Architecture and Technology System Requirements For the NCSC Assessment Platform

Report and Recommendations

ncsc

National Center and State Collaborative

October 31, 2012

measured progress

Prepared in accordance with RFP 2012-01-01 by Measured Progress

Table of Contents

1. Document Overview

1.1 Summary

The Architecture and Technology System Requirements Document describes the fundamental architecture of the NCSC GSEG technology solution. This final product represents an apex of five months of work in identifying the overarching goals, processes, technology and needs for the NCSC Assessment system. This work was conducted in an agile process by holding a series of workshops designed for analysis and consensus building around architectural tenets. Work and documentation continually received reviews from NCSC through showcases and discussions, leading to revision and refinement ensuring the needs would be represented within this architecture document.

It is important to note that this document does not reflect the implementation of the actual components or applications; it represents the enterprise architecture and the unique needs of the NCSC community. Many decisions by the NCSC Community will inform the application development and implementation.

1.2 Purpose

The purpose of the architecture documentation incorporates identifying the technical requirements that the NCSC System must comply with in order to deliver on the promise of the GSEG grant and the aspirations of the NCSC program to dramatically improve the quality and value of the summative alternate assessment provided to students with significant cognitive disabilities. Key objectives that guided the IT architecture include:

1. Following the NCSC vision and theory of action to aim for seamless and continuous access to grade-level academic content towards the goal of increasing college and career readiness.
2. Enabling an assessment delivery that measures a student's understanding in CCSS and allows for evaluation and refinement of best practices.
3. Providing a scoring system, including appropriate evidence collection, with the flexibility of distribute scoring and machine scoring.
4. Providing reporting longitudinally and for lifelong learning.
5. Supporting the highest expectations for all students.

This document is a living document, and as such, it is intended to contain the architectural solution of the NCSC technical design. As decisions are made, this document will be updated with those technical decisions to guarantee its usefulness throughout the lifecycle of the NCSC project work.

Solution providers that will support the NCSC team in realizing the assessment system make up the primary audience for this document. The document provides the framework to design and develop the applications in a unified manner across the enterprise. Secondary audiences include NCSC oversight, Architecture Review Board and NCSC members. These individuals focus on the overall direction, communication, governance, implementation, support and overall guidelines of the solution.

1.3 Document Contents

Various sections comprise this document in defining the overall context and enterprise architecture:

Architecture Approach
Describes the process behind the development of the NCSC Architecture and the architectural principles driving the solution. These principles are agreed upon and drive all design decisions. In addition, this section describes the Assessment Lifecycle, providing an overall, high-level view of an assessment from creation to post-administration of the assessment.

High Level Solution and System Components
Explains the overall NCSC Assessment solution from the categorization of components and the specific relationships between each. A description also demonstrates how each of these components aligns with the Assessment Lifecycle Model.

Interoperability
Identifies a variety of scenarios within the prescribed solution and distinguishes the points of interoperability of components within those scenarios and the standards used to facilitate transfer of data and control.

Data Architecture
Illuminates the general architecture principles, emphasizes the fundamental requirements of data within the overall solution, describes the data source elements found within the various data sources of the solution and explains specific areas of consideration within the solution for acceptance criteria for new technologies.

Component Transport
Describes the necessary interfaces and transport mechanisms to be used for component-to-component communication within the NCSC Assessment System.

Security
Includes the variety of security aspects, including the dimensions of security that the overall solution must support, secure elements, component-to-component communication, user authentication, student and teacher data, a communication pattern between components and security concerns.

Constraints and Non-functional Requirements
Highlights the variety of constraints and non-functional requirements of the architecture, including accessibility, usability, extensibility, scalability and others, as well as identifies any concerns.   Non-functional requirements describe the overall operations of the system and not specific functions or behaviors.

These non-functional requirements provide a framework for the expectations of the architecture and how it performs over time.

Technical Architecture
Covers the following topics: specific technical requirements for test taker workstations, proctor workstations, server requirements, deployment and hosting requirements and bandwidth requirements.

Glossary
Provides a list of terms and concepts helpful to readers of this document.

2. Architecture Approach

The NCSC IT Systems Architecture was designed utilizing an agile process through which the architecture and all principles culminated. Fundamental systems architecture principles lay the foundation for guidelines in the creation of this recommended architecture. Additionally, essential decisions will be made throughout the development and implementation of the NCSC Assessment System and therefore the guidelines must be revisited often. This document is not to remain static, and thus an important component of the architecture includes NCSC developing and adopting a governance model in order to sustain this architecture during implementation and beyond.

2.1 Process

Developing the IT architecture for NCSC enables aligning technology investments defined in the key assessment functions with long-term strategy, while reducing risk, delivering higher-quality information and engineering flexible, evolving assessment solutions and technical services. This illustration depicts the synthesis of the strategy into enterprise architecture deliverables and into governance processes.

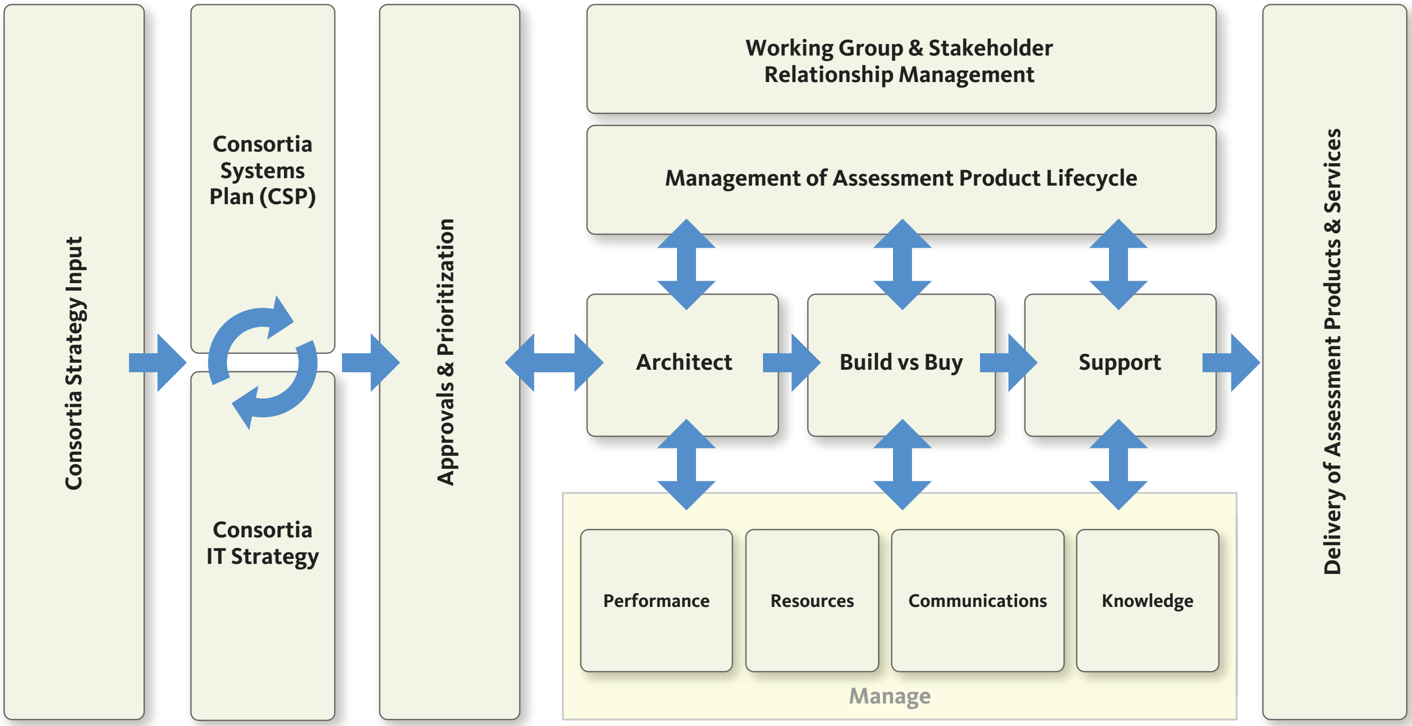
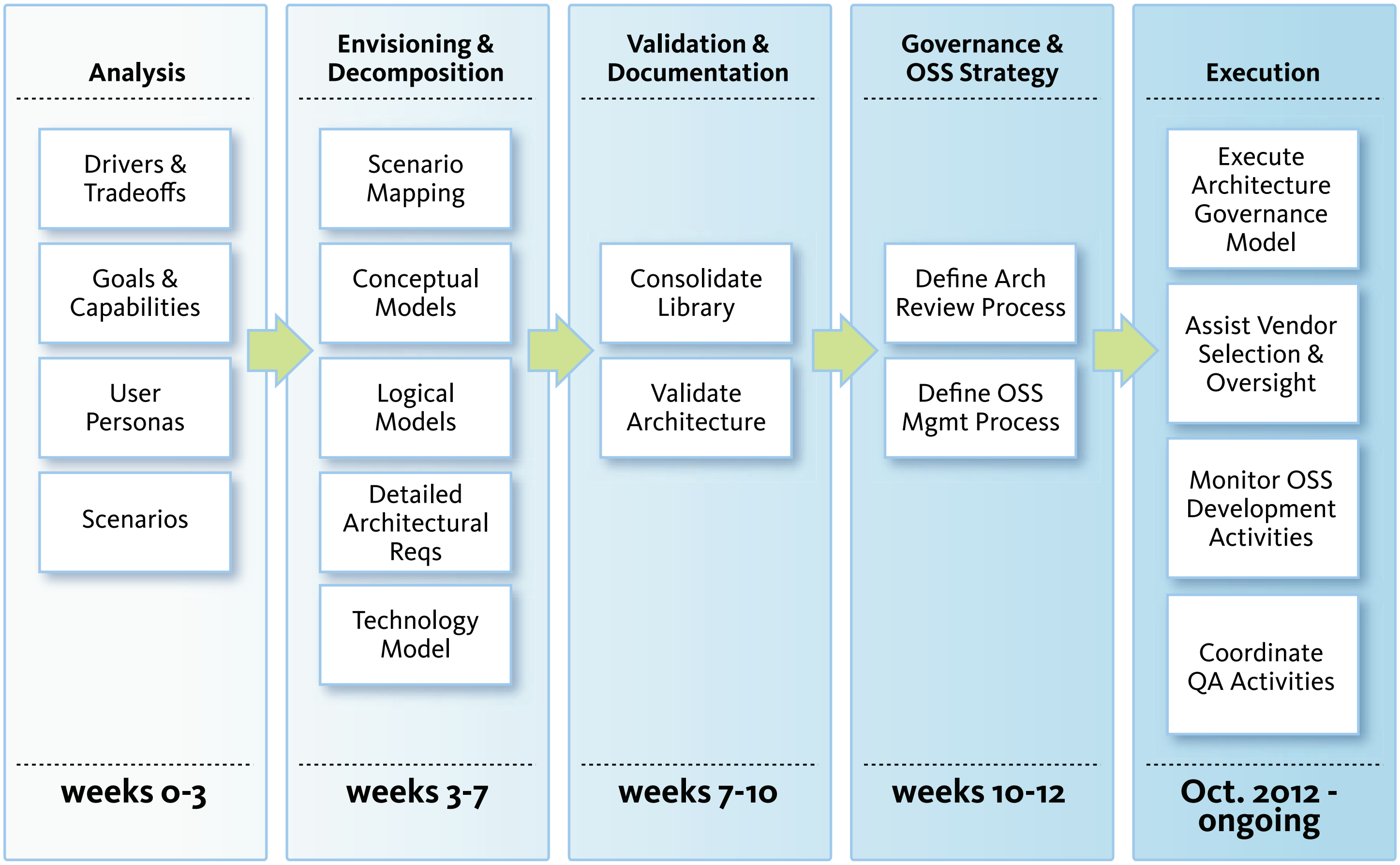

Consortia Strategy Input
Consortia Systems Plan (CSP)
Consortia IT Strategy
Approvals & Prioritization
Working Group & Stakeholder Relationship Management
Management of Assessment Product Lifecycle
Architect
Build vs Buy
Support
Performance
Resources
Communications
Knowledge
Manage
Delivery of Assessment Products & Services

Figure 2.1: Architecture Deliverable Governance Process

The enterprise architecture approach institutes a collaborative, shared planning process with the architecture teams, NCSC, stakeholders and industry experts to define a future-state vision in terms of requirements, principles and models. Though the architecture approach is deeply technical in nature, it is not IT-fixated, but rather focused on the comprehensive NCSC drivers and enablers. This established future-state vision helps coordinate the analysis of, and develops a plan to synthesize required processes to be defined enabling NCSC functions and processes, information and data provisioning, technology capabilities and application solutions. Five overarching phases are included in the entire planning and implementation of the architecture: analysis, envisioning and decomposition, governance and OSS strategy and execution.

The analysis phase included holding two weeks of workshops with NCSC staff, member states, assessment experts and the Measured Progress team. These workshops provided the foundation for the remaining phases of the initiative.



Analysis
Drivers & Tradeoffs
Goals & Capabilities
User Personas
Scenarios
weeks 0-3

Envisioning & Decomposition
Scenario Mapping
Conceptual Models
Logical Models
Detailed Architectural Reqs
Technology Model
weeks 3-7

Validation & Documentation
Consolidate Library
Validate Architecture
weeks 7-10

Governance & OSS Strategy
Define Arch Review Process
Define OSS Mgmt Process
weeks 10-12

Execution
Execute Architecture Governance Model
Assist Vendor Selection & Oversight
Monitor OSS Development Activities
Coordinate QA Activities
Oct. 2012 - ongoing

Figure 2.2: Architecture Development Phases

2.2 Assessment Lifecycle

The SIF Association developed and designed an Assessment Lifecycle Model. The Assessment Lifecycle provides an overall picture as to the existence and iterative nature of an assessment – from item creation to delivery to analysis. No specific entry point exists for the model. The six overarching categories have subprocesses elaborating on the detail of each category. The diagram below demonstrates how the architecture conceptually aligns with this model. More information is available at http://www.sifassociation.org.

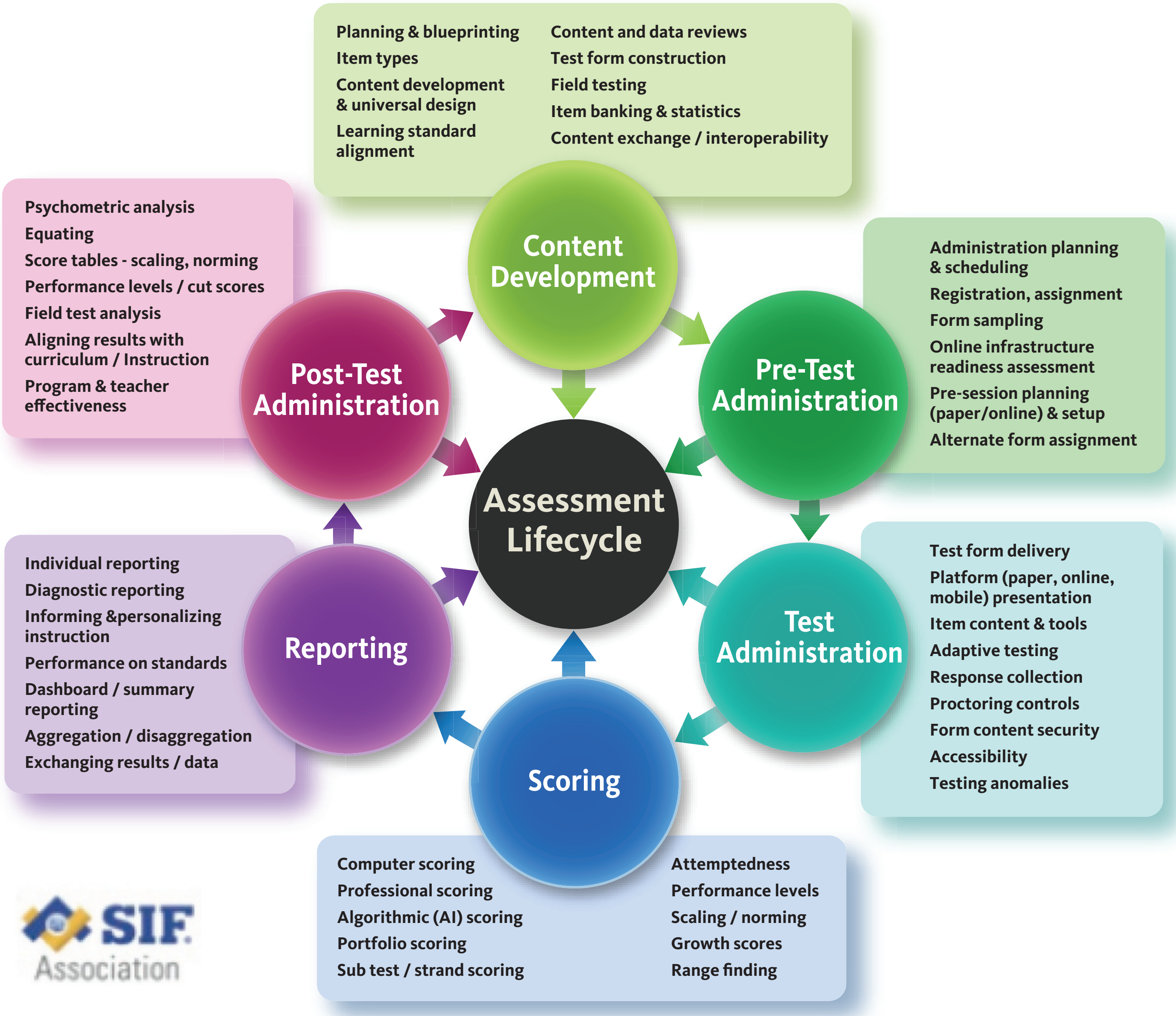


Figure 2.3: SIF Association Assessment Lifecycle

2.3 Conceptual Domain Model

Throughout the analysis phase, several domains, or subject areas, came to fruition. The following diagram illustrates the domains of the NCSC GSEG System Solution. These domains are areas that NCSC focus on and address in the development of the assessment system. These demonstrate an importance in the development of the architecture in order to ensure that each could be represented accurately in the solution.

In the diagram below, the arrows between domains represent requirements relationships between these domains. For example, Test Delivery, Item Authoring and Test Authoring all require the services of Item Presentation to render the items developed for the assessment. This requirement ensures that the items display consistently and completely throughout authoring, review and delivery.

Several domains provide services to multiple domains. These service domains have numerous connections to other domains. For example, as security of item level and student and teacher data is of utmost importance, encryption supports most of the domains identified.

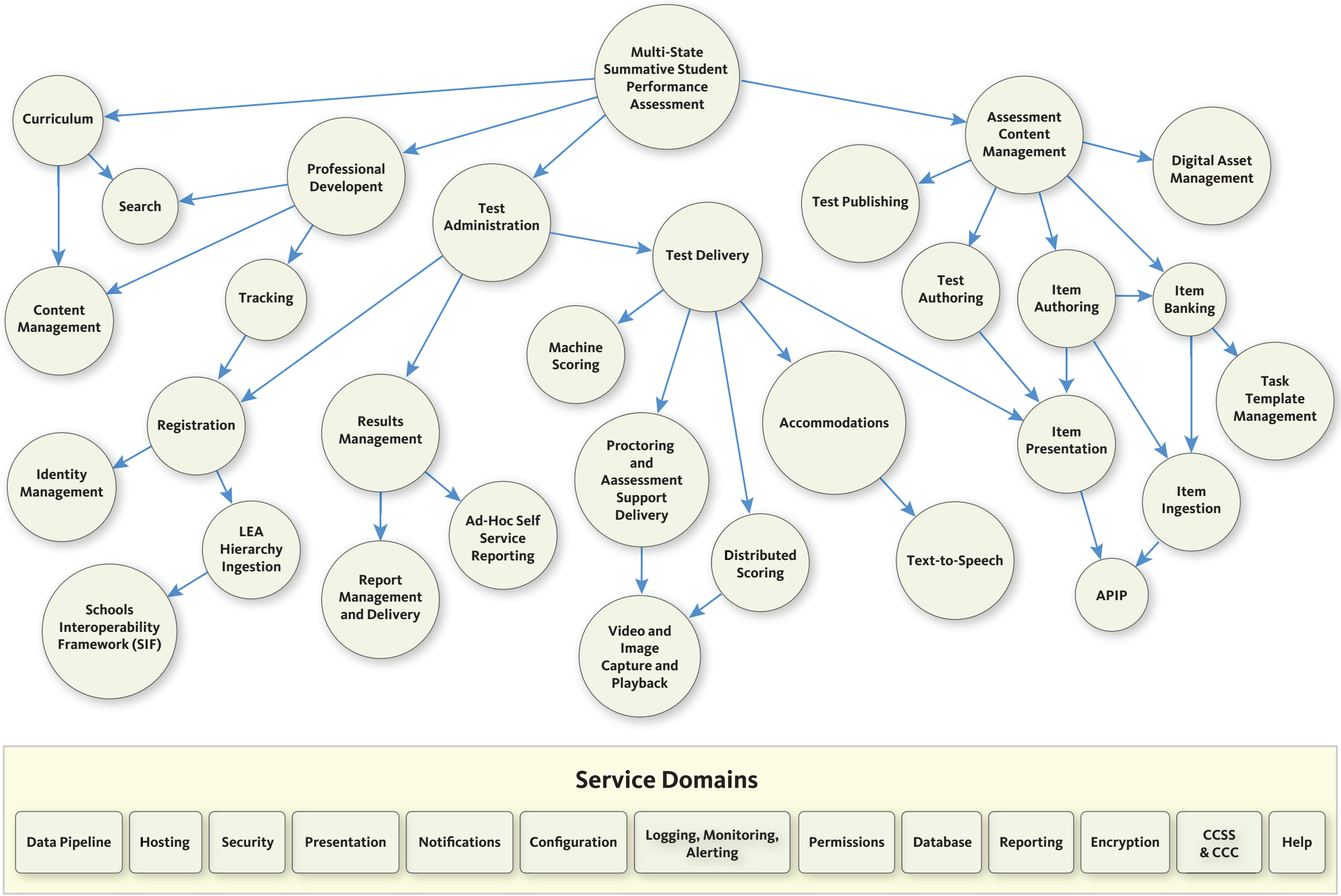


Figure 2.4: NCSC Domain Model

2.4 Principles and Design Goals

During the design process, decisions made require guidelines and rules for deciding between the options. Some basic guideposts apply when defining these principles:

1. The number of principles should be kept to a reasonable number.
2. The principles need to be maintained to support decision-making. These are especially useful when the options presented are equally viable.
3. The principles need to be maintained by the NCSC Architecture Review Board.
4. The principles need to be described at a level that is appropriate for technical decision-making.

2.4.1 Open Systems Development

Software assets created should not be locked up within proprietary systems and contracts with vendors and available for use or extension by other organizations. This is not a mandate for open source development across the board, but rather as a mandate to create a solution that is open to reuse and extensibility and towards more open and accessible implementation models.

A mixture of off-the-shelf (OTS), open source technologies and custom system components are options for NCSC. Open source licensing should be used for both OTS and custom development.

Rationale

- Using open systems lowers the cost of implementation and ownership.
- Providing the opportunity for open development allows other individuals and organizations to extend the work funded here.
- Offering open systems provides more flexibility for NCSC and SEAs when working with service vendors.

Implications

- Vendors must learn, understand and publish software within the new licensing constraints.
- Governance and community management are required for successful execution.

2.4.2 Enforce Componentization and Interface Boundaries

Modularity of components promotes easy adaptation to future business or regulatory requirements. Componentization also makes it possible to switch application providers in a straightforward manner. Adhering to standard interface boundaries advances the ease with which the systems interact with other systems through the consistent protocols. Requiring the components to communicate strictly upon application programming interface (API) boundaries or through shared data exchanges is key to enforcing modularity.

Rationale

- Utilizing components for the system creates opportunities for easier upgrades and deployment.
- Adding new components to the system can be done without disturbing existing system interactions.
- Breaking a system up in a collection of components with unique responsibilities reduces overall system complexity.

Implications

- Give off-the-shelf components or legacy solutions considerations that may not be designed with API boundaries and other standards enforced.
- Take into account security requirements around authentication and authorization as these are typically implemented without modularity in mind.
- Establish and communicate interoperability decisions to vendors and SEAs.

2.4.3 Design for Long-Term Business

Conceiving an assessment system for long-term operations necessitates certain requirements of security, scalability and performance that are absolutely significant and should be considered from the outset. As the system is maintained and extended over time, the non-functional requirements established cannot be violated during deployment or integration activities.

Rationale

- The NCSC Assessment Program intends to self-manage the assessment system. Implementing with this in mind allows for solid long-term decisions.
- Many self-service operations may be done after hours and during periods when assessments are not taking place. Uptime and performance throughout are critical.

Implications

- Developing nonfunctional requirements that are realistic and comprehensive upfront carries significance.
- Establishing long-term governance policies must occur to substantiate long-term operations.
- Generate service-level agreements (SLA) with vendors, ensuring enforcement and measurement.

2.4.4 Interoperate with Existing State and Consortia Systems

In order to support componentization of the NCSC Assessment system, interoperability with existing state systems becomes a requirement. State systems contain the needed data for registration and student and teacher identities and the state systems house the test takers' results.

Likewise, the evolving technical systems under development by other multi-state consortia provide opportunities for NCSC to provide a single point of entry given that some states belong to more than one consortium.

Rationale

- The ability to reuse and share applications and data can be an important cost driver in the assessment systems.
- Systems that use interoperability standards reduce overall integration costs and provide simpler configuration.
- Interoperability promotes an easier on-boarding process for new organizations.

Implications

- Communication, planning and orchestration of the system implementation becomes of utmost importance.
- Open, interoperability standards must be established and utilized by developers.

2.4.5 Utilize Learning Standards

Learning standards serve as the foundation of measuring a student's understanding. The system, from reporting to item creation, should be rooted in the Common Core Standards and the Common Core Connectors.

Rationale

- Standards brought about consortia-level activities and serve as the foundation of the work.
- Common learning standards enable better exchange of data and comparability throughout many areas of the assessment lifecycle.
- All reporting for the assessments are based upon learning standards.

Implications

- Shared ideas, approaches and agreement to the standards must occur.
- Generating an established metadata schema and technical standard must be agreed upon.
- Mechanisms for sharing and collaboration across consortia must be established.

2.4.6 Lower the Cost of Ownership

Striving towards lowering the cost of ownership for NCSC, SEAs and LEAs persists as an essential principle. This manifests itself with upfront costs and ongoing operational costs.

Rationale

- Generating smart, centralized funding, due to the fact that these assessments are for a small population, is necessary.
- SEAs have costs to absorb and keeping these low is of importance for them.
- The funding of NCSC does not possess long-term financing and subject to political uncertainty. Leveraging resources grows in importance.

Implications

- Utilize open source components to reduce initial and ongoing licensing costs.
- Centralizing the deployment and maintenance of the assessment system allows for the consortium to support the full operation of the system in perpetuity.
- Vendors need to align their development with interoperability standards.

2.5 Governance

Technology moves at a rapid pace and educational technology even more so. In order for the architecture to have continuous adherence and evolution, a management process to maintain this solution as the world changes around it must persist.

NCSC will establish a NCSC Architecture Review Board (ARB). This board will consist of members of NCSC and internal and external resources that will maintain the NCSC Architecture as a living solution description. The ARB will establish additional decisions related to the architecture, approve any changes to the architecture and serve as the overall architecture experts for the NCSC architecture.

NCSC will also construct an Architecture Core Team. The Architecture Core Team implements the architecture processes in evaluating, selecting and recommending technology standards for NCSC. The Architecture Core Team also sets documentation standards for maintaining the NCSC architecture over time. The Architecture Core Team membership includes the lead architects from vendors that are selected to develop and implement NCSC architecture.

3. High Level Solution and System Components

With a project of this size and complexity, the system must be broken into clearly defined components. This componentization not only helps NCSC and the development teams manage complexity, but also simplifies deployment, improves reliability and adds the flexibility to incorporate existing products into the system. The specific recommendations below outline the component boundaries with details on the expected responsibilities of each component or set of components.

3.1 Component Categories

This following diagram highlights the broad categories for the components comprising the NCSC Architecture. The desire to separate the components into groups so that requirements, hosting and the source of the components exists such that these might be discussed generally in subsequent design documents. It also creates a simple conceptual model for visualization that helps in successive discussions of the solution.

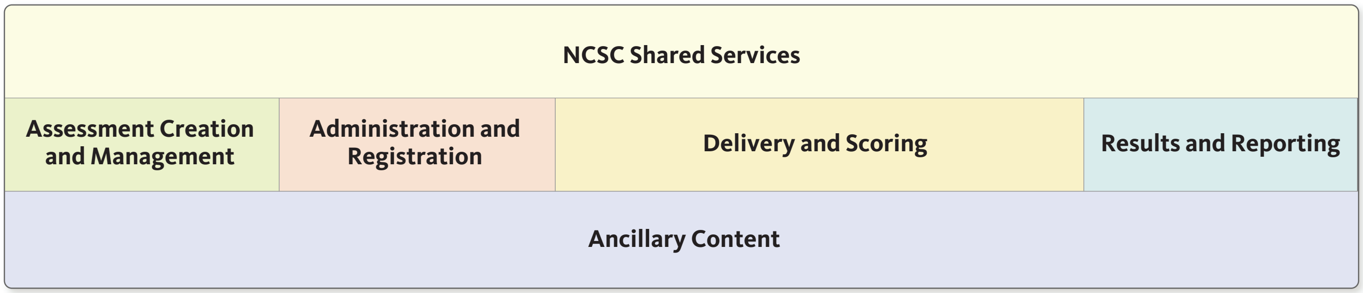


Figure 3.1: High-Level Component Diagram

NCSC Shared Services
Provide core services required by all of the other components and help give the system a more unified feel.

Assessment Creation and Management
Support the creation and management of assessment content including items, tests and supporting assets.

Administration and Registration
Implement the functions needed to set up testing windows and registering students to take tests.

Delivery and Scoring
Deliver a test to a student, including the scoring of that test.

Results and Reporting
Store completed assessment results and reports of those results.

Ancillary Content
Include additional applications that NCSC has plans for developing, but are outside of the scope of this project.

3.2 Logical Component Diagram

The logical component diagram shows the key components required for the NCSC assessment solution. Some components nest within other components. These subcomponents still interact with other components in the broader system using well-defined public interfaces and may also have a tighter coupling within its parent grouping.

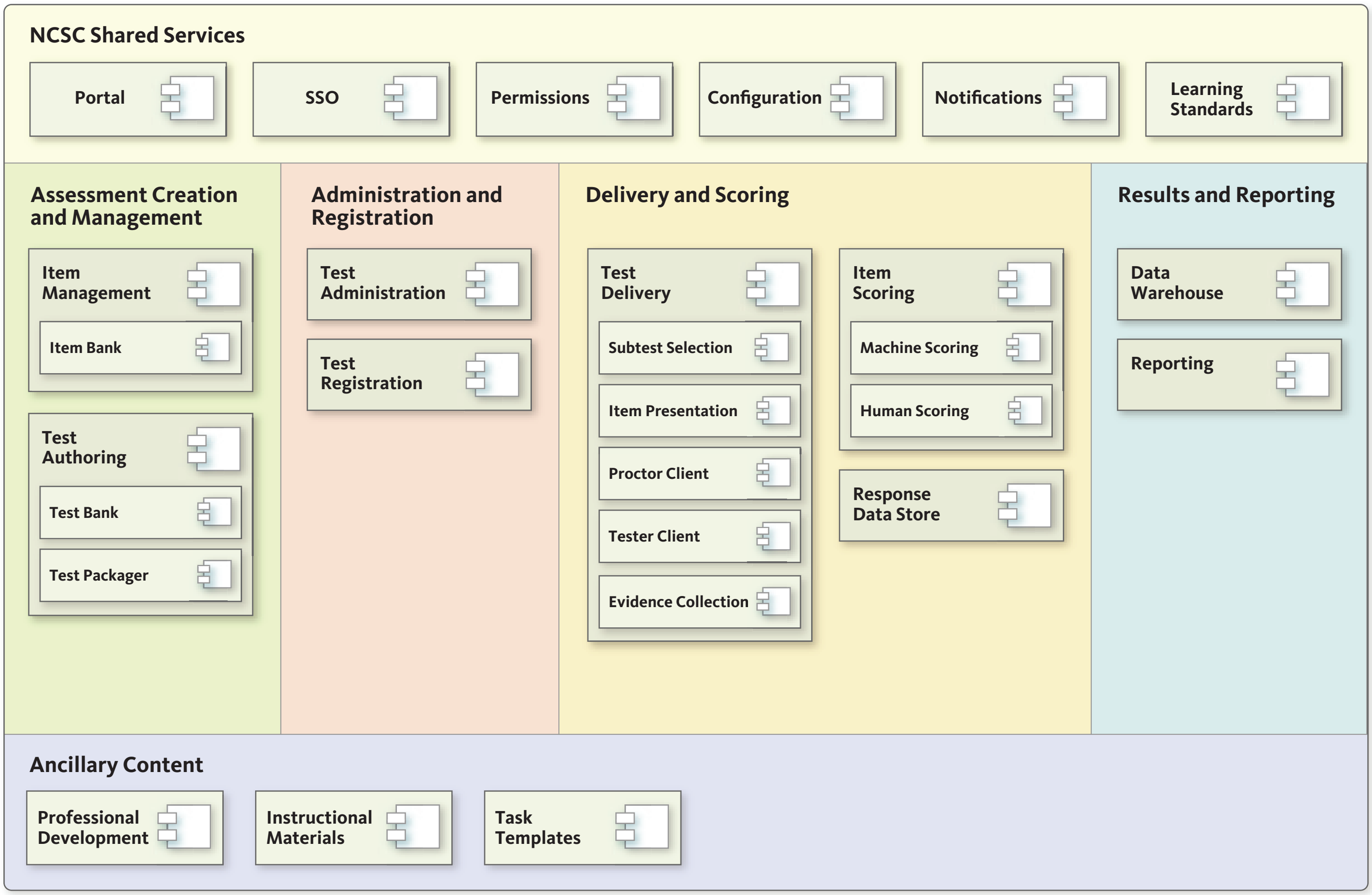


Figure 3.2: Logical Component Diagram

The following sections describe each of the components in more detail. Broken down by component category, each section identifies any required external systems for a complete assessment solution. NCSC does not need to acquire these external components, and these are a loosely coupled part of the solution.

3.2.1 Assessment Creation and Management

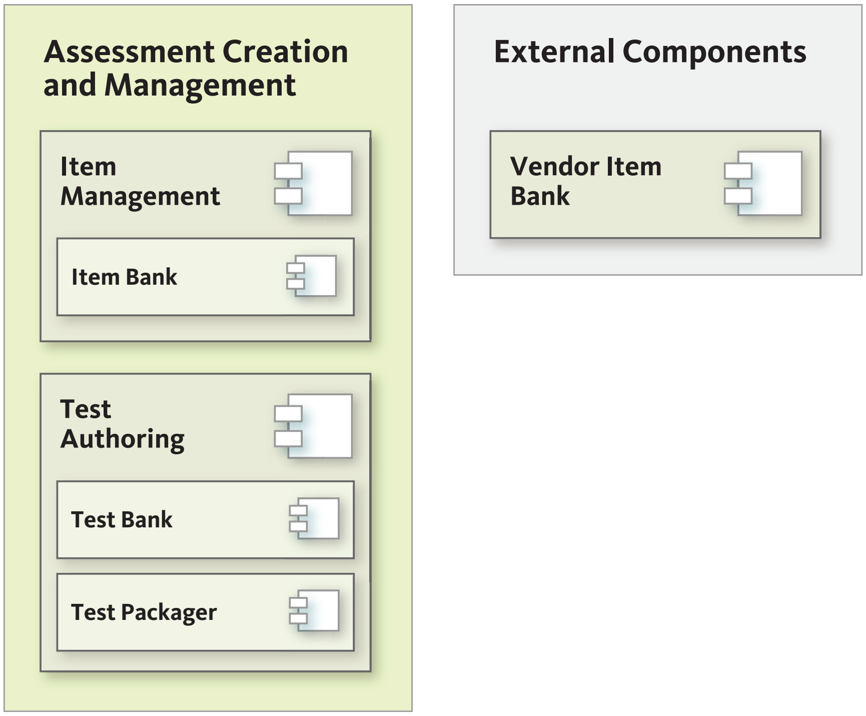


Figure 3.2 Assessment Creation

Vendor Item Bank (External Component)
NCSC works with authoring and banking systems already in use by item development vendors and a build or procurement of an item authoring application is not necessary. The vendor item bank and development exchange data with one or more of the other components. This requires ensuring that vendors agree to package finalized items in the same standard the rest of the system relies on.

Item Management
Managing the items created for NCSC occurs here. This component may only have the very basic editing capabilities and allows NCSC administrators to review the items currently deployed in the system and track item statistics. Basic asset management could also be built into this component.

Item Bank
A subcomponent under item management, the item bank is a data store and web service allowing other components to query and retrieve items. Keeping a versioned history for all items, this component tracks lineage and item relationships and provides other components with the definitive source for an item.

Test Authoring
The test authoring interface allows NCSC test authors to construct tests and deploy them to the test delivery components. Items are pulled from the item bank and grouped into subtests. This component also supports the construction of the multi-stage adaptive algorithms.

Test Bank
The test bank manages tests created by the test authoring component. It provides a repository for tests that manages versioning and exposes tests through a web service. The test bank and test author are developed alongside one another.

Test Packager
Invoking the process of bundling the test and item content, packaging the test in a format optimized and securing the package for test delivery transpires in the test packager. If needed, this component provides a fairly standalone service that deploys easily.

3.2.2 Administration and Registration

Required pre-test components to set up schools in the system and schedule students to take tests ensue in the administration and registration component. Merging these two components into a single component proves acceptable as long as the functions for each remain distinct.

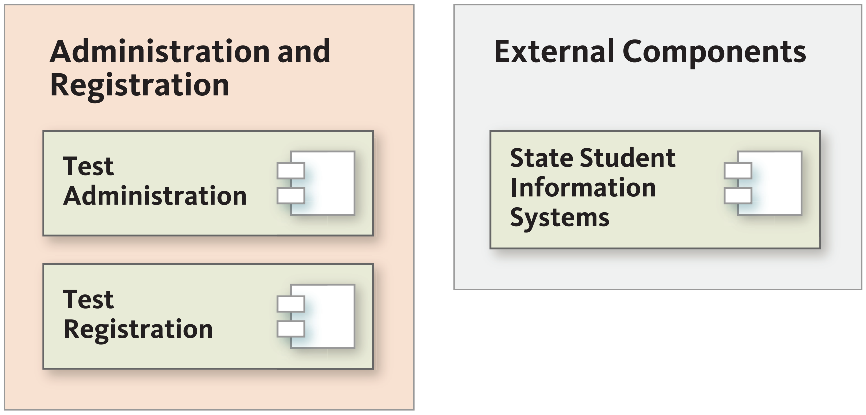

Administration and Registration
Test Administration
Test Registration
External Components
State Student Information Systems

Figure 3.3: Assessment Administration

State Student Information System or Data Warehouse (External)
The NCSC system ingests the following data objects from member states:

- District and school information for scheduling
- Student Information for administration and reporting
- Personal needs data for an accessible assessment

To lower the cost of implementation, define the common formats for these data.

Test Administration
This component manages the capabilities and methods required for assessment scheduling, test windowing, room scheduling, proctor assignment, student assignment and student identification methods.

The responsibility for storing basic state hierarchy information, including defining how schools and LEAs are grouped as well as any information for reporting, retains here. In addition, the assigning of published tests to hierarchy nodes along with the defining of testing windows for each administration remains in this component.

Test Registration

When creating a test window, students and teachers are registered. Registration consists of selecting the students that may participate in the exam period as well as the verification that all student data, from the SIS or data warehouse, imports correctly and has provided any necessary PNP information.

3.2.3 Delivery and Scoring

This section describes all of the components that are leveraged during test taking, from the client application that the student uses to the components required for capturing and scoring the student's responses.

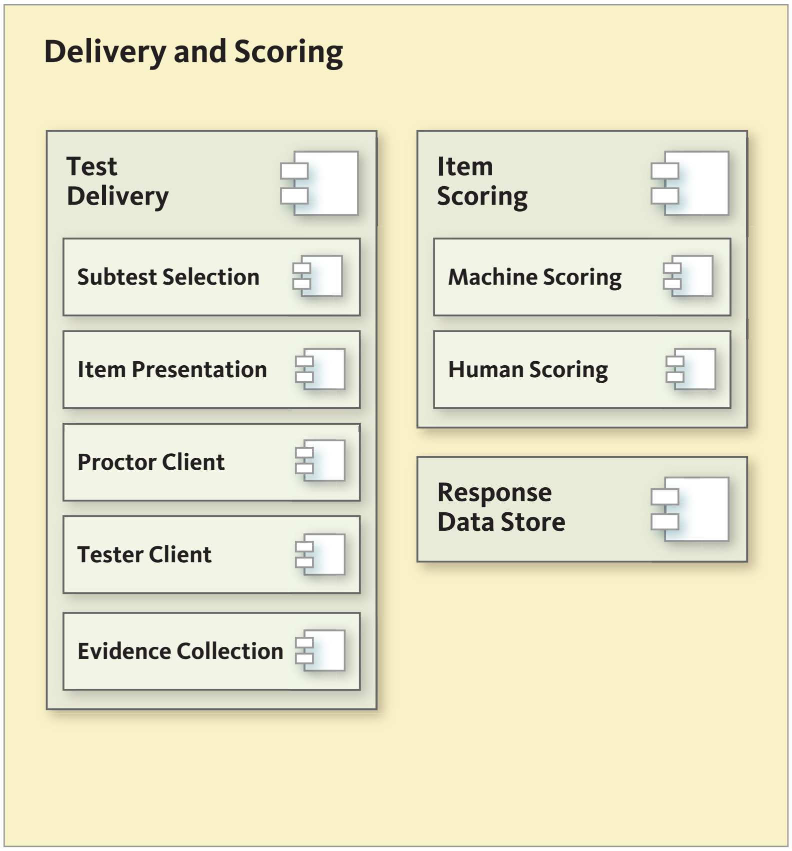


Figure 3.4: Assessment Delivery and Scoring

Test Delivery

Arguably the most unique standalone component within the NCSC assessment architecture includes test delivery. The overall responsibility of this component consists of:

- Delivering the assessment to the student securely
- Storing the student responses
- Storing other information about how the student responded (i.e. time to answer, time to render for the student, etc.)
- Delivering the test items in the proper accessible format that the student needs

Several subcomponents that characterize the innovative solution to this category subsist.

Subtest Selection

The responsibility for the selection of subtests, based upon the test taker responses in earlier sections, occurs in the subtest selection. As opposed to more traditional adaptive item selection algorithms, NCSC will develop a staged adaptive test with a custom selection algorithm. Instead, linear item sections assemble into a pool and the algorithm administers subsequent subtests based upon the earlier responses of the test taker. While still investigating the specific algorithms, the architecturally relevant known decisions include:

- Tests are administered in subtests, oftentimes over a period of multiple testing days
- Not all subtests of a test are administrated to a test taker
- Subsequent subtest selection require machine scoring of items from previous subtests/sections
- During test authoring, test authors define the subtest's items and sequence and probably present linearly to the test taker

Item Presentation

The item presentation component displays a test item to a tester. Build this component in a way that allows it to be reused through out the system. For instance, it may be reused within item management or test authoring giving previews of items. This not only saves development time by not having to develop redundant functionality, but also helps ensure consistency in how items present across the system.

Proctor Client

A proctor uses this component to manage the delivery of a test. It allows the proctor to start, stop, suspend and resume tests. It also provides the proctor with a guide on how to best assess a particular item or subtest.

Tester Client

This subcomponent interacts with the student. It delivers items to the student and gathers the responses and response metadata. It also contains the tools the student needs to take the test. (i.e. calculators, tables, accessibility tooling, etc.)

Evidence Collection

Evidence collection gathers evidence generated during the test delivery process. This substantiation includes a variety of formats (digital images, text, video and audio) collected during the test session. The test delivery system may utilize traditional technology components such as a standard resolution web camera for this collection process. The proctor, who interacts with the delivery system, typically initiates collection activities.

Response Data Store

During test delivery, the student's responses are sent to the response data store after each question is submitted, and on a periodic schedule for longer questions to ensure minimal work loss. This operational data store is accessed by the scoring components and serves as an intermediate collection point of assessment data.

Machine Scoring

Scoring of the summative assessment occurs in three areas:

- Real-time machine scoring of subtests and storage of response selections
- Collection and storage of evidence gathered during the test session
- Distributed scoring that happens with human beings interpreting test taker responses, proctor annotations and evidence collected

Machine scoring programmatically scores an item in real-time while the student takes the test. Due to the adaptive engine's need to either decide the next item or select an item a few items ahead, machine scoring is designated as a high-performing component.

Human Scoring

The human scoring module keeps its own application within the NCSC portal. This allows for the workflow, assignment and functional interfaces for distributed human scorers access to the test content, rubrics, test taker responses and evidence collected.

Evidence Collection

The NCSC system captures evidence of a student response. The evidence is secured by the system as either an image or video that can later be evaluated by human scorers. Depending on usage and the file size of the evidence taken, there could potentially be a significant increase in bandwidth requirements for a school. The diagram below demonstrates a student and proctor conducting an assessment on two separate devices.

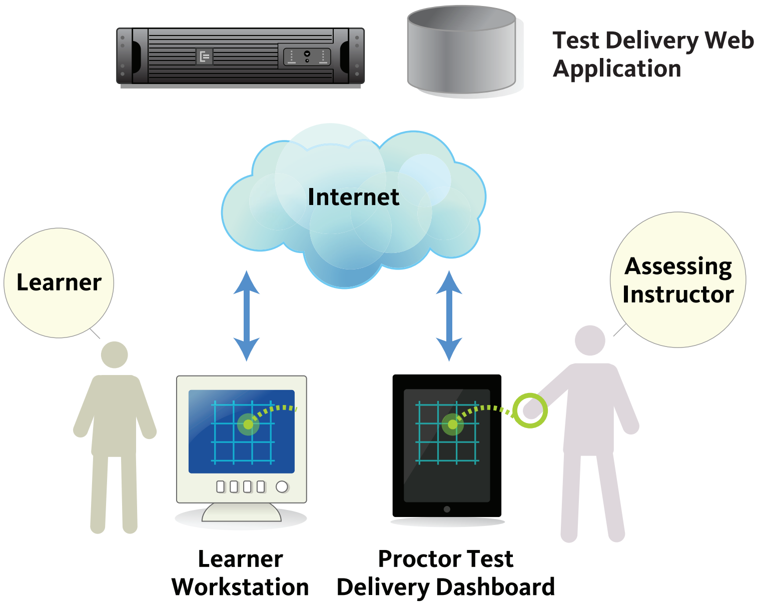

Test Delivery Web Application
Internet
Learner
Assessing Instructor
Learner Workstation
Proctor Test Delivery Dashboard

Figure 3.5: Proctor and Student Delivery

Actual delivery or presentation of the test to the student, through its items, occurs here. The innovated model utilized, facilitated test delivery model allows an assessment to be delivered in one of two modes, as a single workstation, or as a testing workstation coupled with a tablet or workstation devoted to assessing instructor or proctor experience.

The assessing instructor has the capability to annotate the student's response, view student responses as made, or provide an interpreted response with explanatory rationale, etc. In addition, the assessing instructor generates supporting materials, pauses and resumes the test session and initiates a snapshot or video recording evidence collection process.

All delivery sessions can materialize in both modes of delivery. If the assessing instructor interface is not available, all proctor operations may be invoked from the test delivery workstation through a series of keyboard or menu selections. Invoking all proctor operations from the test delivery workstation through a series of keyboard or menu selections may take place depending upon the availability of the assessing instruction interface.

3.2.4 Results and Reporting

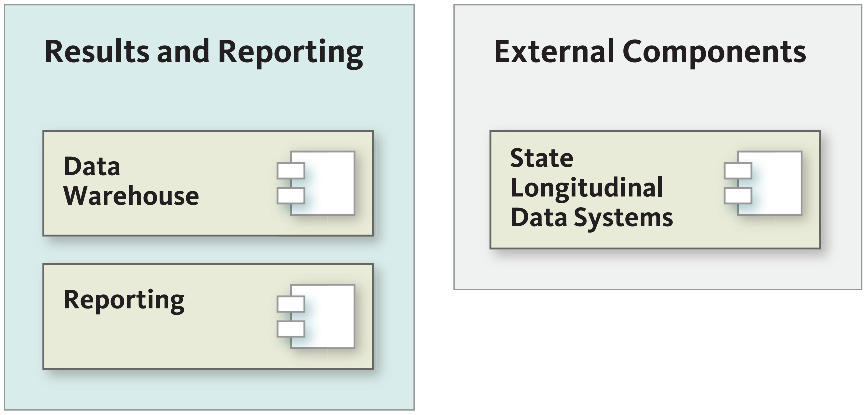


Figure 3.6: Assessment Results and Reporting

The components that implement the assessment results requirements follow a fairly traditional enterprise and big data model. Transactional results are stored in the response data store. These response data, utilized for scoring decisions within the assessment as well as information that drives the distributed scoring procedures, reside in this component. While these data remain in this state, considered unreleased, modifications by the distributed scoring procedures may transpire.

Data Warehouse
This component contains post-scoring information that is moved from test delivery when the assessment window closes. Results are processed and made available for reporting. This component includes the data store that feeds reporting. In addition, store and generate aggregate results here.

Reporting
This component's basic functionality relies on the ability to run NCSC reports against the data warehouse and deliver them in multiple formats to authorized users. To achieve this, several components within the NCSC reporting application fulfill the requirements for reporting and results management.

- Report Output – Generates the report output. This includes data visualization components for the display of interactive or complex datasets. The ability to modify and customize reports based on NCSC standards or customized for member states is also important. NCSC needs to edit and update format, content and logic within reports.

- Report Writing – Permits NCSC staff or its subcontractors to author reports and publish within the reporting services framework.
- Report Distribution – Allows NCSC staff to schedule reports executed and cached for performance reasons, as well as schedule distribution rules where reports may be sent via a variety of mechanisms to users for consumption without logging into the system. Designed to have a configurable pipeline of data, member states can configure the secure delivery of assessment results to their own system via the configuration of a set of rules and authentication parameters. This allows a more seamless delivery of individual and aggregate assessment results into the state longitudinal data systems. Cached reports, distribution rules and configuration settings are all stored in the reporting metadata database.
- Public Report Data – NCSC desires to offer a managed area where it can expose aggregate performance data to its member states, or possibly to the media and other public institutions. These data show multistate performance and allow stakeholders to point and click within the datasets, identifying trends or understanding the performance of student groups on the alternate assessment. The reporting application offers these services and implements them through the ad-hoc reporting services component.
- Psychometric Analysis – The reporting components also provide the foundation for post-assessment psychometric analysis. Most psychometric analysis takes place outside of the assessment platform, within psychometricians' statistical analysis tools, such as SPSS and SAS, and reported back to the assessment system to include necessary data elements with the item bank, assessment, etc. The system provides a handful of pre-defined performance reports, and allows psychometricians with appropriate access to export results data and analyze them within their own tools.

3.2.5 Ancillary Content

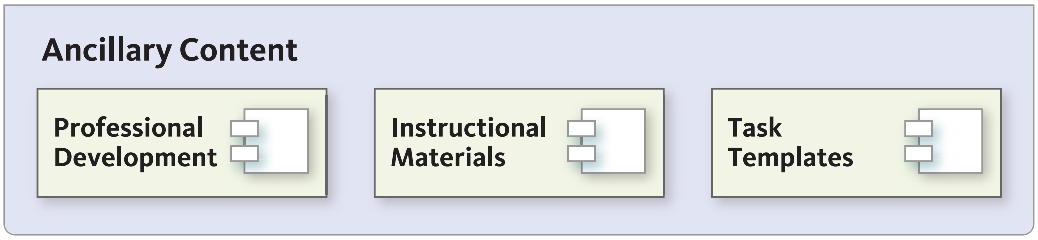


Figure 3.7: Ancillary Content

The ancillary content category contains three free-standing applications and databases. The NCSC System could connect or couple these systems; however, no requirement to do so exists other than house them within the same delivery portal and allow common user authentication via other NCSC components.

Professional Development Content Management
This application allows NCSC staff to publish and manage content used across the consortium in professional development for special education in general, or to specifically provide content that guides and prepares assessing instructors for the summative assessment. Additionally, the professional development system manages some lightweight tracking features, allowing educators to attest that they have learned the content. The system tracks, reports on and uses this attestation as a prerequisite for subsequent test delivery activities.

Instructional Materials Content Management
This application provides a simple, integrated content management system allowing for publishing a wide variety of content types, aligning and searching for educators looking to improve instructional materials or techniques.

Eventually, submitting instructional materials and having community features such as rank, grade and curate the content occur.

Task Template Management
While normally not a standard application in assessment systems, this application allows the NCSC team to author, manage and align tasks to the core content connectors. The designed central repository of these task templates allow access to them during item authoring and use by a distributed team of item writers. Item writers utilize these aligned tasks to guide them in the item writing and item design tasks.

Efficient authoring of items within this system requires the understanding the subtleties of differences among students with intellectual and physical disabilities. Item writers may need additional context and support during the authoring process. Standardizing this process ensures alignment back to the CCSS via core content connectors and task templates.

3.2.6 NCSC Shared Services

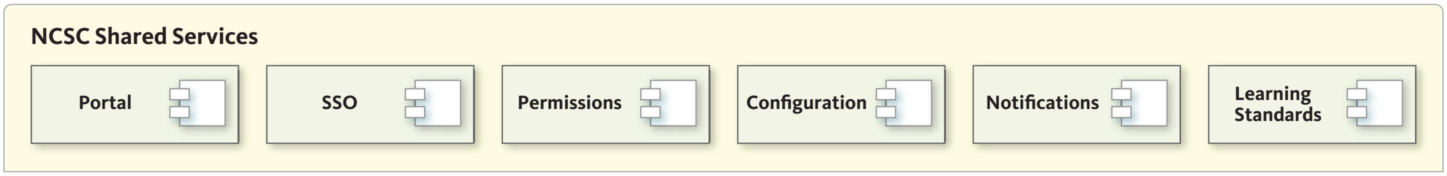


Figure 3.8: Shared Services

NCSC Portal
The NCSC Portal houses all components, or applications, of the NCSC Solution. The portal allows for modular presentation of the different components and, if desired, some configuration and preferences for the layout and permissions of different applications to different classes of users. The portal may be an off-the-shelf (OTS) or open source solution, or it may simply include an application framework provided by NCSC's application development and integration vendor.

Permissions Management
The permissions management application allows NCSC administration personnel to define custom roles and membership of user accounts to roles. It allows function points from applications exposed and assigned within those roles.

Permissions management also includes a second dimension of access control. This access is based upon the organizational hierarchies ingested and managed. Users have a simple access modifier assigned to the hierarchy. This access modifier includes read-only, read/write or no access (R, W, N). During development identify the specific rules around configuration and inheritance of these permissions.

Sign On
Deploy an open source SSO component for the NCSC production stack. A technology such as OAuth2.0 (http://oauth.net/2/), as implemented by Open AM (http://openam.forgerock.org/), represents a typical solution for this component. It allows single sign on capabilities in the case that external systems interoperability require a standards-based SSO component be present to eliminate nuisance logins.

Notification Logging

Consolidate system notifications in a single logging repository here. This allows for more efficient maintenance and diagnostics in a system with a wide variety of application modules. The notifications service configures to allow certain notifications escalation and distribution through typical system notifications channels, including text message, email, etc.

System Configuration

The system configuration application, its data store and attendant services provide centralized configuration options for the operation of the NCSC application portal.

Learning Standards

This application serves two purposes. First, it provides an interface to allow NCSC administration personnel to manage or ingest the CCSS and maintain a copy within the NCSC environment. All components that wish to access or align to the CCSS then have this available. Second, it provides an additional mapping element to the Common Core Connectors (CCC), maintained solely by NCSC. The application development phase of this component may identify additional functionality regarding mapping and reporting.

3.3 Alignment to the Assessment Lifecycle Model

This diagram shows the alignment of the NCSC solution components to the SIF Association's Assessment Lifecycle presented earlier.

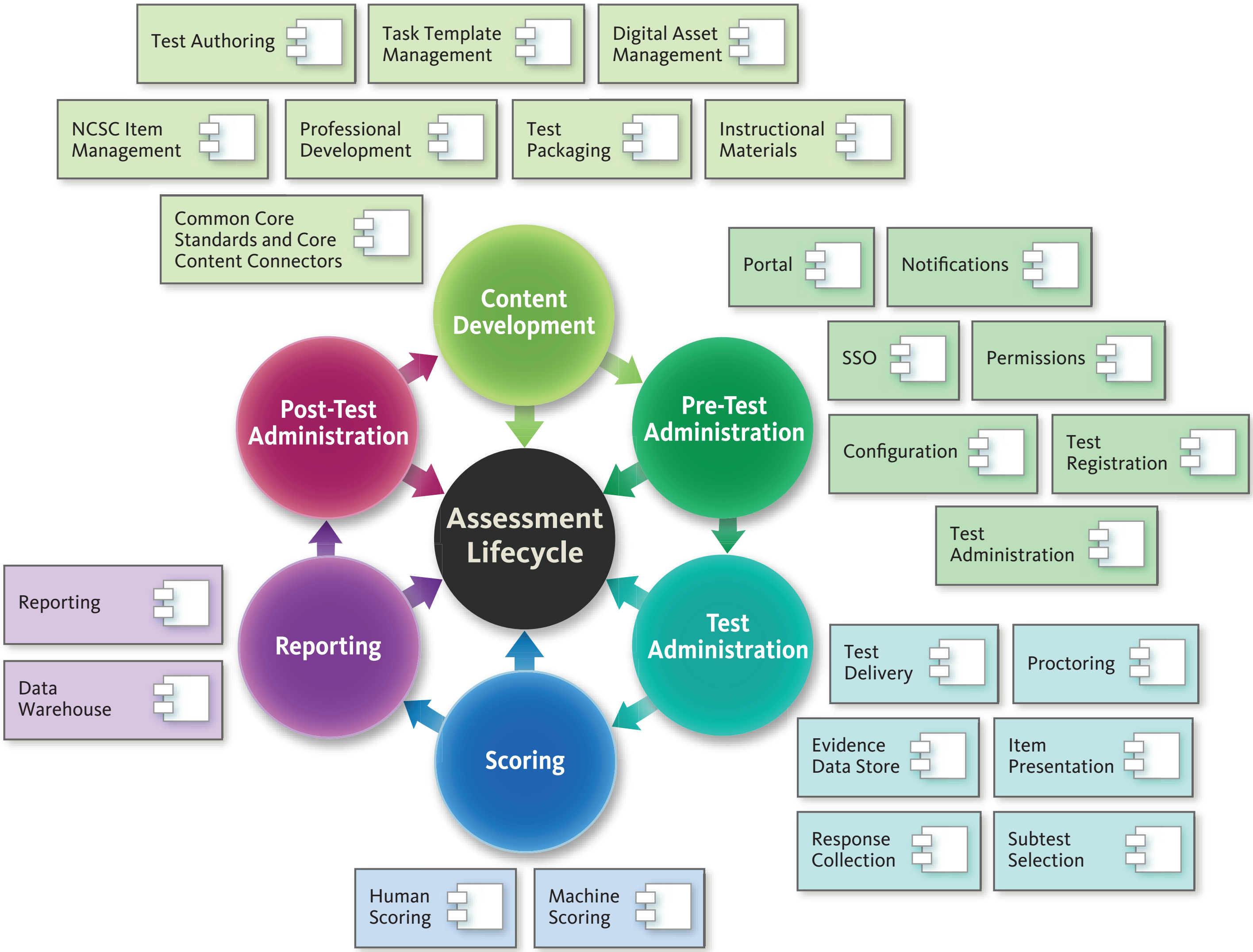


Figure 3.9: Alignment NCSC to Assessment Lifecycle

4. Interoperability

To achieve the benefits of componentization, the architecture must clearly define inter-component communication. When describing the communication between components, the architecture establishes protocols for how:

- Data transports (SSL, FTP, TCP)
- Data formats (APIP, SIF)
- Interfaces are defined (REST)

The NCSC system relies on existing standards as much as possible and only extends the standards where identified gaps occur. Ideally NCSC will work with existing standards organizations to make those extensions part of the industry standard. This chapter gives a summary of the relevant industry standards, identifies the points in the system where interoperability is required and gives recommendations for defining those points.

4.1 Interoperability Standards

This section gives an introduction to the existing standards work currently underway in the assessment space. It is intended to provide the development vendors, who first approach the NCSC project, some context and to give a brief introduction to these initiatives.

4.1.1 CEDS – Common Education Data Standards

CEDS provides a specified set of the most commonly used education data elements to support the effective exchange of data within and across states, as students transition between educational sectors and levels and for federal reporting. A common vocabulary is created to enable more consistent and comparable data used throughout all education levels and sectors necessary to support improved student achievement. The standards are developed by NCES, with the assistance of a CEDS Stakeholder Group that includes representatives from states, districts, institutions of higher education, state higher education agencies, early childhood organizations, federal program offices, interoperability standards organizations and key education associations and non-profit organizations. CEDS is a voluntary effort driven by the intent to increase data interoperability, portability and comparability across states, districts and higher education organizations.

CEDS provides the description for and covers the elements from the domains of Early Learning, K12, Postsecondary, Assessments and Learning Standards. Specifically of interest to NCSC, the CEDS community has worked to include the RTTA required elements in the Domain Entity Schema.

You can find the CEDS data model at http://ceds.ed.gov/.

4.1.2 APIP – Accessible Portable Item Protocol

The APIP standard provides assessment programs and question item developers with a data model for standardizing the interchange file format for digital test items. The idea focuses on accomplishing two goals. First, it allows digital tests and items to be ported across APIP compliant test item banks. Second, it provides a test delivery interface with all the information and resources required to make a test and an item accessible for students with a variety of disabilities and special needs.

APIP assumes that a test delivery system combines two different profiles (item XML and the user profile) to tailor the test delivery to a specific user's needs. The item XML has two main parts: the item information (meta information about the item) and the content XML (the actual content to be presented to the user) based on QTI 2.0. The user profile, called the Personal Needs Profile (PNP), contains information about what the user would need to access the information and may contain specific preferences about that need.

The APIP standard intends to foster interoperability of the content packages and user accessibility needs with PNP files. APIP delivery systems support the content and PNP information and do not directly communicate with other APIP delivery systems. APIP delivery systems support the information supplied in the APIP exchange files, though they do not require the delivery system to use those exchange files at the moment of delivery. Delivery systems use the information supplied in those exchange formats and can elect to use proprietary or other delivery-focused formats during content rendering.

The IMS GLC and APIP specifications can be found on the IMS website. The APIP Best Practices document gives a basic overview: http://www.imsglobal.org/apip/apipv1pocf/APIPv1po_Best_v1pocf.html

4.1.3 SIF – School Interoperability Framework

SIF develops interoperability specifications for the entire education enterprise from food service to learning standards to assessment to student information. This view of standards demonstrates a systemic view of education. Not all applications need to exchange data with one another; for example, a food service application does need to have assessment scores. Portions of the SIF Specification apply to the work for NCSC. Defining students, teachers and the hierarchy requires SIF. In addition, use of components of the assessment data model is necessary.

SIF Implementation Specification 2 provides both an infrastructure standard and a data model. As of SIF Implementation Specification 2.5, the data model exists separately from and developed independent of the infrastructure model. The infrastructure defines protocols for transporting data and messaging. For the purposes of the NCSC solution, sole use of the SIF data model proves sufficient without having to implement the infrastructure or middleware - Zone Integration Server (ZIS) and SIF Agents.

With the release of the SIF v3.0, SIF introduces a choice for the underlying technologies. With SIF v3.0, a direct interface allows an application to connect to another application without the middleware. In addition, SIF v3.0 supports both REST and SOAP, and both bound to a common underlying API enabling bridging between SOAP or REST applications.

The SIF Implementation Specification 2.6 - http://specification.sifassociation.org/Implementation/US/2.6/.

4.1.4 AIF – Assessment Interoperability Framework

When creating assessment systems and following standards, some confusion existed. Both IMS and the SIF Association provide assessment standards. The confusion included deciding which standards to use when and building an entire assessment system trying to utilize only one standard. Using both technical standards is required. The SIF Association (SIF) and IMS GLC (IMS) communities, in partnership with the consortia, LEAs, SEAs and vendors, joined together to develop a standards-based technical solution in support of assessments for deployment in states and schools.

The diagram below provides a high-level overview of AIF. Each component for an assessment system addresses an interoperability standard. The green arrows represent interoperability for IMS, the yellow arrows SIF, the purple arrows for both IMS and SIF and the grey arrows for future development.

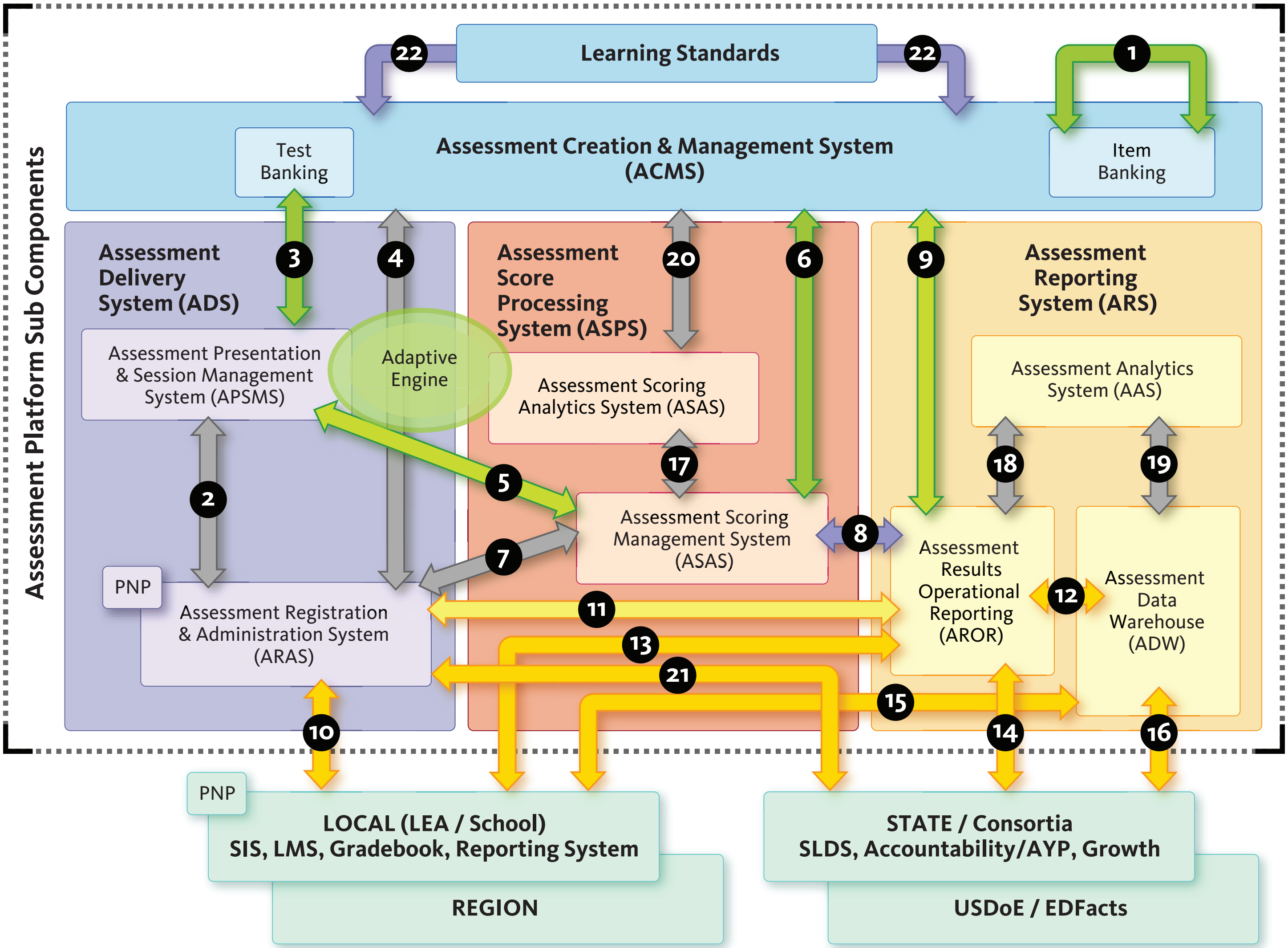

Assessment Platform Sub Components
Learning Standards
22
22
1
Test Banking
Assessment Creation & Management System (ACMS)
Item Banking
Assessment Delivery System (ADS)
3
4
Assessment Score Processing System (ASPS)
20
6
9
Assessment Reporting System (ARS)
Assessment Presentation & Session Management System (APSMS)
Adaptive Engine
Assessment Scoring Analytics System (ASAS)
Assessment Analytics System (AAS)
5
17
18
19
2
Assessment Scoring Management System (ASAS)
8
Assessment Results Operational Reporting (AROR)
7
12
Assessment Data Warehouse (ADW)
PNP
Assessment Registration & Administration System (ARAS)
11
13
21
15
10
14
16
PNP
LOCAL (LEA / School) SIS, LMS, Gradebook, Reporting System
REGION
STATE / Consortia SLDS, Accountability/AYP, Growth
USDoE / EDFacts

Figure 4.1: Assessment Interoperability Framework (AIF)

The AIF defines which standards used based on components. The result of delineating where IMS and SIF each play a role in assessment interoperability resulted in little overlap. Where overlap occurs, use both SIF and IMS. Documentation created will be available in late winter 2012. CEDS 3.0 incorporates most of AIF.

For more information about AIF, please visit https://ceds.ed.gov/aif.aspx.

4.2 NCSC Interoperability Process Flow

Swim lanes identify areas where interoperability needs definition. The following swim lane diagrams denote the process workflows. The area where the workflow jumps over to a different swim lane symbolizes an interoperability requirement.

4.2.1 Creating an Assessment

This swim lane represents the process of creating a test ready for delivery, starting with items being developed by item development vendors and ending by sending a test package to test delivery.

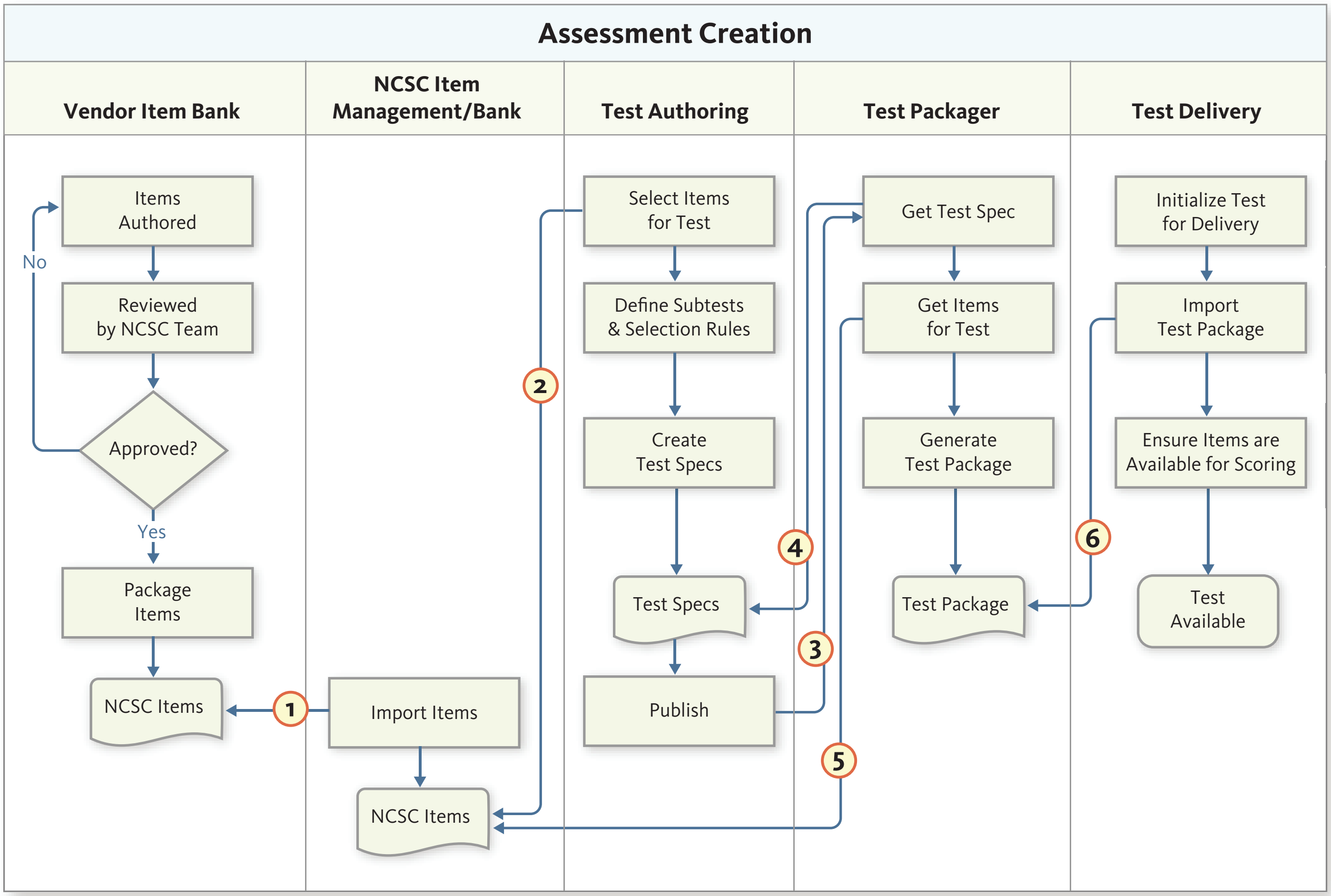


Figure 4.2: Vendor Item Bank to Delivery

Swim Lane Label	Domain Object	Source Component	Target Component	Standard
1	Items	Vendor Item Bank	NCSC Item Bank	SFTP + APIP
2	Item Query	NCSC Item Bank	Test Authoring	REST + XML
3	Trigger	Test Authoring	Test Packager	REST + XML
4	Test Specs	Test Packager	Test Bank	REST + APIP + SIF
5	Items	Test Packager	NCSC Item Bank	REST + APIP
6	Test Package	Test Delivery	Test Packager	REST + APIP

4.2.2 Setting up a Test Administration

This swim lane outlines the process of registering students from the SIS student repository and delivering the test.

Initially student data resides in LEA and SEA student information systems (SIS) or data warehouses. Student personal needs profiles may also reside in the systems, or it may simply be entered during registration if it is not available within the SIS upload.

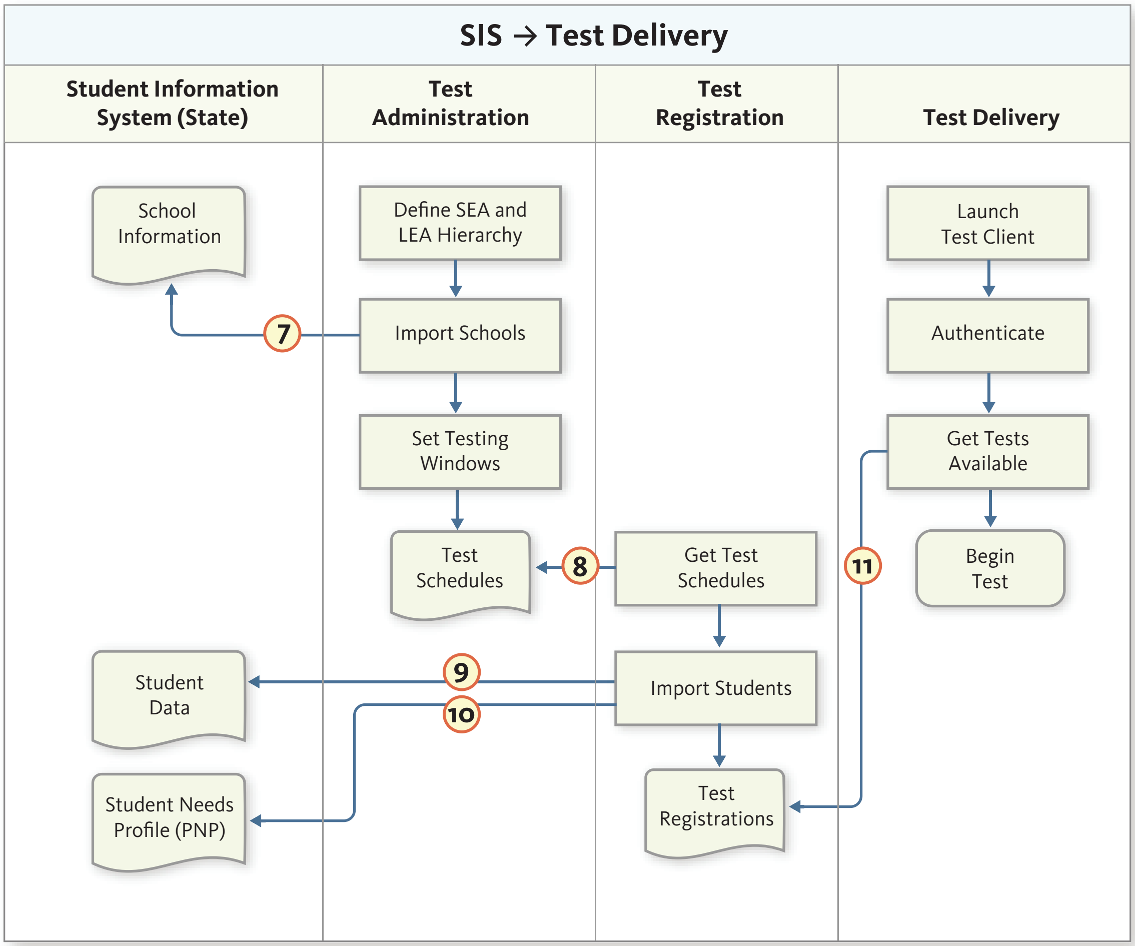


Figure 4.3: SIS to Assessment Delivery

Swim Lane Label	Domain Object	Source Component	Target Component	Standard
7	Schools	State Data Systems	Test Administration	SFTP + XML + CSV, SIF
8	Test Schedules	Test Administration	Test Registration	REST + SIF
9	Student Profiles	State Data Systems	Test Registration	REST + SIF
10	Personal Needs Profiles	State Data Systems	Test Registration	REST + APIP
11	Test Registrations	Test Delivery	Test Registration	REST + SIF

4.2.3 Delivering an Assessment

This swim lane shows the processes around delivering and scoring a NCSC test. Tests are delivered as multistage adaptive with each state or subtest delivered in a single session. Scoring incorporates both machine and human scoring with final test results being stored in the data warehouse.

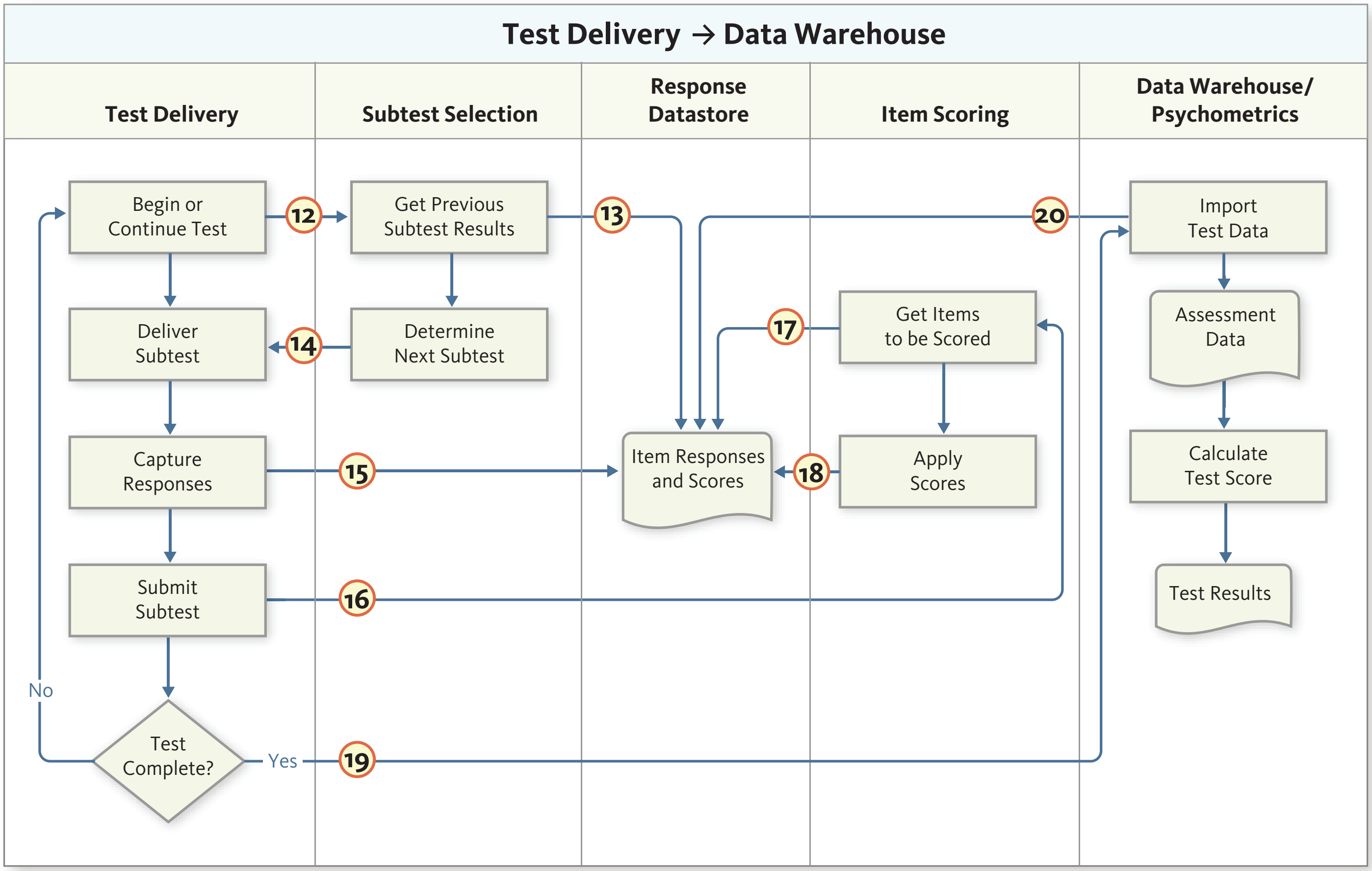


Figure 4.4: Test Delivery to Data Warehouse

Swim Lane Label	Domain Object	Source Component	Target Component	Standard
12	Trigger	Test Delivery	Subtest Selection	REST + XML
13	Item Results	Response Data Store	Subtest Selection	REST + SIF
14	Subtest	Subtest Selection	Test Delivery	REST + SIF
15	Item Responses	Test Delivery	Response Data Store	REST + SIF, APIP
16	Trigger	Test Delivery	Item Scoring	REST + XML
17	Item Results	Response Data Store	Item Scoring	REST + SIF
18	Item Scores	Item Scoring	Response Data Store	REST + SIF, APIP
19	Trigger	Test Delivery	Data Warehouse	REST + XML
20	Testing Results	Response Data Store	Data Warehouse	REST + SIF, APIP

4.2.4 Managing Test Results

This swim lane displays the steps of finalizing and moving test data out to stakeholders. Psychometricians access the test item responses in the NCSC Assessment test system for analysis. In addition, tools for psychometric analysis (such as SAS or SPSS) analyze the item results and transform the results into a statistical item performance set by psychometricians and then finally load them into the NCSC Item Bank.

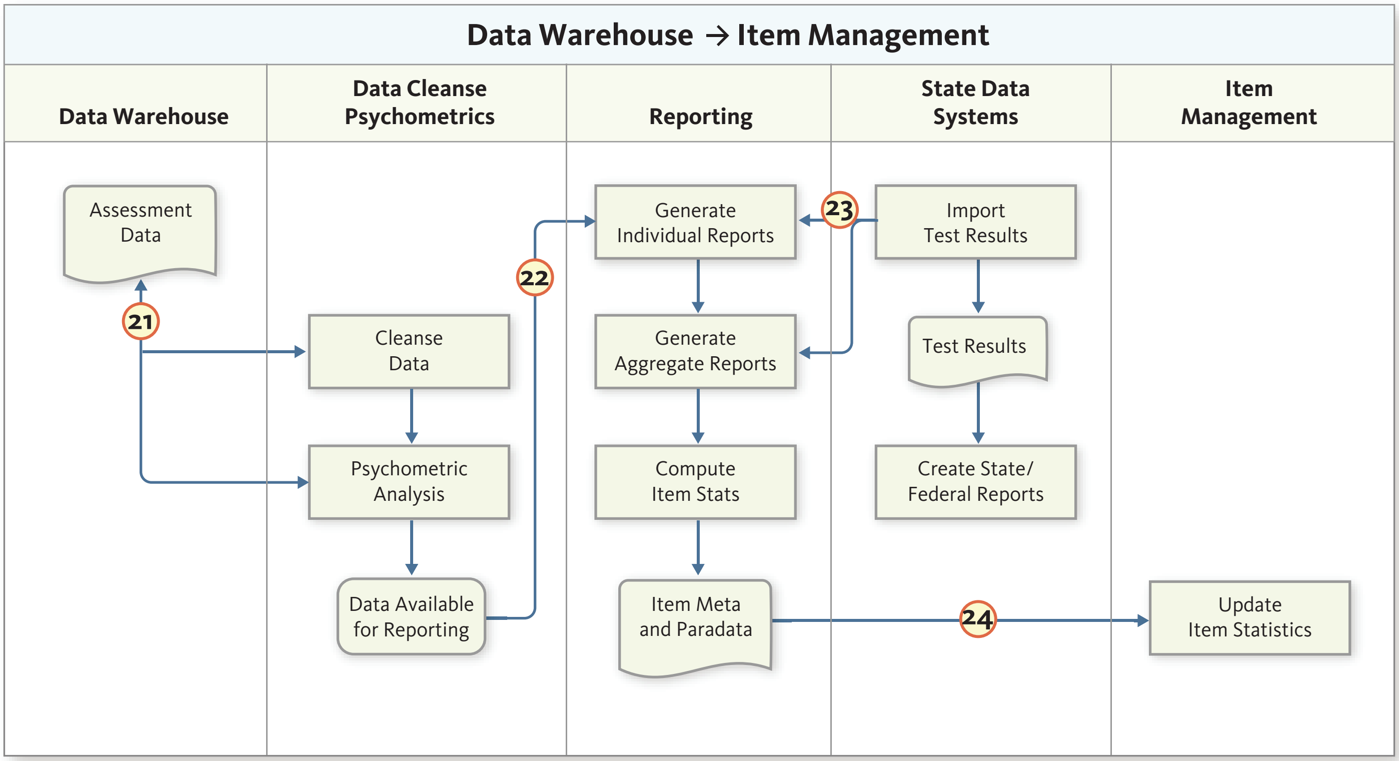


Figure 4.5: Data Warehouse to Item Management

Swim Lane Label	Domain Object	Source Component	Target Component	Standard
21	Assessment Data	Data Warehouse	Psychometric Services	REST + SIF
22	Trigger	Psychometric Services	Reporting	REST + XML
23	Test Reports	Reporting	State Data Systems	SFTP + XML, CSV
24	Item Statistics	Reporting	Item Management	REST + SIF

4.3 Summary of Interoperability Scenarios

The following table provides a summary of the interoperability flows identified in the preceding scenarios:

Swim Lane Label	Domain Object	Source Component	Target Component	Standard
1	Items	Vendor Item Bank	NCSC Item Bank	SFTP + APIP
2	Item Query	NCSC Item Bank	Test Authoring	REST + XML
3	Trigger	Test Authoring	Test Packager	REST + XML
4	Test Specs	Test Packager	Test Bank	REST + APIP + SIF
5	Items	Test Packager	NCSC Item Bank	REST + APIP
6	Test Package	Test Delivery	Test Packager	REST + APIP
7	Schools	State Data Systems	Test Administration	SFTP + XML, CSV
8	Test Schedules	Test Administration	Test Registration	REST + SIF
9	Student Profiles	State Data Systems	Test Registration	REST + SIF
10	Personal Needs Profiles	State Data Systems	Test Registration	REST + APIP
11	Test Registrations	Test Delivery	Test Registration	REST + SIF
12	Trigger	Test Delivery	Subtest Selection	REST + XML
13	Item Results	Response Data Store	Subtest Selection	REST + SIF
14	Subtest	Subtest Selection	Test Delivery	REST + SIF
15	Item Responses	Test Delivery	Response Data Store	REST + SIF, APIP
16	Trigger	Test Delivery	Item Scoring	REST + XML
17	Item Results	Response Data Store	Item Scoring	REST + SIF
18	Item Scores	Item Scoring	Response Data Store	REST + SIF, APIP
19	Trigger	Test Delivery	Data Warehouse	REST + XML
20	Testing Results	Response Data Store	Data Warehouse	REST + SIF, APIP
21	Assessment Data	Data Warehouse	Psychometric Services	REST + SIF
22	Trigger	Psychometric Services	Reporting	REST + XML
23	Test Reports	Reporting	State Data Systems	SFTP + XML, CSV
24	Item Statistics	Reporting	Item Management	REST + SIF

4.4 Interoperability Guidelines

The following set of interoperability guidelines informs the NCSC system.

1. Employ SSL or SFTP for all secure communication.
2. Create REST based internal component interfaces.
3. Define a single source of truth for data elements.
4. Leverage existing standards where available.
5. Avoid tight coupling of components.
6. Use flexible standards for communicating with external systems.

5. Data Architecture

When defining an overall IT assessment architecture, a data architecture lays the foundation for everything related to data. All models, policies and standards that govern data collection, storage and use in data systems and in organizations occur in the data architecture.

Focus on data architecture for the NCSC Assessment System and how the needs and purposes can be met through the design and execution. This section describes the relevant aspects of the data architecture including storage techniques and tools utilized by NCSC. General principles and constraints are identified, as well as specific areas that traditionally represent challenges in education and assessment systems. Finally, any relevant standards are referenced.

An additional discussion needed includes the historical aspect of data architecture. A database stored all, potentially a relational database. This often led to a disconnect between the use and storage of data; therefore, potentially leading to interoperability and sharing of data issues. With a relational database, many tables are used to design the relationships and hierarchies between all of the data.

An innovative technology noticed in the marketplace includes moving away from relational databases to NoSQL (not only SQL) databases. NoSQL databases are non-relational, distributed, open-source, horizontally scalable and web-scalable. Currently over 120 NoSQL databases supporting the numerous types exist, including:

- Graph
- Multimodel
- Object
- Grid and Cloud
- XML
- Multidimensional
- Multivalue
- Key value stores

NoSQL databases, typically created as open-source, have a developed community supporting the work.

5.1 General Principles

The following principles are considered as technical decisions around data repositories for the NCSC Project:

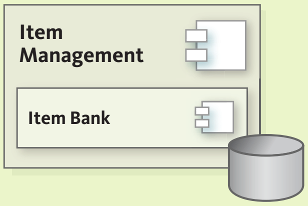
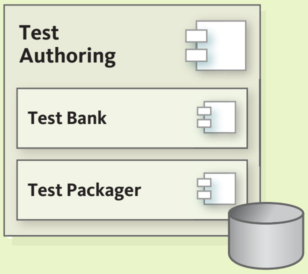
1. Components should not access other components' data stores directly. Services complete this work as it reduces dependencies and allows components to evolve more independently.
2. Select a storage mechanism that fits the intended use of the data.
3. Only one component serves as the sole source of truth for data.
4. Ensure versioning is applied to domain objects only when required. Related objects may capture the version on an as needed basis. For example, an item may change after it was used on an assessment. The assessment would also record the version of the item used so the assessment would pull the accurate items for analysis. Maintain this versioning history (for example, a version of a test).
5. Data are accessible for users of the system to perform their functions and appropriate roles are assigned.
6. When using relational databases, normalize tables to third normal form unless there is a compelling reason to make an exception.
7. Data stores should have appropriate schema documentation and be accessible through enterprise-class database management tools.
8. Data are protected from unauthorized use and disclosure.
9. For any customer development, adopt and use database programming guidelines and naming conventions to ensure professional and consistent database designs.
10. All entities must have globally unique keys generated to keep exclusivity across the system.

5.2 Data Source Inventory

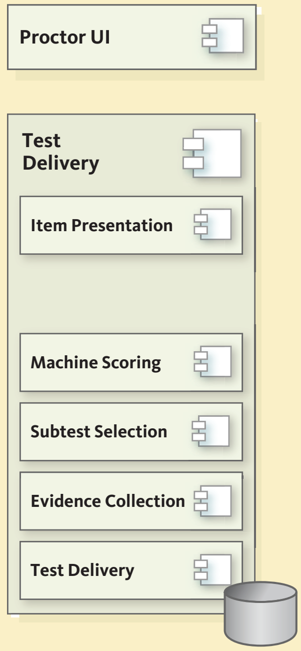
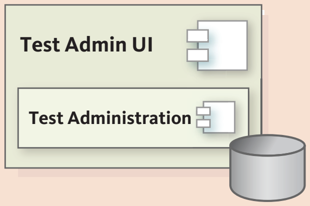
Identifying data sources and naming specific data elements proves critical in a fully functional system. Providing a consistent definition of the data ensures a reliable transfer and representation of the information from component to component. In addition, knowing what data are collected and needed provides for decision-making, reporting and analysis.

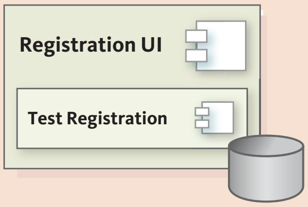
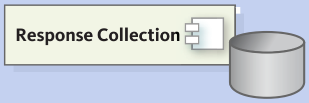
Each category from the NCSC Assessment System Architecture and component within the categories has numerous data sources, data objects and necessary updates. The following table represents this information and provides an overall representation of the data. Each data object needs to have specific elements identified to be fully demonstrative. Additional external data sources may need to be added as the implementation of the assessment system materializes.

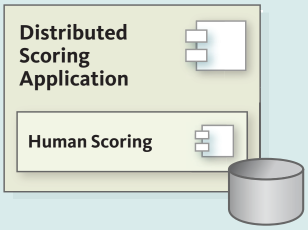
The frequency of the updates may vary depending upon when the test administration occurs. For example, it is critical to receive changes to a student in near-time when an assessment administration occurs; however, when an assessment administration is not being conducted, these data does not need regular updates

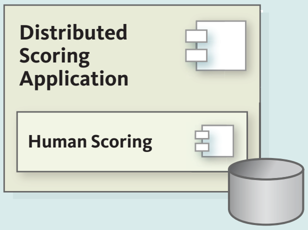
Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
Assessment Creation and Management  	1. NCSC Item Bank	A single repository of all item content and metadata reside here. Item performance statistics are also maintained here.	item version	test administration
			item publish date	test administration
			item ID	test administration
			item name	test administration
			item type	test administration
			item owner	test administration
			item subject	test administration
			item grade level	test administration
			item scoring data	test administration
			item standard	test administration
			stem	test administration
			item statistics	test administration
			item container	test administration
Assessment Creation and Management  	2. NCSC Test Bank	All of the test metadata, including subtest selection criteria and selection algorithms are stored in the test bank. Test blueprints, if used, and publishing history may also be maintained here.	test name	test administration
			test ID	test administration
			test package	test administration
			test descriptions	test administration
			test subjects	test administration
			test grade levels	test administration
			test languages	test administration
			form version	test administration
			form publish date	test administration
			test type	test administration
			form name	test administration
			form ID	test administration
			form accommodations	test administration
			form level	test administration
			period	test administration
			grade levels	test administration
			form subjects	test administration
			form languages	test administration
			subtest reference ID	test administration
			form sections	test administration
			form platforms	test administration
			test asset ID	test administration
			section version	test administration
			section publish date	test administration

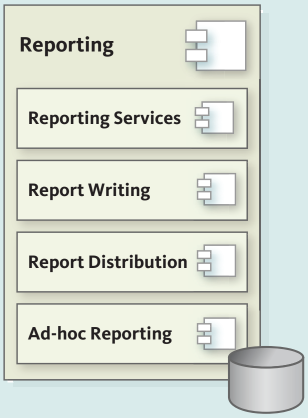
Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
			section ID	test administration
			section name	test administration
			item sequence type	test administration
			item select algorithm ID	test administration
			item select algorithm	test administration
			section time limit	test administration
			section sealed	test administration
			section reentry	test administration
			section assets	test administration
			section items	test administration
			sub test version	test administration
			sub test publish date	test administration
			sub test ID	test administration
			sub test name	test administration
			score reporting	test administration
			sub test subject areas	test administration
			sub test grade levels	test administration
			test ref ID	test administration
			sub test container	test administration
			sub test tier	test administration
			learning standard ref ID	test administration
			evidence ref ID	test administration
			evidence type	test administration
			evidence date time	test administration
			abbreviation	test administration
			description	test administration
			number of items	test administration
			items	test administration
			test asset reference ID	test administration
			meta data	test administration
			item ref ID	test administration
			test meta data	test administration
			test form	test administration
			test section	test administration
			test blueprints	test administration

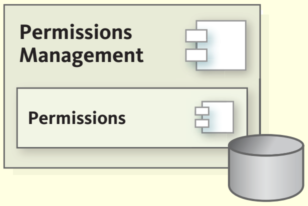
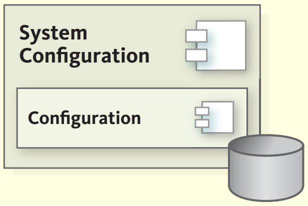
Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
Delivery and Evidence Capture  Proctor UI Test Delivery Item Presentation Machine Scoring Subtest Selection Evidence Collection Test Delivery	3. NCSC Test Repository	This repository houses all published tests and also where the test delivery accesses the released assessments for delivery to test takers. This might include a relational database for accessing published assessments and their metadata and a secure file repository for serving up the assessments and the associated binary assets.	session name	minute
			session type	minute
			unusual events	minute
			scheduled start date	test administration
			scheduled end date	test administration
			actual start date time	minute
			actual end date time	minute
			test admin ref ID	test administration
			test ref ID	test administration
			form ref ID	test administration
			LEA info ref ID	test administration
			school info ref ID	test administration
			staff personal ref ID	month
Administration and Registration 	4. NCSC Testing Administration Repository	Data about testing windows, testing program attributes, associations to published tests and program test delivery constraints reside here.	administration name	test administration
			administration code	test administration
			start date time	test administration
			finish date time	test administration
			administration assessments	test administration
			organizations	test administration
			administration meta data	test administration
			extended elements	test administration

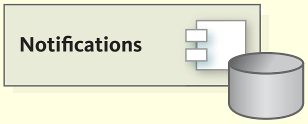
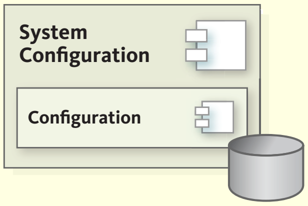
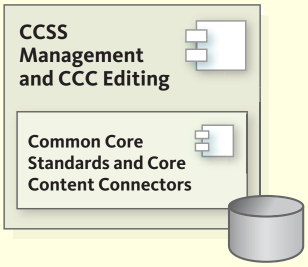
Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
Administration and Registration  Registration UI Test Registration	5. Rostered Students, Teachers and LEA Hierarchy	NCSC shall maintain a subset of the students, teachers and LEA hierarchies. This data is typically stored for the current school year only. These data need to be used for reporting and stored longitudinally as well. Testing assignments and registered entities (i.e., districts and buildings) that are participating in specific assessments are stored here.	ref ID	test administration
			student personal ID	test administration
			test ref ID	test administration
			administration ref ID	test administration
			test session ID	minute
			form ref ID	test administration
			creation date time	test administration
			start date time	test administration
			end date time	test administration
			test platform	test administration
			days of instruction	minute
			retest indicator	minute
			test attempt identifier	minute
			student special events	minute
			testing status	minute
			score publish date	day
			student grade level	test administration
			test grade level	test administration
			student snapshot	minute
			LEA info ref ID	test administration
			school info ref ID	test administration
			staff personal ref	test administration
			section info ref ID	test administration
			PNP	test administration
			LCI	test administration
			meta data	test administration
			extended elements	test administration
Scoring and Data Store  Response Collection	6. Response Data Store	This repository stores results and interim results from tests in process, completed and available for distributed scoring. These results are locked once scoring is completed and the test window closes.	ref ID	minute
			score table version	test administration
			score table publish date	test administration
			score table identifier	test administration
			score table name	test administration
			score value	minute
			meta data	test administration
			extended elements	test administration
			score metric	test administration

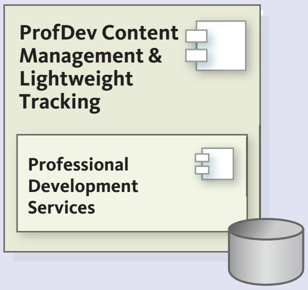
Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
			test item ref ID	minute
			response	minute
			response location	minute
			response correctness	minute
			view status	minute
			attempt status	minute
			number of attempts	minute
			time on item	minute
			item number	test administration
			item name	test administration
			test rubric ref ID	test administration
			item score	minute
			item score code	minute
			comments	minute
			trait score	minute
			feedback items	minute
			item aids	test administration
			item score	minute
			scale score	minute
Scoring and Data Store  	7. Interpretations and Workflow	This database stores the assignments for distributed scoring and any workflow attributes required to complete the scoring process. Any annotations, and interpretations shall also be stored here. Correction activities shall be recorded here.	ref ID	on demand
			rubric identifier	test administration
			scoring guide reference	test administration
			scores	test administration
			meta data	test administration
			extended elements	test administration
			assignments	near real time
			audit history	as required near real time on demand

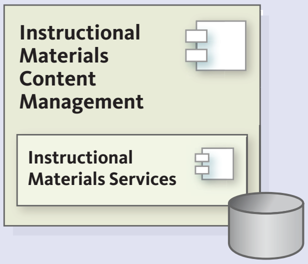
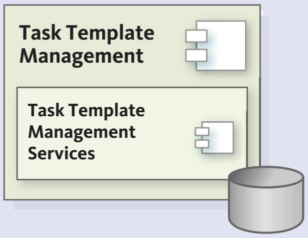
Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
Scoring and Data Store  	8. Evidence Data Store	The evidence data store houses the evidence collected during the assessment sessions. This evidence consists of proof necessary for evaluating a student's performance such as annotations, photographic and audio-visual evidence. An assessment may have several evidence files related to it.	ref ID	on demand

Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
Results and Reporting  Reporting Reporting Services Report Writing Report Distribution Ad-hoc Reporting	9. Reporting Meta Data	All reporting data that are not responses, scores and aggregate scores are kept here. This includes cached reports, report format, specification files, etc.	report data object	as required
			report container	as required
			report package	on demand
			report package type	on demand
			select content type	on demand
			specification file	test administration
			report ref ID	on demand
			test administration ref ID	test administration
			student personal ref ID	test administration
			test registration ref ID	test administration
			letter grade	near real time
			number score	near real time
			response score	near real time
			pass - fail	near real time
			percentile rank	near real time
			T-score	near real time
			Z-score	near real time
			achievement / proficiency level	near real time
			raw score	near real time
			item score	near real time
			scale score	near real time

Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
Results and Reporting	10. Responses and Aggregate Scoring	This data warehouse stores non-transactional data for student performance including individual, aggregate and item performance. Data are also be accumulated year-to year for longitudinal analysis.	transactional results statistic name calculation rule approval date expiration date exclusion rules source effective date discontinue date location measure description definition element name aggregate statistic info ref ID characteristics excluded value meta data extended elements	minute minute minute minute minute minute minute minute minute minute minute minute minute minute minute minute minute minute minute minute
NCSC Shared Services 	11. Permissions	This repository holds all application and LEA permissions and custom roles.	role ID group ID user ID application permission LEA permission permission log file rules file description	test administration test administration test administration minute minute minute minute minute
NCSC Shared Services 	12. System Logs	This repository centralizes log messages from all system components.	system wide log escalation rules app settings assembly string transformer connection strings evaluation context	minute minute minute minute minute minute

Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
NCSC Shared Services  	13. Notifications	System notifications are collected in a single logging repository. This allows for more efficient maintenance and diagnostics in a system with a wide variety of application modules.	notification object	minute
			notification rules	minute
			notification log	minute
			address	minute
			email list	minute
			email type	minute
			phone number list	minute
NCSC Shared Services  	14. Configuration Data	This contains system configuration information for all applications.	aggregation	minute
			name value pairs	minute
			service ID	minute
			rules file	minute
			description	minute
			comments	minute
NCSC Shared Services  CCSS Management and CCC Editing Common Core Standards and Core Content Connectors	15. NCSC Learning Standards Repository	The Common Core Learning Standards, Core Content Connectors and any additional learning standards are maintained here.	CCLS ID	test administration
			CCLS description	test administration
			CCLS type	test administration
			CCLS version	test administration
			CCLS grade level	test administration
			CCC ID	test administration
			CCC grade level	test administration
			CCC complexity	test administration
			CCC description	test administration
			Learning Progression Framework (LPF)	test administration
			LPF ID	test administration
			LPF group	test administration
			LPF indicator	test administration
			LPF description	test administration
			Progress Indicator (PI)	test administration
			PI ID	test administration
			skill sequence ID	test administration

Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
Ancillary Content  ProfDev Content Management & Lightweight Tracking Professional Development Services	16. Professional Development Data Store	All of the content required for professional development, including lightweight completion and certification data are housed here. It can be compared to a simplified LMS data store.	content object	minute
			user ID	day
			user name	day
			certification exam	test administration
			content asset ID	test administration
			rank	minute
			credit	minute
			exit	minute
			entry	minute
			file path	minute
			status	minute
			score	minute
			output file	day
			lesson location	test administration
			time	minute
			comments	minute
			session ID	minute
			objectives	test administration
			interactions	day
			path	day
			performance	minute
			student data	minute
			student preferences	month
			interactions	day
			demographics	test administration

Category and Component	Data Source	Description	Representative Data Object	Frequency of Updates
Ancillary Content  Instructional Materials Content Management Instructional Materials Services	17. Instructional Materials Data Store	This content repository allows a variety of authors to edit, publish and approve materials that are used for instructional purposes.	content	day
			content ID	day
			content name	day
			content type	day
			content description	day
			update date time	day
			content meta data	day
			content languages	day
			content stds alignment	day
			assessment content	day
			assessment ID	day
			assessment status	day
Ancillary Content  Task Template Management Task Template Management Services	18. Task Template Data Store	The task templates that have been mapped to Core Content Connectors are maintained here.	CCC task templates	test administration
			task ID	test administration
			task name	test administration
			test blueprints	test administration
			task description	test administration

5.3 Component Considerations

Each component within the NCSC Architecture requires unique data considerations. Each component outlines the special requirements, implications and constraints. This guidance intends to aid implementation teams and provide a framework for the solution, identifying concepts that are critical for architects.

5.3.1 Assessment and Item Creation

Two primary considerations in this domain exist: metadata and content compatibility and item identification. For metadata and content compatibility, items author in a system outside of the NCSC item repository and ingest when authoring activities complete. With this process, it is important to certify vendors systems in order to export content correctly. This certification should be completed as early as possible to validate that the systems do not pose any problems during export at the end of the item writing cycle. In addition, periodically check the data export after certification as the vendor's item data model may have changed as a result of system updates.

The second issue focuses on item identification. A technical approach must be in place to assure NCSC that item IDs generated by a vendor system are compatible with the inventory of item IDs currently stored or generated concurrently by another vendor.

5.3.2 Assessment Registration and Administration

Assessment registration and administration incorporates two different pieces for consideration in data architecture: student identities and LEA hierarchies.

Registration: Student Identities
Ensuring students have student identities that can cross state lines is vital. As the student population moves, tracking and providing the most up-to-date information about students' assessments should transpire easily. Students are identified with an arbitrary identifier unique to all NCSC students, and contain placeholder attributes for any LEA, SEA or national identifiers provided during student ingestion.

Registration: LEA Hierarchies
Three challenges exist that must be solved for NCSC to have an effective multi-state and multi-year solution:

1. The system must understand the concept of year-over-year equivalence within SEA hierarchies and maintain SEA instance equivalence information in its database.
2. The system must accommodate accumulation of students, registrations and results within regional (multi-state) and national groups.
3. The system must accumulate results over multiple years and report on aggregate performance of all of its SEA entities over those multiple years.

5.3.3 Assessment Delivery

Requiring tighter coupling than other components of the NCSC Assessment System, the test delivery component communicates using direct API calls when necessary for performance reasons. These calls are mirrored with a corresponding RESTful API. For the same reasons, considering a de-normalized or NoSQL data approach provides value. These components have the highest volume of traffic and require the fastest response time. In addition, these components have a custom development requirement, with many of the requirements being unique to NCSC.

This component greatly takes into consideration all of the unique needs of the students for the NCSC assessment system. Carefully designing the assessment delivery components to meet these varying needs proves most critical. The delivery system takes the numerous accommodations based on each student, delivers the assessment with the accommodations, captures all of the data and evidence for the students' responses and scores appropriately.

5.3.4 Assessment Reporting

The data warehouse utilizes a database architecture that facilitates high volumes of analytics across a variety of analytical dimensions. Additionally, the data may be stored in a de-normalized state. This means that traditional rules of third normal form may be broken such that the data is optimized for specific analytical objectives. The data warehouse needs to support SQL based query capability. Finally, most reporting systems and tools access this information with ease provided it is stored in cubes, typical in Online Analytical Processing (OLAP) data stores. This allows for complicated data mining and the support of Pivot Tables.

Two standards provide this functionality. The first is multi-dimensional expressions (MDX). This standard query language for OLAP systems is widely-used. Another standard, XML for Analysis (XMLA), is a SOAP service. Both of these standards may be applied.

5.4 Acceptance Criteria

Database repositories included in the NCSC Assessment System must meet specific criteria for inclusion. These criteria allow the NCSC organization to qualify vendor's database technologies and assure that they meet the technical considerations and the program's short- and long-term objectives:

- The database technology must have open access for third party software and NCSC-developed software. In order for a variety of clients to have data access, the database must implement standards such as SQL-92 and ODBC/JDBC.
- The database technology must deploy on the server hardware and software requirements listed in the technical architecture section. In addition, the technology must meet the requirements of the deployment and hosting section.
- The databases must not carry an initial or ongoing license cost to NCSC nor can the license cost vary by utilization or size of the database.
- The database must allow integration with at least some of the server enterprise management frameworks, such as permissioning, logging and monitoring.
- The database should accommodate contemporary scaling strategies, such as clustering, if the program requires it.

6. Component Transport

This chapter describes the interfaces and transport mechanisms used for component communication within the NCSC Assessment System.

6.1 Component Interfaces

The following diagram describes some of the significant flows of domain objects between component groups. The arrows represent connection points between components and labeled with an action.

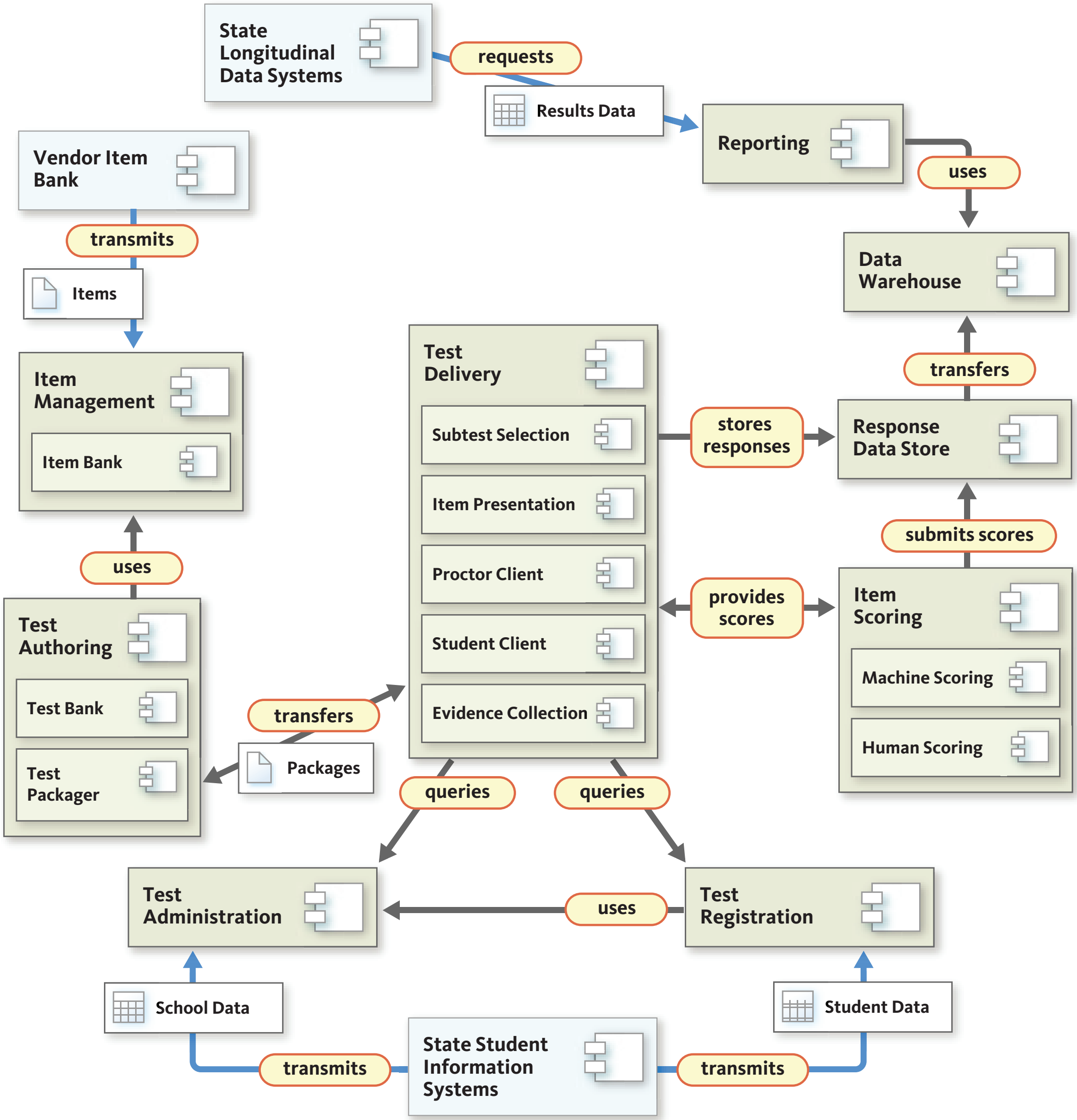


Figure 6.1: Component Transport

6.2 Component Transport Technologies

REST (http://en.wikipedia.org/wiki/Representational_state_transfer) is recommended for use where point-to-point communication is required between components either in a fire-and-forget mode or in a request-response mode. REST uses HTTP(S) for transport and message formats use XML, JSON and the standard HTTP methods. In addition, REST utilizes a uniform predictable interface for stateless communication. Since no client session information is managed on the server, each request contains all of the information needed to supply a response. This makes REST a very scalable, testable solution. Lightweight REST frameworks are readily available for both the .NET and JVM platforms.

The following diagram illustrates an example of using REST API to invoke a server's API methods:

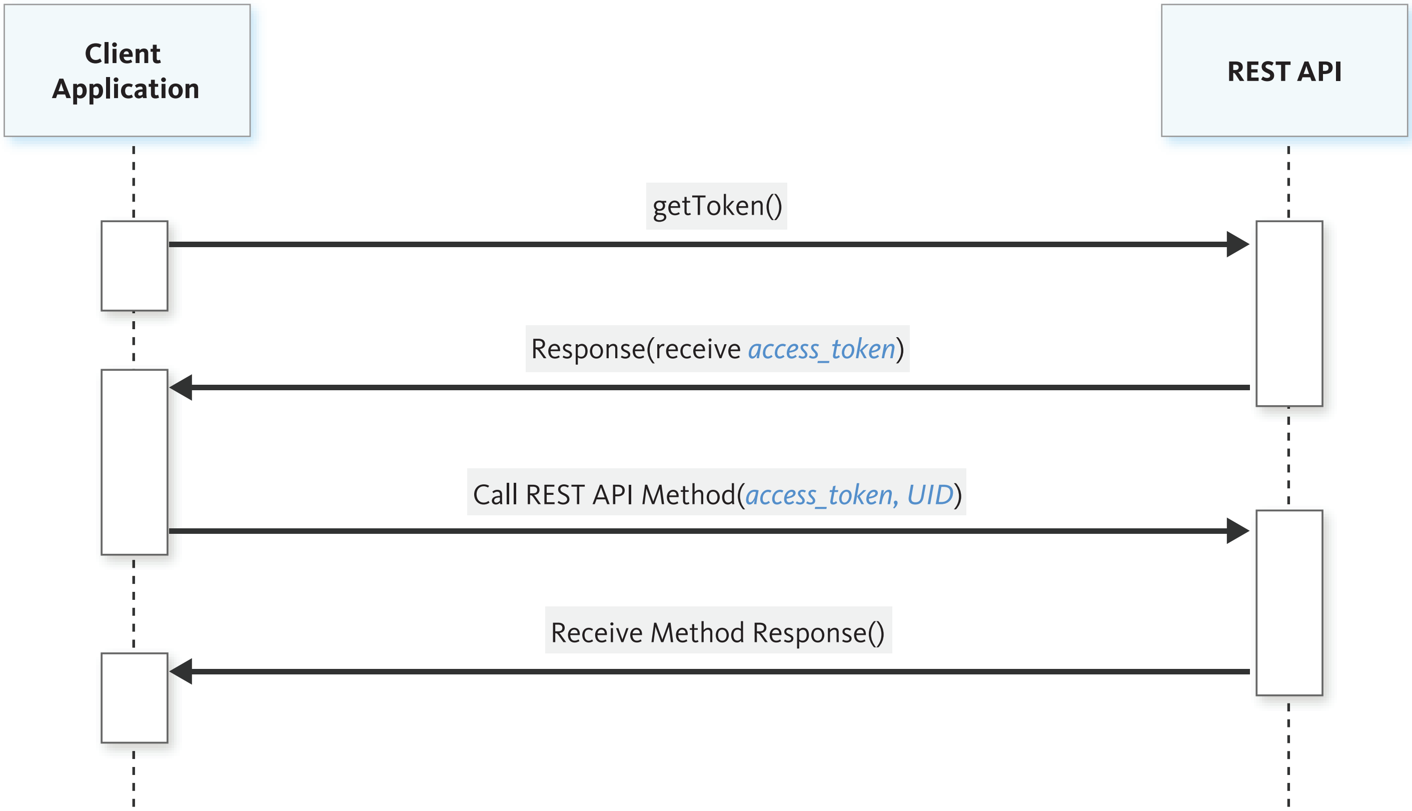

Client Application
REST API
getToken()
Response(receive *access_token*)
Call REST API Method(*access_token, UID*)
Receive Method Response()

Figure 6.2: RESTful Interactions

6.3 Security of Component Interface Communication

All components should utilize SSL for inter-component communication. The diagram below shows a typical pattern when a component communicates with another component. This required set of interactions ensures the component APIs are open and available and enforce the modular goals of the system, and also ensures that authentication, authorization and security requirements are met.

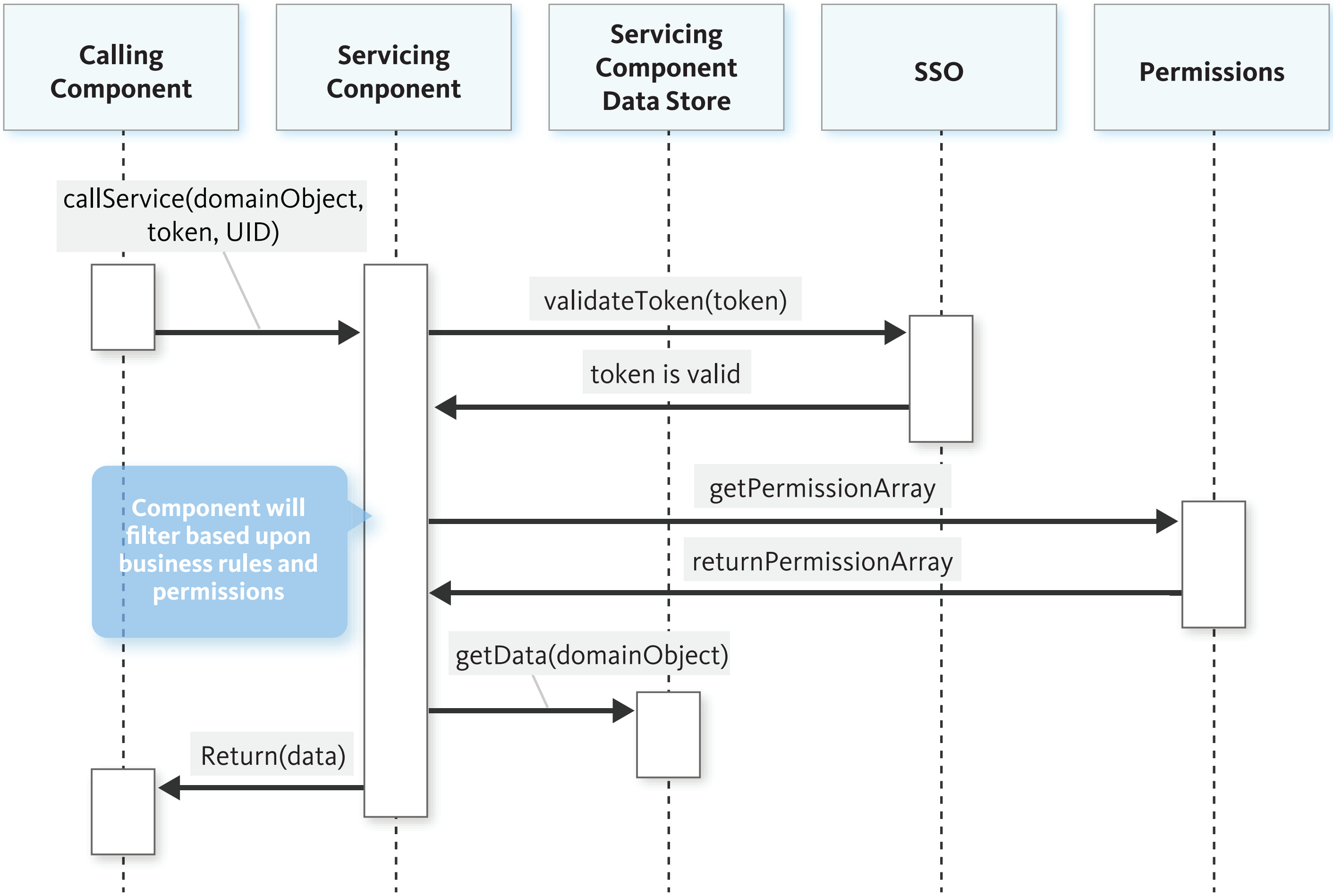


Figure 6.3: Component Authentication

In this pattern, the requesting component passes an identifying token along with the request. Single sign-on validates the token, and then the permissions management components retrieve the permissions for that user. Once the servicing component has the user permissions, the servicing component applies them to the request at hand.

7. Additional Non-Functional Requirements and Constraints

Non-functional requirements are non-behavioral qualities of a system. These exist in the form of execution qualities, such as security and usability, or evolution qualities, such as scalability and maintainability.

7.1 Accessibility

The assessment system components must support access for all students and staff. This section describes the specific compliance requirements of accessibility, how it can be applied to different logical components and some of the tools and standards that are utilized to meet these objectives.

7.1.1 General Web Application Accessibility (Section 508)

The W3C published guidelines for accessibility intended to interpret the Federal Section 508 guidelines for content systems and software applications available on the Internet. These guidelines, the Web Content Accessibility Guidelines (WCAG), provide the controlling document for contemporary web development. These guidelines are available at http://www.w3.org/TR/WCAG.

WCAG defines three levels of compliance - level A, level AA and level AAA. For the NCSC Assessment system all components that expose features to students are required at level AA compliant. All staff, instructor and parent-facing content must include level A compliance. Components for staff-only or components with trivial user interfaces have no compliance requirement unless specified.

7.1.2 Accommodations

The NCSC assessment environment demonstrates innovative accommodations in item presentation and response selection. In order to fully leverage the technological capabilities for accommodations, these are made available on the server-side of the test delivery, item presentation components and on the client-side within a test delivery workstation. Workstation-deployed accommodations define as local accommodations and server-deployed accommodations as hosted accommodations.

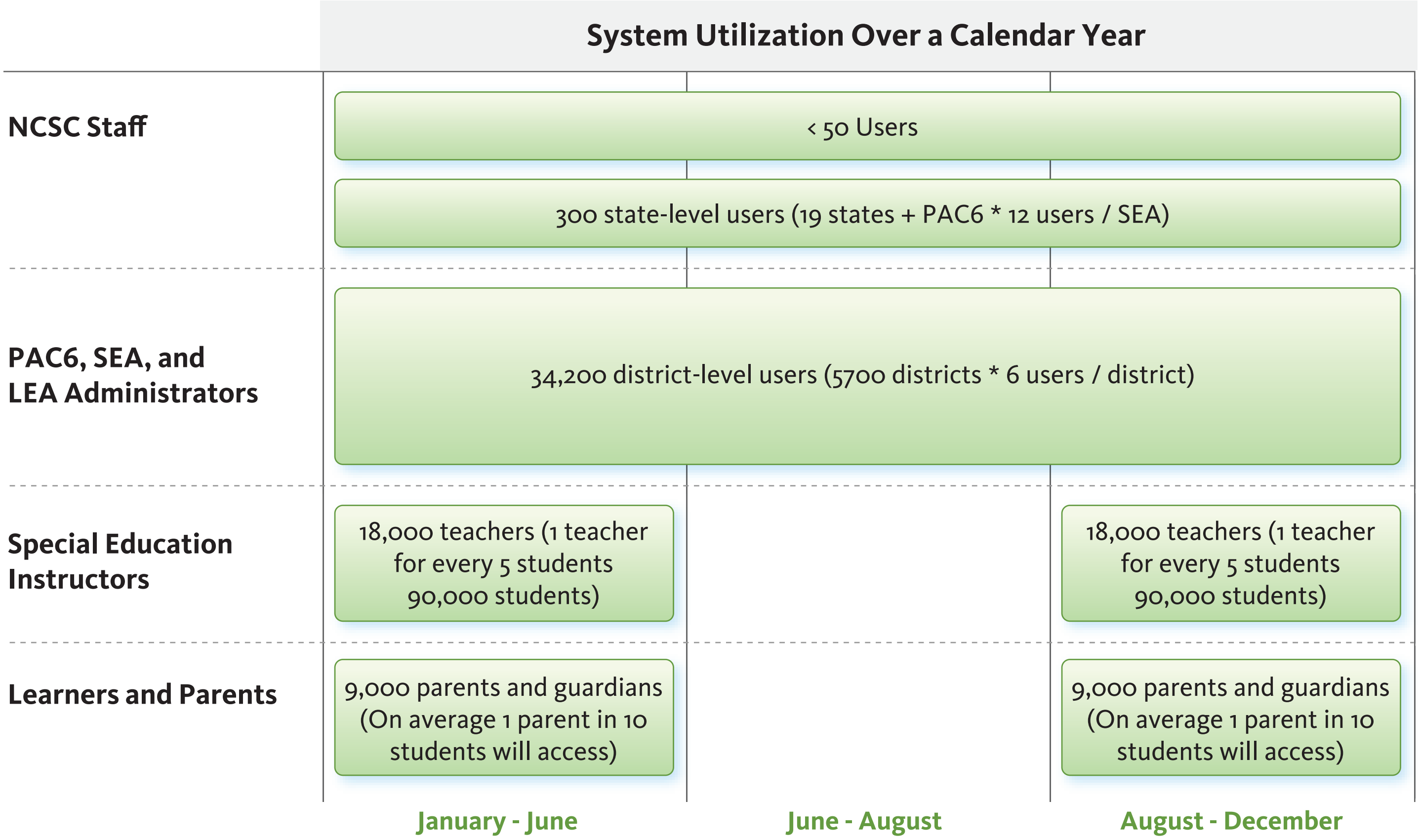
Hosted accommodations comprise functional requirements of the test delivery and item presentation components. These incorporate the accommodations that are developed to and available via the APIP interoperability standard. NCSC will work with the test delivery vendor to specify which accommodation's availability is provided within this system to be acceptable for use. NCSC will also work with the test delivery vendor to establish a linkage between elements of the personal needs profile (PNP) and the enabling or disabling of hosted accommodations.
Local accommodations, typically installed on the test delivery workstation operating system and work to facilitate specific test taker interactions, make the assessment more accessible given an individual's specific areas of disability. NCSC should specify governance procedures around the deployment of local accommodations.

7.2 Availability

The availability of a system takes into account its capability to reliably be online and accessible whenever users need it. Outlined below describes the availability by reviewing some of the usage models predicted for the system, basic requirements for backup, recovery and disaster recovery and finally a discussion on how the system scales as the number of users increases.

7.2.1 Usage / Utilization Model – Yearlong

Through conversations with NCSC, different groups utilize the assessment system at various times throughout the year. Based on these requirements, the table below shows potential utilization for different groups throughout the year:



System Utilization Over a Calendar Year

	January - June	June - August	August - December
NCSC Staff	< 50 Users 300 state-level users (19 states + PAC6 * 12 users / SEA)		
PAC6, SEA, and LEA Administrators	34,200 district-level users (5700 districts * 6 users / district)		
Special Education Instructors	18,000 teachers (1 teacher for every 5 students 90,000 students)		18,000 teachers (1 teacher for every 5 students 90,000 students)
Learners and Parents	9,000 parents and guardians (On average 1 parent in 10 students will access)		9,000 parents and guardians (On average 1 parent in 10 students will access)

Figure 7.1 Yearly Usage Model

This summary information provides NCSC and supporting developers with details in which to determine the general availability for the overall system. Such data informs SLAs for recoverability, concurrent user requirements and bandwidth requirements.

7.2.2 Usage / Utilization Model – Summative Testing Period

The summative testing period sees the majority of use, and similar to having an understanding of yearlong utilization data, comprehending these data points further informs requirements. The table below shows potential utilization for different groups during different phases of the summative test lifecycle:

<table>
<tr><th></th><th colspan="3">System Utilization Over a Summative Testing Period</th></tr>
<tr><td><b>NCSC Staff</b></td><td colspan="3">&lt; 50 Users
300 state-level users (19 states + PAC6 * 12 users / SEA)</td></tr>
<tr><td><b>PAC6, SEA, and LEA Administrators</b></td><td>34,200 district-level users (5700 districts * 6 users / district)</td><td></td><td>34,200 district-level users (5700 districts * 6 users / district)</td></tr>
<tr><td><b>Special Education Instructors</b></td><td colspan="2">18,000 teachers
(1 teacher for every 5 students 90,000 students)</td><td></td></tr>
<tr><td><b>Learners and Parents</b></td><td></td><td></td><td>18,000 teachers (1 teacher for every 5 students 90,000 students)</td></tr>
<tr><td></td><td>Pre-Test</td><td>Testing</td><td>Scoring and Post-Test</td></tr>
</table>

Figure 7.2: Summative Assessment Usage Model

The diagrams above intend to provide a starting point to develop utilization models of system use. Assumptions used in the creation of these tables include:

- Approximately 19 states + PAC6, which roughly covers 5700 school districts
- NCSC staff has less than 50 staff
- 90,000 students receive the assessment
- One assessment given per student at each grade level
- Subtests per an assessment are unknown, but could be between one and five subtests per assessment
- Assessing instructors administer an assessment on a one-on-one basis
- One human scorer for every five students

These models create an approximation for the launch volumes of the testing and non-testing activities on the platform given an immediate rollout in 2014 across all member states and pilot programs in affiliate states.
As decisions about subject areas and grade levels administered undergo further refinement, these models should reflect these decisions in subsequent architectural activities.

7.2.3 Recoverability (Backup and Restore)

Offering components on demand demonstrates an important concept for the NCSC architecture, referred to as high availability. The measure of high availability is calculated by how the user of a system component perceives the availability of the data needed. High availability proves crucial in relationship to recoverability, a main characteristic of a highly available system. Recoverability sets the requirements for the types of failures that may occur and how to recover from them. Service Level Agreements (SLAs) define the monthly downtime allowance.

In order to meet the uptime requirements demarcated in the SLAs, the NCSC architecture must closely monitor exceptions and performance deviations. Fast and efficient error detection and subsequent escalation assists in determining when a component fails so that an appropriate response action may be invoked.

The NCSC architecture must consider data that are backed up and the recovery time to restore these data.
The following requirements indicate the basic recoverability necessary of the NCSC Assessment Program:

1. When the system is being recovered from a catastrophic failure, no losses may occur, including:
   - Assessment results
   - Results previously stored or interpreted
   - Results archived in the warehouse
2. In-process assessments may be terminated early but assessment responses already collected must not be lost.
3. Interim processes such as human scoring, administration, item and test editing sessions may be lost and recovery necessary, but the loss of actions shall not exceed one hour.
4. In the event of a catastrophic recovery, the recovery must take place by the next business day.
5. The system and the vendors maintaining the system must have automated detection of catastrophic errors.

7.2.4 Scalability and Performance

NCSC maintains the hardware architecture of on-boarded states. As more states utilize the NCSC Assessment System, the NCSC architecture should scale horizontally.

Specific performance requirements include that:

1. Each test delivery session must have a deterministic latency between item navigation operations. These may not vary more than 1-2 seconds across the wide variety of client test delivery environments.
2. Test session data should record item navigation latency within it.
3. When navigating web applications, like administration and registration, time to load pages should be under one second on average and under five seconds for the worst case.
4. Report generation may vary by the complexity of the report and an option should exist to schedule and deliver reports.
5. Launch requirements should consider the member states, as well as the affiliate states, that pilot the alternate assessment.
6. A significant amount of student data is collected (for example, item, answer, score, comments, etc.) with test delivery. The architecture must consider data volumes and purging strategies.
7. Network bandwidth and reliability must be taken into account and the architecture must make appropriate recommendations for critical components.

7.3 Extensibility

As part of the interoperability plan, components of the NCSC Architecture provide mechanisms to extend the system to accommodate future needs. The needs required typically addressed in an area of the architecture called extensibility. Extensibility of the system can accommodate future requirements in a logical and cost-effective way. Examples for future considerations include the following components:

- Professional Development and Instructional Learning Materials – provides support for types of media that can be used.
- Item Authoring – offers flexibility in handling new item types and future technologies that can assist the assessment process.
- Reporting Platform – devotes a future need to support advanced custom reporting.
- Test Delivery – yields innovative accommodations for students.

7.4 Open Source Approach

This section of the architecture contains a discussion on some of the areas for consideration as NCSC organizes any open source aspects of their development project. This requires some additional governance, organization and tools to effectively complete.

7.4.1 Open Licensing Requirements

The following design requirements elaborate on the design goal and principle of utilizing open source as a guiding principle within the project:

1. Publish all artifacts describing the architecture under an Open Access License.

2. Distribute all software artifacts produced under an open source software (OSS) license.

3. Consider OSS components for use in building the software systems. This includes, but is not limited to:
   - Operating systems
   - Tools for authoring, building and testing the software components
   - Database software
   - Messaging systems

7.4.2 Recommended Open Licensing Model

The licensing of all supporting materials created for and by NCSC that is not source code shall be licensed under an open access license. The most commonly used one is the Creative Commons license (http://creativecommons.org). Since NCSC does not wish anyone to modify these works, and it is acceptable to utilize these assets in derived commercial systems, this architecture recommends the Attribution-NoDerivs 3.0 Unported license.

If NCSC wishes to change this license approach, there is more information and a calculator for determining the appropriate license variation at http://creativecommons.org/choose.

7.4.3 Open Source Definition

The following text is generally accepted as the standard definition for open source. The Open Source Initiative (http://opensource.org)[1] publishes and maintains it. This architecture accepts this definition and recommends that it is used for the duration of the project as the definition of open source, as it applies to the source code created for NCSC by the organization, its contractors and vendors.

Introduction
Open source doesn't just mean access to the source code.
The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution
The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

2. Source Code
The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

3. Derived Works
The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

4. Integrity of The Author's Source Code
The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

5. No Discrimination Against Persons or Groups
The license must not discriminate against any person or group of persons.

6. No Discrimination Against Fields of Endeavor
The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

7. Distribution of License
The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product
The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

9. License Must Not Restrict Other Software
The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

10. License Must Be Technology-Neutral
No provision of the license may be predicated on any individual technology or style of interface.

[1] Accessed September 20, 2012 (http://opensource.org/docs/osd).

7.4.4 Open Source Software Licensing

The dynamic area of open source licensing constantly reveals new innovations as open source licenses are tested in the business world and legal system. Also, new component categories and deployment and integration techniques stress the existing license inventory and revise constantly. In 2012, there were 69 open source licenses recognized by the Open Source Initiative.

Several licenses have the required features and stability to be reasonable choices for the NCSC program. Apache Version 2 is the recommended open source license. This license provides one of the most liberal offerings, and has already been accepted by other consortia as the open source license of choice. More information and the specific license may be found at http://www.apache.org/licenses/LICENSE-2.0.html.

7.4.5 Open Licensing Requirements

- Release all artifacts describing the architecture and technical documents published publically under an open access license.
- Produce software artifacts under an open-source software license. If a vendor sells a proprietary solution, that solution must be made available under an OSS license.
- Issue a waiver for a component if some components may not be published under an OSS license. Cite the specific rationale and balance the trade-offs to the program's strategic goals.
- Use OSS components, where available, for building the software system. This includes, but is not limited to: operating systems, tools for authoring, building and testing the software components, database software and messaging systems.

7.4.6 Segmentation of Assets

Design the source code should so it can be deployed for another state or consortia's testing program. Segment the assets specific to NCSC so that these are separated from the open source repository in a separate, private source code repository. Examples of this include program logos, testing data files with NCSC-specific data and any files or configuration settings (XML, etc.) that might specific connectivity or security settings that NCSC would not want to expose to the general public.

The NCSC Architecture Review Board will perform periodic audits of the source code repository to ensure that none of these constraints have been violated.

7.4.7 Ownership and Community

Offering a project to a community as an open source asset, as this is the proper and legal selection and specification of an open source license. The overarching vision for the development and maintenance of the NCSC open source repositories is that NCSC will initially utilize its 2012-2014 funding to jump-start the program by selecting vendors who build and publish their components as open source under the license that NCSC has selected.

During that time period, emphasis focuses on delivery and integration in preparation for 2014 testing. This phase also includes a thread for planning the sustainability of these assets. In parallel, NCSC will begin interacting with a community of users and contributors that will also provide changes to the source code. NCSC will utilize its access to funding and organizations that seek to leverage the NCSC investments to hold user groups, sponsor code-a-thons and generally maintain interest in the platform and also continue to provide governance to the source code assets.

NCSC should also look for opportunities to transition stewardship of certain components to other community members if the opportunity arises. Having a publicly-funded organization drive an open-source community is an emerging phenomenon, and some learning will be expected along the way.

8. Security

The NCSC Assessment System meets the highest level of current industry standards for information security. This section describes the relevant aspects of data security including its principles, component-to-component concerns, user authentication and authorization and sensitive data (item level security, security of results, and student data security). Additionally, this section explains an acceptable approach for components to assure that security, authorization and user sign on components can all blend to form a secure system with acceptable user interactions.

8.1 Security Principles

The NCSC Assessment System architecture expresses three key principles of information security.

1. Confidentiality
Information contains elements of data that are private and/or restricted from specific or public eyes – thus the term confidential applies. These data are carefully safeguarded so that no unauthorized access occurs, whether by accident or mischievous/malicious intent. When an unauthorized person accesses confidential data, a breach in security has occurred. The architecture must be designed to provide the best security possible and to minimize security breaches.

2. Integrity
Integrity means that data cannot be modified in an unauthorized or undetected manner. For example, an email message must not change between the pressing of the "send" button and its arrival at the intended recipient's inbox. The architecture is designed in such a way that prevents accidental changes to data in storage or in transmission.

3. Availability
A system can only function when the information it requires resides in the system. Design the architecture in such away that the storage and processing of information, the security around it, the communication between components and transmission methods of data all function correctly. This principle also refers to the prevention of service disruptions. (e.g. system maintenance)

8.2 Security by Design

One of the best ways to ensure the development and efficiency of system security involves designing it well from the beginning. Determining the required components, the component's requirements (such as data inputs and outputs) and how these interconnect with other components in the system and out of the system, presents a challenging up front task. However, the benefit of having this roadmap from the start eliminates many problems down the road. Defining the roles, functions, interactions and permissions of users is part of the design process as well. The overall system integrity and security remains an important part of the design and development processes.

Experience tells us that solid architecture is built using industry-standard technologies and by following best practices in security. Build the NCSC Assessment System based on these principles and for high availability and fault tolerance. Redundant hardware at all layers within the system ensures no single point of failure; a server could fail but the system would continue operating with no perceptible change to the end user. System monitoring allows for immediate alerts of hardware or software failures, permitting rapid resolution and restoration of the failed components.

8.3 Security Policy

A solid security policy, accepted by all parties involved, is essential to the optimum performance of the NCSC Assessment System. In general, a security policy governs the expected behavior of its members who interact with a system or other entity. A software security policy also addresses the privileges or restrictions that its users will encounter when interacting with various applications, components or modules. Finally, it governs how data are securely and safely stored, transmitted or used (for example, network and data security, virus protection and security logs).

The policy would contain, but not be limited to, the following:

- Access control – deciding who can do what when
- Computer security – securing workstations, user accounts
- Information protection policy – defining secure materials and the established school, district and state security policies, along with federal and state regulations
- Network Security – determining what devices connect to a network or system and access to data within it
- Physical security – establishing hardware and printed materials (tests, identifying student information) guidelines
- Remote access policy – agreeing whether or not a user can access the system when outside the network (e.g. working from home)

To ensure security by design, include the requirement that all account management and user authorization (permission) data are stored in a centralized repository. This eliminates the need for redundant editing, which can cause gaps or duplicate specification that may lead to errors in the system.

8.4 Component to Component Security

Each component within the system needs the ability to communicate consistently. Accordingly, configure each component to allow only authorized components to use some or all of its services. The communications channel between components may be secure and prevent unauthorized listening, component access or forged entry by a mischievous or malicious third party. Some items to consider include:

- IP Filtering: The ability to grant specific systems access to a system. Note: The larger the system (for example, four states vs. all 50 states), the more complex this becomes.
- Secure Sockets Layer (SSL): A mandatory level of security for authentication and items.
- Pretty Good Privacy (PGP): A data encryption and decryption protocol providing privacy and security for the data communication of files. Users authentication must occur before granted access to encrypted files.

8.5 User authentication and Access Control

User authentication and authorization is a key component to information security. This section describes the recommended approach.

8.5.1 Authentication, Authorization and Sign On

Authentication and authorization are two key concepts for granting users access to a system.

- Authentication involves the manner in which a system securely identifies the user. For example, whenever you sign into your email, you authenticate who you are to the system with your user credentials.
- Authorization is the manner in which a system determines what level of access a user has to a system and what they can do; it is concerned with their role in the system. For example, as an administrator, or master role, of the NCSC Assessment System, you are authorized to perform all functions. As a proctor, you possess permissions to view the materials, administer the assessment to the student, enter in scores for the student (if part of the accommodations) and upload evidence. The proctor could not access student level data for other students, unlike the master role.

The NCSC Assessment System must authenticate users and accommodate multiple roles (i.e. student, teacher, building administrator, district administrator, state administrator, etc.). Each authorized role uses certain features and functionalities of the system components. The user authentications and access control must be standardized across components to allow seamless access to all of the authorized systems.

Authenticate users with single sign on (SSO). SSO uses a protocol such as SAML, OAuth or OpenAM. This allows the user to sign on once with their credentials and access multiple systems without having to re-enter the credentials. This approach becomes more complicated when multiple systems and vendors are part of the entire suite.

Single sign on for all platform components creates boundaries and opportunities for platform security and confidentiality. Architectural considerations that promote modularity also encourage a variety of best practices in terms of data integrity and availability, and of a flexible role and permission scheme.

8.5.2 Roles and Permissions Authorization

The NCSC System could accommodate one of two methods of authorization:

1. Role Determined Authorization
In this method, a predefined role (i.e. student, teacher, building administrator, district administrator, state administrator, etc.) grants each role a certain level of access or functionality. Users are assigned one or more roles by a system administrator or other manner. Once the user has logged in, the system's components allow the user with a specific role(s) to access functions granted for that role. This requires a defined list of permissions or functions that each role within a component can do. Complex code changes may be required when new roles are identified after implementation.

2. Permissions Determined Authorization
In this method, the component's functionality or permissions are defined and associated with groups. Users are then placed into these groups, allowing an unlimited number of group creations without complex code changes. This method may require each component to poll an external system for user permission verification.

The desire to enable access only to certain data sets an important piece of the authorization puzzle. For example, a district administrator should only have access to data or results in his or her own district. Thus, the system must allow for the assignment of user permissions such as read, write or no permissions for any role in the system. Communicate these hierarchy assignments to components where the user has permissions and utilize them to constrain operations on certain subsets of data.

8.6 Sensitive Data

The following section addresses specific data security considerations for NCSC.

8.6.1 Item-level security

The validity of the results relies on the security of summative items. Test takers must not see the summative items in advance of the assessment's administration. Strict access control to secure test content ensures test security. For example, security can be managed by the use of confidentiality and nondisclosure agreements, the prohibition of camera-enabled cellular telephones in rooms with secure materials present and the daily collection of all secure materials before students and proctors leave.

Because item exposure remains critical to the validity of a summative assessment, item security must consider the following:

- Item storage and who has access to the items.
- Item transmission to other systems, including authentication and authorization and permissions.

8.6.2 Student data security

State and federal regulations such as the Family Education Rights and Privacy Act (FERPA) requires states and LEAs to protect student data. Each state must apply their policies around privacy as seen fit and only de-identified data may be used in aggregate by NCSC. Given this, real student identification data (name, grade, school, etc.) is replaced with false data or with an identification number so that no one can identify a student from the data. Similar to a security policy, publish a data use policy and make it available for review by any state participating in the NCSC Assessment Program.

Student data security must comply with:

- **Family Educational Rights Privacy Act (FERPA)** – a federal law that protects the privacy of student education records. The law applies to all schools that receive funds under an applicable program of the U.S. Department of Education. For more information, visit http://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html.
- **Children's Online Privacy Protection Act** (COPPA) – a federal law that places parents in control over what information is collected online about or from their children under the age of 13. For more information, visit http://www.ftc.gov/privacy/coppafaqs.shtm.
- State laws, including data breaches. For example, California has a law called the "California Data Breach Notification Law" (SB 24 – Sep 2012), which requires companies, institutions and government agencies to provide key details in data breach notification letters and to notify the state attorney general about the data breach.

8.6.3 Encryption of Sensitive Data

Data that gets stored (for example, student identifiable data, password field in the database, export file, SSN in an XML file, etc.), or data at rest, in a database or repository that NCSC either has direct control over or indirect control through a contractor or vendor relationship must be encrypted. Similarly, transmit data securely. Encryption of data also must occur to prevent exposure to a mischievous or malicious third party that has broken the network and server security measures of the file system.

9. Technical Architecture Definition

Technical architecture describes the different components, platforms, deployment and hosting models, deployment environments required for the NCSC system, definition of data management and transport mechanisms and monitoring.

9.1 Workstation Requirements

Test Delivery Workstation Hardware and Software
The test delivery systems should align with the State Educational Technology Directors Association (SETDA)[2] recommendations. The following table represents the minimum hardware and software requirements of the test delivery workstation:

Operating System	
Desktop Windows Vista, Windows 7 or better 32-bit or 64-bit OS X 10.6+ Linux: Ubuntu 10.04+; Debian 6+; OpenSuSE 11.3+, Fedora Linux 14+	**Tablet** iOS 5 (iPad2 or better) Android 2.3+

Browser	

Desktop
IE 9 or better;
Firefox 4 or later;
Google Chrome;
Opera 11+;
Safari 5+
	Tablet
	Safari Mobile iOS 5+
	Android 2.3+

Evidence Collection		
Desktop Webcam (320 x 240 or better) Flash 11	**Desktop** 1024 x 768 or better	**Tablet** iPad 1024 x 768 or better Android 320 x 480

Figure 9.1: Test Delivery Workstation Requirements

[2] Accessed October 15, 2012 [http://assess4ed.net/sites/default/files/techrequirements_june22_combined_o.pdf]. Levin, D., Fletcher, G. & Chau,Y. (2011). Technology Requirements for Large-Scale Computer-Based and Online Assessment: Current Status and Issues. Washington, DC: State Educational Technology Directors Association (SETDA).

Proctor Workstation Hardware and Software

The proctor workstation may require a separate deployment and therefore may have different hardware and software requirements. The following table represents the minimum supported hardware and software of the proctor workstation:

Operating System	
Desktop Windows Vista, Windows 7 or better 32-bit or 64-bit OS X 10.6+ Linux: Ubuntu 10.04+; Debian 6+; OpenSuSE 11.3+, Fedora Linux 14+	**Tablet** iOS 5 (iPad2 or better) Android 2.3+

Browser	

Desktop
IE 9 or better;
Firefox 4 or later;
Google Chrome;
Opera 11+;
Safari 5+
	Tablet
	Safari Mobile iOS 5+
	Android 2.3+

Evidence Collection			
Desktop Webcam (320 x 240 or better)	**Tablet** Integrated Camera (2MP or better)	**Desktop** 1024 x 768 or better	**Tablet** iPad 1024 x 768 or better Android 320 x 480

Figure 9.2: Proctor Workstation Requirements

Bandwidth

Bandwidth becomes an important consideration as new technology solutions deploy to an existing network infrastructure like the U.S. public school system. The test delivery system assesses and adapts to the network bandwidth available during the test delivery session. The test delivery system must elegantly handle different bandwidth availability and if the bandwidth is compromised mid-session, it must not interrupt the testing session. SETDA[3] provides some overall minimum recommendations as a baseline:

Broadband Access for Teaching, Learning and School Operations	2014 - 15 School Year Target	2017 - 18 School Year Target
An external Internet connection to the Internet Service Provider (ISP)	At least 100 Mbps per 1,000 students/ staff	At least 1 Gbps per 1,000 students/staff
Internal wide area network (WAN) connections from the district to each school and among schools within the district	At least 1 Gbps per 1,000 students/staff	At least 1 Gbps per 1,000 students/staff

9.2 Establishing Language and Platform Guidelines

Many programming languages and platforms are used today. The two most popular platforms include Oracle Java Platform (JVM) and Microsoft's .NET Platform (CLR). While C and C++ are still frequently used, the JVM and CLR platforms support for multiple programming languages and hosting multiple target platforms make these the logical choices for this system. These allow programming an application in multiple languages while running on the same platform, enabling developers to use the most efficient language when solving specific issues. For example, Scala [http:// www.scala-lang.org] could be used for parts of a component that require concurrent processing. However, Clojure [http://clojure.org] would serve well for concurrency, and also works properly for functional style programming, supporting more mathematical features.

.NET, often considered only available for Microsoft platforms, has an open source project Mono [http://www.mono-project.com/Main_Page] built on it. .NET technologies run on Linux and OSX operating systems. Commercial versions of Mono are also available for Apple IOS and Google Android, both platforms suitable for components of the NCSC system. Some components can be built in one platform, and others built in the other.

Mono does extend the platforms on which the application may be deployed. However, it does not support all features of the .NET 4.0 platform, may introduce increased support costs and lags behind .NET updates and new features. Xamarin, a company founded in May of 2011 by some of the originating Mono developers, now supports Mono.

JVM was started by Sun in the mid 1990s and acquired by Oracle in April of 2009 with the purchase of Sun. It is supported on most platforms, except IOS, and has been a dominant enterprise platform since the beginning of the millennium. JVM has wide support in cloud technologies. .NET is still competitive, although it is not currently as broadly supported.

[3] Accessed October 15, 2012 [http://www.setda.org/c/document_library/get_file?folderId=353&name=DLFE-1517.pdf]. Fox, C., Waters, J., Fletcher, G., & Levin, D. (2012). The Broadband Imperative: Recommendations to Address K-12 Education Infrastructure Needs. Washington, DC: State Educational Technology Directors Association (SETDA).

The NCSC Program requires that all programming languages deployed meet certain critical criteria in order to be acceptable within the overall solution and technical architecture. The following criteria have been established to assist NCSC stakeholders in the evaluation of these languages:

- Maturity: The language has been around long enough to be a stable and a full-featured language.
- Adoption: The language is widely adopted.
- Community: An active development community exists.
- Portability: The language runs on a wide variety of operating systems.
- Scalability: The language allows for horizontal scaling and handles large amounts of traffic.
- Maintainability: Enough developers in the market to support future development exist.
- Interoperability: The language supports interoperability requirements established.
- Consistency: The language fits in well with the other languages used in the system.

9.3 Deployment and Hosting

In order to define how and where the components are deployed, the constraints of NCSC need classification. These define the hierarchy from the school up to the consortia level, the data integration points between them and potential deployment models.

Constraints

1. NCSC procures a centrally hosted solution and works with the vendor to meet the requirements of the assessment architecture.
2. The NCSC assessment system maintains compatibility with the varying levels of technology within the NCSC states, districts and local schools.
3. To improve scalability, the NCSC system relies upon virtualization and contemporary hosting strategies, such as private-clouds as necessary.

Deployment Hierarchy and Local Reporting

All solution components are deployed at the consortium level. Additionally, the reporting components required for the NCSC assessment system may live at different physical locations. Possibilities include:

- Other Consortia
- Region
- State

A state may choose to utilize all non-reporting components at the consortium level, while still leveraging its own local reporting solution. Should a state maintain its own reporting solution, those reporting platforms must be compatible with the consortium level components through a common technical standard.

Deployment Models

Deployment models considered must handle the varying technical capabilities of the LEAs and state-specific components. Example deployment models considered include virtualization, cloud deployment and traditional dedicated physical server implementations.

Virtualization

Virtualization is the concept of partitioning hardware to create virtual entities. These entities consist of a server, storage unit, an operating system, etc. The goal is to provide maximum utilization of actual physical hardware while providing a central point of administration and increased scalability. With virtualization, sync virtual components on multiple physical hardware spaces thus enhancing the overall redundancy of the NCSC assessment system. Virtualization stands as an essential component of a cloud deployment strategy.

System VM [http://en.wikipedia.org/wiki/Virtual_machine] / Virtual Private Server (VPS) [http:// en.wikipedia.org/wiki/Virtual_private_server] based hosting allows the NCSC assessment system to bypass the usual hardware procurement needs. By VM hosting through a hosting provider, the NCSC assessment system can procure the required server and data storage in a just-in-time manner. VM providers usually have a higher speed network backbone to support inter-server communications. These providers can support the hardware management and network support. This frees up the NCSC assessment system to solely provide the software support.

Cloud Deployment

Deploying components in the cloud may yield many advantages while providing support of the NCSC assessment requirements. These benefits include:

1. **Low cost initial investment** – Because of the dynamic nature of cloud computing, there is no need to purchase the real estate, hardware and staff required to run a large-scale system.

2. **Self-provisioning** – A cloud infrastructure provides elasticity, which means that the computing resources between the components at all levels scale to demand and increase overall utilization while keeping costs low.

3. **On-demand cost structure** – A cloud deployment keeps support costs down by billing NCSC only for the infrastructure used. During non-test times when components may be less busy, then there is less of a cost to maintain.

4. **Reduced time to market** – On-demand parallelization means that the cloud architecture can utilize more than one machine for resource intensive tasks in a cost effective manner.

5. **Redundancy and failover mechanisms** – By maintaining Disaster Recovery (DR) servers in the cloud, an environment can be replicated within minutes should some type of failure occur.

Given the concerns over security, it is not recommended that NCSC utilize what may be called the public cloud as a deployment target. Public clouds provide environments to customers through services like Amazon Web Services, Rackspace or Heroku where application deployment environments are bought and made available on a month-to-month basis. These environments are easy to build and run but often lack strict compliance with security requirements. No guarantees exist that the physical hardware the systems run on will not co-mingle with other physical systems that have very different security considerations.

Private clouds include environments sequestered by hosting companies where the same tools for cloud server management and some of the same opportunities for content delivery and the like are available, but the cloud services stipulate dedication to a specific customer. These private clouds are utilized for enterprises with higher concerns for security and physical management and still wish to employ cloud tools and management for scalability and reliability goals. Private clouds operate within the data center or hosted and managed by the cloud provider. As an example, Rackspace offers a private cloud hosting option and hosts these servers in its data center environment.

If a cloud deployment model is considered, this architecture recommends using a private cloud or hybrid solution to realize the benefits of cloud deployment without compromising security considerations.

Traditional Dedicated Physical Server Deployment
Another option for deployment offers a dedicated physical server model. The designed solutions ought to include the same reliability, scalability and cost-effectiveness as a cloud deployment. Traditionally, a dedicated physical implementation contains the following characteristics:

1. Fixed and known pricing for initial and long-term costs of ownership
2. Security capabilities that are managed by a trusted vendor, auditable and accessible by the purchasing organization
3. An established performance profile that is deterministic and testable
4. Compatibility with the structure of intact systems

Other Requirements
Additional specific requirements for the deployment solution involve:

1. Providing support for the NCSC solution deployed to a cloud provider or traditional deployment models
2. Assuring the consuming services requirements for security and privacy are met by the consuming system and the transport method when employing distributed components, such as reporting
3. Supporting a dashboard showing application, component and service availabilities
4. Offering flexibility on underlying platforms over tight coupling to the operating system or hardware

9.4 Development Operations

In order to have the highest quality and most reliable deployments, the NCSC solution should implement a DevOps solution to facilitate deployment activities. DevOps from development and operations (http://en.wikipedia.org/wiki/DevOps). This involves the intersection of three disciplines: development, quality assurance (QA) and operations. Each organization is responsible for different disciplines: the QA organization for quality, the development organization for functional changes and architecture and the operations group for deployment of the solution within the appropriate environments.

DevOps as a practice recognizes that quality software does not happen unless all three of these areas are communicating and coordinating their activities.

This architecture recommends the following:

1. A set of processes that create a DevOps awareness and culture
2. A focus on automated, repeatable processes for deployment
3. A mechanism to document shared responsibilities and tasks
4. A mechanism to coordinate operations within deployment cycles

Logical Environments

Software development uses many environments. These environments, managed by NCSC in conjunction with the selected component vendors, provide the for the best solution and ensures the developed software stands up to rigor.

Environment	Description
Production	The environment where live data resides. All systems that reach students, teachers and administrators are housed within this environment. Against this environment, assess for uptime, reliability and performance requirements. Security tests are performed against this environment as it can house actual student demographics and high-stakes summative test content.
Staging	Staging provides an exact replica of production for the server configuration. This utilization ensures no issues exist upon deployment. Since production data constantly changes, a recent snapshot of production data captured assures no issues arise when new functionality is deployed on top of production data.
Integration Testing	As the integration testing environment does not have the performance requirements of production, an abbreviated version is permitted. These data typically contain a variety of test data provided by the QA organization. These data aid in perform integration and regression testing to ensure that new component changes do not break the system as they interact with other components.
QA	This environment also provides for an abbreviation of production. To ensure functional changes fulfill the requirements of the release, these components are assessed.
Development	These environments may be shared or individualized. Oftentimes developers only require a subset of the whole environment to work on their component. As these environments tend to become chaotic and ad-hoc if each developer defines the development tool chain and the environments upon which there is building and testing, management proves necessary.
DevOps	The DevOps environment allows for an integrated set of features and data that facilitate the DevOps procedures and automation techniques designed.

9.5 Database, Data Storage and Archiving Requirements and Approach

Each component can have different storage and archiving requirements. The application architecture of those components needs to define those requirements. The following list of principles should be observed:

- Data storage needs point-in-time recoverability. Time resolution depends on the criticality of the respective data object.
- Student assessment responses must never be lost. If a student or proctor submits an answer to an item and is presented with another item or test section completion page, the system must recover all responses including that response.
- Base item and test authoring requirements on NCSC policy. NCSC needs to define what an acceptable loss in case of system failure (i.e. 1 day, 1 hour, 15 minutes, etc.) includes. The shorter time recovery point, the higher the development and support costs to implement.
- Keep student responses and other data warehouse data for longitudinal use. NCSC needs to define the retention lengths for this data.

- Item metadata guarantees delivery from the delivery and data warehouse components to the item bank (at least to the once guarantee as opposed to the once and only once guarantee). The item bank accepts “re-sent” metadata and gracefully handle redundant data.
- Components seamlessly recover from single data node failures. When this occurs, components switch to other data nodes. Data storage for a component needs to have a minimum of two nodes to support single node failure.
- Provide explicit NCSC policies about archiving lengths for specific data objects. State and federal law may drive some of these policies.

9.6 Data Management

The NCSC architecture contains a segregation of where data is modified. The component that owns the data only modifies the data. For example: The item bank owns all Item related data and no other component can update this data. Other components consume parts of this data, but never update it. A simple custom solution should be used instead of a commercial Master Data Management [http://en.wikipedia.org/wiki/Master_data_management] product since the NCSC requirements do not demand a sophisticated system with features like “single version of truth” [http://en.wikipedia.org/wiki/Single_version_of_the_truth] and data governance.

9.7 Systems Management and Monitoring Requirements

All components use a logging framework that configures outside of the component. This allows components to write log and tracing information in a consistent and configurable way.

Here are some commonly used tools:

- JVM - log4j [http://logging.apache.org/log4j],slf4j [http://www.slf4j.org]
- .NET - Log4Net [http://logging.apache.org/log4net]

For components built on the JVM, the component uses a capability such as Java Management Extensions (JMX) [http://www.oracle.com/technetwork/java/javase/tech/javamanagement-140525.html]. Applications can expose information about performance, load and other information though a standard interface. Many management solutions support JMX through direct support or through JMX to Simple Network Management Protocol (SNMP) [http://en.wikipedia.org/wiki/ Simple_Network_Management_Protocol] adapters.

Similar to JMX, components on the Windows .NET platform implement Windows Management Instrumentation (WMI) [http://msdn.microsoft.com/en-us/library/windows/desktop/ aa394582(v=vs.85).aspx]. It performs the same capabilities for Windows components and is built into the operating system. Windows itself uses this protocol, so all tools capable of monitoring Windows can monitor the components.

Cloud vendors usually offer monitoring capabilities to their solutions. By following JMX / WMI and SNMP standards while implementing components, NCSC can choose management and monitoring solutions without being tied to a specific vendor.

These components expose information as to the status of the component (e.g. test delivery component needs to expose the number of connected students). These components also need to be monitored for preventative issues. The machine or VM that they run on must monitor for low-memory issues, disk-full issues, processor overloading issues and exceptions. These cause alerts in the system management software to notify support personnel of possible issues.

9.8 Middleware and Integration Software Requirements

This section details the main integration patterns and technology recommendations for messaging, communication and data transfer.

The following principles drive these recommendations:

- Favor lightweight integration and simple message queue frameworks over centralized hub and spoke models or messaging systems. In addition to being less expensive, these are easier to test, integrate, extend and have very low requirements on hardware and software.
- Resist adding business logic in centralized service buses, since they are harder to test and troubleshoot.
- Favor lightweight stateless services over integrating via a shared database. Database integration leads to tightly coupled systems that are difficult to test and extend. Stateless offers scalable and predictable services.

10. System and Acceptance Test Plans

System and acceptance testing are necessary in order to fix any bugs in the software coding as well as ensure that the software meets the functionality desired. System testing includes the process of testing an integrated system to verify that it meets specified requirements; whereas acceptance testing conducts formal testing to determine whether a software satisfies it's acceptance criteria and to enable the user or authorized organization to determine whether to accept the system.

Without broad testing, an array of problems can arise during an assessment causing financial and political expenses. With sufficient QA, testing provides the opportunity to eliminate the majority of these problems. Each of the testing types require comprehensive plans and testing occurs multiple time and during different stages as you do not wait until the software development completes. A part of the process comprises comprehensive tracking of any bugs through a formalized process. NCSC must have a clear testing strategy in place from the beginning of software development.

Some tenets of a testing strategy include:

- Confirming that all components meet the defined requirements and function as expected.
- Minimizing the risk of introducing major bugs to the operational system.
- Ensuring that performance issues are caught in time to fix them.
- Certifying that all components are successfully integrated.
- Guaranteeing that applications are compatible and function correctly on all supported configurations.
- Identifying issues early in the process so they are less expensive to fix.
- Minimizing the chance of major incidents once the system goes live.
- Giving confidence that the system can handle the expected traffic.

10.1 Testing Approach

A formal testing strategy is recommended for NCSC in that it provides greater prominence as to the overall state of the system, a greater understanding of the potential barriers that lie ahead and a better chance of meeting overall goals and objectives. The development teams procured by NCSC should be included in the strategy development; however, owned by NCSC. Consider the following guidelines for the test approach:

- Recognize efficiencies and don't add process unless it adds real value.
- Define clearly all roles and responsibilities.
- Create a definition of done. Testing never ends, so determine when a component or a system is ready and provide a clear definition of "done" for all releases.
- Establish entry and exit criteria for all environments.
- Promote quality from the beginning and get the development team to take ownership.
- Have common tools and a testing framework. This includes defect trackers, continuous integration, testing frameworks and automation tools.

- Ensure a common language with key terms and concepts for the assurance process.
- Provide standard, quality metrics to help determine the quality of each component. Compile the metrics for each sprint to enable the viewing of trends.
- Define QA artifacts by establishing guidelines for unit tests, test coverage, test cases, defect attributes, etc.
- Strategize platform compatibility testing, describing how to test all supported devices, operating systems, and browsers.
- Develop integration testing strategy describing how to verify that components are successfully integrated.
- Provide performance testing strategy to verify, with defined metrics, that the system can handle the anticipated load and still be responsive to end-users.
- Verify that sensitive data is being adequately secured both at rest and in transport between components with a security testing plan.

10.1.1 Functional Testing

To ensure and satisfy all requirements are me, functional testing occurs. NCSC should require all development vendors to keep an up-to-date test plan for each component with test cases aligned to individual requirements. These test plans and test cases need to be reviewed and approved on a regular basis. Execute test cases manually or using automation software and test results made transparent to all stakeholders.

10.1.2 Performance Testing

Performance testing ensures that the NCSC system handles the anticipated volume of traffic. The first step in this process includes making estimates on expected capacity. After each successful administration, NCSC more accurately predicts future capacity. With an estimate of the number of concurrent users, load tests must be conducted to test this upper bound. The end goal of this testing consists of having a clear picture of how system response times change as the number of users increase. It is important to know how many users the system can support while keeping response time below an acceptable threshold as well as understand what happens to the system under extreme loads.

10.1.3 Integration Testing

NCSC needs to ensure that all components work seamlessly as a system. This requires integration testing after each component has successfully completed unit testing. This type of testing guarantees that component interfaces function as expected, and that no new issues arise when connecting components. A single group, potentially one of the development vendors, oversees integration testing and maintaining the integration test environment.

10.1.4 Acceptance Testing

NCSC needs to establish a team responsible for User Acceptance Testing (UAT) and that it be done on a regular schedule. This opportunity certifies that the system works as intended and requirements satisfied in a way that is acceptable to the UAT team. This process needs to happen early and often in the development phase of the project.

10.2 Roles and Responsibilities

As mentioned previously, clearly defining roles and responsibilities proves critical with testing. Each team needs to understand which assurance activity for which they are responsible. The Architecture Review Board should approve the outlined roles and responsibilities, after review from the Architecture Core Team.

Development Vendors

Development Vendors, secured by NCSC, create the components for the assessment system. Most likely, multiple vendors will develop the assessment system. As such, each vendor has their own QA process and these processes must align to the overall NCSC strategy. The vendors will also need to work together to ensure that testing between components verifies.

Responsibilities

1. Ensure that staff has the skills necessary to successfully perform testing activities.
2. Maintain all environments needed for testing.
3. Manage a defect-tracking database and keep it up-to-date.
4. Develop test plans.
5. Create and maintain test cases and automated test scripts.
6. Execute test cases and generate test summary reports.
7. Generate traceability reports.
8. Facilitate User Acceptance Testing (UAT).
9. Perform system, integration, regression, compatibility, and load testing.

Architecture Core Team

The make-up of the Architecture Core Team includes the lead architects from the development vendors procured by NCSC. This cross-collaborative group is vital in ensuring communication and collaboration between all development parties. Many issues and items need discussion and resolution in order to bring the NCSC assessment system to fruition and this forum proves critical to the success of the development and deployment of the overall system.

Responsibilities

1. Review and approve all test plans.
2. Establish best practices to be followed by all testing teams.
3. Resolve disputes that arise from integration testing.
4. Define test summary report standards.
5. Review and approve all test summary reports.
6. Record and document any risks that arise from the testing process.

Architecture Review Board

The Architecture Review Board (ARB) provides one of the most important functions of the ongoing maintenance of the architecture. This board, made up of NCSC members, establishes policy and ensures that the development vendors adhere to the overall architectures. For testing, the ARB oversees the process and serves as a reviewer of the testing artifacts. In addition, the ARB reviews and provides actionable feedback from the testing process that reports back to the other teams.

Responsibilities

1. Accept or reject any testing standards used.
2. Accept or reject test summary reports.
3. Provide guidance with any integration issues.
4. Record and document any risks that arise from the testing process.
5. Assist with user acceptance testing.
6. Accept or reject changes to the testing strategy itself.
7. Accept or reject quality metrics.

NCSC Work Groups

The NCSC Management Team monitors and evaluates attainment of goals, objectives, and timelines, identifies barriers and solutions to problems encountered by workgroups or individual collaborative members and ensures that the research-to-practice efforts honor the contributions, insights, needs, and unique concerns of all collaborative members. For the architecture, the work groups are responsible for the development of the assessment system.

Responsibilities

1. Investigate alternative options to improve practice and develop methods to explore and test feasibility.
2. Develop work plan and resource requirements to guide vendor and work group activities.
3. Oversee and direct Consortium work in assigned content area.
4. Work with development vendors and Architecture Core Team to mitigate risks.
5. Sign-off on test plans.
6. Sign-off on test summary report requirements.
7. Sign-off on quality metrics used.

10.3 Escalation

Generating an escalation process is necessary for issues that arise. All phases of the escalation process focus on the involvement of the QA Test Lead and the Technology Project Manager who in turn (if required) alerts others in order to manage critical or high priority issues that arise.

Generate a rubric to determine defect severity and priority levels for escalation.

Severity	Condition	Impact	Probable Resolution	Expected Resolution Turnaroud
1 - Critical	• Large # of users impacted. • Critical capability failure, code used frequently. • Accuracy: Incorrect data may have legal impact. • Security issue impacts majority of users. • Prevents implementation of the business solution. • Jeopardizes Service Level Agreement (SLA). • Data corruption holding up further processing. Unable to complete cycle. • Unable to perform critical transactions with no work-around. • System is down/crashes/ hangs or product terminates abnormally.	Loss or inability of QA to transact in the environment will severely impact their ability to perform their primary functions in a timely manner.	Requires introduction of new code to resolve problem.	Fix ASAP. Continuously worked. Once fixed, code is installed.
2 - High	• Problem with essential functionality/Key functionality disabled. • Function doesn't work or doesn't work as intended. • Accuracy: Data corruption or impairment requiring remedy prior to further QA phases. • Security of issue impacts moderate # of users. • Greatly impacts business processing.	Data issues or environment loss would have significant impact to the testing schedule.		Code/data fix is worked on and is targeted for the next build.

Severity	Condition	Impact	Probable Resolution	Expected Resolution Turnaroud
3 - Normal	• Function does not satisfy business requirements/ processes This may require a work-around. • Limited impact to functionality. • Minor impact to standard Operations. • Data corruption with no impact to test. • Documentation issues that jeopardize accurate delivery of code.	Routine data or system loss would cause areas of the application not to perform as expected.		Prioritized by Technology Product Manager for resolution.
4 - Low / Cosmetic	• Misspelling or incorrect grammar • Problem with minor functionality: - Minor impact to limited functionality with a work-around. Minimal impact.	Documentation or cosmetic issue. Minimal impact.		Could be deferred to a later release/build depending on time remaining to project conclusion.

10.4 Tools and Templates

To maintain some consistency across development efforts it is recommended that NCSC establish templates for creating test documentation and a common toolset for test execution.

10.4.1 Testing Tools

Numerous testing tools exist for use in tracking, testing frameworks and load testing. These tools help organize the extensive process and support the testing plan. The following summary table outlines a list of suggested tools and the purpose for the use.

Tool	Purpose
Jira	Defect tracking, issue log for missing or unclear requirements and triage
Zephyr	Develop, store and execute test cases
Selenium, Webdriver and Java	Automation code
Jython and The Grinder	Performance code, test scripts and load testing
Zephyr	Test case design, assignment and execution

10.4.2 Templates

This section gives an outline for some of the testing documentation required for this project. These outlines offer a start for creating templates used across teams.

10.4.2.1 Test Plan

The overall test plan requires a well thought out strategy. The components of most testing plans are universal. The NCSC test plan should include these sections:

1. Project Description
2. Test Objectives
3. Test Scope
4. Technical Requirements for Testing
5. Supporting Documents
6. Entry and Exit Criteria
7. Assumptions and Risks
   a. Assumptions
   b. Risks
8. Schedule Summary
9. Test Logistics and Resources
   a. Resources
   b. Software and Tools
10. Test Case Design/Defect Management/Status Reporting
    a. Test Case Design
    b. Defect Management
    c. Status Reporting
11. Test Results
    a. List of outstanding incidents
    b. Impact
    c. Measurement criteria and lessons learned
12. Test Plan Sign-Off
13. Sign-Off (after Test Completion)
14. Outstanding Questions

10.4.2.1 Logging Bugs

Field Name	Comment
Project name	
Issue Type: Bug	
Summary	
Priority	
Severity	Blocker, critical, major, minor, trivial, etc.
Due Date	
Component	Blocker, critical, major, minor, trivial, etc.
Assignee	
Original Estimate	Time to fix
Attachment	Supporting documents
Description	
Fix in build	Date of the build the bug was fixed
Found in build	The date of the build in which bug was found
Target Resolution	
Browsers	
Operating Systems	
Functional Area	
Fix versions	
Resolution	Status; completed, not fixed, fixed, unresolved, etc.
Department	
Participants	

11. NCSC Glossary

The NCSC Glossary contains terms and phrases used within this document. These are provided within this appendix for ease of use for readers of the NCSC Architecture Document.

AA-AAS

Alternate assessments based on alternate achievement standards. The focus for NCSC is to build an alternate assessment based on AA-AAS for students with the most significant cognitive disabilities.

AAS

As a part of No Child Left Behind, alternate achievement standards (AAS) allowed states to set expectations of performance differing in complexity from grade-level achievement standards. These are aligned to CCSS.

Apache License

Apache license is a free software license authored by the Apache Software Foundation (ASF). The Apache License requires preservation of the copyright notice and disclaimer. All software produced by the ASF or any of its projects or subjects is licensed according to the terms of the Apache License. Some non-ASF software is also licensed using the Apache License.

API

An application programming interface (API) is a set of standards that defines the communication points for software components allowing components to communicate with each other and aiding in interoperability.

APIP

The Accessible Portable Item Profile (APIP) is a technical standard developed by IMS Global that focuses on accessibility in assessment items.

Application Architecture

Application architecture includes the design of the internal structure of an application.

Application Development

Application development is the development of a software product.

Architecture

The practical art of selecting and interconnecting hardware components to create computers that meet functional, performance and cost goals, to formally model those systems.

Apache Software Foundation (ASF)

A non-profit organization, made up of a community of decentralized software developers, who maintain a collection of open-source projects. Started in 1999, with their most notable project, the Apache web server.

ASP

Active server pages (ASP), a web-scripting interface developed by Microsoft, allows for dynamically generated web pages.

Bandwidth

A rate of data transfer, bit rate or throughput, measured in bits per second (bps).

Binary Transport

Binary transport is a transport implementation based on TCP or SSL/TSL. Distributed applications often utilize this transport.

Branching

Branching makes up a software development methodology. Branching typically involves workflow decisions and changes based on input or output.

Cardinality

In database design, cardinality is the defining of the relative size of elements used to describe a relationship between two elements.

CCC

Core Content Connectors (CCC) is the pinpoint of primary content for CCSS.

CCR

College and career readiness (CCR) signifies the knowledge and skills students should possess when they graduate from high school to be successful.

CCSS

The Common Core State Standards (CCSS) were created by the Council of Chief State School Officers and the National Governors Association to provide a consistent meaning as to what students should know and be able to do.

Component

A component includes one of multiple applications that may make up a system.

Concurrency

Concurrency is a property of technological systems in which several computations are executing simultaneously, and are potentially interacting with each other.

COP

Communities of practice (COP) often are included as a part of professional development. Individuals form a COP to collectively learn about a given topic.

CSS

Cascading style sheets (CSS) are used most often in web page design to determine the presentation of content.

DAM

Digital asset management (DAM) consists of management tasks and decisions surrounding the ingestion, annotation, cataloguing, storage, retrieval and distribution of digital assets. Digital photographs, animations, videos and music exemplify the target areas of media asset management (a sub-category of DAM).

Data Accountability Center

DAC's mission is to support the submission and analysis of high-quality IDEA data by reviewing data collection and analysis and providing technical assistance to improve state capacity to meet data requirements. The DAC's mission includes assisting the Office of Special Education Programs (OSEP) at the U.S. Department of Education by taking a leadership role in the Technical Assistance and Dissemination network to support the vision of high-quality data.

Data Warehouse

A database used for reporting and analysis.

Database

An organized collection of data for one or more purposes, usually in digital form.
Deployment
Deployment involves the process of making a software system available for use.

Domain

A set of common requirements, terminology and functionality for any software constructed to solve a problem.

Domain model

A domain model in problem solving and software engineering is a conceptual model of all topics related to a specific problem. It describes the various entities, their attributes, and roles and relationships, plus the constraints that govern the problem domain.

EBSR

In assessment, evidence-based selected response (EBSR) involves a specific item model where students, in part, are required to demonstrate the ability to cite evidence from text. It is used primarily in English Language Arts.

ECD

Evidence centered design (ECD) is an approach to creating educational assessments in terms of evidentiary arguments and is built using a framework.

ELA

An abbreviation for English Language Arts.

Epic

An epic includes a large feature, or a grouping of smaller features or user stories. Also see User Story.

ETL

The Extract, Transfer, and Load (ETL) process involves extracting the data from the source systems. The transform stage applies to a series of rules or functions to the extracted data from the source to derive the data for loading into the end target. The load phase loads the data into the end target, usually the data warehouse (DW).

Federation

A Federation is multiple computing and/or network providers agreeing upon standards of operation in a collective fashion.

Focal KSA

A focal knowledge, skill, and ability (KSA) is an essential part of the targeted standard to be evaluated. A focal value is often established that includes the value that the test taker must have to possess a high level in the targeted standard.

Friends and Enemies

In adaptive testing, friends and enemies are a way of defining the relationship between items. If an item is presented to a tester, its friend items should also be presented, and its enemies should not.

GSEG

A General Supervision Enhancement Grant (GSEG) is a grant program funded by the US Department of Education.

Hosting

Hosting involves the deployment of software in a physical environment that makes an application available on the World Wide Web.

Identifier

A unique name given to a specific object or a specific class of objects.

IEP

An individualized education plan (IEP), mandated by IDEA, is designed to help teachers and students meet the unique educational needs of a student with disabilities with the intention of enabling the student to achieve improved educational results.

IMS

An instructional management system (IMS) provides information to teachers, administrators and others in order to improve instruction in the classroom. This information can include data about student performance, content resources to use in the classroom, and analytics.

IMS GLC

IMS Global Learning Consortium (IMS GLC) is a global, nonprofit, member organization that strives to enable the growth and impact of learning technology in the education and corporate learning sectors worldwide.

Interface

An interface is a tool and concept that refers to a point of interaction between components and is applicable at the level of both the hardware and the software elements.

IP

IP refers to internet protocol, or intellectual property.

Item

An item is a composite object that is made up of many assessment item parts, metadata and paradata about that item.

JSON

JavaScript Object Notation (JSON) is a lightweight, text-based, open standard designed for human readable data exchange.

JSP

Java Server Pages (JSP) is a technology that helps software developers serve dynamically generated web pages based on HTML, XML or other document types.

LCI

The Learner Characteristics Inventory (LCI) was created to study the learning characteristics of students participating in alternate assessments based on alternate achievement standards (AA-AAS).

Learning Registry

The Learning Registry was created by the US Department of Education and the Department of Defense. According to the website, it is an open source technical system designed to facilitate the exchange of data behind the scenes and an open community of resource creators, publishers, curators, and consumers who are collaborating to broadly share resources, as well as information about how those resources are used by educators in diverse learning environments across the Web.

LGPL

The Lesser General Public License (LGPL) is a free software license published by the Free Software Foundation (FSF). The license explains how the software and its source code can be freely copied, distributed and modified.

LPF

The research-based Learning Progression Framework (LPF) describes a curricular sequence for how typical students develop and demonstrate more sophisticated understanding in each content area over time. From these LPFs for mathematics and English language arts (ELA), NCSC is developing grade-level assessment content targets and alternate achievement standards linked to the CCSS for students with the most significant cognitive disabilities.

Metadata

Metadata includes information about data that is usually associated with content for ease of search and description of the data.

NAAC

The National Alternate Assessment Center (NAAC) is a five year project funded under the US Department of Education, Office of Special Education Programs focused on alternate assessments.

NCEO

The National Center on Education Outcomes (NCEO) provides national leadership in designing and building educational assessments and accountability systems that appropriately monitor educational results for all students, including students with disabilities and English Language Learners (ELLs).

NCIEA

The National Center for Improvement of Educational Assessment (NCIEA) provides support for the technical, practical and policy issues of large-scale educational assessment and accountability programs nationwide.

NoSQL

NoSQL is a broad class of database management systems that significantly differ from the classic model.

Ontology

An ontology represents knowledge as a set of concepts within a specific domain, content area, and the relationships among those concepts.

Open Content

Open content refers to content that can be copied and/or modified. Licensing can still impact open content and often Creative Commons licenses are applied.

Open Source Software

Software that has its source code available is referred to as open source software. Some rights can apply depending on licenses to allow users to change, improve and distribute the software.

Organizational Partners

Several partners working together make-up NCSC, including: University of Kentucky, University of North Carolina Charlotte, edCount, and the state departments of education of Arizona, Alaska, Connecticut, District of Columbia, Florida, Georgia, Indiana, Louisiana, Massachusetts, Nevada, New York, North Dakota, Pacific Assessment Consortium (PAC-6), Pennsylvania, Rhode Island, South Carolina, South Dakota, Tennessee and Wyoming.

Paradata

Paradata includes reference information about data. An example of paradata would be the number of times a piece of content was used with a variety of students.

Progress Indicators

When proceeding through an assessment, a progress indicator may include a graphic or text image to inform the student that he/she is in progress. It may also show the student how much further until the end of the assessment.

QTI

The Question and Test Interoperability (QTI) is a specification created by IMS Global to define questions, test and results.

RTI

Response in Intervention (RTI) includes a formal process to provide support for students at the appropriate level as a prevention measure and maximize student achievement.

RTTA

The Race To the Top Assessment (RTTA) program is a grant funded by the US Department of Education to develop innovative assessment systems. Two consortia made up of states were awarded the grants – the Smarter Balanced Assessment Consortium and the Partnership for Assessment of College and Career Readiness (PARCC).

SAAS

Software as a service (SAAS) is a software delivery model in which the application and all data is hosted, often in the cloud.

SCORM

The Shareable Content Object Repository Model (SCORM) is a reference model that brings together several standards to create content interoperability.

SIF

The Schools Interoperability Framework Association (SIF Association) is a nonprofit organization with over 3,200 member organizations that produces open technical standards, SIF Specifications, for interoperability in the education ecosystem, including everything from student information systems to assessments to learning resources.

SLA

The service level agreement (SLA) defines the level of service. This often includes performance, failure recovery and time.

SME

A subject matter expert.

SSCD

Students with significant cognitive disabilities.

Support vs. Accommodation

A support is provided to a student that has a disability in order for them to reach their fullest potential. Some examples of support include accommodations, modifications or adapting instruction. An accommodation is a type of support and includes a change that helps a student overcome or work around the disability. The accommodation does not change performance expectations. For example, a student who is visually impaired may need large print.

TEI

A technology enhanced item (TEI) involves a computer-delivered item that includes specialized interactions for collecting response data. These interactions are typically beyond constructed-response or selected-response.

Tenant

In architecture design, an instance of the software that runs on a server, serving a single client organization. Multi-tenancy is an instance of the software that runs on a server, serving multiple client organizations (tenants).

Thin Client

A thin client is a computer that relies on another computer or server to operate fully.

UDL

Universal Design for Learning (UDL) was created by CAST and is a set of principles for curriculum development that gives all individuals equal opportunities to learn. UDL provides a blueprint for creating instructional goals, methods, materials, and assessments that work for everyone, regardless of their abilities and individual needs.

User Story

A user story is a small piece of a requirement that accomplishes a single identified goal in software development.

UX

User experience (UX) takes into account how a person interacts with the overall system or software program.

UXD

User experience design (UXD) is based on what the overall UX a software program would like to convey.

Validity Argument (VA)

According to Kane, 2006, the validity argument provides an evaluation of the interpretative argument.

WIDA

The World-Class Instructional Design and Assessment (WIDA) is an assessment grant, funded by the US Department of Education, provided to a consortium for the development of innovative assessment for English language learners.

Wireframes

A wireframe provides a visual mock-up of a web page or series of screens for a software application.

XML

Extensible Markup Language (XML) offers a standard set of rules for encoding documents in machine-readable form.