Identity Integration Solution Overview

SLC Project Documentation

May 4, 2012

Revisions

Date	Version	Name	Change Summary
11/6	1.0	Janko Bazhdavela	Internally reviewed Draft
12/16/11	1.1	Janko Bazhdavela	Updated Draft
1/10/12	1.2	Janko Bazhdavela	Updated Draft
5/4/12	1.3	Stephen Cole, Janko Bazhdavela	Major edits in the Federation process. Added details for Application On-boarding, Application Authorization. Minor edits in Roles and Permissions overview and Default Roles and Permissions.

Table of Contents

Table of Figures

1 Purpose and Scope

The purpose of this document is to describe how SLI will implement identity integration with State Education Agency (SEA) and Local Education Agency (LEA) directories through Federation, how the users are authenticated and authorized on the system, and to outline the application authentication and authorization process. In addition, the Identity Integration Solution document outlines the default SLI roles and permissions that will be available to SEAs and LEAs for assignment to users.

The content of this document addresses the scope for the SLI Alpha release.

The intended audiences of this documentation are mainly District/State IT Administrators, Application Developers, and SLI Operators.

This document extends the *Identity Integration and Management Overview Whitepaper* (published 9/17/11) in the areas of SEA/LEA Directory integration through Federation and Federated Authentication design, including web application Authentication, user data access Authorization, and default user Role and Permission management.

For an overview of the SLI Architecture, SEA/LEA registration, onboarding and opt-out, creation and management of Super-Administrator Roles, application approval and deployment, custom Roles, and Delegated integration strategy, please refer to the *Identity Integration and Management Overview Whitepaper* as a primary resource.

2 Definitions

This section contains definitions of terms that are used in this document.

Authentication The process of verifying that a user is who they claim to be. For example, I am "*jsmith*" because I entered the correct username and password for the *jsmith* account. SLI supports Federated Authentication.

Authorization The process of determining what resources a user has permission to access and what actions the user has permission to do. For example, the user *jsmith* is allowed to view data associated with students in his classes.

Data Category Predefined list of student data fields

Delegated Authentication SLI uses an Institution's Directory to authenticate a user; however, that identity is not shared with other applications such as the District portal. With delegated authentication, user identity is controlled in one place (the Institution's Directory), but users will need to login for each application.

Directory A service that manages user identities and those users' Roles. Roles in turn control access to specific resources through their assigned Permissions. Examples of commonly used Directories include Microsoft Active Directory, OpenLDAP, and Novell's eDirectory.

Federated Authentication A means of linking an individual's identity and attributes across multiple distinct systems. Federation is a common means of providing Single

Sign-on functionality. For example, by using Federation, the identity *jsmith* may be used to login to an SLI application and a District portal. Federation standards allow *jsmith* to login once to the District portal and have that identity automatically shared with the SLI application so that the user need not re-enter their password. SLI enables Federation through the industry standard Security Assertion Markup Language (SAML).

Identity Provider A computer system that stores user identity information such as user names, passwords, and roles, and provides a way for other computer systems to access this data. An Identity Provider may be a Directory or it may be a service acting on behalf of one or more Directories. In a SAML federated environment, a SAML Identity Provider produces SAML assertions.

Institution A School, a School District, or a State.

Permissions A set of actions that an actor is allowed to take in SLI (*for example*, "Can see student assessment data for students the user teaches" or "Can change administrative setting for an account"). Permissions are assigned indirectly to specific users through Roles.

Role A pre-defined relationship between a user a specific set of Permissions. Roles within SLI will generally correspond to an individual's function within an Institution, for example, "teacher" or "principal."

Service Provider (SP) An entity that controls access to services and resources. In a SAML federated environment, a SAML SP creates and consumes SAML assertions.

SLC Shared Learning Collaborative, LLC

Super-Administrator The Role description for a user who has been assigned a set of Permissions within an Institution that grant the user complete administrative control over all data within SLI that is associated with that Institution.

Trust When the SLI identity provider accepts the validity of users logged into a SEA/LEA Directory, it is said to Trust this Directory.

UID Unique Identifier

3 Directory and Identity Integration

This section provides a conceptual overview of the directory and identity integration process.

3.1 Overview

In order for an SEA/LEA to achieve authentication integration and/or Single Sign-On (SSO) with SLI, they need to have a Directory (or set of Directories) that stores all of the user identities and roles that will access SLI and govern their permissions within SLI (as

illustrated in Figure 1). This Directory will need to be integrated with SLI and must contain identifiers for each user that map to the identifiers of those users in the SLI data store. After Directory integration is established, when end users log in to the SLI portal or any SLI application, their identity will be authenticated by the SEA/LEA Directory, not by the SLI system itself. The SEA/LEA Directory will verify that the username and password credentials that were supplied are valid and will return this information to SLI for further access control and authorization/auditing purposes.

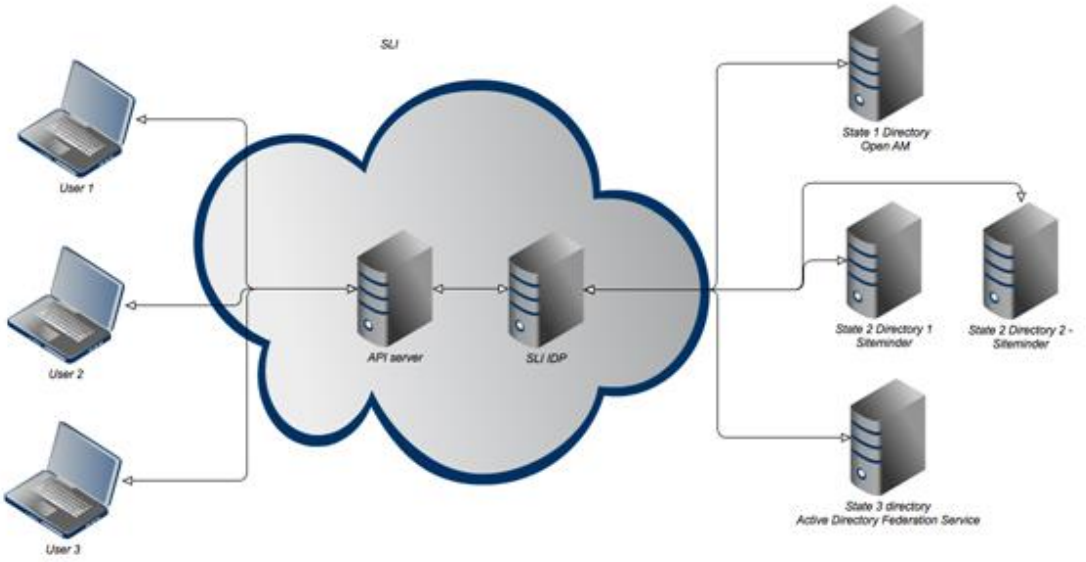

Figure 1 – Overview Diagram

It is expected that the Alpha Release Institutions will provide one or more Directories that they manage and host. However, future SLI versions may provide an option for SLI to host a Directory that contains the Institution's users. This scenario may be appropriate for Institutions that do not have Directory servers that can be easily integrated into the SLI system.

3.1.1 Guiding Security Principles

Ensuring that only authorized individuals can gain access to the information that they are authorized to view or change is of the upmost importance. Therefore, the following security controls will be strictly enforced by SLI:

- All traffic between the SLI platform and the SEA/LEA directories will be encrypted.
- All login activities that include the user id and password will be encrypted (SEA/LEA identity providers are largely responsible for this).

- All login activity (both success and failure) will be logged and reviewed (SEA/LEA identity providers are largely responsible for this).
- All failed access attempts to data will be logged for audit purposes and to identify access model issues.
- All application and user access to the SLI API will require authentication. No anonymous access will be allowed.
- Controls will be established to ensure that session identification mechanisms will be protected to ensure that a session cannot be compromised.

3.2 Federated Directory Integration Strategy

SLI will support end-user authentication through Federation. In this approach, the SEA/LEA is responsible for authenticating the end user and managing the authentication credentials. SLI will serve as both a native Service Provider (application that requires authentication) and conduit for other Service Providers (applications) that make use of SLI. SLI will also provide interfaces/standards and suggest business rules and best practices for authentication to authorization mapping. When logging in to SLI, a user is prompted to provide the username and password directly to the SEA/LEA identity provider (the combination of Directory and SSO). This is done by using the SAML 2.0 protocol. The identity provider authenticates that the information is correct, and returns a SAML 2.0 assertion to the service provider (SLI) that indicates that the user supplied the correct credentials. Role information for that user, as well as a state-wide UID, are provided as an attribute of the SAML assertion.

Delegated Authentication will not be implemented for the SLI Alpha release, but will be provided in a future release based on site need.

3.3 Federated Authentication Design

The federation is a group of identity providers that the SLI identity provider trusts. All communication within the circle of Trust is provided by SAML 2.0, which abstracts the idiosyncrasies of individual Directory providers and configurations from the application developer and the SLI API. The Trust relationship is one-way, that is, S/LEA directories will not allow SLI users to log in to S/LEA-specific systems that are not related to SLI.

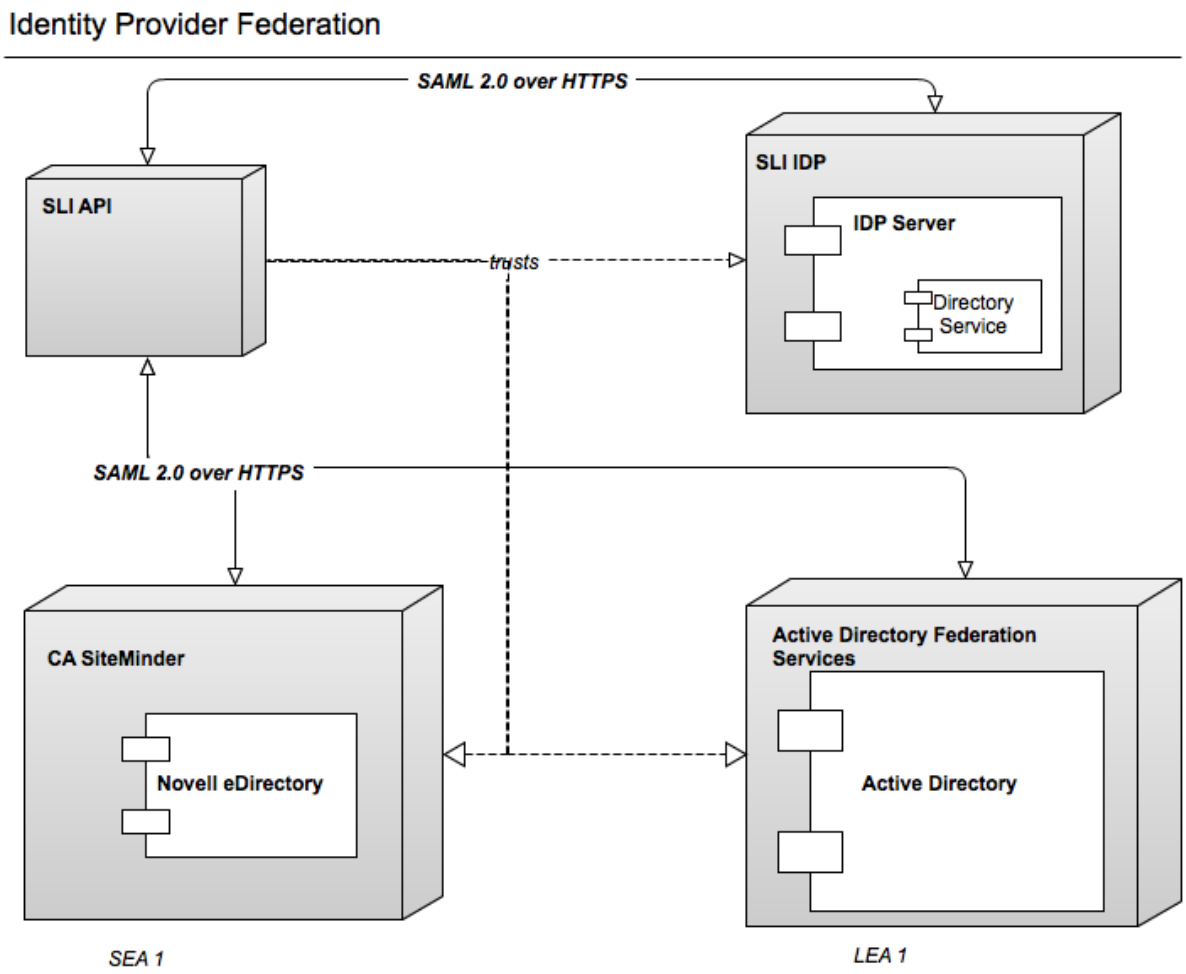

Figure 2 – Example of Identity Provider Federation

3.4 Overall Directory Integration Implementation Process

The overall process of setting up SEA/LEA Directory integration involves two main steps:

- Provisioning and validating the SEA/LEA Directory and Identity Provider with which SLI will integrate
- Mapping student information systems data to data entities within the SLI data store that have been ingested through the data integration process

For the Federated approach, these steps are described in greater detail through an example below. For information about the process of ingesting data, see *SLI Data Ingestion Specification*.

3.5 SEA/LEA Directory information requirements

This section describes requirements and processes for setting up an SEA/LEA Directory.

State-wide UID

SLI's authorization process requires that each user is uniquely identified within the system so that SLI can identify all of the entities associated with the user (students, assessments, and so forth) within the SLI data store. At login time, SLI is only aware of the user's realm, for example, State1 or State2 > District6. Thus, a state-wide UID enables SLI to achieve uniqueness of users across SLI by concatenating the user's realm and state-wide UID.

SEA/LEA user role

For SLI to apply appropriate filtering/permissions on the associated entities, SEA/LEA directories need to provide the user's role info. This role info will be mapped to one of the default SLI roles (Aggregate Viewer, Educator, Leader and IT Administrator). Default SLI roles and permissions are described in more details in Section 4.

Setting up federated authentication for SEA/LEA

To set up a federated authentication for a SEA/LEA, follow these general guidelines:

1. Set up the Identity Provider for SEA/LEA:

    a) Verify that the pilot SEA/LEA already has a SAML 2.0 "ready" Directory service, such as Active Directory.
    b) Install and configure SAML 2.0 Federation.

    In an Active Directory environment, this may be done with Active Directory Federation Services (ADFS). Any other SAML 2.0 based SSO solution, such as CA SiteMinder or OpenAM, may also be used.

    c) Enable the SLI and the SEA/LEA Identity Provider to communicate securely.

    Note that it is necessary to configure firewall setting at the SEA/LEA to allow the SLI Identity Service to access the Identity Provider server URL. All network traffic in this example is performed over HTTPS.

2. Configure Federation

    a) Configure the SEA/LEA SAML 2.0 Identity Provider to enable Trust from SLI.
    b) Configure the SAML 2.0 Identity Provider to include the required attributes about the user in the SAML assertion response. An example is provided further down that demonstrates how the attributes should appear using the AttributeName and AttributeValue XML tags. Note that for the roles attribute there can be many AttributeValue tags.

        - username – User's full name

- userid – staffUniqueStateId in edfi Student Interchange
- roles – roles or group memberships that the user has

c) Within the SLI Administrator Tool for Realm Management, configure a Realm which requires the SAML 2.0 SEA/LEAD organizations Identity Provider identifier and end point URL.

3. Map Roles

   a) Use the SLI Role Mapping tool to map incoming role list entries from the IDP to the corresponding default SLI Roles

4. Map SLI Entities

   a) Ensure entity IDs (for example, Staff ID) within the SLI core entity data store (or data to be ingested) map to attributes within the Directory record.
   b) Initiate the SLI data ingestion process to which will include the ingestion of staff and teacher users that will map to users in the IDP via the staffUniqueStateId

3.6 Federated Application Authentication Flow

Applications that use the SLI API must use the SLI's OAuth 2.0 implementation to access the SLI. OAuth 2.0 is used to authenticate the application and to allow the SLI API to authenticate the user for the application. In the diagram below steps 1, 5 and 6 are all part of the OAuth process and are discussed in more depth in the developer guide.

Authentication of the user is done with Single Sign-On (SSO) identity integration with SEA and LEA Identity Provider. The SLI API takes care of interacting with the SEA LEA IDP, this frees the application developer from the complexities of SAML integration.

The SAML request sent to the SEA and LEA IDP (step 3 in the diagram below) includes the "ForceAuth" attribute. This attribute determines whether the user must authenticate even if an existing session exists in the IDP. The SLI API sets this value to true if the user has no existing SLI session. If the user opens another SLI Application and has a valid SLI session already then the SLI API still sends the SAML assertion to the SEA or LEA IDP to authenticate but the ForceAuth attribute is set to false. The SEA or LEA IDP can then decide whether the user needs to authenticate again or not.

After successful authentication at the SEA or LEA IDP, the IDP returns the required attributes about a user in the SAML response (userId, username, roles) (step 4 in the diagram below). A single session is then created in the SLI API for that user using that application.

If the user starts a different SLI application while an existing SLI session exists, the SLI API will still send the user to their IDP to authenticate. SLI API sessions are for one user using one application. Whether the user has to authenticate with their IDP or not depends on the configuration of the IDP, specifically the session timeout.

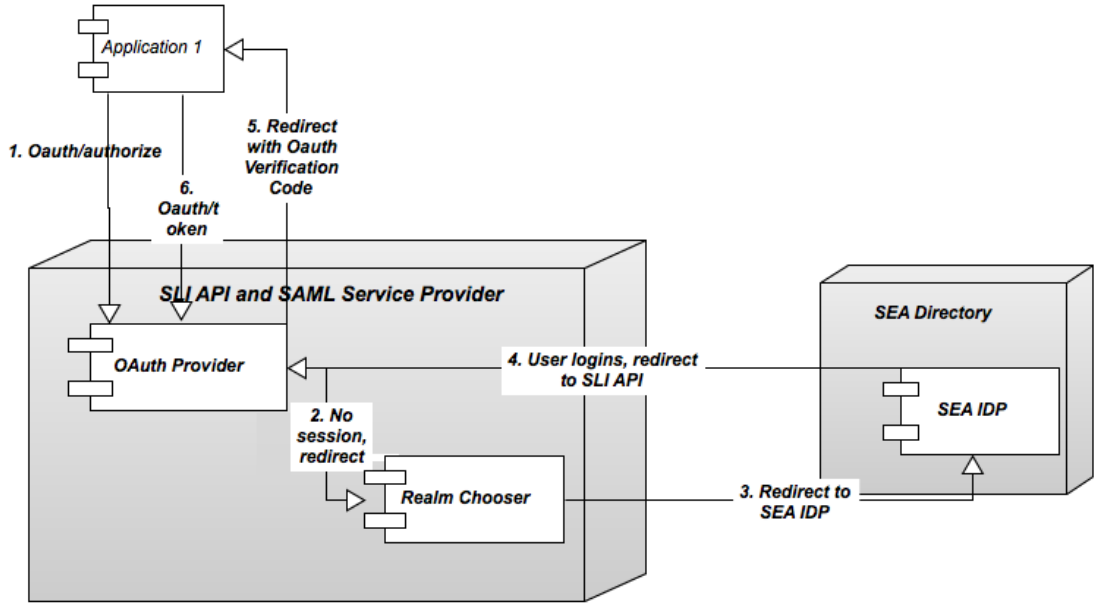

Figure 3 – Application Authentication Flow

Metadata URL

The metadata URL can be used to retrieve the SAML metadata for the SLI Service Provider. Some IDP's can be configured using this URL. In some cases the IDP may call this URL at some interval to check if metadata about the Service Provider has changed. The URL will be `$BASE_URL$/api/rest/saml/metadata`. Here is an example of a response from this call:

```
EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
  entityID="">
  <SPSSODescriptor AuthnRequestsSigned="false"
    WantAssertionsSigned="false"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="
      ResponseLocation=" />
    <SingleLogoutService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="
      ResponseLocation=" />
    <SingleLogoutService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location=" />
    <ManageNameIDService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="
```

```
      ResponseLocation=" />
    <ManageNameIDService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="
      ResponseLocation=" />
    <ManageNameIDService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="
      ResponseLocation=" />
    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
    </NameIDFormat>
    <NameIDFormat> urn:oasis:names:tc:SAML:2.0:nameid-format:transient
    </NameIDFormat>
    <NameIDFormat> urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress
    </NameIDFormat>
    <NameIDFormat> urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified
    </NameIDFormat>
    <NameIDFormat>
      urn:oasis:names:tc:SAML:1.1:nameid-
format:WindowsDomainQualifiedName
    </NameIDFormat>
    <NameIDFormat> urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos
    </NameIDFormat>
    <NameIDFormat>
      urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
    </NameIDFormat>
    <AssertionConsumerService index="0"
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
      Location=" />
    <AssertionConsumerService isDefault="true"
      index="1" Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location=" />
    <AssertionConsumerService index="2"
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS" Location=" />
  </SPSSODescriptor>
</EntityDescriptor>
```

Example AuthN Request

The following is an example AuthN request sent by the SLI Service Provider to initiate and authenticaton with a remote IDP:

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  Destination="
  ForceAuthn="false" ID="sli-b931632c-9706-4c8e-bd27-0a0a1f8dc408"
  IsPassive="false" IssueInstant="2012-03-27T17:56:55.486Z"
  ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  Version="2.0">
```

```
  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://nxbuild.slid
ev.org
  </saml:Issuer>
  <samlp:NameIDPolicy AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"
    SPNameQualifier=" />
  <samlp:RequestedAuthnContext Comparison="exact">
    <saml:AuthnContextClassRef
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">urn:oasis:names:tc:S
AML:2.0:ac:classes:PasswordProtectedTransport
    </saml:AuthnContextClassRef>
  </samlp:RequestedAuthnContext>
</samlp:AuthnRequest>
```

Example SAML Response from ADFS to SLI SP

The following is a snippet of an example SAML response that is sent to the SLI SP, pointing out the SAML attributes that SLI requires.

```
<AttributeStatement>
<AttributeName="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/u
pn">
                <AttributeValue>john.doe</AttributeValue>
            </Attribute>
            <Attribute Name="userId">
                <AttributeValue>john.doe</AttributeValue>
            </Attribute>
            <Attribute Name="userName">
                <AttributeValue>john.doe@slidev.org</AttributeValue>
            </Attribute>
            <Attribute Name="roles">
                <AttributeValue>Domain Users</AttributeValue>
                <AttributeValue>Teacher</AttributeValue>
            </Attribute>
</AttributeStatement>
```

3.7 Authenticating Non Web-based Applications

The OAuth specification supports long-life tokens, which can be granted to an application on behalf of a user after the user authorizes the actions/permissions requested by the application. Investigation into the configuration of such a feature to meet the stringent security requirements of SLI is currently under way.

3.8 Application Onboarding

To become available to a District end user, an application needs to be onboarded with SLI and then authorized by the respective District Super-Administrator (3.7 Application Authorization).

The Application on-boarding is a two-step process that requires joint effort between vendors and SLC.

3.9 Application Registration

In the first step, the Vendor submits a registration request to SLC, where the basic Application information is specified. The platform operator should then approve the request. After the Application registration request is approved, the vendor will receive Application-unique information that needs to be integrated in the application, so that the (SLI) platform can authenticate and “recognize” the application.

SLC should assess whether the application meets SLC terms and conditions and applicable SLC privacy and security policies.

3.10 Application Enablement

When the vendor is ready to release the Application, the vendor will choose to whom (States, Districts, everyone) the Application should be available. As noted earlier, making the Application available for a particular District does not provide access to the District’s data.

3.11 Application Authorization

To access a District’s data within the SLI, an application must be granted permission by the designated District Super-Administrator or other District user who has the appropriate delegated permissions. SLI recognizes the District as the ultimate arbiter of who is able to view or manipulate its data. Ultimately, the authorization will be configurable based on particular data entities (or collections thereof).

3.12 Data Access Authorization

Authorization to protected resources in the SLI APIis a multistep process that starts by determining if any of the roles provided in the SAML assertion response correspond to the default SLI roles. If not then no authorization is granted. If a role is matched this determines the permissions the user has, next the SLI API must determine which data the user has access to. This is done by using the userId attribute passed in the SAML assertion response to locate the core entity for that user. Now a determination can be made as to whether the current user has a valid association to the requested data via the allowed security paths in the entity model and the specific data the user is trying to access.

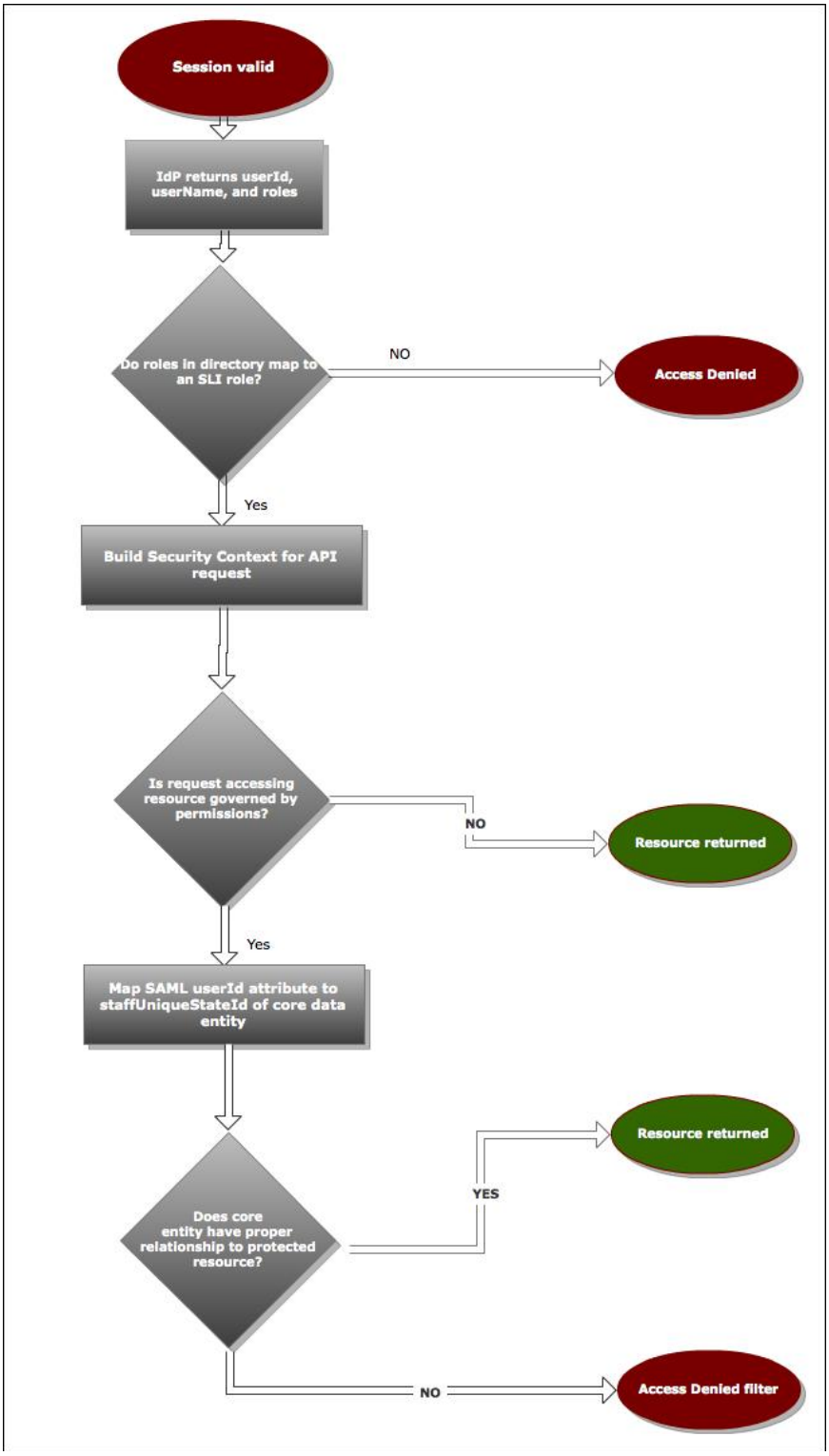

Figure 4 – Accessing Protected Resource Workflow

3.13 Mapping entities to Directory entries

In the previous diagram, one of the final steps is to resolve the mapping of the ID attribute in the Directory that corresponds to the SLI data entity that is used to determine the relationship of the user logged in to the resource that is protected. For example, if a teacher can only view student assessments for students that they are teaching, the relationship of the Staff entity to the student (through the Section entity) must exist in the Core Entity Model.

The attributes in the Directory entry that can be used to resolve the relationship will most likely need to be configured by the SEA/LEA, because different Directory setups can use different attributes to represent the UID within the enrollment data (typically sourced from an SIS).

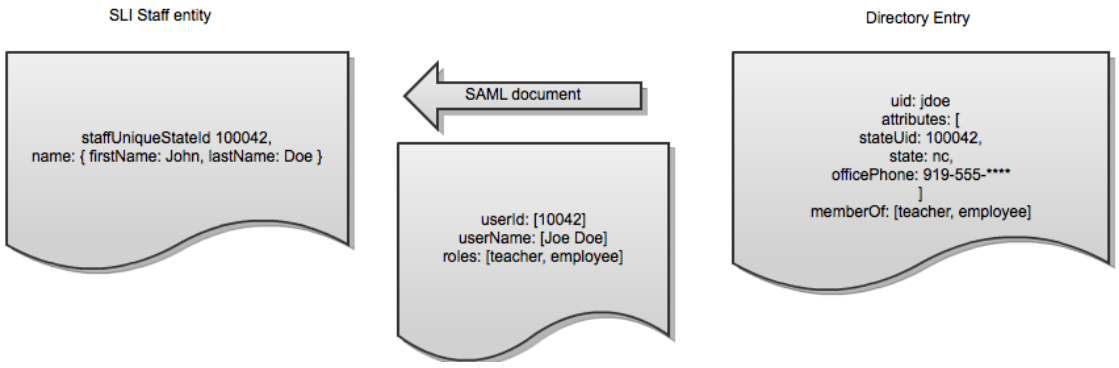

Figure 5 – ID Mapping

The following figure shows an example of how the relationships within the core entity model must exist to provide the context-based permission evaluation that SLI will support.

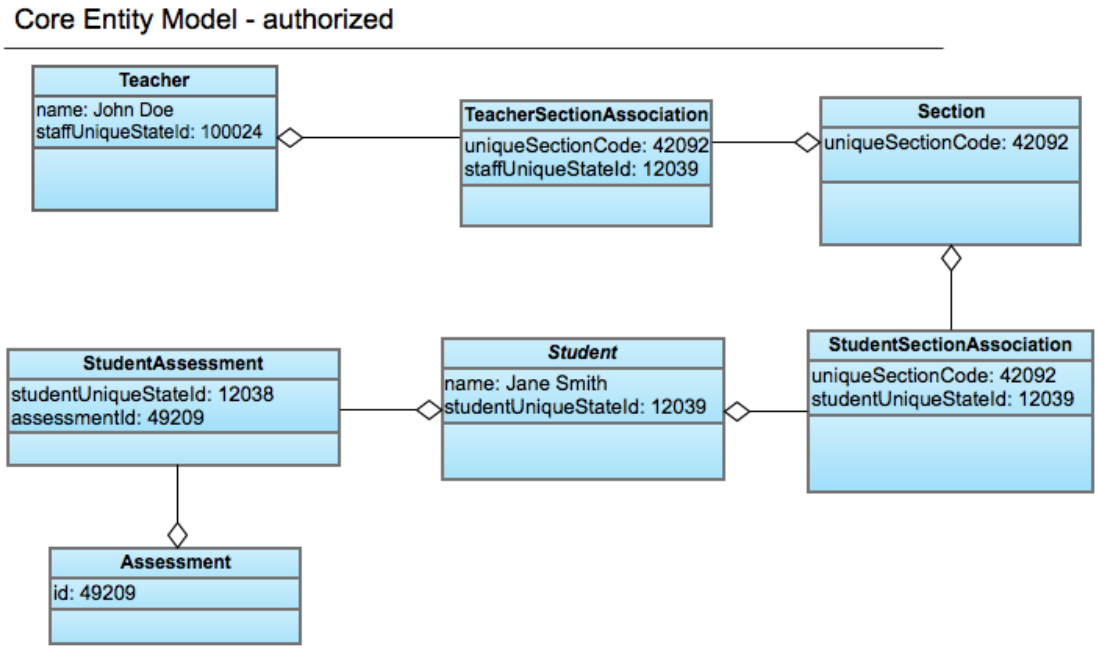

Figure 6 – Core Entity Model Relationships

If no relationship exists and the data trying to be access is protected, the access would be denied.

The following figure shows an example of a lack of access granted for a different student that is not being taught by the teacher John Doe.

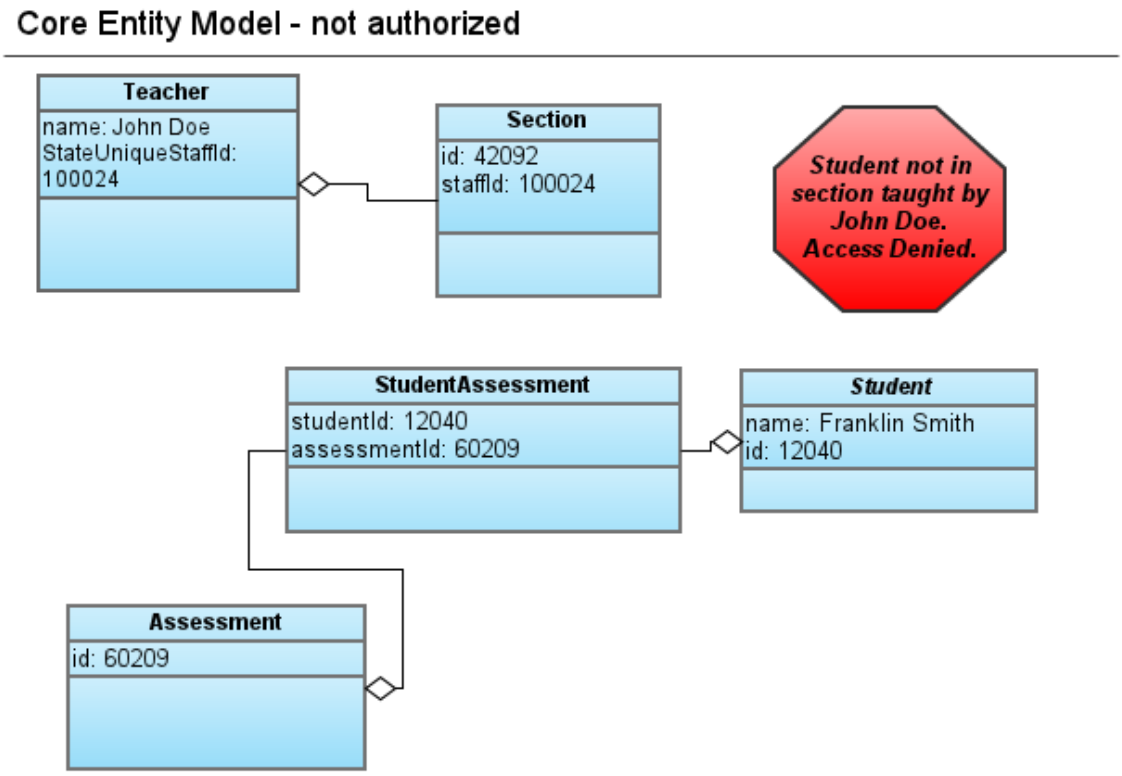

Figure 7 – Non-Existent Relationships in Core Entity Model

4 SLI User Role and Permission Management

This section describes the business rules that govern access to SLI data. It also outlines the default SLI Roles planned for the Alpha release, as well as the future (1.0 Release and beyond) plans for the SLI Roles implementation.

4.1 Data Stewardship

SLI recognizes an organizational hierarchy consisting of Schools, Districts, and States, as shown in Figure 1. It is assumed that all Schools within SLI belong to exactly one District, and that all Districts belong to exactly one State. Student data storage and access-control is aligned with this same hierarchy. In other words, the data of a given student is directly associated with their School, and, in turn, associated with the School's District. This hierarchy is used by SLI to determine data access and permission management. For example, the hierarchy allows a District IT Administrator to manage data and permissions within any School within the District's portion of the hierarchy.

Figure 8 – State/District/SchoolHierarchy

SLI recognizes the District (LEA) as the ultimate arbiter of who is able to view or manipulate the District's student data. While SLI is built to facilitate District control over data access, the SLC recognizes that Districts may have contractual or regulatory arrangements enabling States to view and administer data and permissions on their behalf.

4.2 Roles and Permissions Overview

Access to an Institution's data is controlled by the Permissions of a user who is using a particular Application. While the Application Permissions for Alpha release are simple and they define if the Application can or cannot access the District's data (at all), the user Permissions define what data a given user is allowed to see, and what the user is allowed to do with the data.

SLI determines the user Permissions based on a combination of user role and a particular context.

Each SLI Role (Leader, Educator, IT Administrator and Aggregate Viewer) determines what Permission the users who are assigned that Role will have. Permissions will determine what operations a user is allowed to perform and, in the context of the Institution with which they are associated, what data they are allowed to access.

One key context element is the Institution of the user. For example, an IT Administrator (role) may be associated with a particular District, and hence he/she can access the District's data in a way that is defined by this role and Institution context. For some actions, other information from the data model is needed. For example, some of the rights that are associated with the Educator (role) will depend on the sections that an Educator is teaching, and not only the institution he/she is associated with. Likewise, a principal (Leader role) at a given School may be able to view all student data for students in his/her School, but would not have permissions to view the student data of students in other schools in the District.

4.3 Default Permissions and Custom Roles

SLI defines four default Roles that correlate to common roles within the educational domain. In the Alpha release, SLI roles cannot be modified. Custom Roles, which will be supported in future, can be defined by associating the Role with any combination of (only) the existing SLI Permissions.

4.4 Default Roles and Permissions (Alpha)

This section describes the default roles and permissions as they are defined for the Alpha release.

4.5 Default SLI Roles

This section describes the default SLI Roles for the Alpha release.

- Aggregate Viewer–- Interested in the aggregate/summary data as well as the trends, but cannot see individual-level record information.
- Educator – A person who works for education organizations and is interested in student outcomes. Generally, educators are the people who interact directly with students on a daily basis. Examples of derivative roles: Teacher, Classroom Assistant, Athletic Coach.
- Leader – Interested in students in their own School or many Schools within a District/State. In addition to the individual, they are interested in summary/aggregate data. Examples of derivative Roles: District/LEA Leader, State Leader, School Principal, Dean, Department Head, Special Ed Expert, School Psychologist, Guidance Counselor.
- IT Administrator – May not have a direct educational interest in particular students, but administer systems that help educators and leaders by making data available to them.

4.6 Permissions by Student Data Category

Permissions for each of the above Roles are defined for the following student Data Categories:

- General Student Data: By default, all data not marked as Restricted falls into General, including: Name, Address, Phone Number, Email, Date and Place of Birth, Grade, Transcripts Information, Attendance, Discipline, and so forth
- Restricted Student Data: By default, this includes: Free Lunch Entitlement, Reduced Lunch Entitlement.
- Aggregate Data*: School Attendance Rate, Class Profile and so forth
- Public Data: School Address, School Phone Numbers and so forth

* In Alpha release, Aggregate Data will not exist in SLI.

These categories are pre-set and not configurable for Alpha.

4.7 SLI Roles to Student Data Category Mapping

The following table shows the mapping of the SLI's default Roles with the student Data Categories (of fields) and the context.

SLI Role	Examples	Aggregate/ Public	Individual Record Data	General Student Data	Restricted Student Data
Aggregate Viewer	State Data Analyst, State DOE Representative	yes	none	none	none
Educator	Teacher, Athletic Coach, Classroom Assistant	yes	student enrolled in my sections	R	none
Leader	School Principal, District Superintendent, State Superintendent	yes	student enrolled in my district(s) or school(s)	R	R
IT Administrator	SLC Operator, SEA IT Admin, LEA IT Admin	yes	student enrolled in my district(s) or school(s)	R/W	R/W

Table 1 - Default Roles and Permissions for Alpha Pilot users

4.8 Roles and Permissions (1.0 and Beyond)

This section describes the future plans for the SLI Roles and Permissions implementation, for the 1.0 release and beyond.

- Default Roles and Permissions will be added for parents and students that correlate to their common roles within the educational ecosystem

- Ability for SEAs and LEAs to define and configure Custom Roles, and assign Permission/Permission groups that exist in the SLI
- Granular Permission configuration:
  - The ability to define Entity and Attribute-level Permissions for custom roles. For example, restrict access to students' Special Ed program participation info for a Classroom Assistant.
  - The ability to define Dataset-level Permissions for Custom Roles. For example, restrict access to specific Benchmark Assessments within a District for State-level Academic Administrators.