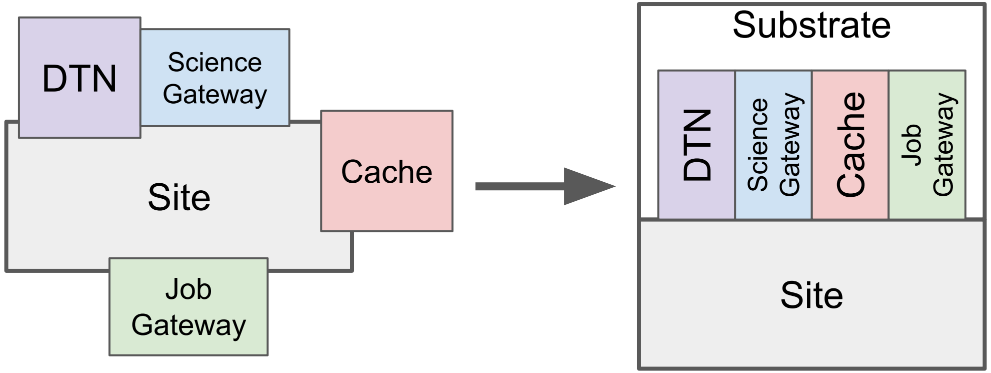
Slate—Technical, Security, and Operations Concerns

Chris Weaver for the SLATE Team

WLCG Pre-GDB Meeting
December 9, 2019

Standardizing a Service Substrate

- Currently, each piece of infrastructure added to a site tends to require
  - a person located at the site to advocate for setting it up and to manage it
  - a 'hand-built' custom installation
- By adding a consistent substrate that is common to sites and modular service components which use it, labor can be reduced
- Security challenges change, though, because instead of considering one service to permit (at a time), the site must consider the whole substrate

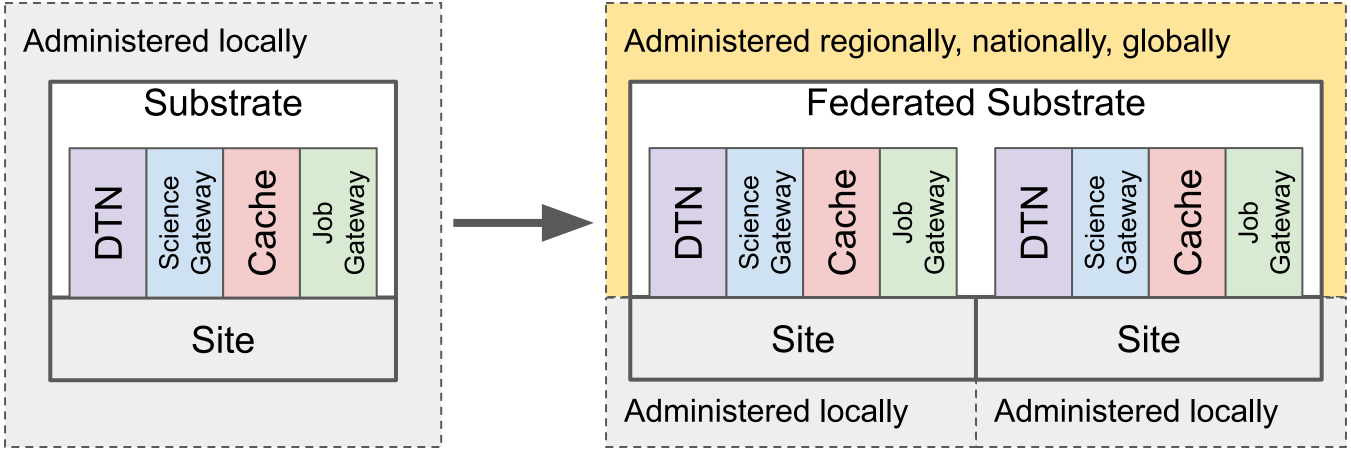


Possible Approaches

- A substrate can do more than just standardize single sites: It can be distributed, giving a single interface to address many sites
- A distributed substrate can be federated in different ways:
  - Hardware deployed at each site may be managed centrally
    - This is, as we understand it, broadly the approach taken by the Pacific Research Platform (PRP)
  - Hardware may be controlled by local site admins, who then grant fine-grained permissions to external organizations
    - This is the approach taken by SLATE
- Different methods may be better suited to different collections of sites and different end uses
  - A simpler, centralized platform probably works better for direct science uses
  - Some sites (national labs, for example), have indicated that they would require greater local control

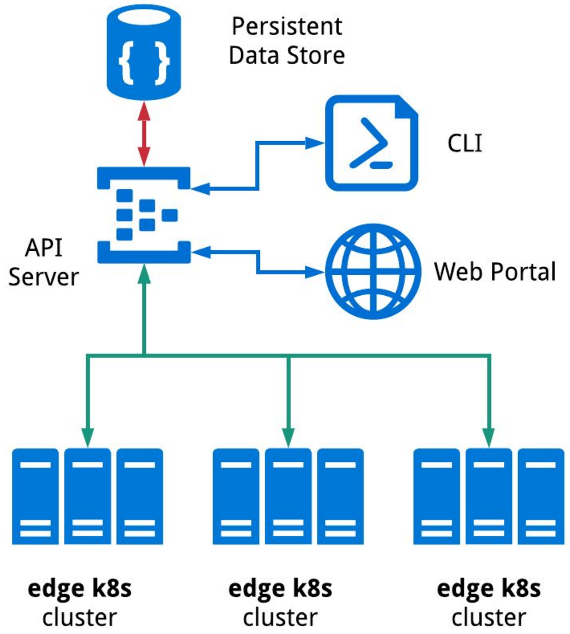
Federated platforms present new challenges

- Services have traditionally only been the responsibility of the local admin and security teams
- Broad interest in building multi-site platforms for orchestrating services means that:
  - Sites need to define or review policies for external administration of services
  - Platforms need to establish their policies for interacting with sites and define how they will use resources



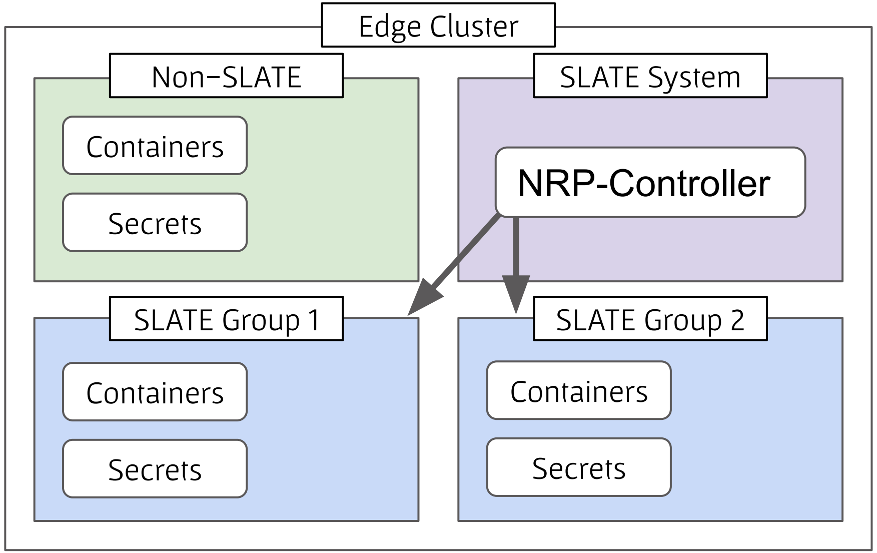
The SLATE Platform for Edge Services

- SLATE (Services Layer at the Edge) provides a substrate for this type of infrastructure
- Docker, Kubernetes, and Helm are used to package and deploy service applications
- A central server component is used to mediate user requests being sent to participating edge Kubernetes clusters
- State is stored persistently in DynamoDB, with sensitive data encrypted while 'at rest'
- Command line and web interfaces are provided



Approach to Multi-tenancy

- SLATE uses Kubernetes' namespaces, secrets, and implementation of Role-Based Access Control (RBAC)
- The SLATE API server is granted access only to its own subset of namespaces
- SLATE places applications belonging to different user groups into separate namespaces
- Kubernetes forbids containers in one namespace from reading secrets in other namespaces



https://gitlab.com/ucsd-prp/nrp-controller

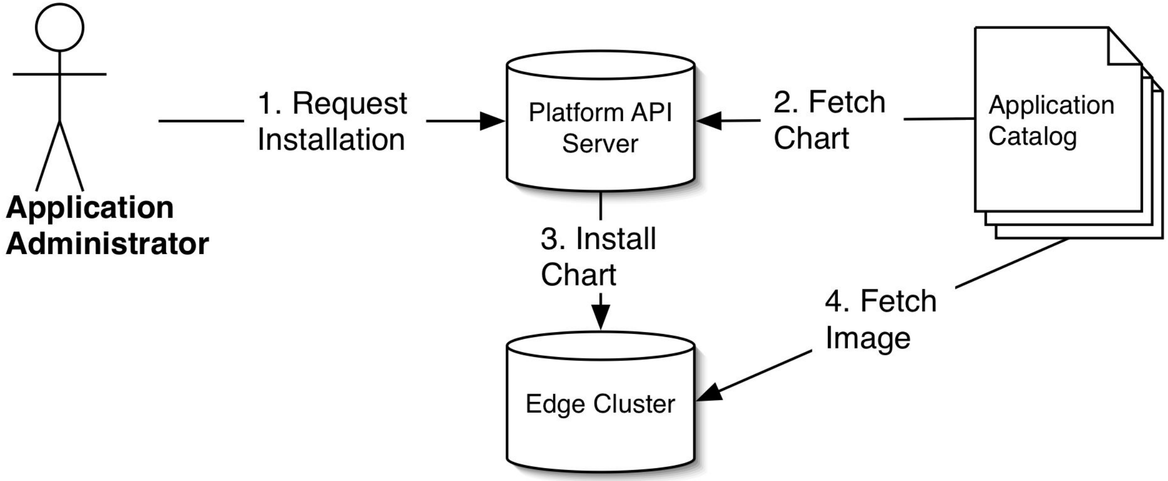
Internal Permissions Model

- SLATE organizes users into groups, and permissions apply per-group
- Every participating cluster is administered by a group
  - When a cluster first joins the federation, only its administering group has access
- The administrators of a cluster can:
  - Grant access to other groups to deploy applications on their cluster
  - Set up per-group whitelists of which applications guest groups are authorized to deploy
- The site administrator always retains the capability to directly work with the underlying Kubernetes layer to perform actions beyond what SLATE directly supports
  - This means that local admins have no restriction on inspecting, editing, or removing components if needed

Application Packaging

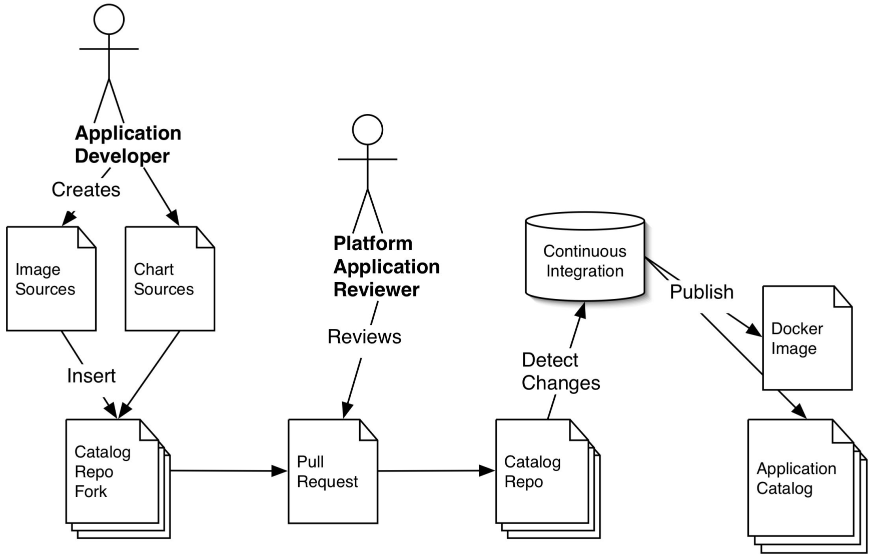
- SLATE makes use of Helm to package applications for Kubernetes
    - Helm is commonly used in the broader Kubernetes community
    - Helm enables templating Kubernetes YAML manifests for more convenient reuse
- Only limited configuration settings for each application are exposed by its Helm chart
    - Hides complexity users don't want to see
    - Can be used to enforce required aspects of configuration
    - Provides a consistent interface which all participants in the federation can inspect and agree on
- SLATE maintains its own catalog of charts, and allows only those applications to be installed

Application Install Process



- The SLATE API server mediates requests to install applications
  - Fetches applications only from the curated catalog
  - Enforces rules set by the administrators of the target cluster

Application Curation



- Much of the value of the centralized application catalog derives from the overesight applied to the applications added to it
- Some amount of human attention is required, but maximizing automation is highly desirable

Special Challenges of Container Images

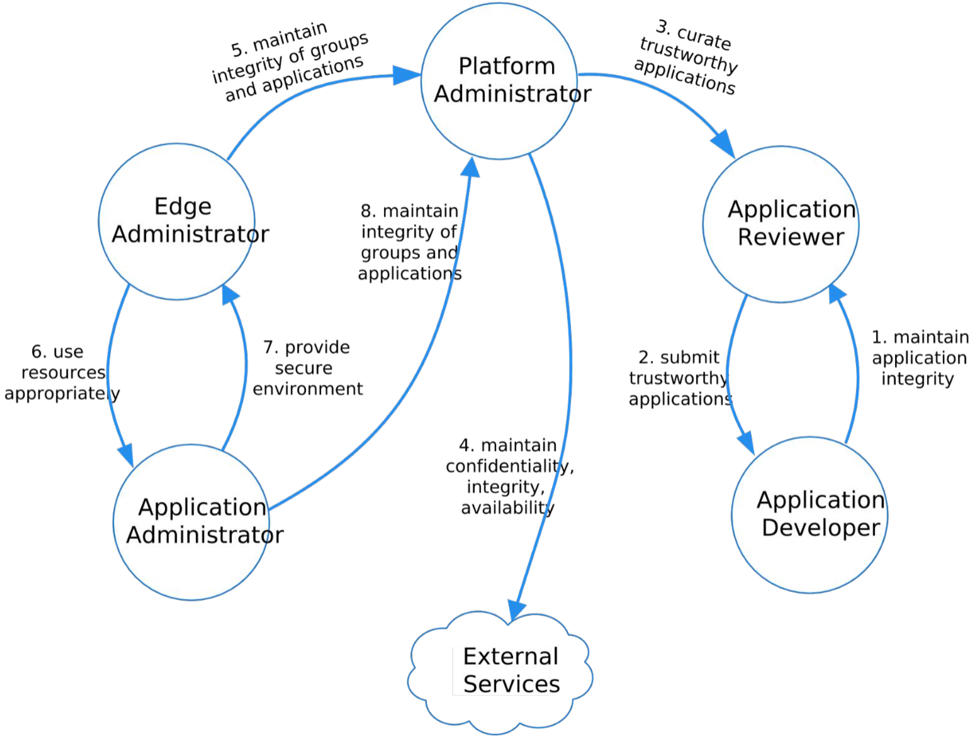
- Container images are a snapshot of a system state, so they do not tend to be aware of security patches since their creation
  - This implies that periodic rebuilding of images is necessary, and possibly that containers should be periodically restarted
- Typical distribution mechanisms (Docker) allow the data referred to by a particular image 'tag' to be replaced—an image which was previously reviewed may be replaced by one with different contents
  - This is why we prefer to have SLATE manage image sources, build and publish the images to a repository itself
- Automated image scanning tools can help with review, but are not a complete answer
  - Only images containing package manager data can be scanned
  - Scans may find large numbers of low-importance vulnerabilities for which no patched packages are available from the base distribution

Policy Development

- The SLATE Team has worked on an engagement with TrustedCI, with one major goal being to design security policies and procedures
    - The final engagement report will be available very soon
- Incident Response and Disaster Recovery have been identified as particularly critical areas
    - Incident Response, in particular can involve multiple sites, and a need to share information in a timely manner
- We think that getting these policy areas structured correctly is key building a useful platform
- Eventually, we hope to have policies which can themselves be considered sufficiently standard for broad adoption by the community
    - This means that we need to form a clear picture of what sites' concerns are

Trust Relationships for Federated Operations

As part of the work with TrustedCI, we have identified the principal actors in the system, and the trust relationships between them.


Platform Administrator
Edge Administrator
Application Reviewer
Application Administrator
Application Developer
External Services
1. maintain application integrity
2. submit trustworthy applications
3. curate trustworthy applications
4. maintain confidentiality, integrity, availability
5. maintain integrity of groups and applications
6. use resources appropriately
7. provide secure environment
8. maintain integrity of groups and applications

On-going Work

- We are still actively developing SLATE security policies. Feedback from groups like WLCG, and particularly resource providers, is key to doing this well.
- We are still adding new features to SLATE itself. Implementing resource quotas and better lifetime management for resources like storage are planned or under investigation.
- We have a WLCG working group to address edge platform security. Interested parties are encouraged to join the mailing list: https://cern.ch/simba3/SelfSubscription.aspx?groupName=wlcg-security-SLATE-wg
  - See also the WG Charter

thank you

slateci.io