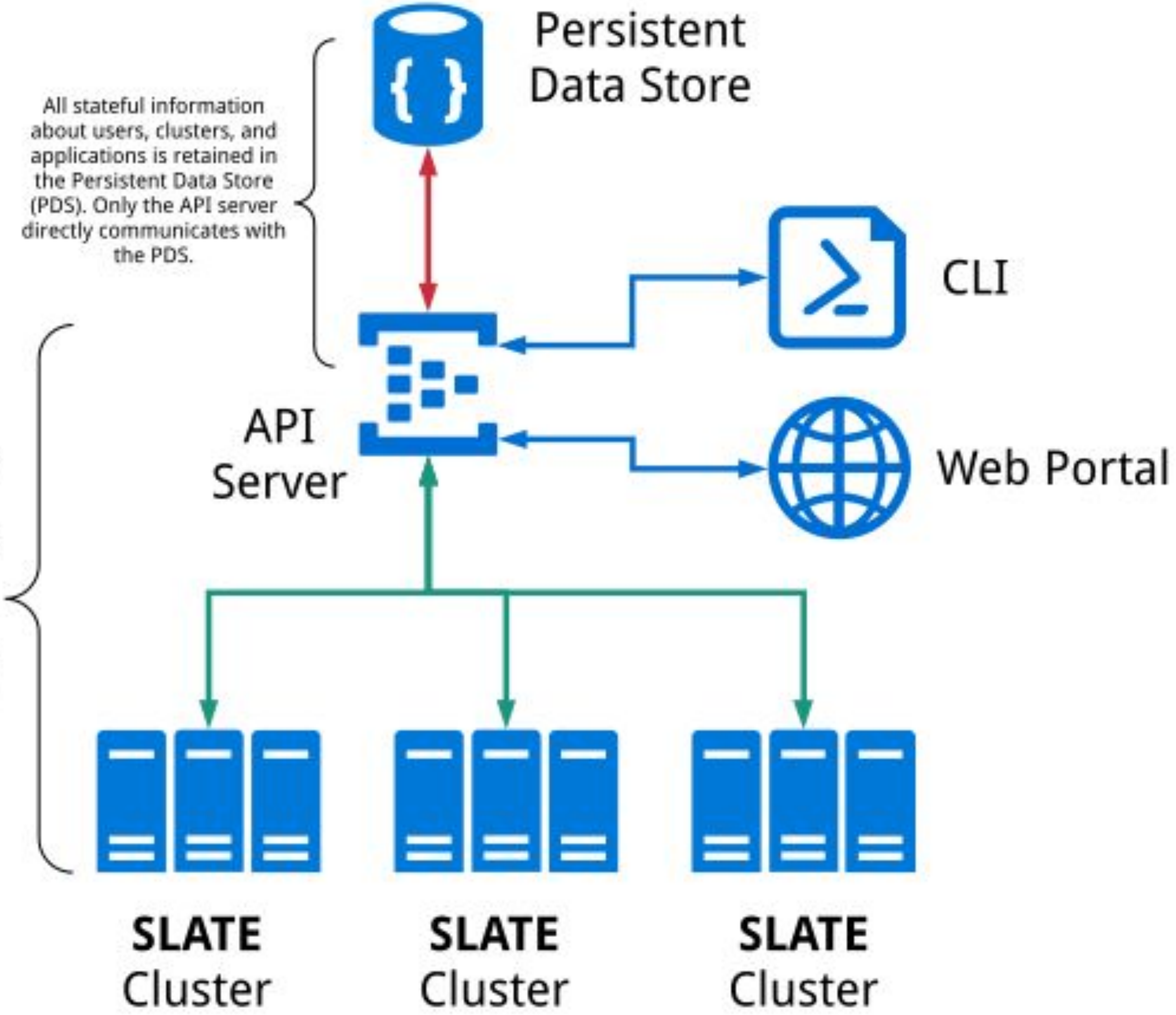
SLATE Client and Server Architecture

Chris Weaver

August 29, 2018

Architecture



All stateful information about users, clusters, and applications is retained in the Persistent Data Store (PDS). Only the API server directly communicates with the PDS.

User-facing CLI and Web Portal allow users to manage clusters and applications.

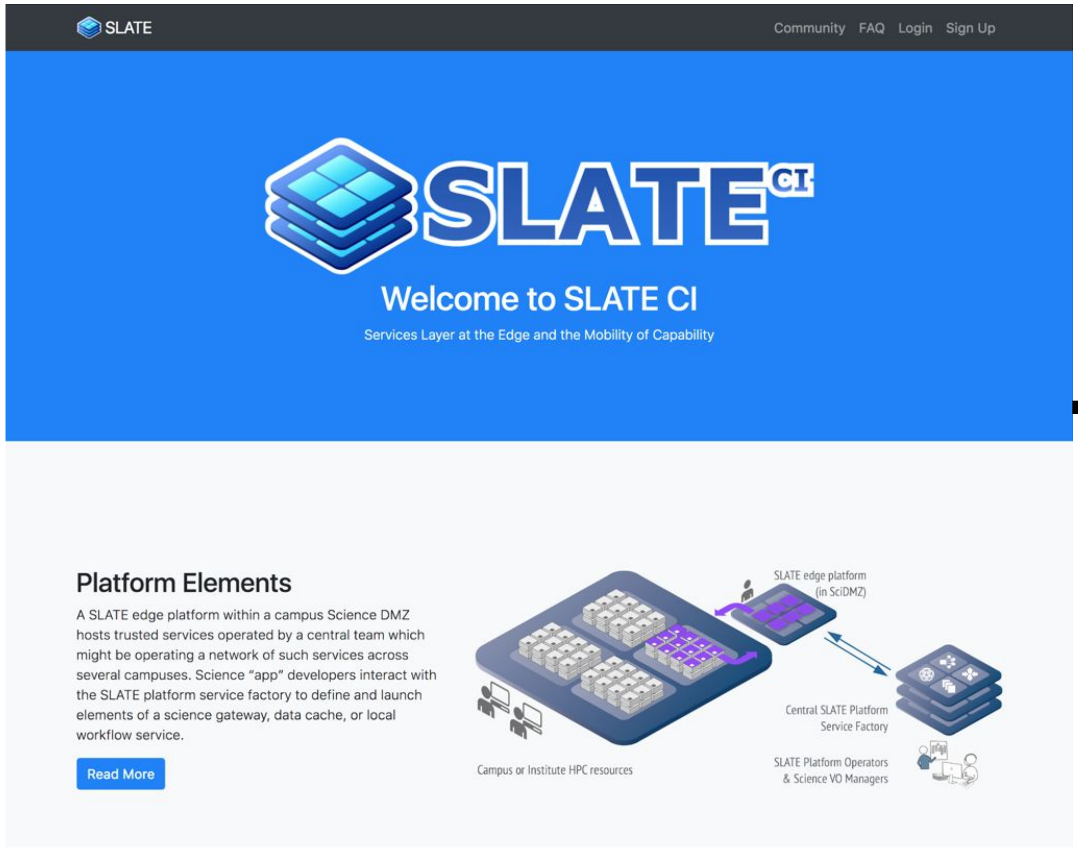
Clusters are registered with SLATE API Server, which then applies base configuration to the clusters (e.g. VO, Monitoring, etc)

API server reconciles state between container orchestration software and data in the Persistent Data Store (e.g. in the event of cluster failure).

SLATE Goals

- Simplicity
  - Many potential users have no experience with Kubernetes. Well written Helm charts should enable them to install applications anyway.
- Security
  - Allowing only curated charts simplifies what we ask site administrators to trust
  - Limited access to edge clusters—SLATE is a regular, limited cluster user
- Flexibility
  - Both dedicated and otherwise existing clusters should be able to participate in SLATE federation

Web Portal



Welcome to SLATE CI

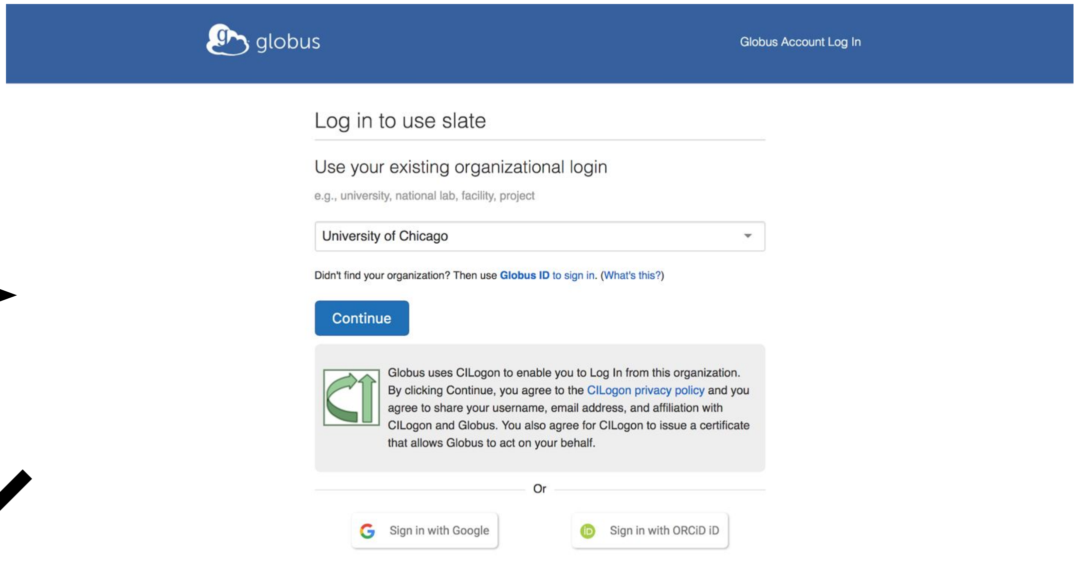
Services Layer at the Edge and the Mobility of Capability

Platform Elements

A SLATE edge platform within a campus Science DMZ hosts trusted services operated by a central team which might be operating a network of such services across several campuses. Science "app" developers interact with the SLATE platform service factory to define and launch elements of a science gateway, data cache, or local workflow service.

Read More
SLATE edge platform (in SciDMZ)
Central SLATE Platform Service Factory
SLATE Platform Operators & Science VO Managers
Campus or Institute HPC resources

Globus Auth
Signup/Login



Log in to use slate

Use your existing organizational login

e.g., university, national lab, facility, project

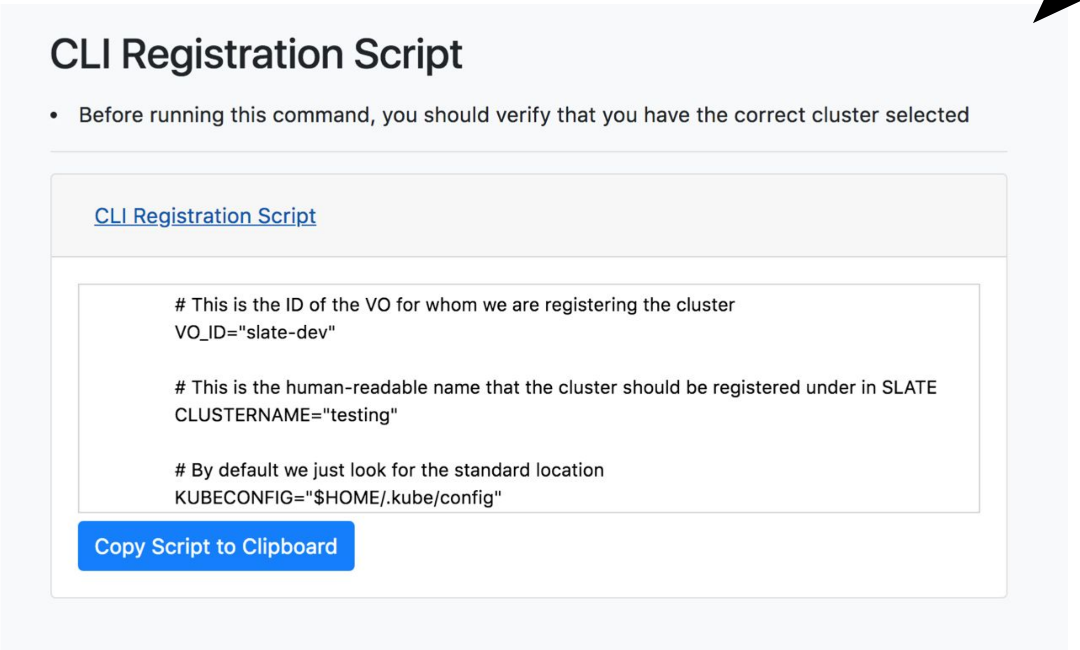
University of Chicago

Didn't find your organization? Then use Globus ID to sign in. (What's this?)

Globus uses CILogon to enable you to Log In from this organization. By clicking Continue, you agree to the CILogon privacy policy and you agree to share your username, email address, and affiliation with CILogon and Globus. You also agree for CILogon to issue a certificate that allows Globus to act on your behalf.

Application and
Cluster Admins

Cluster Admins



CLI Registration Script

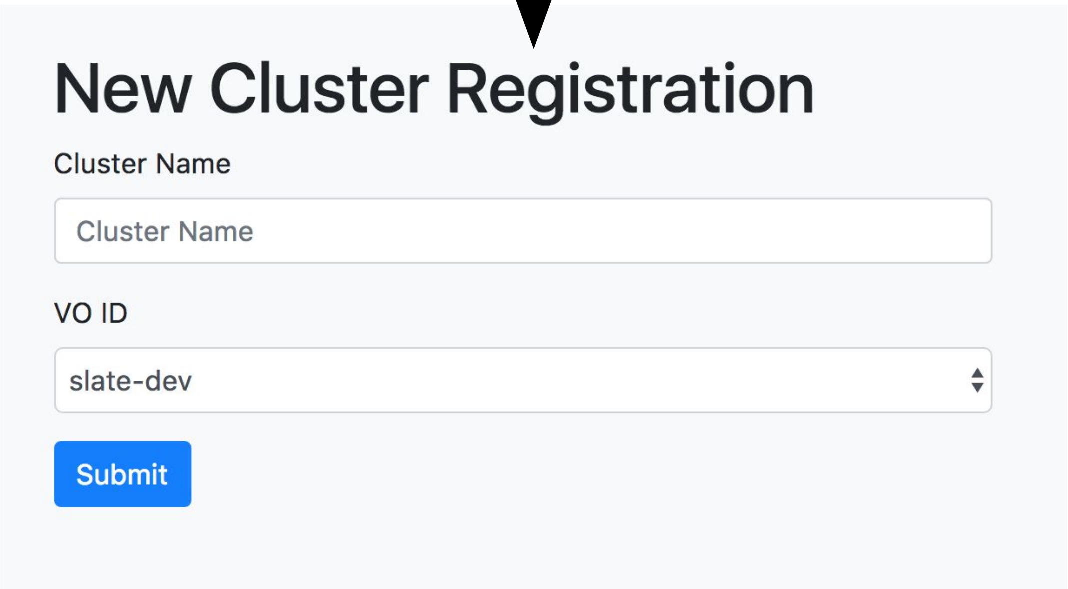
- Before running this command, you should verify that you have the correct cluster selected

CLI Registration Script

```
# This is the ID of the VO for whom we are registering the cluster
VO_ID="slate-dev"

# This is the human-readable name that the cluster should be registered under in SLATE
CLUSTERNAME="testing"

# By default we just look for the standard location
KUBECONFIG="$HOME/.kube/config"
```



New Cluster Registration

Cluster Name: Cluster Name

VO ID: slate-dev

Command-Line Interface

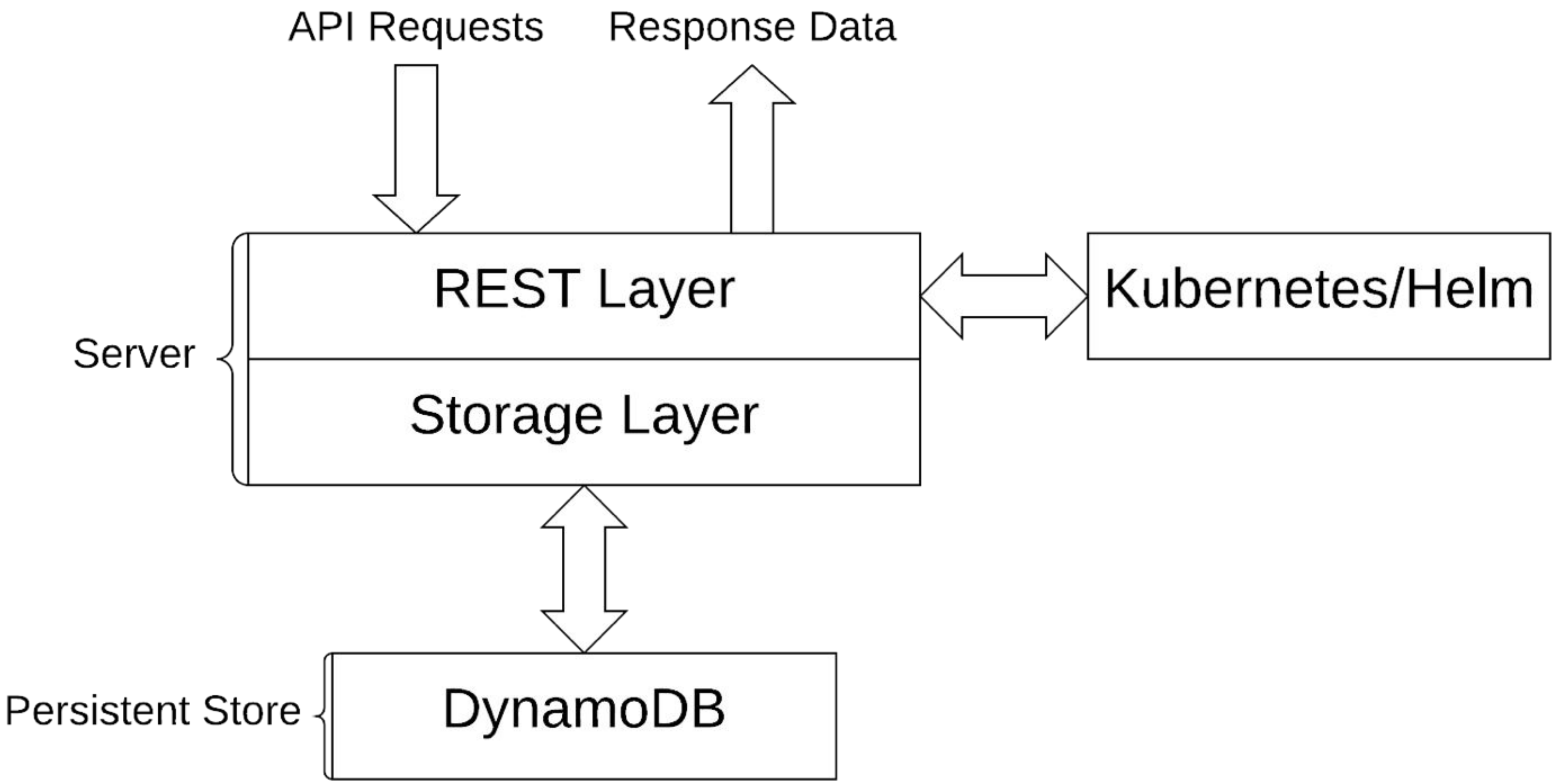
```
Niobium-III $ ./slate-client cluster list
Name                ID
us-chicago-minikube Cluster_71da4abf-6340-4169-aec9-446816d36d79
Niobium-III $ ./slate-client app list --dev
Name                         App Version Chart Version Description
slate-dev/elasticsearch                  0
slate-dev/fluentbit-kibana               0.1
slate-dev/jupyterhub         v0.8.1      v0.7-dev      Multi-user Jupyter installation
slate-dev/osg-frontier-squid squid-3     0.2.0         A Helm chart for configuration and deployment...
slate-dev/perfsonar          1.0         0.1.0         A Helm chart for Kubernetes
Niobium-III $ ./slate-client app install --dev --vo test-vo --cluster us-chicago-minikube
osg-frontier-squid
Successfully installed application "osg-frontier-squid" as instance "osg-frontier-squid" with ID
"Instance_e4357b8c-8176-482e-be2e-5917b686159d"
Niobium-III $ ./slate-client instance info Instance_e4357b8c-8176-482e-be2e-5917b686159d
Name               Started                  VO      Cluster             ID
osg-frontier-squid 2018-Jul-13 16:24:45 UTC test-vo us-chicago-minikube
Instance_e4357b8c-8176-482e-be2e-5917b686159d

Services:
Name                      Cluster IP   External IP ports
osg-frontier-squid-global 10.96.65.120 <pending>   3128:30822/TCP

Configuration: (default)
```

- Interface very much inspired by kubectl
- Similar, but simpler, set of capabilities

API Server Structure



- API server performs authentication and authorization checks on all user requests
- If authorized, carries them out by issuing commands to kubectl, helm, or recording information in the Persistent Store
- Server itself has no important state

Design Choices

- SLATE uses RBAC and a Kubernetes plugin developed by the Pacific Research Platform group to isolate user groups (VOs) while requiring minimal privileges on the Kubernetes cluster
  - This enables SLATE to work within cluster admins' security choices
  - A single cluster can belong to more than one SLATE federation
- The VOs which may access a cluster can be controlled by that cluster's admins via a whitelist
  - Cluster admins can likewise choose to limit exactly which applications an admitted VO may deploy on that cluster

Design Choices

- Using Helm and curating charts can boost confidence in available applications and make getting started easy for new users
  - Charts need to be written, which isn't always easy
    - We are still figuring out best practices ourselves
  - Requiring Helm means that we can't directly use applications containerized in other ways
    - Are Helm charts useful to Kubernetes user outside SLATE?
  - Helm itself can be a somewhat awkward tool, which is part of why we wrap it up as an internal detail of our API