Building a Global Supercomputer with Virtual Kubelet

Dmitry Mishin, University of California San Diego
Adrien Trouillaud, Admiralty

Outline

Part I: Building a Global Research Platform

from Nautilus, the Global Kubernetes Cluster, to a Decentralized Cluster Federation (Dmitry)

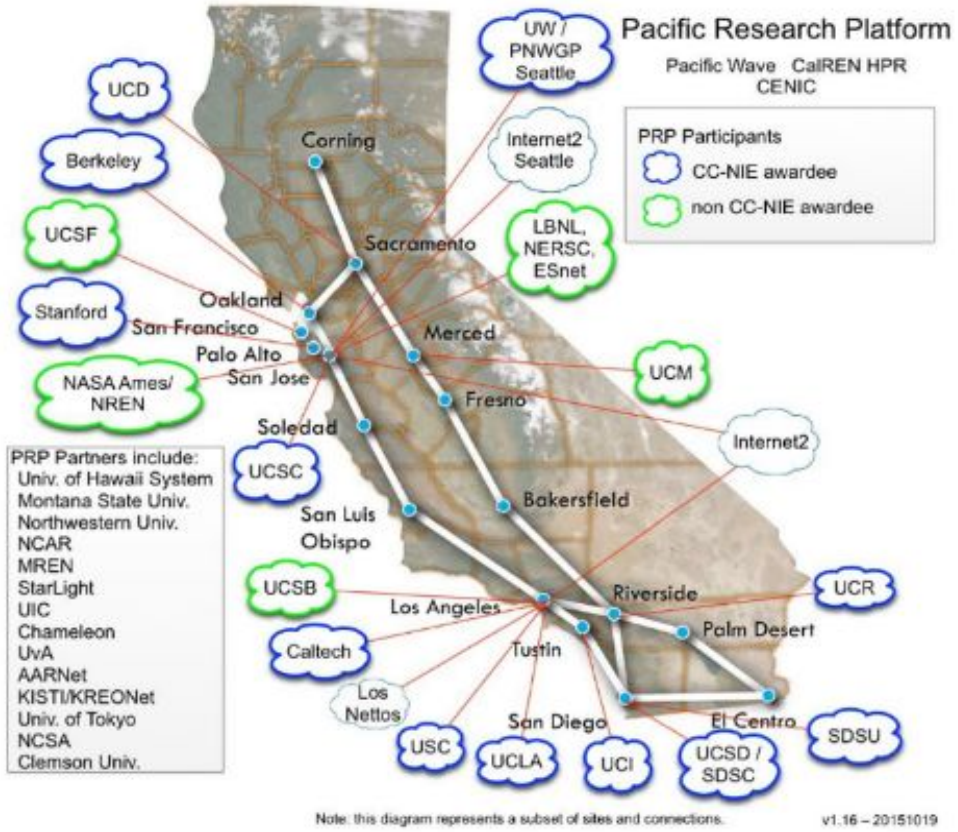
Demo (Dmitry)

Part II: How It Works

Virtual Kubelet and the Scheduler Framework, among other Kubernetes patterns (Adrien)

The Early Days of the Pacific Research Platform

- The PRP was originally created as a regional networking project
  - Establishing end-to-end links between 10Gbps and 100Gbps



PRP PACIFIC RESEARCH PLATFORM

Nautilus

More Than Just Network Measurements

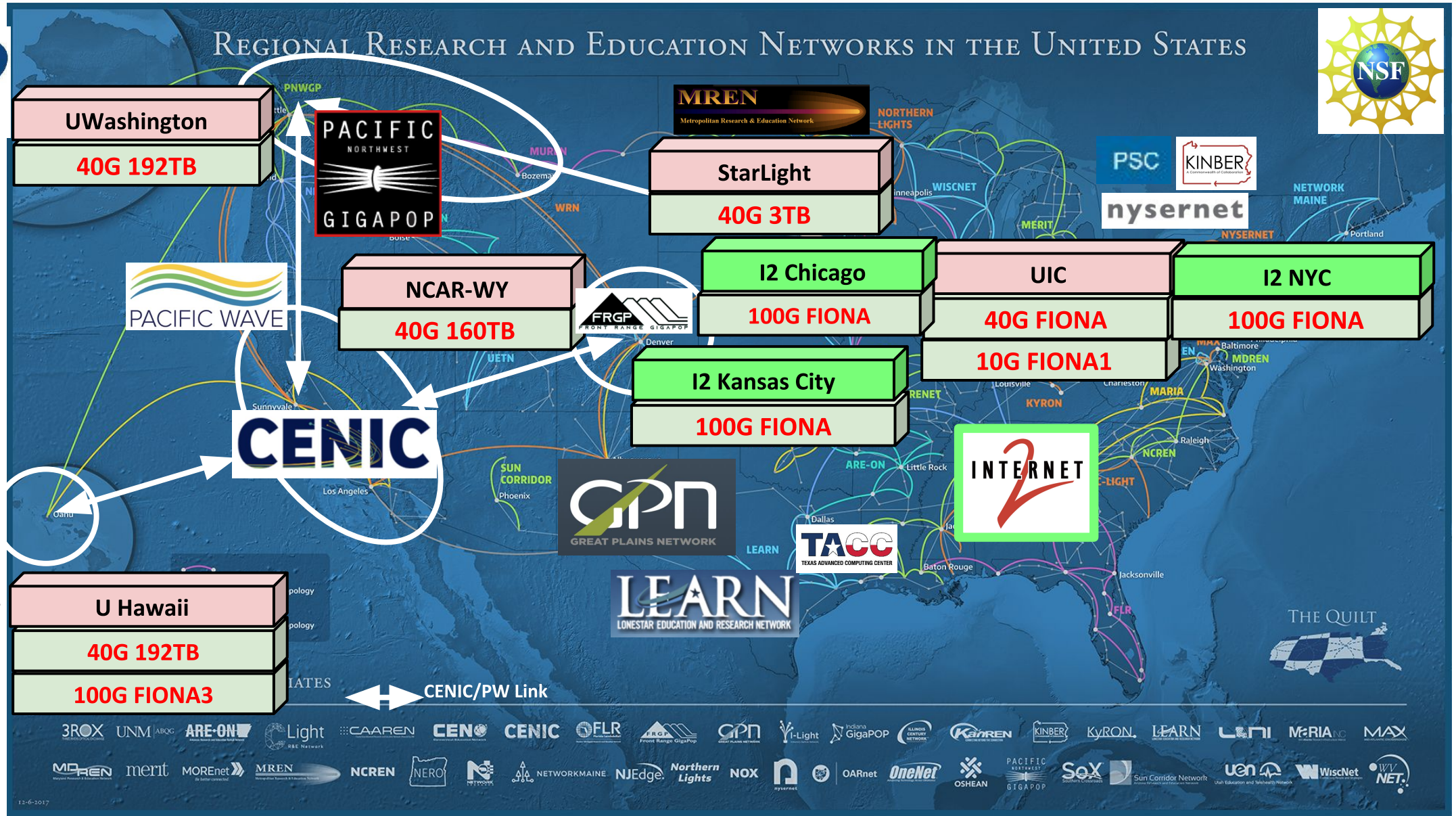
- PRP v2, with CHASE-CI (Cognitive Hardware and Software Ecosystem Community Infrastructure) and Partners, added disks and GPUs
  - Because scientists really need more than bandwidth tests
  - They need to **share their data** at high speed and **compute** on it, too

- Internet2 added nodes with disk at the POPs
  - Partnering with Open Science Grid to create a national Content Delivery Network (based on XRootD caching technology)

Open Science Grid

Pacific Research Platform / Toward a National Research Platform's United States Nautilus Hypercluster FIONAs Now Connect More Regionals and 5 Internet2 Storage & Test Sites

KubeCon CloudNativeCon North America 2020 Virtual



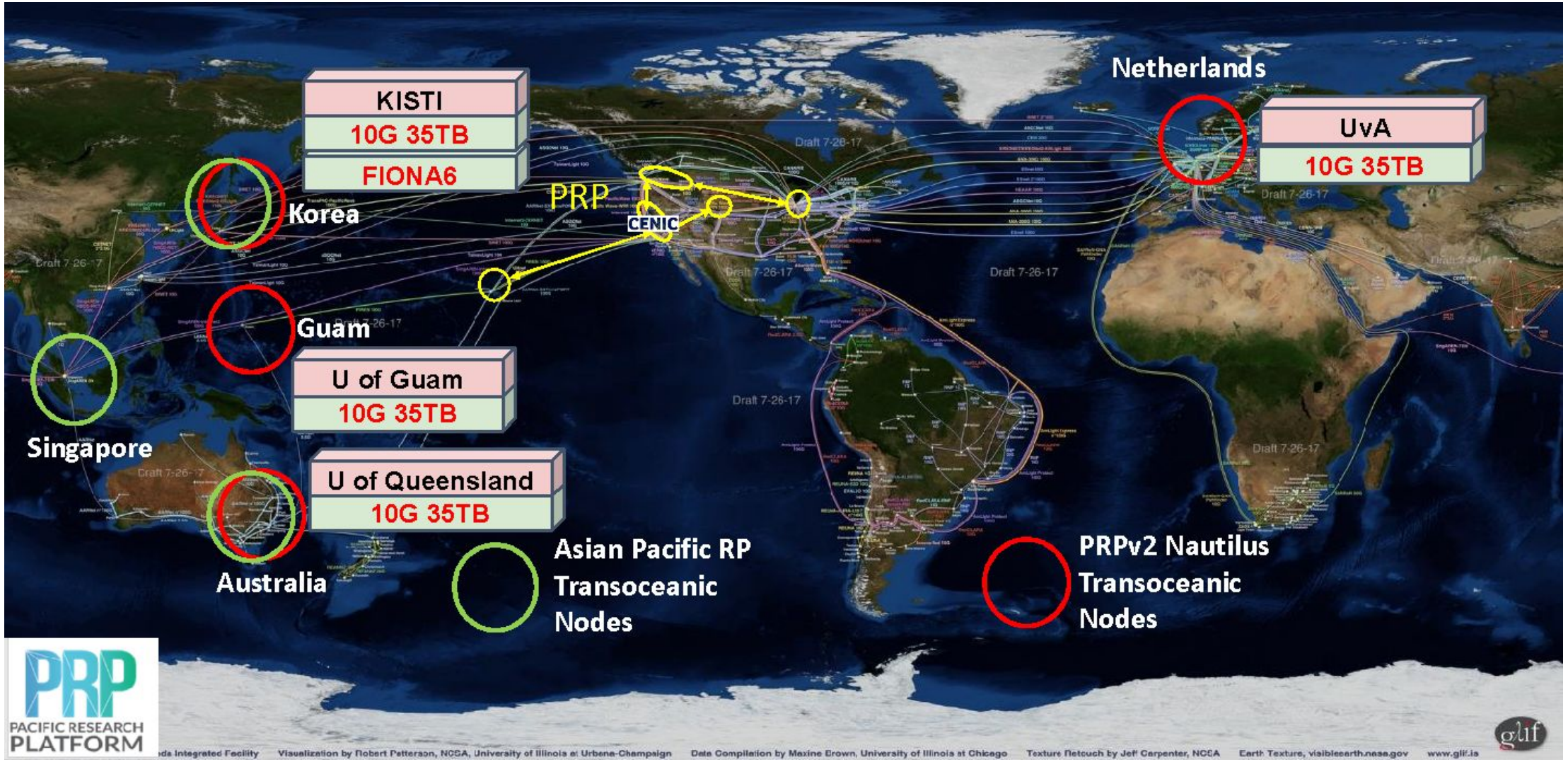
ESnet

CENIC

SDSC SAN DIEGO SUPERCOMPUTER CENTER

CITRIS AND THE BANATAO INSTITUTE

PRP/TNRP International Expansion: European and Asian-Pacific Regions



Data Transfer Nodes (DTNs): Flash I/O Network Appliances (FIONAs)

UCSD-Designed FIONAs solved the disk-to-disk data transfer problem
at near full speed on best-effort 10G, 40G and 100G networks

Two FIONA DTNs at UC Santa Cruz: 40G & 100G
Up to 192 TB rotating storage

Add up to 8 Nvidia GPUs per 2U FIONA
to add machine learning capability

FIONAs Designed by UCSD's Phil Papadopoulos, John Graham,
Joe Keefe, and Tom DeFanti

SDSC
SAN DIEGO SUPERCOMPUTER CENTER

These Days:
Federated Scientific Clusters

- PRP Nautilus, US national
  - 7000 cores / 500+ GPUs / 2.5+PB storage
- SSL-University of Chicago-RIVER
  - 2784 cores / 23 TB storage
- SSL-University of Chicago-RIVER-dev
  - 432 cores / 3.2 TB storage
- University of Washington - Tiger
  - 368 cores / 1.5 PB storage
- University of Nebraska
  - ? cores / 4 TB storage
- Metropolitan Research and Education Network Research Platform (MREN)
  - 260 cores / 3TB storage
- Expanse Supercomputer (coming soon) University of California San Diego
  - 93000 cores / 200+ V100 GPU / 12PB storage

+ dev K3S ARM
+ dev Windows
+ dev Agones GPU
+ AWS cloud bursting

Nautilus

How can we use resources between clusters?

- Universities want to control their clusters themselves
- Want to set policies
- Scientists need to set federations on namespaces level, without bugging admins too much

EXPANSE
COMPUTING WITHOUT BOUNDARIES

These Days:
Cloud Bursting for Wider Reach

- We like on-prem compute, but Clouds do have advantages
  - Elasticity
  - Many locations
  - High-end HW configurations
- All major Clouds provide native k8s support
  - Installing a custom k8s not too hard, either
- Federation makes it easy on the users
  - Submit in a single cluster
  - Run anywhere
- Federated PRP Nautilus with
  - Google Cloud
  - Microsoft Azure

Microsoft Azure

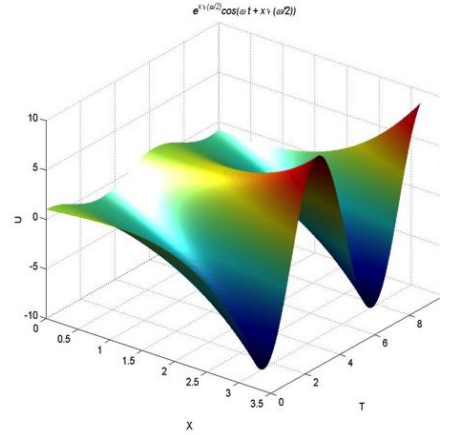
Google Cloud

Nautilus

Immediate Use Cases of Federation

- GitLab CI builds on different architectures
  - Windows, ARM
- Network monitoring
  - Deploying measurement pods, running measurements between nodes in different clusters
- Jobs bursting to clusters with unused resources
  - Scientific computations at scale
  - Data storage is still a challenge
- Medical data use
  - Some data should stay in the cluster, but can be used to create products that are shared anonymized
- Special devices (IoT) can’t join regular cluster
  - IoT devices are small appliances controlled by different groups and are not general enough to get all monitoring from large cluster

perfSONAR



Future of Federation

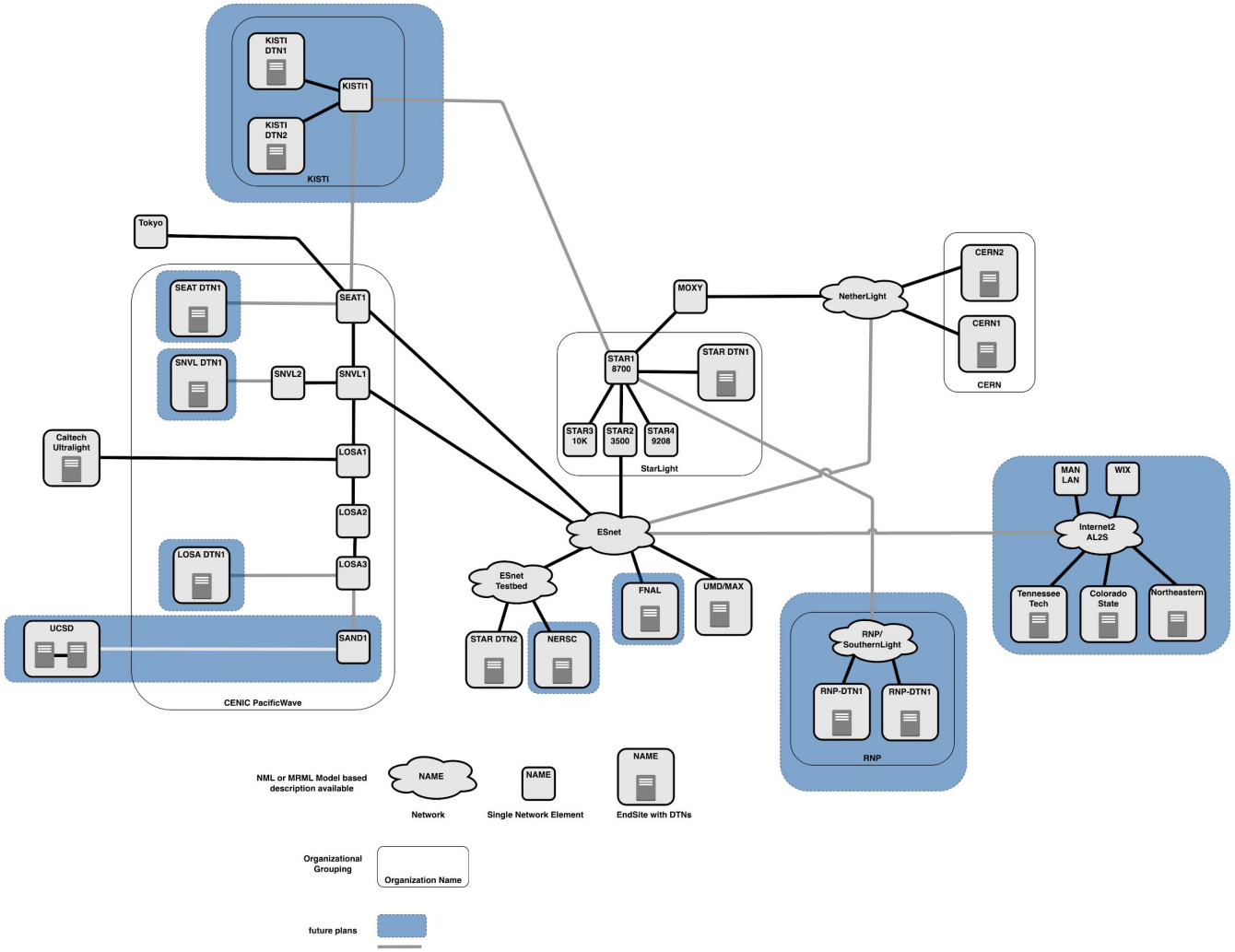
OpenNSA AutoGOLE

provisioning Layer 2 paths

between clusters controlled by

kubernetes CRDs

(in progress)



NORDUnet
Nordic Gateway for Research & Education

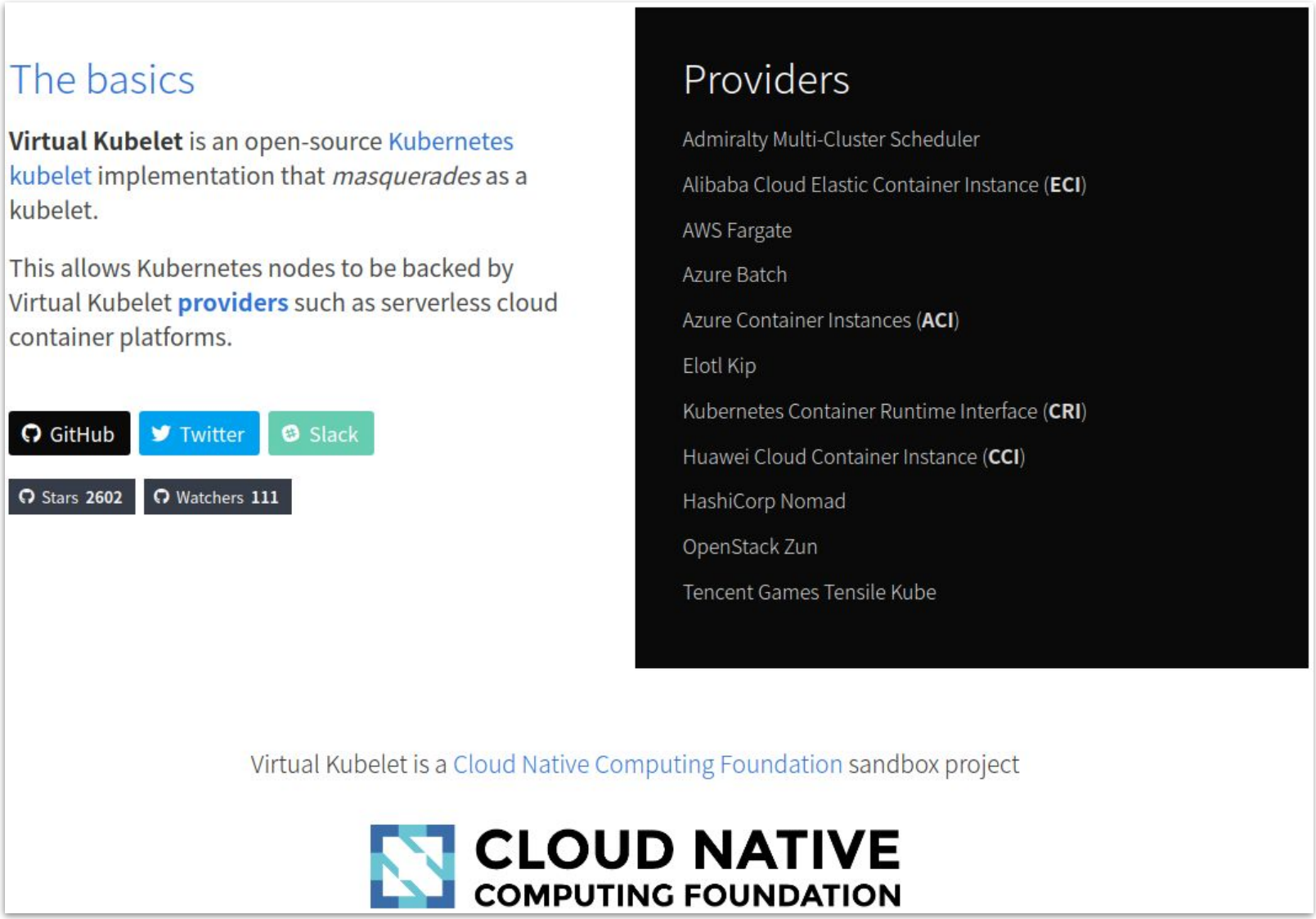
Part II

Virtual Kubelets
+ Admission Webhooks
+ Schedulers
+ Controllers
= Admiralty

a decentralized multi-cluster control plane

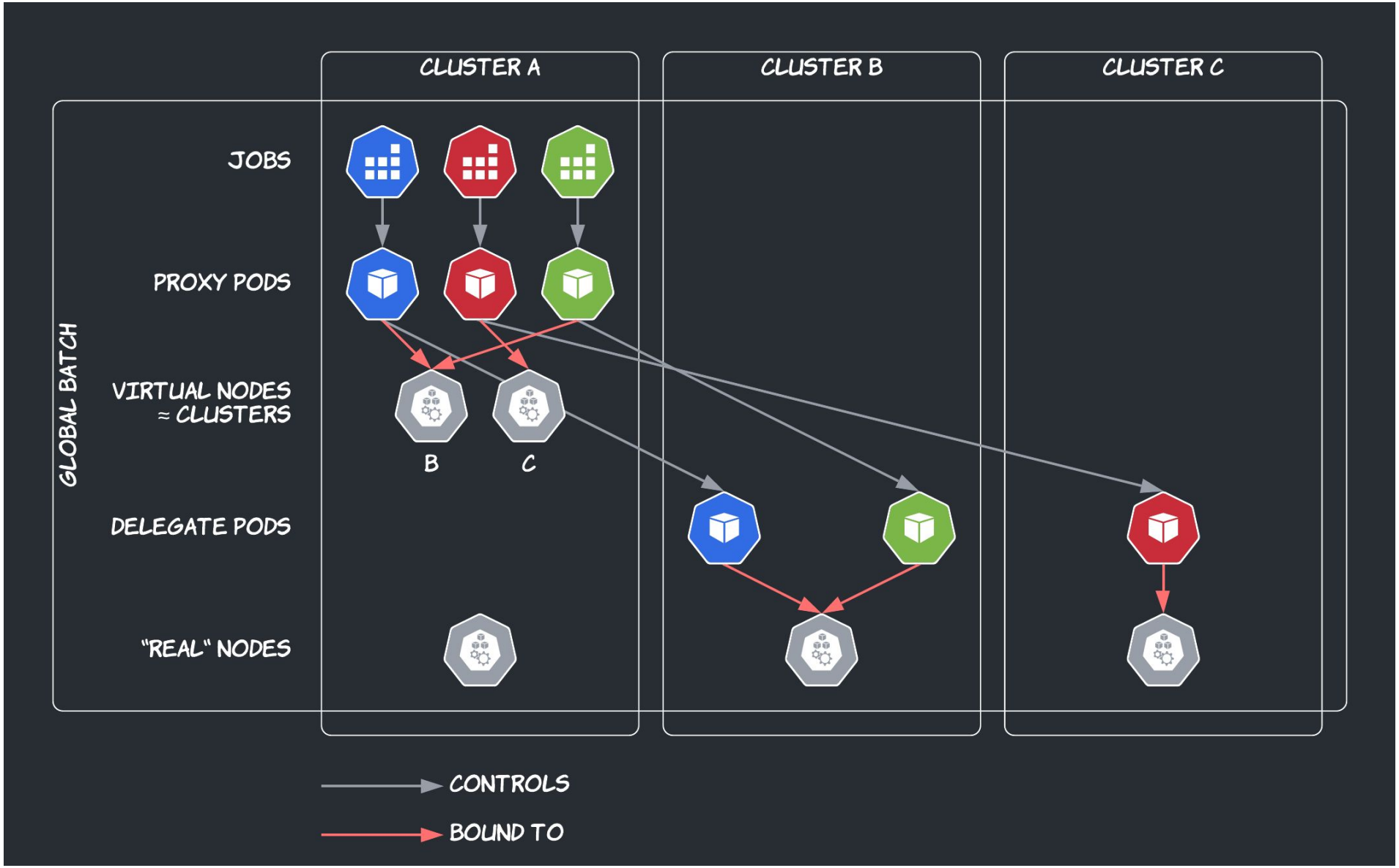
https://github.com/admiraltyio/admiralty

Virtual Kubelet



Screenshot from https://virtual-kubelet.io

Example Use Case



Virtual Kubelet Responsibilities & Admiralty Implementation

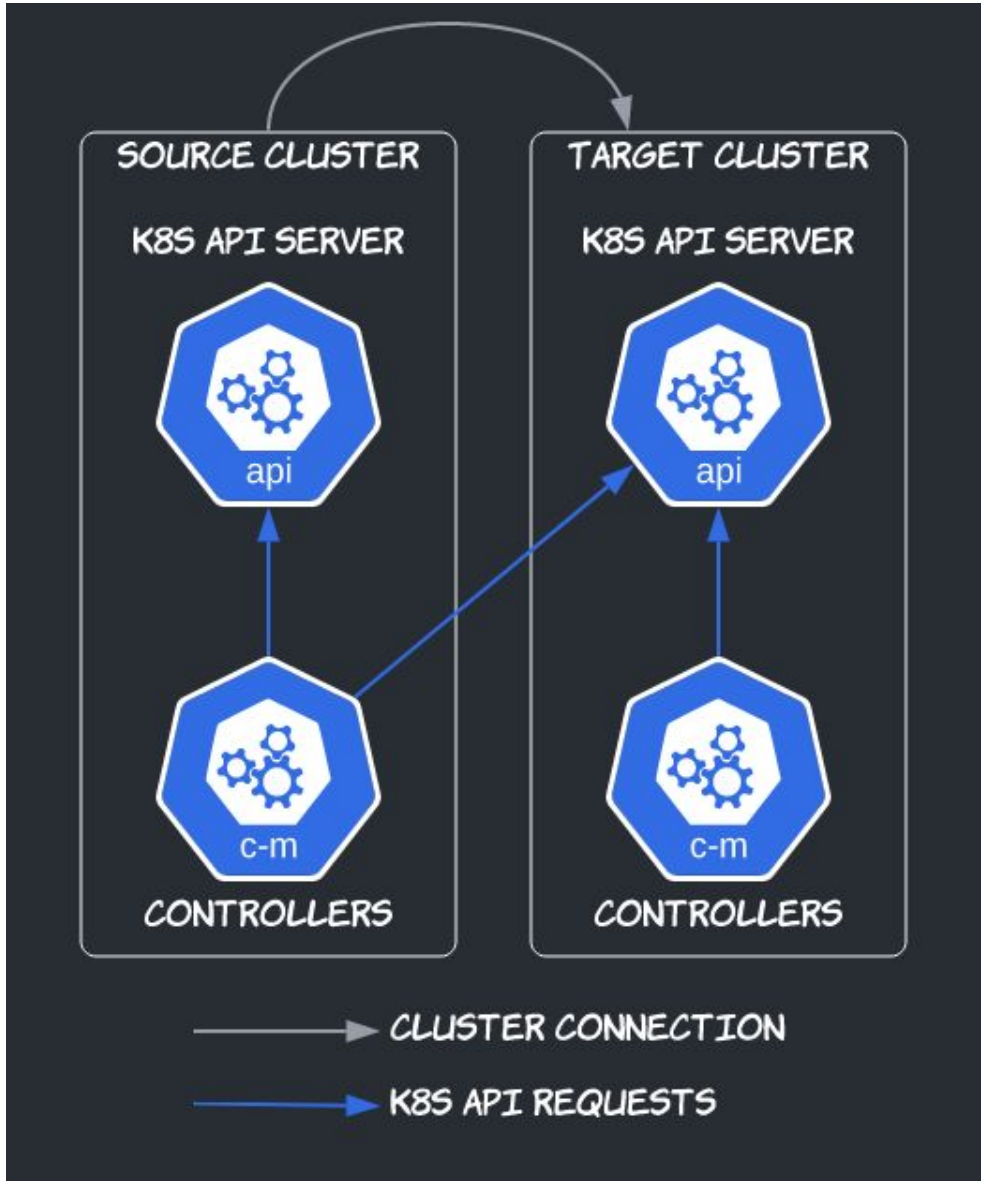
Self-Registration Create Node object	Create one Node per Target/ClusterTarget (namespaced/cluster-scoped CRDs mapping virtual node names to kubeconfig secrets)
Heartbeat Update NodeStatus and Lease object regularly	NodeStatus condition Ready based on target cluster health check
Pod Lifecycle Handle Pods bound to its Node (run containers somewhere and update PodStatus)	Multi-Cluster Scheduling PodStatus Feedback Cross-Cluster Garbage Collection
Extras Handle pod logs and exec requests	Forward requests to target cluster Kubernetes API

Cluster Connections

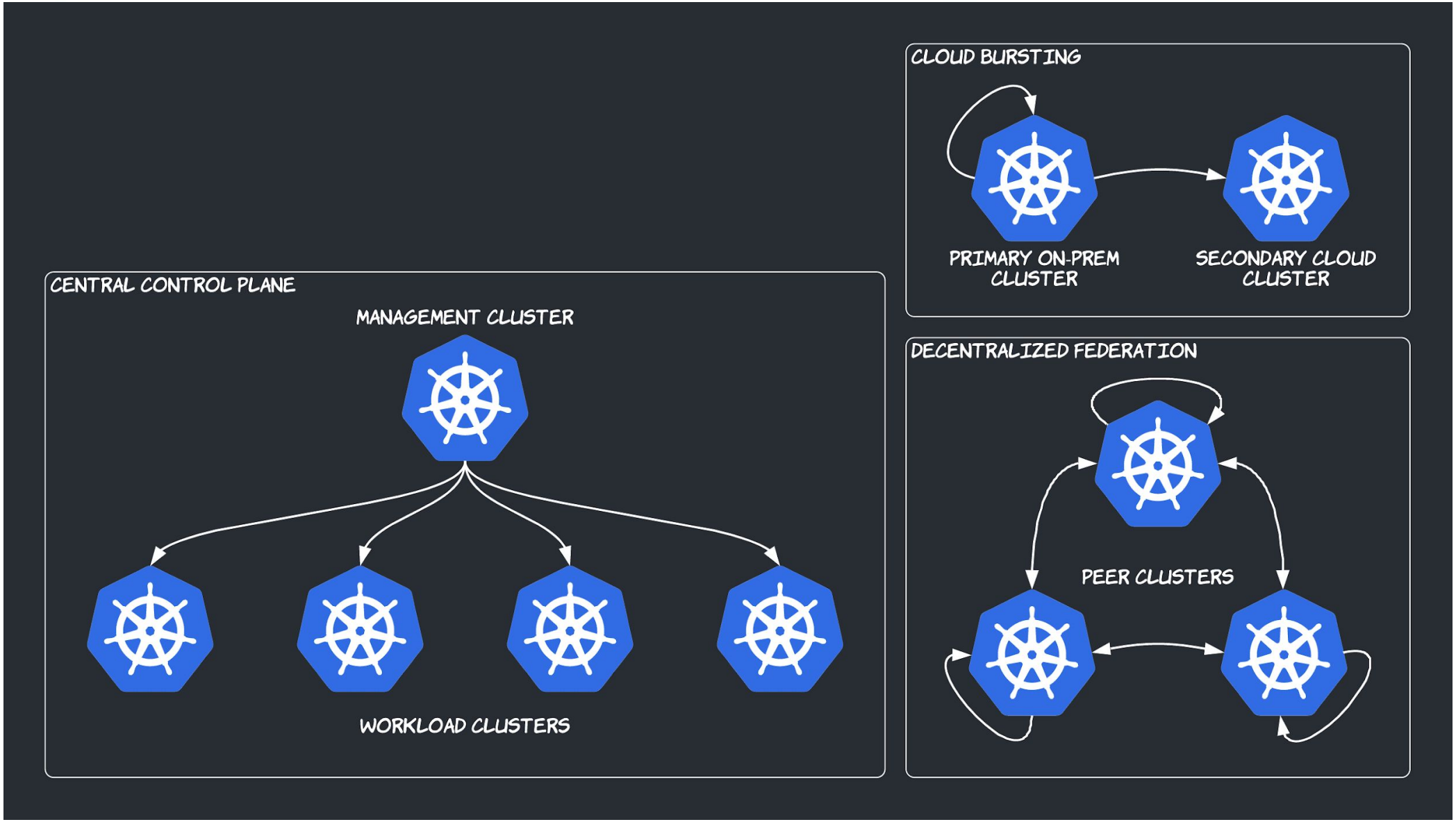
Target cluster Kubernetes API server must be routable from source cluster controller pods (VPN or tunnel may be needed)

Authentication (several methods—next slide)

Authorization (target cluster RBAC)



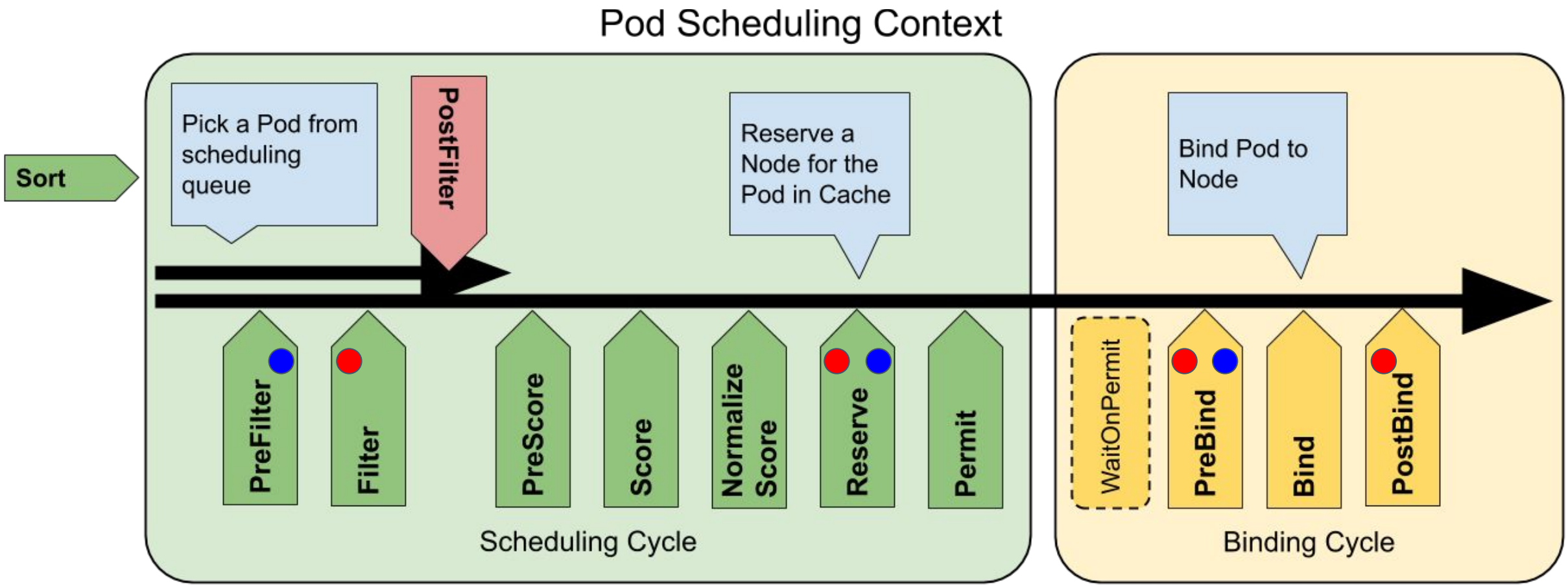
Cluster Topologies



Cross-Cluster Authentication Methods

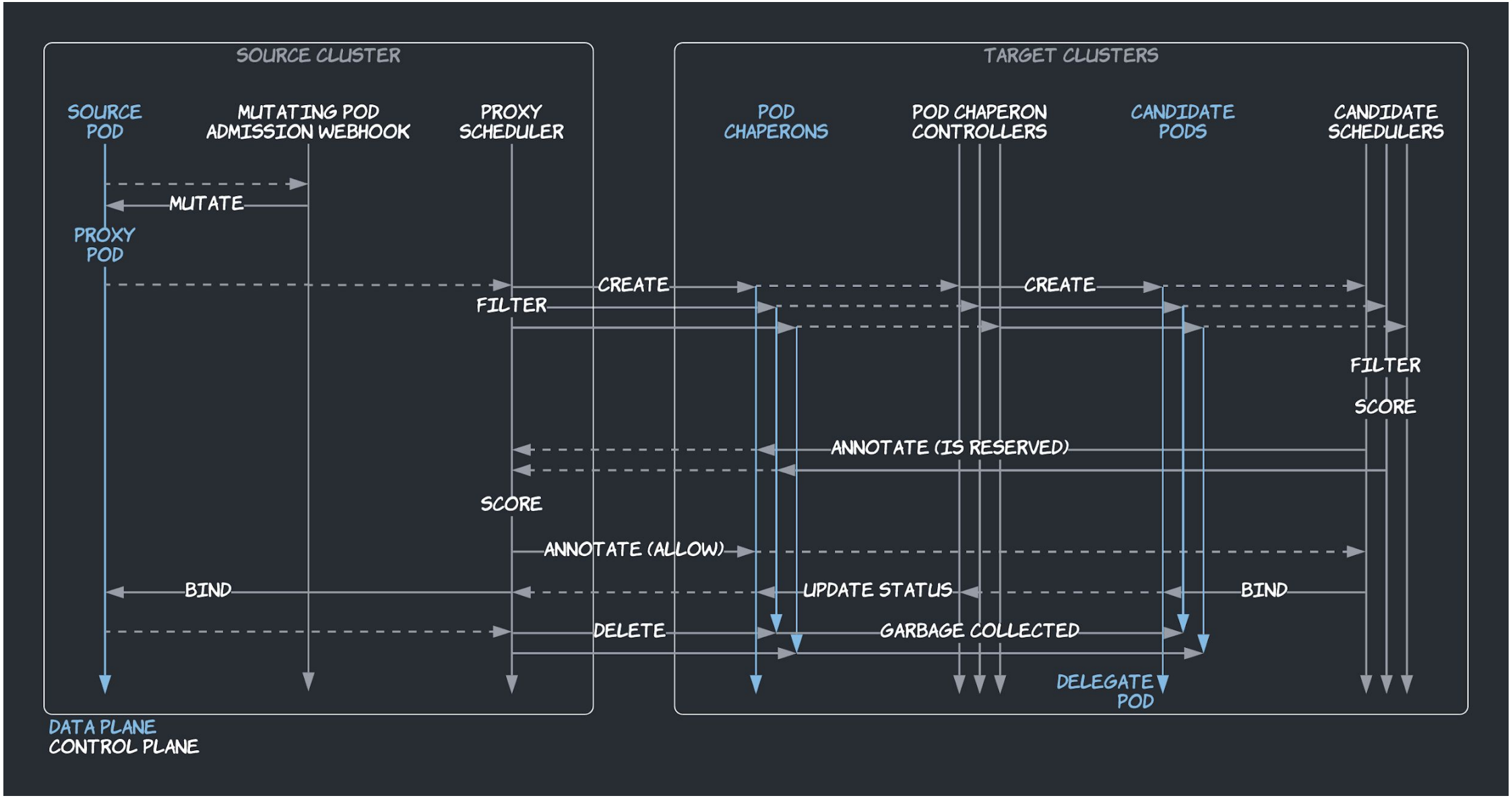
Method	Secret distribution and rotation, shared/federated identity provider support	Kubernetes API server flag changes needed	Example Implementations
Kubernetes Service Account token from target cluster	no	no	n/a
Certificates API of target cluster	no	no	n/a
Cloud identities	yes	no	AWS IAM roles Google Cloud service accounts Azure service principals
Webhook Token Authentication	yes	yes	
Authenticating Proxy	yes	yes	
Impersonating Proxy	yes	no	jetstack/kube-oidc-proxy Admiralty Cloud

Kubernetes Scheduler Framework



https://kubernetes.io/docs/concepts/scheduling-eviction/scheduling-framework/

Admiralty's "Look-Ahead" Multi-Cluster Scheduling Algorithm



Summary

Nautilus, a global Kubernetes cluster, is now federated with multiple other clusters in peer-to-peer relationships, totaling 10,000+ cores (soon 100,000+).

Admiralty, the open source fabric of this federation, was built upon Virtual Kubelet, the Scheduler Framework, among other Kubernetes patterns.

Join the federation: https://pacificresearchplatform.org/userdocs

Build your own: https://admiralty.io

Connect with the speakers:

- dmishin@ucsd.edu
- adrien@admiralty.io, @adrienjt

This work was partially funded by US National Science Foundation (NSF) awards CNS-1456638, CNS-1730158, ACI-1540112, ACI-1541349, OAC-1826967, OAC 1450871, OAC-1659169 and OAC-1841530.