



Federated Service Deployment on the Edge

Rob Gardner
University of Chicago

Snowmass CompF4, April 7, 2022



Motivation

- HEP **at all scales** requires collaboration and sharing
- Sharing of data, resources, and expertise **democratizes access, optimizes discovery**
- At every scale institutional boundaries are crossed
 - Accessing resources, data transfer, scaling applications
- Data-intensive, distributed, high-throughput computing systems to share data and cycles is a common pattern
 - Open Science Grid and PATH are pioneers in building **national-scale shared cyberinfrastructure to pool resources**
 - New facilities, for example **low latency analysis platforms**, are being explored by projects such as the NSF IRIS-HEP software institute

How to address these challenges?



- 1. Extend our computing centers and augment the ScienceDMZ to include an open and secure platform for running long-lived services.**

How to address these challenges?



1. **Extend our computing centers and augment the ScienceDMZ to include an open and secure platform for running long-lived services.**
2. **Establish a catalog of applications trusted by resource providers, and give them fine-grained control over *who* and *what***

How to address these challenges?



- 1. Extend our computing centers and augment the ScienceDMZ to include an open and secure platform for running long-lived services**
- 2. Establish a catalog of applications trusted by resource providers, and give them fine-grained control over *who* and *what***
- 3. Build teams of application experts who can build and run services for a whole collaboration or experiment**

Enter Federated Operations



- These concepts became the foundation of a model we call Federated Operations (FedOps)
- Worked with NSF [Trusted CI](#) cybersecurity center of excellence in 2018-2019 to establish a **trust framework** and set of policies for implementing Federated Operations
 - Also worked with CERN and the Worldwide LHC Computing Grid to develop this further (c.f. [CHEP2019 article](#))
- The SLATE platform **implements FedOps**
 - SLATE and the Mobility of Capability, NSF DIBBS Award # [1724821](#)





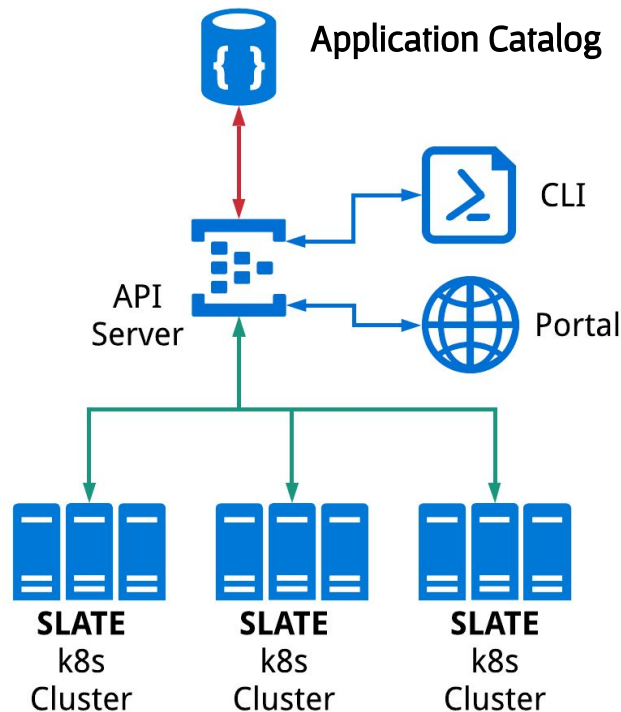
SLATE in a nutshell

- Services Layer At The Edge
- Platform for programmatically deploying applications to sites in a secure and easy-to-use way
- Three fundamental pieces:
 - Centralized service to manage users, groups, clusters, and authorization thereof
 - REST API with a fully supported web portal and commandline client
 - Curated catalog of applications, a la 'the App Store'

SLATE Architecture



- Lightweight federation and application catalog layer on top of Kubernetes
 - Security-conscious, site autonomous
 - Sites retain administrative control
- Single endpoint using institutional identity
- Simple UNIX-like permissions model (Users + Groups)
- Application catalog abstracts away much of the Kubernetes details, let's users think about the important parts of their deployment



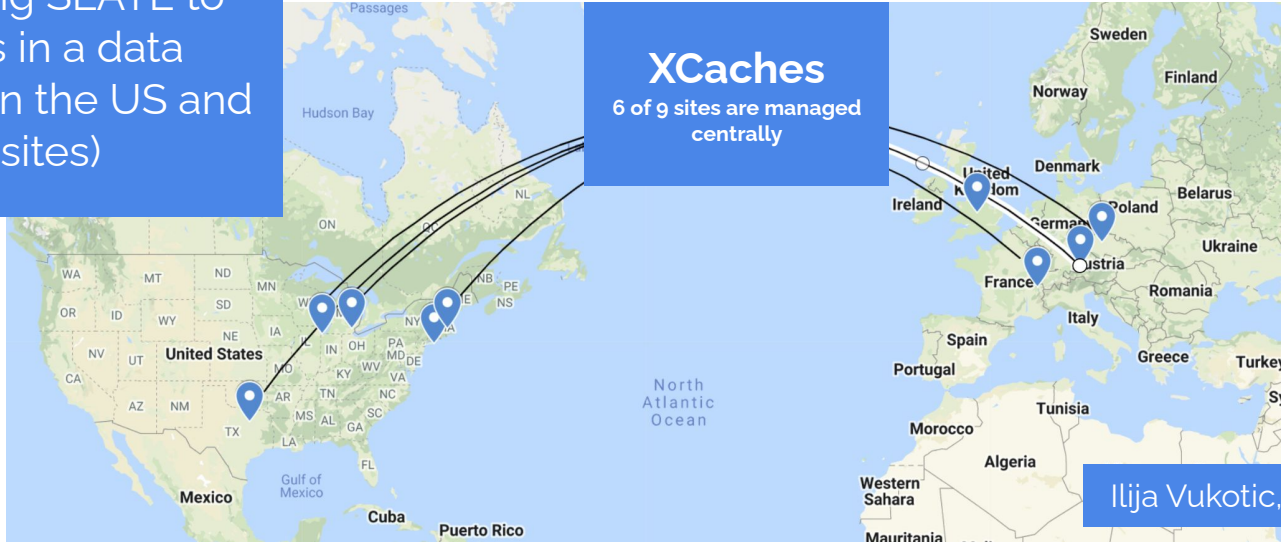


Examples of Federated Operations

Content Delivery Networks

US ATLAS is using SLATE to deploy caches in a data delivery network in the US and Europe (5 sites)

XCaches
6 of 9 sites are managed centrally



Ilija Vukotic, UChicago

Easy XCache Deployment

Upload & encrypt certificate

```
slate secret create xcache-cert-secret \  
  --from-file userkey=xcache.key.pem --from-file usercert=xcache.crt.pem \  
  --group atlas-xcache --cluster uchicago-prod
```

Deploy instance to a cluster

```
slate app install xcache --group atlas-xcache \  
  --cluster uchicago-prod \  
  --conf uchicago.yaml
```

Restart or Delete instances

```
slate instance restart instance-3JDNXA  
slate instance delete instance-3JDNXA
```

Read logs

```
slate instance logs instance-3JDNXA
```

```
Instance: global  
  
Service:  
  # Port that the service will utilize.  
  Port: 1094  
  ExternalIP: 192.170.227.128  
  
SiteConfig:  
  Name: UChicago  
  AGISprotocolID: 999  
  
Monitoring:  
  Collector: http://uct2-collectd.mwt2.org:8080  
  
XCacheConfig:  
  # Local directory to be used as a cache  
  CacheDirectories:  
    - /scratch/1  
    - /scratch/2  
    - /scratch/3  
    - /scratch/4  
    - /scratch/5  
    - /scratch/6  
    - /scratch/7  
    - /scratch/8  
  MetaDirectory: /scratch/meta  
  # Space usage  
  HighWaterMark: 0.95  
  LowWaterMark: 0.90  
  # The amount of memory XCache is allowed to use (in GB)  
  RamSize: 60g  
  # Minimal chunk size that will be prefetched/cached.  
  BlockSize: 1M  
  # to prefetch even chunks that were not requested  
  Prefetch: 0  
  # Write queue parameters  
  WQBlocksPerLoop: 10  
  WQThreads: 2  
  
  # The name of the secret created for XCache certificate  
  CertSecret: xcache-cert-secret
```



Frontier Squid FedOps for US ATLAS

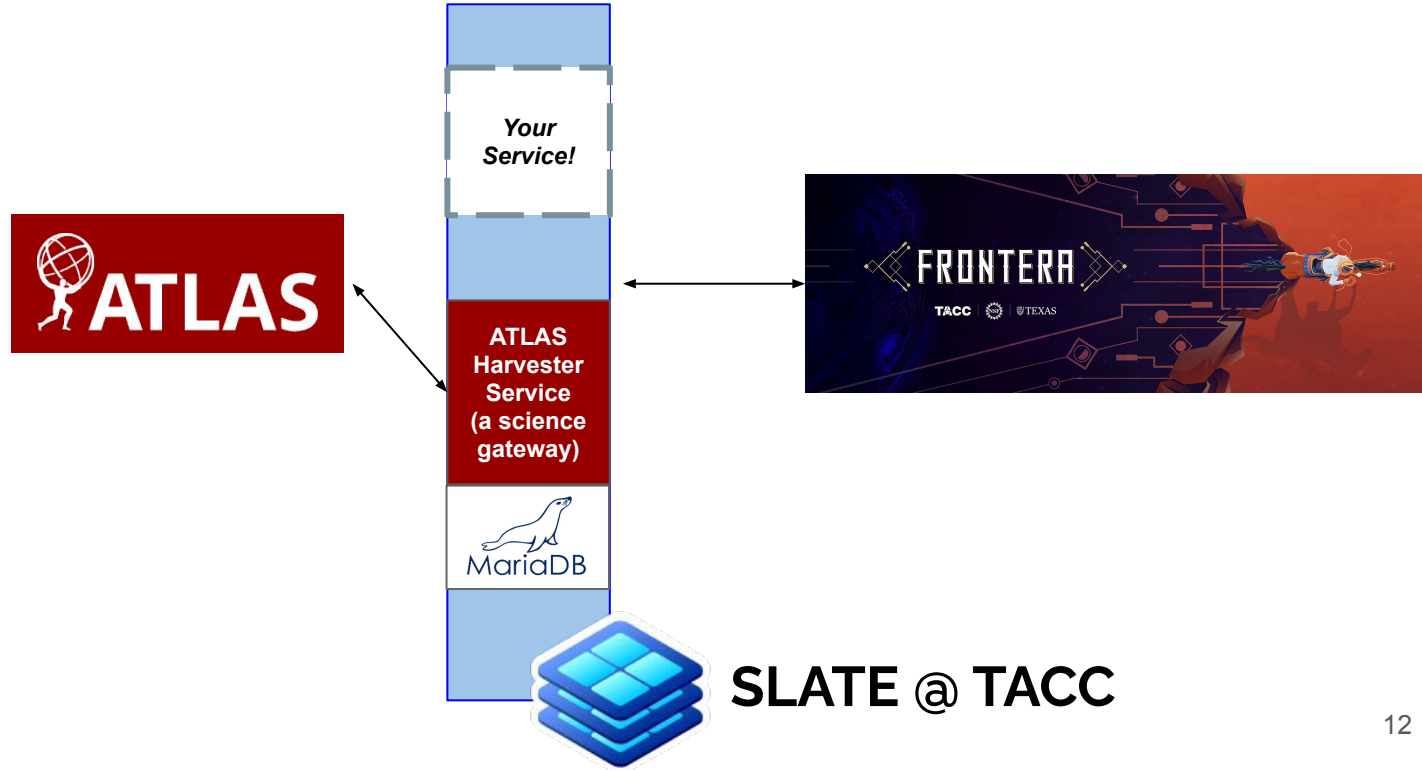
- Squid container from OSG, Helm chart from SLATE/US ATLAS
- Meetings in the past year to develop Federated Operation policies
 - responsibilities
 - priviledges
- Deployment options:
 - web interface (useful for testing)
 - command line interface (direct deployments)
 - GitHub actions/automation (GitOps - declarative deployments) (**chosen for production**)
- Currently have SLATE-managed instances at all US ATLAS Tier2s
- A FedOps operator's document ([link](#)) gives you as sense of how its done

Deployment by a Central Expert Team



A SLATE cluster at TACC uses the Singularity Container Runtime Interface (CRI) to deploy long-lived services adjacent to Frontera

Bringing long-lived services to multiple HPC centers





Contact information

- Homepage: <https://slateci.io/>
- Slack: <https://slack.slateci.io/>
- Drop us a line [here](#)

We have working calls Wednesdays 3PM CST
open invite, [ask us](#) for contact details



Extras



Anatomy of a SLATE Application

- Dockerfile describing the installation of your software
- Templated YAML file (Helm chart) describing the service in terms of Kubernetes objects (Pods, Ingress, Services, etc)
- Configuration parameter file exposed to the operator at deploy time
- Scrutinized & stamped with approval by the SLATE team and merged into the catalog



SLATE Catalog



- The SLATE Catalog ensures quality and integrity of Docker containers following a rigorous security policy for curation
- Helm Charts in the SLATE Catalog refer to container images from **trusted repositories**, such as those operated by SLATE, Open Science Grid, CERN, etc.

