Trusted CI SLATE Engagement

Final Report
January 6 2020
For Public Distribution

Andrew K. Adams, Christopher Weaver, Jim Basney,
Joe Breen, John Zage, Kay Avila, Lincoln Bryant,
Mitchell Steinman, Reinhard Gentz, Robert Gardner, Sean Peisert, Shawn McKee, Tom Barton

---

Trusted CI is supported by the National Science Foundation under Grant #1547272. SLATE is supported by the National Science Foundation under Grant #1724821. The views expressed do not necessarily reflect the views of the National Science Foundation.

Cite this work using the following information:
Andrew K. Adams, Christopher Weaver, Jim Basney, Joe Breen, John Zage, Kay Avila, Lincoln Bryant, Mitchell Steinman, Reinhard Gentz, Robert Gardner, Sean Peisert, Shawn McKee, and Tom Barton, "Trusted CI SLATE Engagement: Final Report", January 2020.

Table of Contents

1 Introduction

In the second half of 2019, Trusted CI and the Services Layer at the Edge (SLATE) projects collaborated on developing a cybersecurity plan for the SLATE system.

SLATE[1] is funded by an NSF grant from the Office of Advanced Cyberinfrastructure (Award #1724821). SLATE aims to accelerate collaborative scientific computing through a secure container orchestration framework focused on the Science DMZ, enabling creation of advanced multi-institution platforms and novel science gateways. The ATLAS collaboration at the CERN Large Hadron Collider has an R&D program utilizing SLATE to centrally operate a distributed data delivery network having service endpoints at multiple computing facilities in the U.S., CERN, the UK, the Czech Republic and Germany, and has evaluated a cache deployed using SLATE within the ESnet backbone in Sunnyvale, California. Similar approaches are already in production (the Open Science Grid data federation using StashCache which is implemented in part using the Pacific Research Platform and Internet2 supporting LIGO, DUNE and other scientific collaborations) but as yet lack generalized trust frameworks. While innovation of the new trust model initially is occurring in the context of the OSG and the worldwide LHC computing grid (WLCG), trusted federated edge infrastructures enabling operation of advanced computing platforms will in future be necessary to sustain a wide range of data intensive science disciplines requiring shared, scalable national and international cyberinfrastructure.

The deployment and operation of software through containerized edge services raises issues of trust between many stakeholders with differing perspectives. Resource providers require guarantees that services running within their infrastructure are secure and operated within site policies; platform service developers and operators require flexibility to continuously deliver and compose new cyberinfrastructure supporting their scientific collaborations; edge cluster administrators need visibility and operational awareness while delegating some of their traditional deploy and operate responsibilities to centralized platform teams, following a "NoOps" model[2]; and finally, the application workloads from end-user science communities rely on the foundational capabilities implemented by platform services to realize the full potential of shared cyberinfrastructure. This engagement broadly focused on developing SLATE's cybersecurity program in a way that balances these needs.

The remainder of this report is organized as follows. In Section 2, we summarize the results of our community engagement activities, including discussions at the NSF Cybersecurity Summit and in the newly formed WLCG SLATE Security Working Group[3]. In Section 3, we review the security policies that we developed during the engagement and those that we identified for future work. In Sections 4-5, we provide a security analysis of the SLATE platform, with a discussion of trust relationships (Section 4) and system workflows (Section 5). In Section 6, we discuss the challenge of assessing container security and

[1] https://slateci.io/ , https://www.nsf.gov/awardsearch/showAward?AWD_ID=1724821

[2] The meaning of "NoOps" by the SLATE team relates to a federated edge operations model that reduces the effort required by local site administrators operating services as part of a multi-site infrastructure (e.g. a "grid"). Similar terminology has been coined in cloud computing, c.f. https://searchitoperations.techtarget.com/definition/NoOps

[3] At the time of this report, there is discussion within the WLCG Security team to rename the group to the "WLCG Federated Operations Security Working Group", generalizing the scope beyond the specific implementation of the SLATE platform.

review available container security tools that can help. In Section 7, we discuss related work and identify the novel security aspects of the SLATE federation versus those aspects that can borrow from existing security policies in scientific computing (e.g., in OSG and WLCG). Section 8 provides conclusions and next steps, and the Appendix provides additional details on roles in the SLATE federation.

2 Community Engagement

An important goal for the SLATE security program is to address the concerns and requirements of SLATE stakeholders. To help meet this goal, we solicited community input on our plans via multiple channels.

In collaboration with WLCG, OSG and ESnet, we formed the WLCG SLATE Security Working Group.[4] In a working group kick-off meeting[5] at Fermilab, co-located with the WLCG Grid Deployment Board meeting, we set the charter for the working group with the following charge:

> The main challenge to be addressed is to document a trust model for centralized service orchestration capability across WLCG centers ("federated NoOps") to enable efficient operation of WLCG computing services and innovation of new platforms in support of HL-LHC software development.
>
> This Working Group aims to clearly articulate entities and processes which implement such capabilities. The methods and trust relationships will be described in documents (both existing and to be written) such as service level agreements and security policy documents, including security incident response and traceability. The trust model enables delegation of the service operator responsibility by the resource provider.

The Trusted CI SLATE engagement provided input to the ongoing efforts of the Working Group by documenting the SLATE trust model and drafting security policy documents. Trusted CI personnel will continue their involvement in the Working Group going forward.

At the NSF Cybersecurity Summit,[6] we presented during the WISE (Wise Information Security for collaborating E-infrastructures) community workshop[7] and during the Summit plenary session. The presentations about the WISE security model and security program kicked off a productive discussion among stakeholders representing NSF projects (LIGO, OSG, IceCube, Gemini), DOE (ESnet), and international collaborations (CERN, WLCG). Discussion take-aways included:

- Some attendees noted that they have found Trivy to be a useful container security scanning tool (see Section 6).
- The SLATE "governance model" is key to acceptance. The model should clearly articulate trust relationships and responsibilities (see Section 4).

[4] https://trustedci.org/slate
[5] https://indico.fnal.gov/event/21485/
[6] https://trustedci.org/2019-nsf-cybersecurity-summit/
[7] https://wiki.geant.org/display/WISE/WISE+@+NSF+Cybersecurity+Summit+2019

- High priority areas for security program documentation are the application vetting process and the logging/traceability policy (see Section 3).

We also led a discussion about SLATE on the October 2019 Large Facilities Security Team[8] call. Discussion topics included the sustainability plan for the SLATE platform beyond grant funding and the variety of application needs across the different NSF facilities (with common software like CVMFS and Globus used by multiple facilities).

These discussions also identified the need for an ongoing forum for stakeholder engagement. While the WLCG SLATE Security Working Group has been initially focused on WLCG requirements, the working group chairs agreed to open the group to participation by the wider scientific community. Interested parties should visit https://trustedci.org/slate for details on joining the group.

3 Security Policies

To develop policies for the SLATE security program, we adopted templates from Trusted CI's Guide to Developing Cybersecurity Programs for NSF Science and Engineering Projects.[9] In particular, we filled in the Master Information Security Policy & Procedures (MISPP) Template to provide a top-level view of the SLATE security program and to identify underlying policies that should be part of the overall program. We also supplemented the list of needed policies based on community input (with the application review policy and the logging/traceability policy being identified during the WISE workshop discussed in Section 2). All public portions of SLATE's security policies and procedures will eventually be published at https://slateci.io/docs/security-and-policies/, in addition to being displayed as needed during particular workflows on the platform.

Work on policy documents was prioritized largely by which were judged to be the most critical to defining how the federation will operate and outline necessary trust relationships. This specifically included the asset management policy (which was incorporated into the MISPP), incident response policies, disaster recovery policies, and Edge Administration/Federation policy. The last of these policies was an addition to the types of policies contained in the Trusted CI guide which was determined to be necessary for this project. The Application Review Policy is another type of policy specific to SLATE, which had been previously identified as a necessity, but due to lack of resources it was not further developed during the engagement with Trusted CI.

Table 1 summarizes the current state of SLATE security policies as of December 2019. During the engagement, we focused on developing the four highest priority policies: 1) the Master Information Security Policy & Procedures (MISPP) document that gives a high-level organization security program, 2) the Acceptable Use Policy that provides a policy framework for users, 3) the Asset Management Policy (included in the MISPP) that identifies the assets being protected by the security program, and 4) the Incident Response Procedures that prepare the SLATE operations team for handling any violations of the security policy. As of December 2019, these four policies are mostly drafted and are under review by

[8] https://trustedci.org/lfst
[9] https://trustedci.org/guide

SLATE leadership. 13 additional policies are planned for development in 2020, including policies identified by the Trusted CI Guide plus those identified through community input (Edge Admin/Federation Policy, Application Review Policy, and Logging/Traceability Policy). Included in that list are the Mobile Compute Policy (about managing sensitive credentials on laptops and other mobile devices) and the Remote Access Policy (about using secure methods for remote system administration). Of all the policies recommended by the Trusted CI Guide, we decided that a formal Training and Awareness Policy was not an immediate priority, due to the small size of the SLATE team.

Under Review	Planned	Not Planned
Master Information Security Policy & Procedures Acceptable Use Policy Asset Management Policy Incident Response Procedures	Disaster Recovery Policy Access Control Policy Edge Admin/Federation Policy Application Review Policy Information Classification Policy Logging/Traceability Policy Personnel Exit Checklist Mobile Compute Policy Network Security Policy Password Policy Physical/Environmental Security Policy Privacy Policy Remote Access Policy	Training and Awareness Policy

Table 1. Four policies were drafted during the engagement and are now under internal review. Additional policies are planned for 2020. A formal Training and Awareness Policy is not currently prioritized due to the small size of the SLATE team.

Defining Security Roles

A separate development which arose from work on SLATE's security policies was the definition of a new operation role within the SLATE project/platform, namely that of 'Security Operations Staff'. The addition of this role was desirable in order to support the role of Information Security Officer being a part time and primarily advisory role within the project. The expectation is that the Information Security Officer will apply broad expertise to guide the development and application of the security policies and procedures, while the Security Operation Staff will use specific technical knowledge of SLATE's component systems to to design and implement the policies and procedures. In particular, the latter group will hold credentials to carry out security procedures on production systems, but will coordinate with the Information Security Officer to do so.

Defining the Audience for the Acceptable Use Policy

One issue which was raised in connection with the draft Acceptable Use Policy for SLATE and has not been fully resolved is the scope of the phrase 'using resources associated with SLATE', which is used to describe the conditions requiring adherence to the policy. The phrase, and the policy, is intended to cover the use of the particular information systems which make up the SLATE reference platform, and

this is reasonably clear from context, except that public source code for SLATE software components could also be interpreted as such a resource. It is not the intention of the policy that individuals or organizations who use the SLATE software to operate their own platform(s) should be required to agree to this particular policy, or be bound to the SLATE developers in any way besides the software licensing of the SLATE components themselves. The SLATE team concluded that some language should be added to clarify this distinction.

Maintaining Policy Documents

For the purpose of integrating the Acceptable Use Policy specifically and other policies generally with existing SLATE workflows, such as the new user sign-up process, the SLATE team concluded that keeping policies in the form of Google Documents (as provided by Trusted CI templates) is not suitable, or at least not sufficient. The primary concern is that Google Documents are not a suitable format from which policy text can be programmatically obtained for display at appropriate times to users. The shared Google Documents are, however, a convenient form for drafting policy documents, including commenting and review by team members. The proposed solution is that when a policy is adopted or updated, it will be translated into a markdown document and stored in a version control repository from which SLATE software can programmatically obtain the current versions of all policies. No new material would be added to the 'published' markdown forms of the policy documents; instead, changes will be made to the original Google Documents and merged into the markdown version stored in version control only when a formal policy update is decided upon.

4 SLATE Trust Relationships

In addition to beginning to develop security policies and procedures for SLATE, we also performed a security analysis of the SLATE system, with particular emphasis on the federated nature of SLATE, where system components and operational responsibilities are distributed across multiple organizations and roles. In this section, we provide a summary of our analysis of the trust relationships between the SLATE Roles, and in the next section we identify the key “workflows” that illustrate how the SLATE Roles work together to fulfill SLATE use cases. We expect this analysis to be useful input to the ongoing risk assessment discussion in the WLCG Slate Security Working Group. A detailed description of each role is provided in the Appendix.

As explained in the Introduction, the deployment and operation of software through containerized edge services raises issues of trust between many stakeholders with differing perspectives. Understanding how trust is distributed across the SLATE Roles helps the security analyst understand how security requirements are met across the system and identify where security controls should be applied to fulfill trust obligations. Identifying the trust relationships between components also helps to identify impacts of security risks, i.e., the downstream effects on system security if one component in the system is compromised. We identified 8 trust relationships, as illustrated in Figure 1 and specified in the following listing.

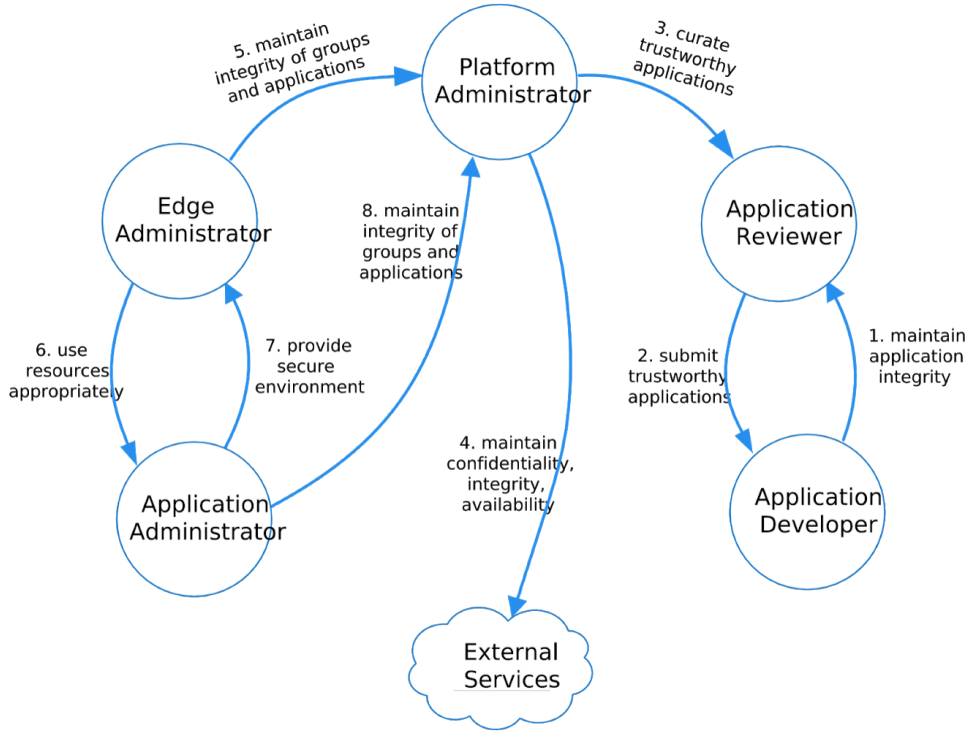

Figure 1. This entity relationship diagram provides an overview of the trust relationships between SLATE Roles.

Application Developer

1. Application Developer trusts the Application Reviewer to maintain the integrity of submitted applications.

Application Reviewer

2. Application Reviewer trusts the Application Developer to submit trustworthy applications.

Platform Administrator

3. Platform Administrator trusts the Application Reviewer that applications submitted are non-malicious.
4. Platform Administrators trusts external services (e.g., Cloud file-stores) to maintain the availability and integrity of backed-up data, as well as the confidentiality of that data requiring it (based on its classifications).

Edge Administrator

5. Edge Administrator trusts the Platform Administrator to maintain the integrity of groups and the integrity of applications committed by Application Reviewer.

6. Edge Administrator trusts the Application Administrator to only execute applications and groups previously white-listed.

Application Administrator

7. Application Administrator trusts the Edge Administrator to maintain the integrity of data, to maintain services availability, and to adhere to security best practices.
8. Application Administrator trusts the Platform Administrator to maintain the integrity of applications committed by Application Reviewer, to maintain the integrity of groups, and to maintain service availability.

In addition to the above trust relationships between SLATE stakeholders, we also note that members of each Group are trusted equally, so for example, all members of an Application Administrator group trust each other to administer the application in a consistent manner.

5 SLATE Workflows

To better understand the security properties of the SLATE system, in this section we document key SLATE workflows, i.e., user and administrator actions, that have trust implications. For each workflow, we identify potential risks that may be explored in a follow-on, in-depth risk assessment.

Workflow 1: Edge Provider Registers CI with SLATE

A SLATE Edge Administrator desiring to add their Kubernetes cluster creates an account on the SLATE Dashboard, then obtains their access token from the dashboard. They then download the SLATE client software from the Jenkins artifact server and install it on their Kubernetes cluster. Using the client software, they submit a request through the API Server to join the cluster to the federation, specifying which existing group will administer the cluster. This process registers the SLATE Edge Provider to SLATE to be administered SLATE Group and cuts self-signed certificates for said Edge Provider.

Risks:

- For CI owners, who are trusting that their CI is used appropriately & securely:
    - DoS, AUP violation, unauthorized access, network bandwidth (perhaps related to DoS), reputation, contention of resources (not really concerned with side-channel issues)

Workflow 2: SLATE Application Developers Submit Applications or Application Updates to SLATE

SLATE Application Developers desiring to add their containers to SLATE first must produce a Helm chart. They then submit their Helm chart to GitHub as a Pull Request to place the chart into the 'incubator' directory within the repository, which functions as a sort of 'holding area' for charts which are not yet finalized. This triggers an automated continuous integration process running on the Jenkins VM that vets the Helm chart for correct syntax. A SLATE Application Reviewer is notified of the pull request, and can begin the process of manually vetting the container(s) Helm chart specifices. The chart is manually moved from the incubator to the stable folder via a commit after the vetting process is complete and

any necessary changes have been made and tested. At which point, the CI tool builds and publishes the chart (making it available for download via HTTPS on the Jenkins VM), as well as building and publishing the container image to Dockerhub.

Risks:

- To Application Developer:
  - Reputation (based on code being modified after upload, i.e., integrity)
- To SLATE Core Team/Application Reviewer
  - Buggy/malware code being submitted
  - Reputation

Workflow 3: SLATE Group Requests Application Service to Execute

When a SLATE Group desires to execute a Service Application, they connect to the SLATE Platform's API Server via Globus and request execution of said service. Note, once a user has an active account, a token can be obtained and used through the CLI client. If the SLATE Group is whitelisted at the SLATE Edge Provider specified in the request, then the API server (i) installs the Application Service to the kubernetes cluster(s) if the service is either not available or a new version exists, and (ii) executes the Application Service. SLATE Group users can check on the status of their requests via the API Server.

Risks:

- For Application User:
  - Loss of science due to tampered/buggy software
  - Reputation (if application misbehaves)
  - Confidentiality (of embargoed data)
- For CI Provider:
  - DoS
  - Reputation
- For SLATE:
  - Reputation
  - Lost Functionality
  - Lost time of work troubleshooting

Workflow 4: Group Creation and Management

Any SLATE user can create a Group (or multiple groups). All members within a group have equivalent privileges to administer the Group and its resources, which include any Edge Clusters it has joined to the Federation and any Application Instances it has launched. Any member of a Group may add Users to or remove users from the Group.

Risks:

- Users could be incorrectly added to/deleted from a Group

Workflow 5: Adding Group to Whitelist to Run Resources on Edge Cluster

After registering a group with a platform's API server, the Application Administrator queries the API server to find the contact(s) for the edge cluster(s) the individual is interested in running platform applications on. After finding out the contact email addresses, the group administrator requests out-of-band permission to the Edge Cluster administrator to be added to the whitelist that controls the platform's operation on the edge cluster. The Edge Cluster administrator adds the group to the whitelist using the platform API server, potentially restricting what applications are allowed to be run.

Risks: Edge Administrator could whitelist a malicious actor.

Workflow 6: Maintenance of SLATE Federation

Platform Administrator configures and operates the federation, including API server, Jenkins, DB service, etc. The meta-data involved with the platform must be backed-up on a file-store for redundancy. This workflow is behind-the-scenes for the other stakeholders, but they still have a stake in it being done securely.

Risks:

- The file-store exposes or loses critical/sensitive data.

Workflow 7: DR & State Preservation Via Cloud

SLATE Admins preserve the *state* of SLATE users, groups, and application instances by leveraging DynamoDB service within AWS. The uploads are done through HTTPS and AWS programmatic authorization using credentials dispersed by Amazon. Admins with superuser privileges and specific applications on the API server have access to the AWS keys. All Edge Provider information, Platform Provider and Application Administrator requests are stored within the database.

Risks:

- For SLATE:
    - Sensitive credentials
- Application Developers:
    - Sensitive credentials
        - Could allow unauthorized access to science

6 Container Security Scanning Tools

As described in Section 5, the Application Reviewer has the important responsibility of vetting containers submitted by an Application Developer for inclusion in the SLATE application catalog. Container security scanning tools can potentially help the Application Reviewer with this task, by automating some of the vetting, but this is only effective if the output of the tool(s) is not overwhelmed by false positives (non-issues or very low impact issues) that make it too tedious for the Application

Reviewer to review. Ideally the Application Developer could review the tool output prior to submitting the container for review. Scanning tools can also perform the function of identifying newly discovered vulnerabilities (Common Vulnerabilities and Exposures and similar catalogs) in existing containers, prompting the responsible Application Developer to submit an update.

Prior to the engagement, the SLATE team already had some initial experience with the Clair[10] and Trivy[11] tools. As part of the engagement, we performed additional tests with these tools two tools plus Anchor.[12] All three tools were effective at identifying CVEs when package metadata was available in the container, with Trivy producing the most complete results. While our experiments were not exhaustive, our initial conclusion was that integrating Trivy into the Application Reviewer workflow would provide value. During the NSF Cybersecurity Summit we learned that Trivy was positively evaluated by other WISE workshop participants, which provided an additional vote of confidence for selecting that tool.

7 Related Work

When analyzing the SLATE security model and developing the SLATE security program, we found it valuable to compare with existing security infrastructures for distributed scientific computing, to identify similarities (i.e., not reinvent the wheel) and differences (i.e., what is uniquely new about the SLATE security model). Comparing with current Open Science Grid (OSG) and WLCG security models was most valuable, so we summarize our findings in this section.

Like SLATE, Open Science Grid is formed of a distributed set of Resource Provider sites operating software provided by a central team and coordinated by federation-wide security policies and security operations. Currently, OSG distributes software in RPM packages that system administrators at the sites manually install, configure, and maintain. This is a key difference from SLATE, which allows Application Administrators to launch containers at the sites (called Edge Providers). Existing OSG policies[13] that are applicable to SLATE include the Service Container Security Policy and the Security Incident Response Policy.

WLCG is an international scientific computing infrastructure that operates on top of OSG, the European Grid Infrastructure, and other scientific computing infrastructures. Existing WLCG policies[14] that are applicable to SLATE include the Security Incident Response Policy, Security Traceability and Logging Policy, and Security Policy for the Endorsement and Operation of Virtual Machine Images. These WLCG policies are an important reference point for SLATE as the project continues to develop its corresponding policies.

8 Summary and Next Steps

[10] https://coreos.com/clair/docs/latest/
[11] https://github.com/aquasecurity/trivy
[12] https://anchore.com/kubernetes/
[13] https://opensciencegrid.org/security/OSGSecurityPolicies/
[14] https://wlcg.web.cern.ch/computer-security

In the Trusted CI SLATE engagement, we performed an overall security analysis of the SLATE platform, identified trust relationships and key user/administrator workflows, identified a set of needed security policy documents, and began drafting the security policies. We also evaluated container security tools, explored existing applicable OSG and WLCG security policies, and gathered community input on the SLATE security program, resulting in initial consensus around the security policies and procedures needed to enable wider adoption of the SLATE platform.

Community-driven work on the SLATE security program continues through the WLCG SLATE Security Working Group, which is open to all who are interested. Visit https://trustedci.org/slate for pointers to current status of the working group and https://slateci.io/docs/security-and-policies/ for pointers to current SLATE security policies as they are developed.

Appendix A. Terms and Definitions

Organizational Roles

These are the organizations that come together to participate within the project.

SLATE Platform Provider

The organization that manages a particular installation of a SLATE platform, comprised of an instance of the SLATE API server, the SLATE web portal, and some number of federated edge clusters. Any organization can create an independent SLATE platform with its own edge clusters.

Examples:

- The SLATE team for the SLATE Reference Platform.
- A computing consortium or virtual organization which has adopted the SLATE software to assist with service orchestration across multiple sites, creating their own operational fabric of services.

Responsibilities:

- Informs resource providers, application administrators, and application developers of operational roles, responsibilities and risks associated with participation in the service federation.
- Handles review and curation of applications.
- Define and apply policies governing use and operation of the federation.

SLATE Edge Cluster Provider

An organization that hosts an edge cluster which has been federated with a given instance of a SLATE platform.

Examples:

- A university research computing organization
- An experiment facility which may operate edge cyberinfrastructure to deliver data to distributed processing or analysis centers.

Responsibilities:

- Identify the SLATE edge cluster administrator(s) who will administer the resources within the federation.

SLATE Group

The SLATE platform handles permissions and authorization through the use of SLATE Groups. A SLATE User's group membership will determine the resources that user is authorized to administer. Authorized resources might be Edge Clusters, Application Instances, or SLATE Secrets. Each member of a SLATE

group has the same access to that group's application instances, SLATE Secrets, and Edge Clusters. A SLATE User may belong to multiple groups which may have related, or unrelated purposes.

Examples:

- A research collaboration team which deploys a computing infrastructure service application
- Edge Cluster Administrators who manage one or more Edge Clusters for a single Edge Cluster Provider

Responsibilities:

- Authorizing other Groups to act within namespaces on any Edge Clusters administered by the Group
- Managing any Service Applications operated on other Groups' Edge Clusters

SLATE Core Team

The organization that develops the SLATE reference platform, software, and its associated policies.

Responsibilities:

- Providing software solutions, hardware integration, documentation, and support
- Operating and maintaining the SLATE Reference Platform
- Developing and continually testing the security posture of the SLATE platform
- Supporting other SLATE Platform Providers

Individual Roles

These are individuals that are part of the aforementioned organizations.

SLATE Core Developer

A person within the SLATE team carrying out development efforts.

Responsibilities:

- Developing the suite of SLATE services
- Evaluating and integrating suitable hardware (servers, switches)
- Evaluating and integrating suitable software

SLATE Platform Administrator

A person (possibly one of several) designated by a Platform Provider to operate the components which make up the SLATE platform.

Responsibilities:

- Monitoring the platform for software and hardware failures, notifying Edge Cluster Administrators of local issues as necessary

- Upgrading SLATE platform components as appropriate
- Deploying and maintaining additional central services, such as out-of-band logging and backups, as required by policy
- Serves as operations point of contact, including platform security

SLATE Platform Security Administrator

A Platform Administrator specifically tasked with carrying out security-related operations.

Responsibilities:

- Serves as a point of contact for security
- Carries out security policies in conjunction with the Information Security Officer

SLATE Platform Information Security Officer

The Information Security officer is the central coordinator of platform security efforts. This role reports to the leadership of the Platform Provider, and directs the Security Administrators as needed.

Responsibilities:

- Advising on the content of platform security policies and procedures
- Serving as a point of contact for security concerns
- Advising and coordinating activity by the Platform Security Administrators to carry out security policies and procedures

SLATE Edge Cluster Administrator

A person (possibly one of several) designated by an Edge Cluster Provider to administer an edge cluster which participates in a SLATE federation. This administration includes management within the SLATE federation using SLATE-provided tools, and traditional administration of the cluster.

Responsibilities:

- Supporting hardware and local software, up to and including Kubernetes on edge clusters owned by the Edge Cluster Provider to which the administrator belongs
- Joining the Edge Cluster to the SLATE federation running the installer
- Managing access permissions for other groups on the SLATE platform to the Provider's cluster(s)
- Responding to support issues raised by Application Administrators who have been granted access to run applications on the Edge Cluster
- Responding to security issues raised by other entities involved in the SLATE Platform, particularly the Platform Provider

SLATE Application Developer

A person who takes an existing application and packages it in a way that it can install on a SLATE platform and submits it for inclusion in the platform's application catalog. Being an application developer is independent of other possible roles in the SLATE platform.

Responsibilities:

- Submitting the application for review, and responding appropriately to questions or concerns raised in the review process
- Responding to support or security issues raised by other SLATE users or by the Platform Provider in connection with the application

SLATE Application Administrator

A person who operates an application on a SLATE platform. This requires being a member of a SLATE group, so administration is done on behalf of that group.

Responsibilities:

- Operating an application according to the needs of a group (which represents some large entity, such as a computing center or research collaboration)
- Responding to support issues raised by the end users of the service provided by the application
- Responding to security issues raised by the Platform Provider or the Edge Cluster Provider on whose cluster the application runs

SLATE Application Reviewer

A person who reviews a new or modified SLATE application for inclusion in the SLATE platform catalog after it has been submitted by an Application Developer. Application reviewers are selected or granted authority by the Platform Provider.

Responsibilities:

- Ensure that submitted applications meet quality and security requirements
- Approve (or provide feedback on) applications in a timely manner

Hardware Components

These are physical resources provided by the different organizations within the project.

SLATE Reference Platform (SRP)

The SLATE platform managed by the SLATE Core team which serves as a technical reference and testbed for all production SLATE platform installations.

SLATE Platform

A collection of SLATE edge clusters used as a single entity via federation with an instance of the SLATE API. There can be more than one SLATE platform, and an edge provider may decide to join more than one.

SLATE Edge Cluster

An edge cluster at a particular site in a single administrative domain which participates in a SLATE federation, owned by a SLATE Edge Cluster Provider and managed by one or more SLATE edge administrators.