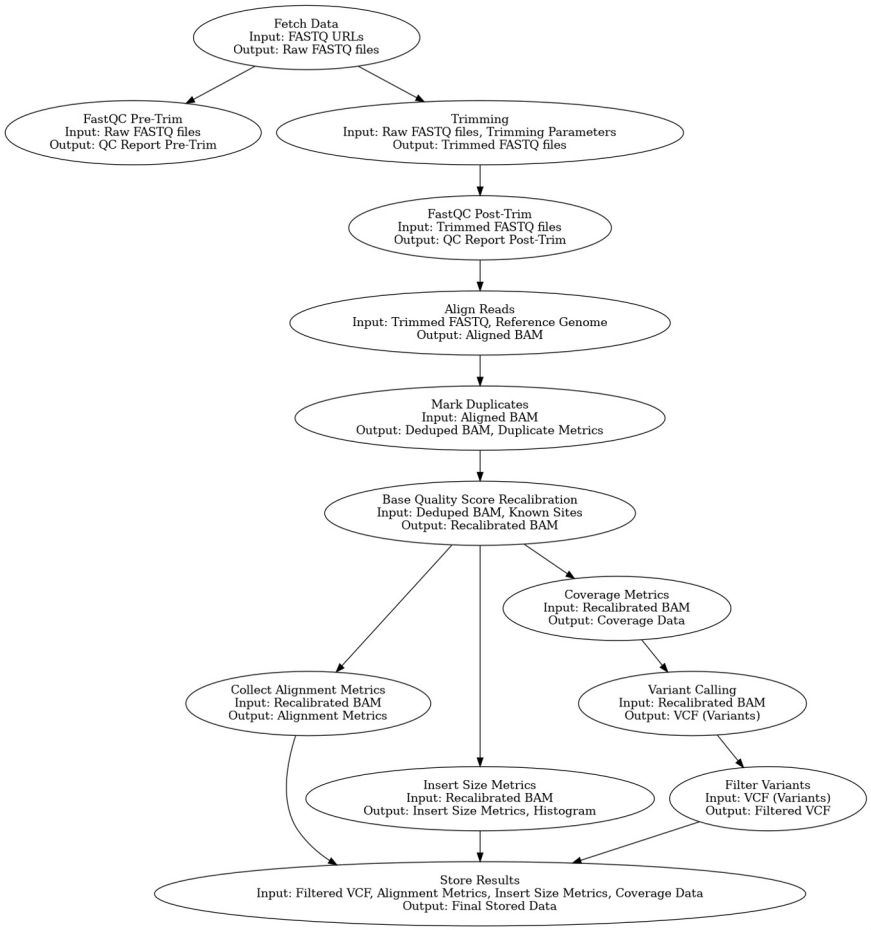


SASW Core Architecture: Built for Decentralized Execution

- Workflows broken into tasks (DAG model)
  - Each task has:
    - Semantic input/output names
    - Resource requirements
    - Execution metadata
  - Nodes learn of tasks via synced task table
  - Nodes claim tasks based on local resource state
  - Execution proceeds as data becomes available—no centralized trigger

What does such a DAG look like?

- Workflows can be mapped to a DAG



The other aspects: Nodes and Security

- **Federated Nodes:** Nodes can belong to different organizations; no central control.
- **Trust Bootstrapping:** Each entity gets a certificate from a domain controller (trust anchor).
- **Semantic Naming:** Entities and data are named uniquely and meaningfully (e.g., /sasw.domain/user/alice).
- **Authenticated Identity:** Workers use DNS; users use email for identity verification.
- **Data-Centric Security:** Tasks and data are signed/encrypted using named public keys (e.g., /entity/KEY/<id>).

Decentralized Mechanisms in Action

Task Dissemination
No global queue

all nodes see the same DAG/task table

Task Claiming
Nodes independently match tasks to available resources

Competing nodes resolve via local policy

Data Publication
Tasks publish output under semantic names

Results flow to next eligible tasks in DAG

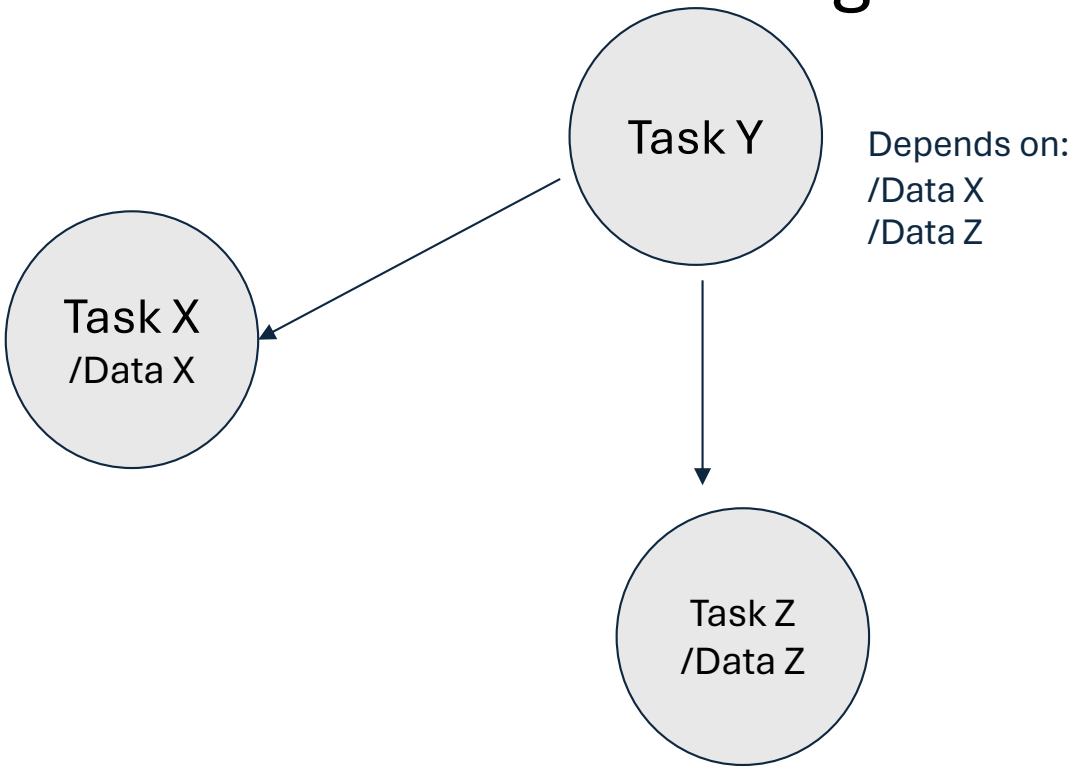
Next-stage nodes discover data, trigger task claim

Security Without Central Trust
All data is signed and semantically named

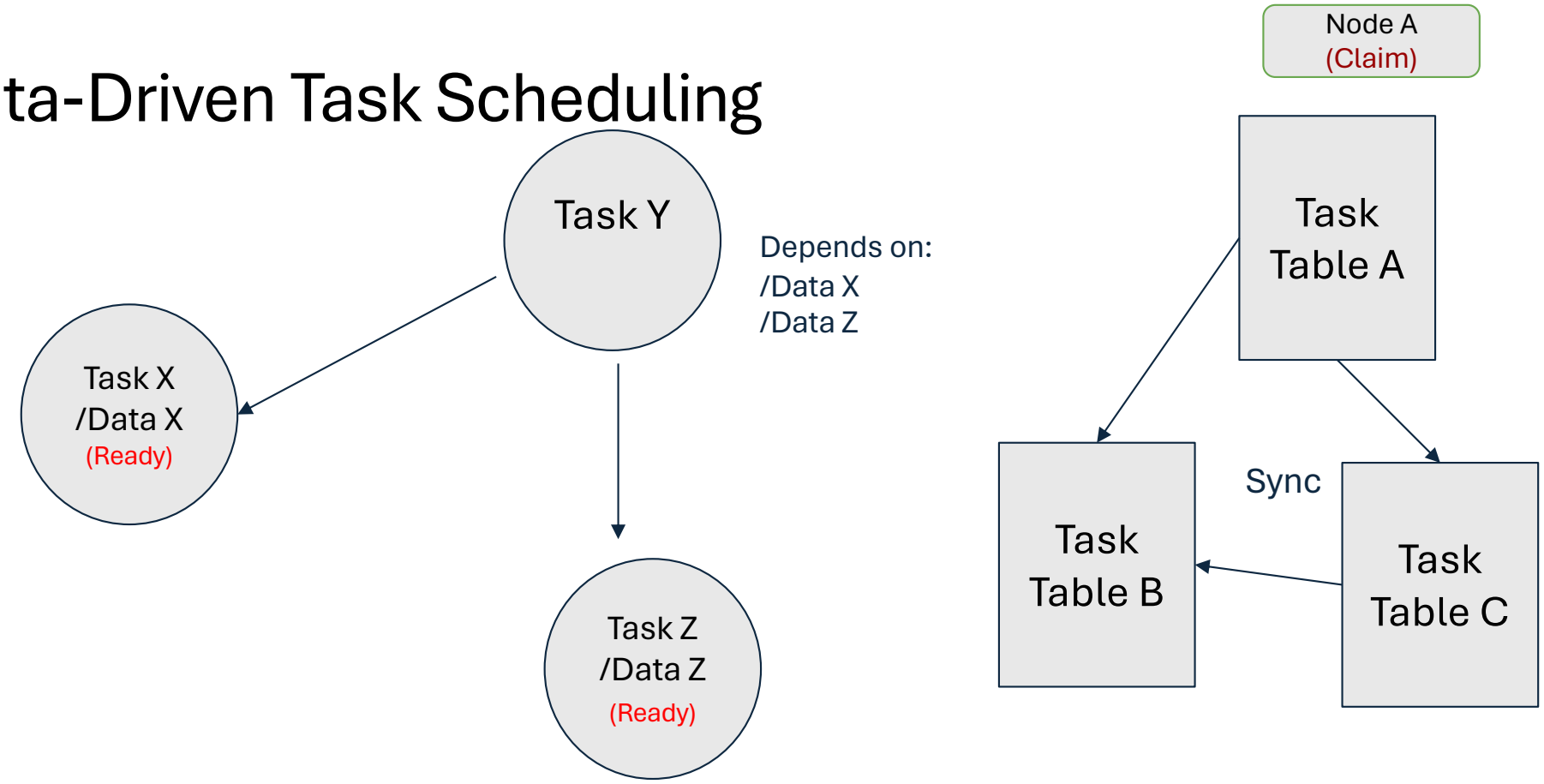
Access policies enforced per-node, not per-system

No centralized policy enforcement needed

Data-Driven Task Scheduling

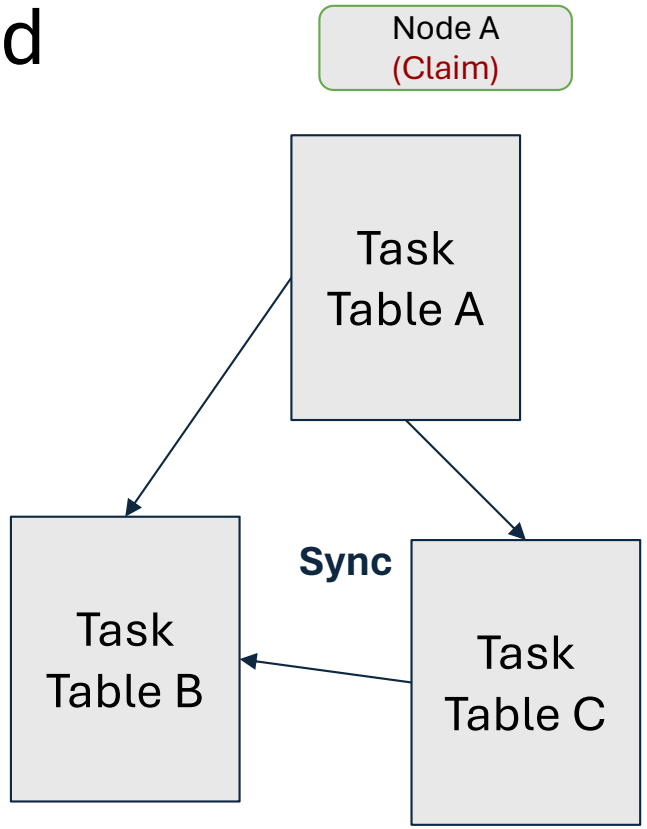


Data-Driven Task Scheduling

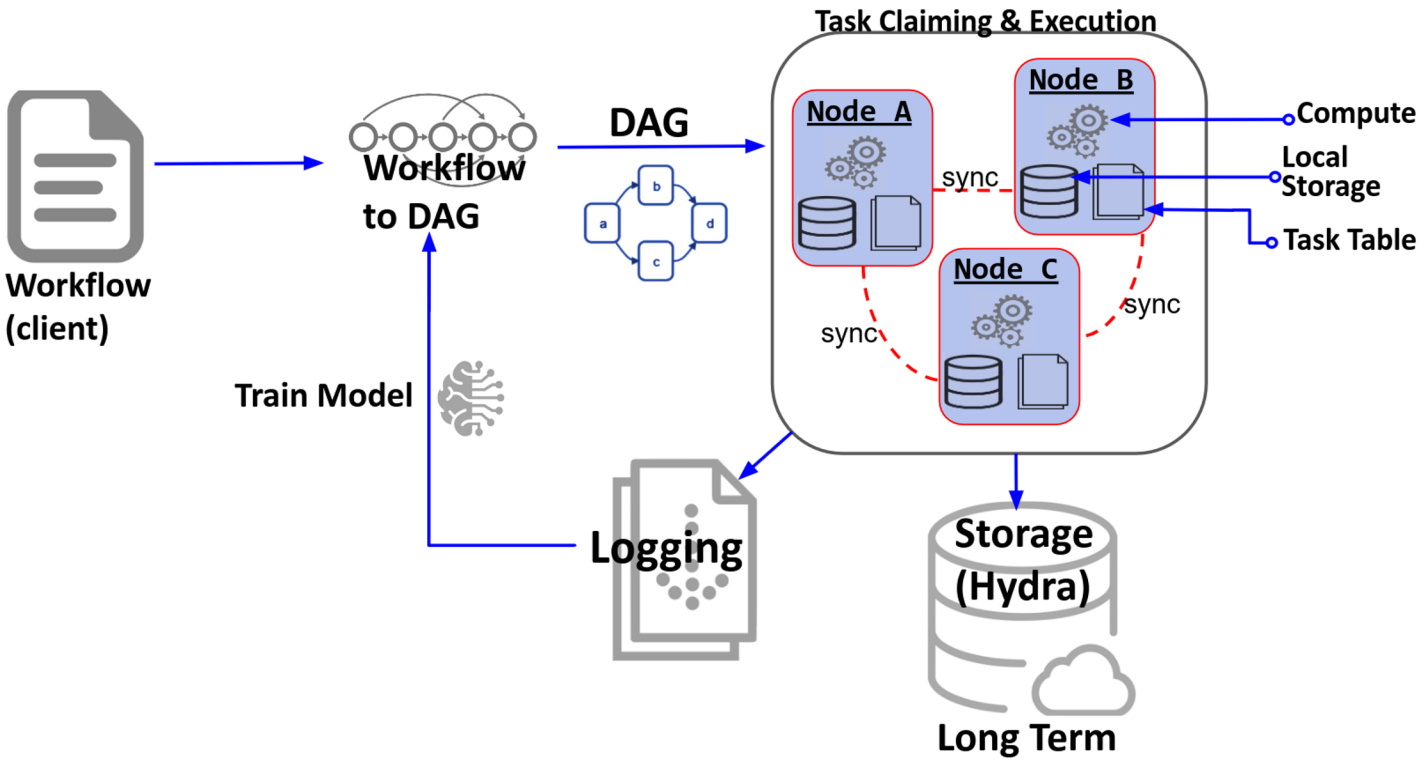


This is where we can do intelligent and decentralized scheduling

- Nearest
- Optimized
- Policy based
  - Institutional/geographical constraints
  - resource constraints



Once Results are available, name and store them



Challenges and Research Opportunities

Efficient decentralized task dissemination

Handling conflicting task claims

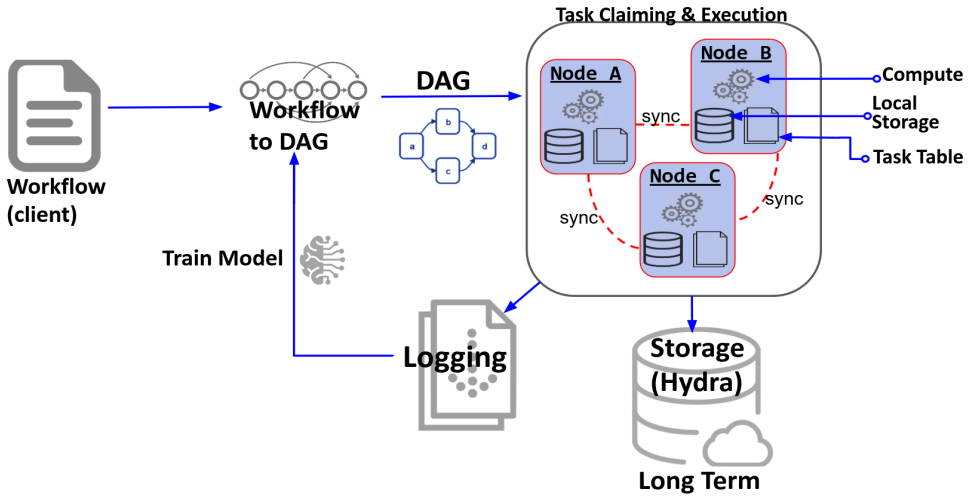
Renders well to multi-objective optimization problems

Monitoring and observability without global control

Federated debugging and logging

How do we provide safety and liveness in such a system?

Conclusion



- SASW is a step toward **truly decentralized scientific computing**
- No schedulers, no controllers—just tasks, data, and nodes
- Enables flexible, secure, large-scale scientific workflows
- Aligns with real-world scientific collaboration needs

Thank you!