Future Galaxy developments for distributed data access

INCOB 2019 Breakout
Thursday, 12 September, 11am-12pm

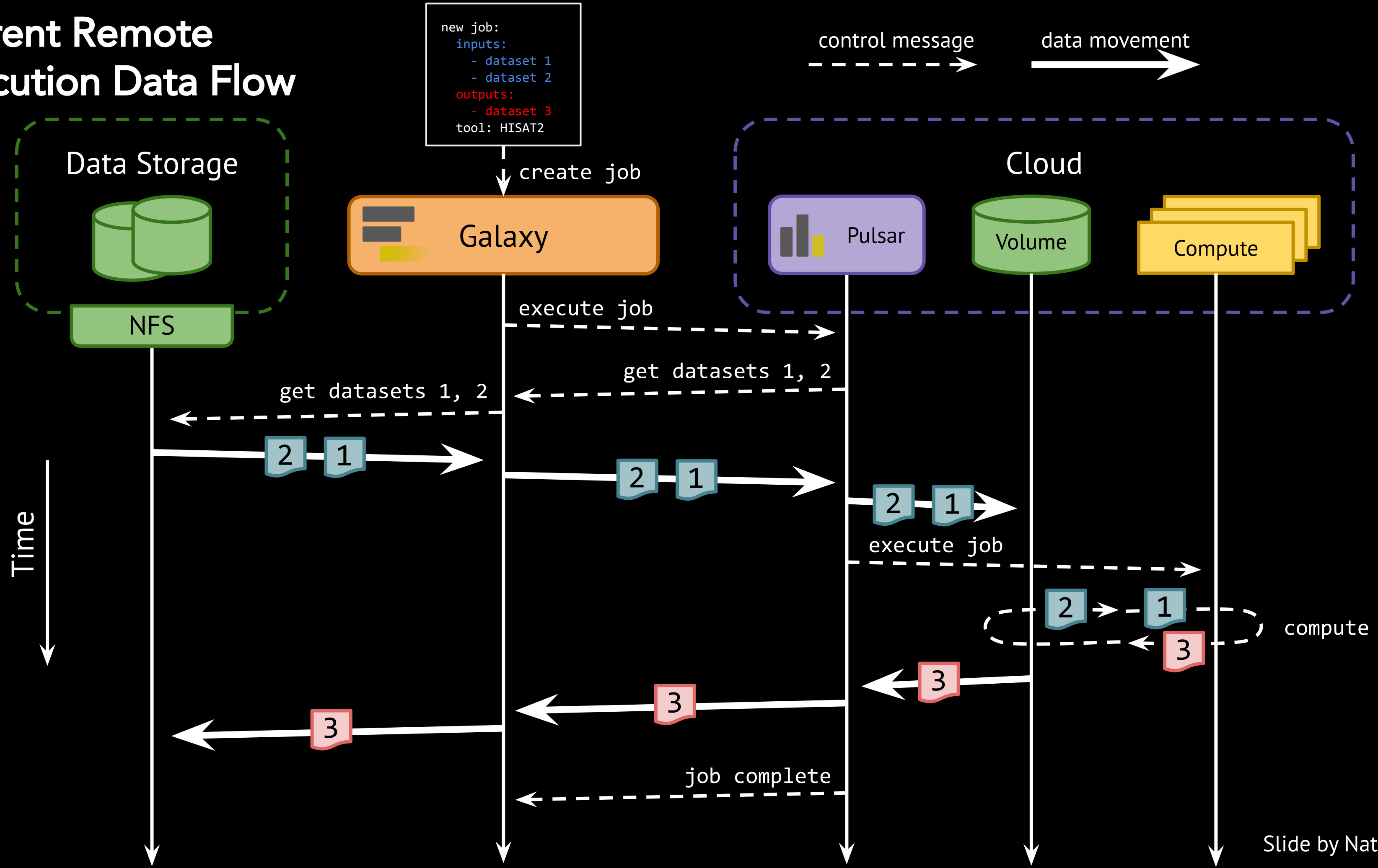
Nuwan Goonasekera
Galaxy Australia
University of Melbourne
@nuwang

bit.ly/gxy-incob-2019 #UseGalaxy

The current state of Galaxy data storage

1. Data is stored centrally on a shared file system (e.g., NFS)
2. Expensive to move around
3. Yet we do to utilize remote computing resources
4. Not a scalable solution, increasingly large and immovable datasets
5. Reference data is solved, thanks to CVMFS

Current Remote Execution Data Flow



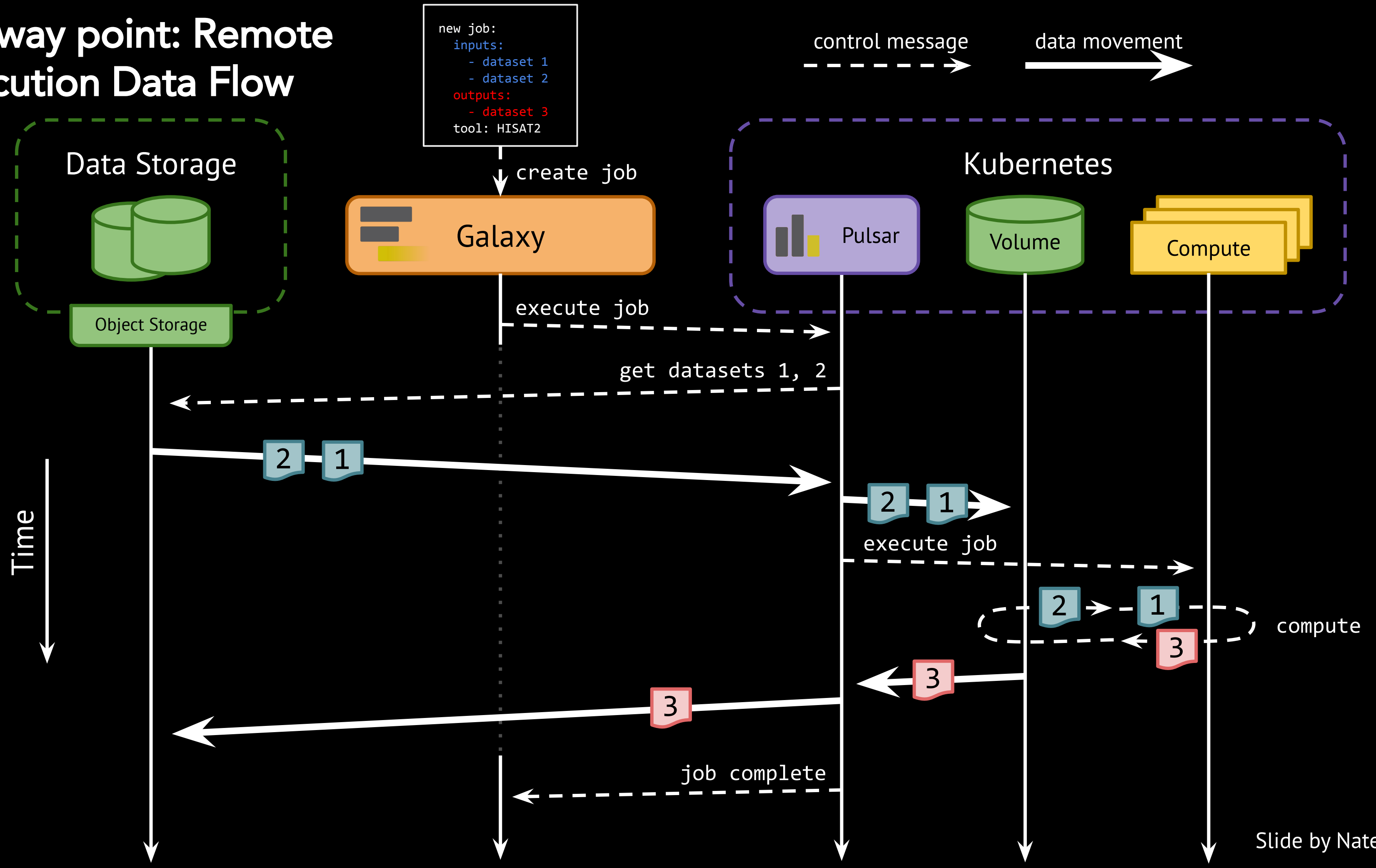
```
new job:
  inputs:
    - dataset 1
    - dataset 2
  outputs:
    - dataset 3
  tool: HISAT2
```

Slide by Nate Coraor

Where we want to get to

1. Store user data in object storage (like Swift or S3)
2. Remove shared file system - single point of failure, potential bottleneck, not geographically distributed
3. Move computation close to the data. Let local compute fetch data - provide federated view of compute and data.
4. Federated authentication - allowing individual institutional login, but centralized control

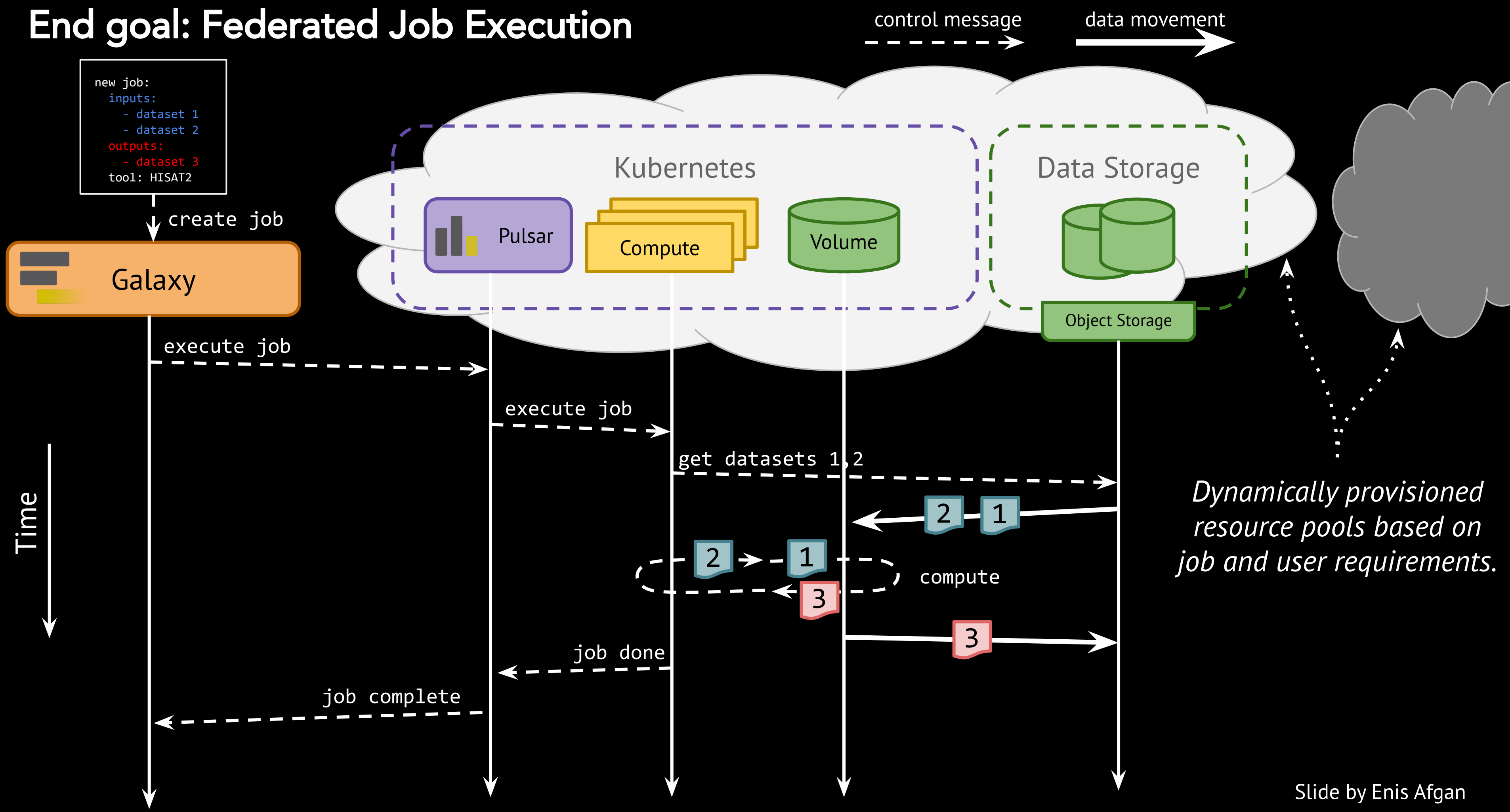
Halfway point: Remote Execution Data Flow



```
new job:
  inputs:
    - dataset 1
    - dataset 2
  outputs:
    - dataset 3
  tool: HISAT2
```

Slide by Nate Coraor

End goal: Federated Job Execution



```
new job:
  inputs:
    - dataset 1
    - dataset 2
  outputs:
    - dataset 3
  tool: HISAT2
```

Dynamically provisioned resource pools based on job and user requirements.

Slide by Enis Afgan

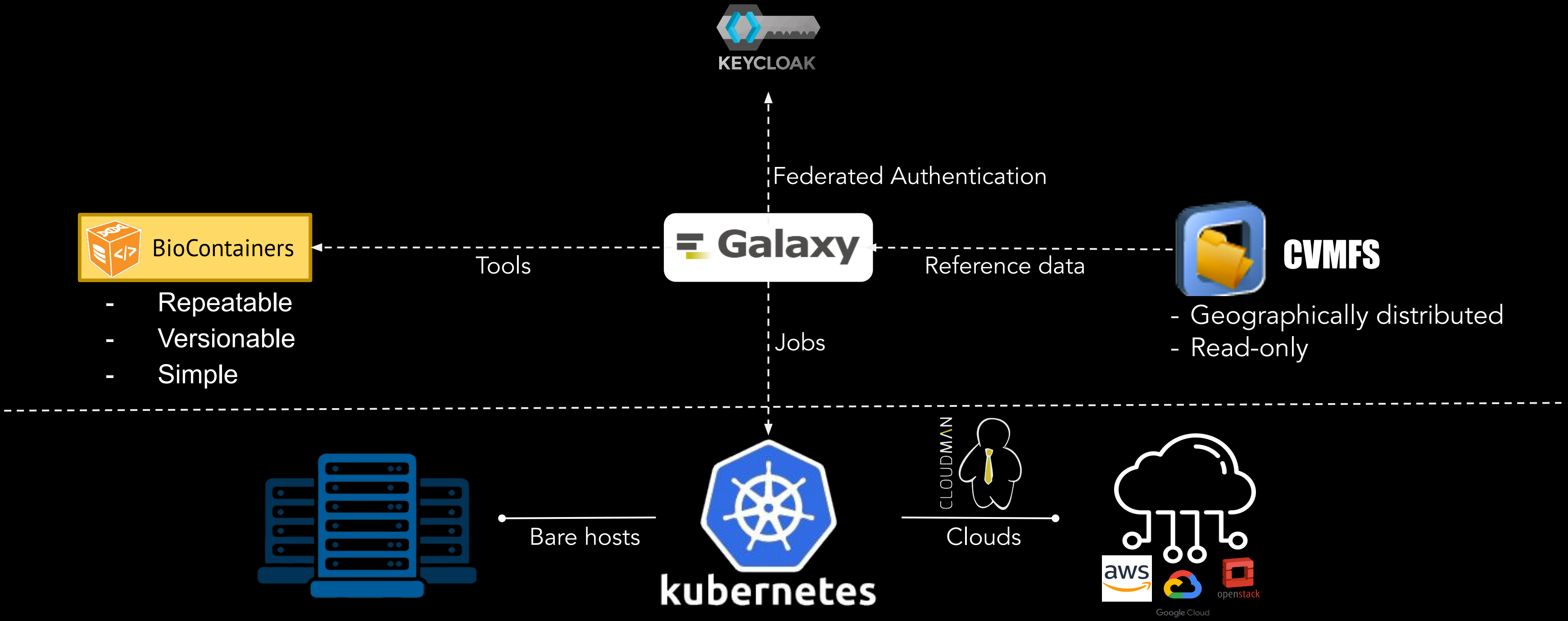
What remains to be done

1. Pulsar needs to be able to fetch and store data directly from/to object storage
2. Users need to be able to connect their storage
3. Authentication and authorization between Galaxy, users, and providers
4. Resources needs to be dynamically provisioned and torn down

Strategy

1. Evolve the current model
2. Start off with getting Pulsar staging working with a single distributed object store
3. Integrate authentication and authorization with users
4. Add support for user specified object stores

What’s been done so far



- Repeatable
- Versionable
- Simple

- Geographically distributed
- Read-only

Kubernetes as the container orchestrator - reliable, scalable, portable

Relevance and benefits to a regional Galaxy

For users

- Single Galaxy instance vs. many → more accessible
- Easier sharing, publishing, and collaborating
- No manual data transfers

For providers

- Lower administrative burden
- Pool resources together → better resource utilization
- Keep data local → efficiency and compliance
- Broader recognition and easier integration with similar efforts → more impact on funding and policy