ADL DAU Sandbox

Final Report

March 2021

Contract information:

IDIQ #: OPM2615D0001

PowerTrain Project #: 4781-01

Sae Schatz, Ph.D.
sae.schatz@adlnet.gov

Val Feemster
Valerie.Feemster@opm.gov

Juli Tompkins
jtompkins@powertrain.com

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 01-03-2021	**2. REPORT TYPE** Research Paper	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE
ADL DAU Sandbox - Final Report

5a. CONTRACT NUMBER
IDIQ #OPM2615D0001

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
0603769D8Z

6. AUTHOR(S)
Sae Schatz, Ph.D.
Val Feemster
Juli Tompkins

5d. PROJECT NUMBER
PowerTrain Project #4781-01

5e. TASK NUMBER
N/A

5f. WORK UNIT NUMBER
N/A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Advanced Distributed Learning Initiative
USA Learning
PowerTrain

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
ASD Personnel & Readiness
Advanced Distributed Learning Initiative
13501 Ingenuity Drive, Suite 248
Orlando, Florida 32826

10. SPONSOR/MONITOR'S ACRONYM(S)
ASD/P&R/ADLI

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In May of 2020, the Advanced Distributed Learning Initiative (ADL) and the Defense Acquisition University (DAU) partnered with USALearning/PowerTrain to create a practical application of ADL's 2019 TLA Reference Implementation. The purpose of this project was to recreate key components of the Reference Implementation with commercially-available, open source, and customized solutions to demonstrate the value of organizations choosing to adopt the architecture to improve their ability to track competency-based learning.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Sae Schatz
U	U	U	UU	78	**19b. TELEPHONE NUMBER** *(Include area code)* 5714804640

Contents

Executive Summary

In May of 2020, the Advanced Distributed Learning Initiative (ADL) and the Defense Acquisition University (DAU) partnered with USALearning/PowerTrain to create a practical application of ADL's 2019 TLA Reference Implementation. The purpose of this project was to recreate key components of the Reference Implementation with commercially-available, open source, and customized solutions to demonstrate the value of organizations choosing to adopt the architecture to improve their ability to track competency-based learning. The DAU Sandbox, as it came to be known, contained multiple hardware and software components, including:

- Learning Records Stores (LRSs): collect, store, and route data about learners and their activities
- Activity providers: provide data to LRSs
- Learner records: learning and demographic data about learners
- Competency management system: stores competencies that are mapped to learning events
- Dashboards: provide users with varying access permissions to view data specific to their needs

One important aspect of the DAU Sandbox is that it is designed with interoperability in mind. While the Sandbox is a closed system in a test environment, the ultimate goal of the architecture is to be able to connect with activity providers and learner record data across multiple enclaves. For this connectivity to occur, it was vital to build the system using recognized standards, such as the IEEE 1484.12.1 Learner Object Metadata 2.0 draft standard. Where possible, these standards or subsets of them were used to provide a future capability to connect to additional data providers.

The DAU Sandbox served as an abbreviated version of the envisioned end architecture and was credibly able to create the competency-based learning environment as designed. During testing, data was able to be exchanged and "passed through the pipes" as desired. Course completions for users were successfully tied to competencies and the users who achieved them. Relevant sections of the standards were successfully applied. Opportunities to enhance system performance moving forward were identified, particularly in the areas of component maturation, security, and the Competency and Skills System (CaSS) that was used. Other successes included identifying opportunities to collect data at a more granular level, and additional business rules that could be established for score competency and experience.

Background

Training organizations for years have been struggling with ways to identify the total picture with regard to learners and learning within their organization. Learners often achieve skills and knowledge in disparate ways that are not connected or otherwise shared. This stove piped existence hampers organizations in understanding workforce needs and finding value in their learning opportunities. Without looking at the whole picture of an individual's learning, the organization cannot have accurate data on the skills and skills gaps within their workforce. In order to view of the whole picture of an individual's competencies and learning experiences, we are developing an intricate competency-based learning system.

Competency-based learning refers to systems of instruction, assessment, grading, and academic reporting that are based on students demonstrating that they have learned the knowledge and skills they are expected to learn. In theory, this extends beyond taking instructor-led or web-based training. It would also capture outside education or job experience. What is unique about competency-based learning is that it focuses on what students learn and not on the time spent in the classroom completing credits. In this approach, students work at their own pace to demonstrate mastery in the competencies necessary for their chosen field of study.

ADL has undertaken a five-year effort (to date) to create a Total Learning Architecture (TLA) designed to help organizations to better understand their workforce knowledge base through the interpretation of meaningful data. The TLA is an R&D project to design a business enterprise architecture for learning (training/education) systems. The project began as a theoretical approach to the architecture design and has evolved to a working sandbox using real-world components. The TLA R&D project also includes Reference Implementations, which are non-functional models as well as sandboxes that allow DoD organizations to test and evaluate the architecture.

A driver for the development of the architecture lies within The DODI 1322.26 Policy.

> **"**To implement DoD policy affecting distributed learning, the DoD Components will adhere to several guidelines. When developing or acquiring distributed learning, they must search for existing distributed learning content that may be reused or repurposed, and should make existing distributed learning assets, content, and other reusable resources visible and accessible to other DoD Components. They should strive to design and develop distributed learning that leverages learning science, technology, specifications, and standards to produce state-of-the-art, affordable, effective, and convenient education and training. It is also essential to consider the security of networks, data, and personal information in all distributed learning content and systems development and comply with all applicable policies and requirements for the protection thereof."

The DODI 1322.26 Policy Framework:

- Dictates there be a federated data strategy for all education and training related data.
- Is derived from internationally accepted technical specifications and standards.
- Ensures portability of learning data between enclaves.
- Provides auditability and non-repudiation of competencies and credentials.
- Facilitates enterprise analytics, artificial intelligence, and automation.

Conceptually, the project serves as a data strategy for the education and training community. This data strategy includes**:**

- Digital learning data strategy
- Data and software interoperability standards
- Specifications for microservices
- Specifications for technology architecture implementations
- Recommendations for business rules and governance

A critical feature of the DAU Sandbox is the Experience Application Programming Interface (xAPI). xAPI is an eLearning specification that enables data collection on the wide range of experiences a person has within online and offline training activities. xAPI's use of a shared format for both the receiving and sending of data makes it an ideal tool for sharing learning between multiple systems, making it possible to track experiences that happen in many different environments and systems. There are two key elements within the xAPI specification: statements and the LRS.

Statements dictate the format for specific learning activities and follow an "[actor] [verb] [object]" structure. The LRS is where xAPI statements are stored and its portion of the specification defines the communication method for sending, receiving, and requesting data.

In order for the DAU Sandbox to function across the greatest user base and meet future security needs, it must follow the Identity, Credential, and Access Management (ICAM) process. It also requires the organization to instrument an immutable identification (ID) for learners so that ALL data for each learner can be consistently tracked throughout the system's various components.

The DAU Sandbox was built following the above-described principles over the course of nine months. The development process included conducting in-depth research, employing ADL and open-source components, adapting existing USALearning components, and importing necessary courseware and competency frameworks from DAU. As components were added, they were configured to emulate practical systems that DAU currently has as closely as possible.

Significant actions completed during the course of the project included:

- Creating a system design that leveraged work completed on prior iterations of the TLA
- Analyzing capabilities of candidate systems for the Sandbox
- Identifying sources of data
- Installing Sandbox components
- Creating custom coding where necessary to connect Sandbox components
- Conducting interim testing as components were added
- Developing a testing plan
- Conducting requirements and functionality testing

The paragraphs below contain descriptions of the logical, hardware, and software architecture used to implement the DAU Sandbox. This architecture generally follows the logical, hardware, and software architecture of ADL's 2019 Reference Implementation.

DAU Sandbox Architecture

The DAU Sandbox is a practical testbed application based on the 2019 ADL Total Learning Architecture. The purpose of this project is to advance the theoretical generic application of the 2019 Reference Implementation to a constrained real-world environment. The system specification for the DAU Sandbox is initially scaled for a small user base using a limited number of activity providers, with the potential to be expanded for additional learners and activity providers given an increase in hardware capacity and additional efficiencies in software performance, particularly in the competency management system.

Figure 1 contains a graphic representation of the Sandbox components and how data moves through the system.

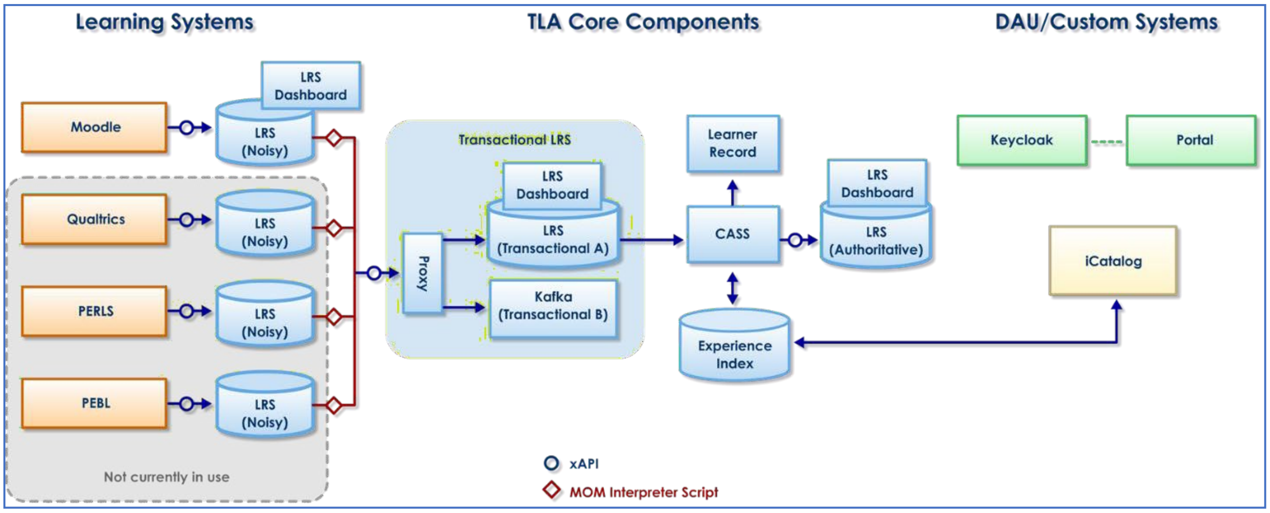


Figure 1. DAU Sandbox Components. *This figure outlines the high-level arrangement and flow of information for the DAU Sandbox.*

Logical Architecture

The 2019 Reference Implementation Logical Architecture was based on the Common Education Data Standards (CEDS) data model for understanding the relationship between learning data and the processes used to create or manipulate those data. The DAU Sandbox followed this model, as its structure essentially mirrored the Reference Implementation.

Under the CEDS model basic processes of student registration and tracking, content presentation, and performance tracking is captured in a Learning Management System (LMS). Relying on an LMS alone, however, can potentially create a vendor lock for all functions within that product (or suite of related products if developed by the same vendor). It also limits the workflow options to those supported by the LMS User Experience, which is derived from the "factory model" of learning and the "Shannon/Weaver model" of learning transfer inherent in traditional models of content delivery.

The DAU Sandbox attempts to decouple LMS functions from different learning activities and segregate management of student and instructor experiences within learning environments from the actual act of providing learning. The factory model of learning starts with a set curriculum and pushes as many students through as possible; standardized testing is the quality control mechanism for measuring educational progress. Following the Shannon/Weaver model of communication theory, a transmitter (instructor) delivers a message (instruction) to a passive receiver (student); successful communication of that message is verified using traditional assessments. The TLA Concept of Operations (CONOPS) replaces the "factory model" with a learner-centric integrated supply chain model based on a core/edge paradigm. The edge systems are the devices used to provide learning, which may include courses held in a traditional LMS, as well as handheld devices, intelligent tutors, electronic publications, simulators, and any other evolving learning technologies.

In order to conduct ledgering of the data types that pertain to planning and controlling individual learning events, a mix of core systems is necessary. These core systems are configured to collect and make sense of data from the intersection of learners, resources, and organizations. Core services manage the learner bookkeeping functions, while back-end core services manage the virtual network bookkeeping functions necessary to operate in a distributed, cloud-based environment. Ancillary functions, like an access portal, data visualization tools, and attached learning devices are edge systems that communicate to the core. Within the DAU Sandbox, learning content and the activities that host it are edge systems.

The core service features are separated into service groups tied to the data structures they support. These four data structures represent aspects of the learner and their possible learning paths.

The DAU Sandbox employs a Service Layer that acts as the bridge between learning devices, other TLA components, and shared data stores. Each service exposes the stored data to an application so that information can be transformed into other meaningful data used by other components. Each service group includes control logic and user interfaces for a set of functions. The data contracts between data and service layers are based on the nature of the data exchanged. The behavior and functionality of each service is defined and aligned with TLA business functions. The Service Layer includes:

1. Core Services
    - Competency Management – tracks the overall competency state for selected goals and makes assertions based on evidence provided via xAPI. An instance of the CaSS v1.39 hosted the competency framework definitions for the DAU Sandbox. The DAU Contracting competency, which was selected for the DAU Sandbox, maps to 10 selected courses from the DAU course catalog.
    - Experience Index – is managed by an activity and resource registry service which manages the Experience Indices, containing metadata for all content in the 2020 content database.
2. Back-end Services
    - Identity Management – handles protection of PII, login credentials, and identification. Keycloak is an open-source identity and access management solution used in the DAU Sandbox. Note that all data in the Sandbox are fictitious so there is no actual PII to protect; however, Keycloak has the capability to handle PII for further iterations of the Sandbox.
3. Edge Services
    - Portal – displays basic data and provides a redirect service for the otherwise protected-access user interfaces native to each of the services listed above. For the Sandbox, the Portal is limited to Learner and Admin interfaces.
    - Decision Management – is based on data dashboards generated by Veracity LRSs.

The key enabling technology of the Sandbox is xAPI. Using xAPI for decoupling learning content delivery from the planning and tracking mechanisms allowsfor a broader array of trackable learning experiences. The xAPI specification uses a client-server paradigm of Learning Record Providers (LRPs) that generate xAPI statements and Learning Record Consumers (LRCs) that use them. Learning Activities are always acting as LRPs, though some are also LRCs.

In the current iteration of the Sandbox, a version of Moodle using the Community Logstore xAPI Plugin generates xAPI statements capturing learning events. Additional LRPs, such as PeBL and PERL will serve the same role as they are integrated in later iterations. The xAPI statements are normalized to “actionable information” that propagates through the core services to provide evidence of learner competence and are eventually archived to LRSs.

In order for the Sandbox to function properly, the Learning Activities xAPI data must be translated to the TLA data contracts (specifically xAPI and the TLA Master Object Model (MOM), which are used to normalize data) within the learning environment by acting as boundaries between the learner and the core services. The composable arrangement of web-based services, data, and devices operating with strongly typed data contracts provides these planning and tracking functions.

The performance of these microservices can be extended horizontally by cloning the processes on multiple server instances using cloud-based technology.

Hardware Architecture

The DAU Sandbox uses 10 virtual machines, listed in Table 1. The DAU Sandbox is installed in an Azure virtual private cloud hosted via contract to USALearning. Azure provides the back-end platform hosting, virtualization, and Domain Name Service (DNS) resolution functions. Each machine was procured under contract to USALearning and maintained by PowerTrain.

The server instances communicate between themselves using either HTTP/S over TCP/IP or by producing and consuming messages to the centralized Kafka cluster, internally to the Azure campus. External clients accessing the portal, the hosted content, or the service redirects may be located outside the Azure campus and connect via REST. The application ports and protocols used to access each service are listed in Table 2.

Table 1. DAU Sandbox Server Provisions. *Computing and storage presets for each machine used during the DAU Sandbox.*

DAU Sandbox					
Primary Component	**VM Size**	**Operating System**	**Volumes**	**Volume Type**	**Storage**
ADL-CaSS	**Standard D2s v3 (2 vcpus, 8 GiB memory)**	**UBUNTU 18.04**	**2**	**Premium SSD**	**30 GB 32 GB**
ADL-ExperienceIndex	**Standard D2s v3 (2 vcpus, 8 GiB memory)**	**UBUNTU 18.04**	**2**	**Premium SSD**	**30 GB 32 GB**
ADL-Kafka	**Standard D2s v3 (2 vcpus, 8 GiB memory)**	**Linux (redhat 7.8)**	**2**	**Premium SSD**	**32 GB 128 GB**
ADL-Keycloak	**Standard D4s v3 (4 vcpus, 16 GiB memory)**	**Linux (centos 7.7.1908)**	**2**	**Premium SSD**	**30 GB 32 GB**
ADL-LearnerRecord	**Standard D4s v3 (4 vcpus, 16 GiB memory)**	**UBUNTU 18.04**	**2**	**Premium SSD**	**30 GB 64 GB**
ADL-Portal	**Standard B2s (2 vcpus, 4 GiB memory)**	**Linux (centos 7.7.1908)**	**2**	**Premium SSD**	**30 GB 32 GB**
ADLSBAPP1 (LMS)	**Standard D2s v3 (2 vcpus, 8 GiB memory)**	**Linux (centos 7.7.1908)**	**2**	**Premium SSD**	**30 GB 32 GB**
ADLSBAPP2 (LRS)	**Standard D2s v3 (2 vcpus, 8 GiB memory)**	**Linux (redhat 7.8)**	**2**	**Premium SSD**	**32 GB 32 GB**

ADLSBDB1 (LMS)	Standard D2s v3 (2 vcpus, 8 GiB memory)	Linux (redhat 7.4)	2	Premium SSD	32 GB 64 GB
ADLSBDB2 (LRS)	Standard D2s v3 (2 vcpus, 8 GiB memory)	Linux (redhat 7.4)	2	Premium SSD	32 GB 64 GB

Table 2. Service, Container, and Port Details. *Service, port usage, and container layouts of each machine.*

Auth Server	https://keycloak.azure.usalearning.net				
Components	**Service**	**Local Port**	**Public Port**	**Public Path**	**Description**
Keycloak	Keycloak	8080	proxied	N/A	Learning Experience Index core service and UI
MySQL	Mysqld	3306			Keycloak's database
Nginx	Nginx	80/443	80/443		Reverse proxy handling ports / SSL
Certbot	Certbot				SSL certificate management and automation

Content Server	https://portal.azure.usalearning.net				
Components	**Service**	**Local Port**	**Public Port**	**Public Path**	**Description**
Nginx	Nginx	80/443	80/443		Reverse proxy handling ports / SSL
Certbot	Certbot				SSL certificate management and automation

LRS Server	https://lrs.azure.usalearning.net				
Components	**Service**	**Local Port**	**Public Port**	**Public Path**	**Description**
USAL LRS	USAL LRS	N/A	proxied	Root	Service USAL LRS
MongoDB	Mongod	27017	N/A	N/A	LRS primary database
Elasticsearch	Elasticsearch	9200	N/A	N/A	LRS analytics and dashboard acceleration database
Nginx	Nginx	80/443	80/443		Diverts xAPI traffic to proxies, all else to LRS directly. Also handling ports / SSL
Certbot	Certbot				SSL certificate management and automation

Moodle Server	https://lms.azure.usalearning.net				
Components	Service	Local Port	Public Port	Public Path	Description
Moodle	Moodle	N/A	proxied	root	Moodle instance running through Apache
MySQL	MySQL	3306			
Nginx	Nginx	80/443	80/443		Reverse proxy handling ports / SSL
Certbot	Certbot				SSL certificate management and automation

Kafka Server	https://kafka.azure.usalearning.net				
Container	Service	Container Port	Public Port	Public Path	Description
Kafka Broker 1	Kafka	19092	19092		Clustered Kafka broker instance
Kafka Broker 2	Kafka	29092	29092		“
Kafka Broker 3	Kafka	39092	39092		“
Zookeeper 1	Zookeeper	12181			Clustered Zookeeper instance
Zookeeper 2	Zookeeper	22181			“
Zookeeper 3	Zookeeper	32181			“

Metadata Server	https://xi.azure.usalearning.net				
Container	Service	Container Port	Public Port	Public Path	Description
Experience Index	Experience Index	5000	proxied	experience	Experience index mapping activitiesto competencies
MongoDB	Mongod	27017			Learning ExperienceIndex's database
Nginx	Nginx	80/443	80/443		Reverse proxy handling ports / SSL
Certbot	Certbot				SSL certificate managementand automation

CaSS Server	https://adlcass.usalearning.net				
Container	Service	Container Port	Public Port	Public Path	Description
CaSS	CaSS	8080	80	root	Instance of CaSS
CaSS	Apache2	8080			Hosts CaSS on port 80
CaSS	SkyRepo	8000			CaSS’s Database
Nginx	Nginx	80/443	80/443		Reverse proxy handling ports / SSL

Learner Data Server	http://learner-data.azure.usalearning.net				
Container	Service	Container Port	Public Port	Public Path	Description
DAU Learner App	dau-learner-api	3005	proxied	/	DAU Learner UI and API
MongoDB	mongod	27017			DAU Learner App's database
Nginx	nginx	80/443	80/443		Reverse proxy handling ports /SSL
Certbot	Certbot				SSL certificate management and automation

Learning Activities in the DAU Sandbox include ten of DAU's Contracting courses loaded into the USALearning Moodle. The Sandbox can be scaled to allow additional LRPs as identified.

Software Architecture/Data Strategy

This section describes the major components and high-level responsibilities for each DAU Sandbox software component, as shown in Figure 2. While the DAU Sandbox is structured in a way that allows the test and evaluation of individual components and capabilities, in practice, many of these components will be integrated into a larger system.

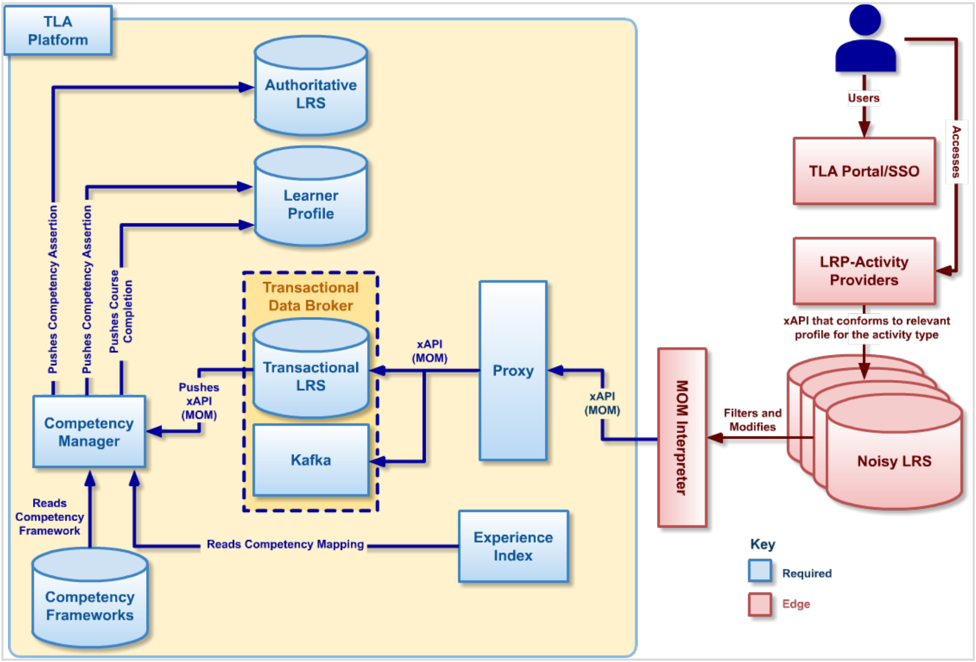


__Figure 2. DAU Sandbox Component Architecture.__ Each software "component" refers to a high-level service group that typically consist of several smaller services, each "microservice" performing a specialized job for that component.

The goal of the DAU Sandbox is to integrate data from different sources in order to make connections and inferences between the data. The data strategy is built from the four data pillars that serve as the basis of ADL's Total Learning Architecture. These four pillars are:

1. A **Learner Profile** to record learner credential history, aptitudes, local and global preferences, and local state -- which can be shared at the enterprise level (leveraging federated identity to protect privacy) to provide a complete portrait of human performance.
2. A **Competency Framework** to define the elements, relationships, standards, contexts, and levels of mastery required to perform jobs and to certify the credentials used to define job placement.
3. A **Learning Record Store** to store the records of learning experiences that improve performance and operational data indicating effective learning transfer and impact on mission effectiveness.
4. An **Experience Index** to list and describe the activities that provide a context for a learning or assessment opportunity, and the content that is provided within that context. Activity-Content tuples are linked with competencies that they trigger into "experiences" which can be scheduled or launched for learners.

Each of these pillars have an associated standard that was used to drive the connections between the Sandbox systems. Many of the data elements in the draft standards are intended to aggregate into dashboards and are designed to incrementally increase the fidelity for how different items are tracked.

Where appropriate, the data strategy for each these four pillars is discussed below with its associated software.

Experience Index

Description

An Experience Index is used to list and describe the activities that provide a context for a learning or assessment opportunity, and the content that is provided within that context. Metadata contained in the Experience Index is used by other Sandbox components. An Experience Index also stores information on the relationships among content, competencies, activity and content metadata, and evaluated paradata. Together, these data describe the learning experiences represented within the activity-content-competency triplet. The metadata are used for activity scheduling and evaluating the impact of a given learning experience on competence. The Experience Index houses the metadata that describes content resources, instructional activities, and competency objects (describing educational alignment) that can ultimately link to lessons, courses, and credentials.

Data Strategy

The Experience Index used in the Sandbox is intended to be built following the IEEE 1484.12.1 Learner Object Metadata 2.0 draft standard. Descriptions of learning activities and their associated content are stored in the Sandbox's Experience Index and use a modified version of the Learning Resource Metadata Initiative standard.

Activity Providers

Activity providers are the edge system LRP that represent the boundary between learning technology and the TLA core services. Activity providers can include LMS servers, simulation hosts, device managers, or direct connect devices that host the content, in the form of files, e-publications, scenarios, etc. Together, the content and activity form an "experience" which preserves the intent of the learning and context under which it occurs. Each LRP is uniquely situated to produce learning records that connect a user to learning experiences within an activity. The LRP is responsible for the formatting and population of a learning record that meets xAPI requirements. These learning records are then transmitted to a Noisy LRS.

For the DAU Sandbox, these activity providers are limited to 10 DAU Contracting courses contained in the Moodle LMS and data provided by an activity provider.

Learner Record Store(s)

xAPI-enabled learning activities generate statements, or records of learning that include a basic triple structure consisting of Actor, Verb, and Object. Services transmit these statements over HTTPS. An LRS serves as a repository for learning records stored by connected systems or content where learning activities are conducted. Its main function is to store and retrieve the data that are generated by LRPs such that all other Sandbox components may access those data without being dependent on direct communication with each other. LRSs generally can store more granular data than that held in an LMS. For example, an LRS can store information such as “user watched video X for 5 seconds”, as opposed to an LMS, for which data is at the completion/pass level. The LRS can store data from learning that takes place outside of LMS; however, these data have to be configured to track to the LRS.

For the DAU Sandbox, *Noisy*, *Transactional*, and *Authoritative* LRSs are established to store and retrieve xAPI data. The communication process for the three types is as follows:

1. All learning experiences that can be tracked with xAPI send data to *Noisy* LRSs that store all types of xAPI data. In the Sandbox, this xAPI data comes from the Moodle LMS; however, as indicated above, any learning activity that sends xAPI statements can be tracked in an LRS. The Moodle collects all sorts of information about users and courses and sends those data to the Noisy LRS. The Noisy LRS collects many data points, but the current configuration of the DAU Sandbox only acts on “completions”. MOM Interpreter Scripts translate the “noisy” data points to a corresponding MOM statement every time a completion statement is received. The completions now covered to the MOM profile are then forwarded to the Transactional LRS.
2. The *Transactional* LRS contains two components: the LRS and Kafka. Statements of completion are pushed through Kafka (see detailed discussion below) and stored in a traditional xAPI-conformant LRS.
3. The *Authoritative* LRS contains competency assertions stored by CaSS. CaSS receives completion data from the Transactional LRS, determines if a competency should be asserted and if so, stores the assertion in the Authoritative LRS.

Data Strategy

LRSs, by definition support xAPI data. In the current DAU Sandbox environment, xAPI Version 1.0.3 is used to represent learning experience data.

Learner Record

The Learner Record contains any information that is captured about a learner, and is maintained in the Learner Profile. The Learner Record aims to capture and hold data that often transcends the information in a traditional transcript. Items that a Learner Record can hold include:

- Course completions
- Demographics
- Credentials

- Position
- Geographic location
- Physical attributes
- Competencies

Data Strategy

Numerous efforts have been underway to create standards for what a learner record should contain and how it can be shared. The draft IEEE 1484.2 Integrated Learner Record (ILR) is an attempt to align these different standards. It contains in excess of 400 attributes, grouped by categories such as Person, Organization, Employment, Course, Credential, Competency, Career, and Issues. A subset of this standard was used for the Sandbox Learner Record:

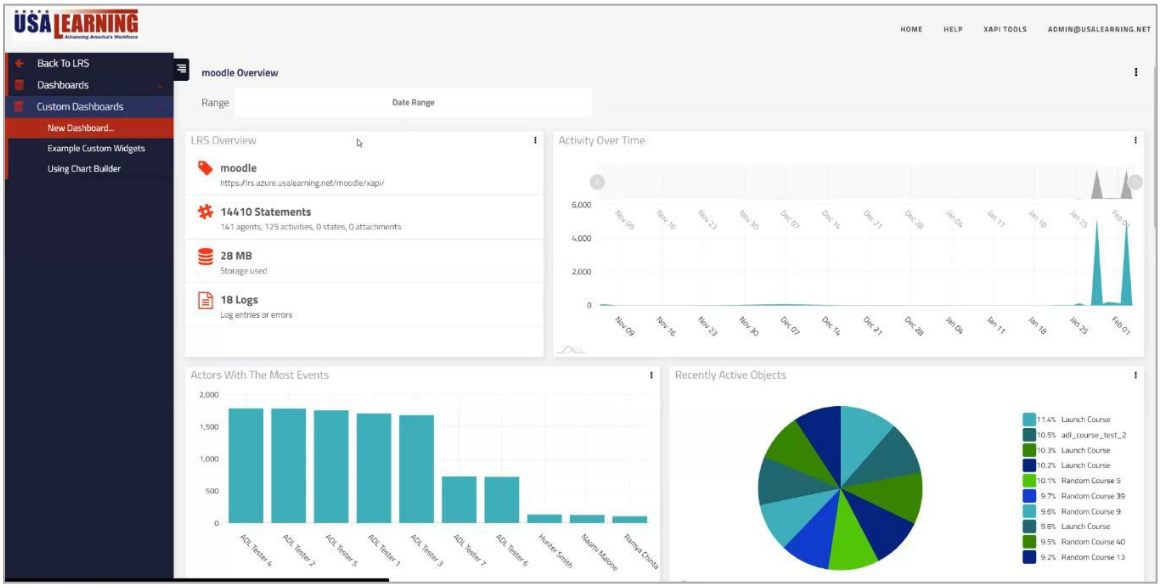
1. StudentIdentifier: <person ID>
2. PersonName: <string person name>
3. LRSEndpoint: <URL of the Authoritative LRS>
4. Courses
    a. CourseIdentifier: <CourseIRI>
    b. CourseTitle: <CourseTitle - OPTIONAL>
    c. CourseType: <ILT or VILT or DL - OPTIONAL>
    d. CourseCredits: <Number of credits for this course - OPTIONAL>
    e. CourseCreditBasisType: <Type of enrollment for this course - OPTIONAL>
    f. CourseAcademicGrade: <Grade for this course - OPTIONAL>
    g. CourseMetadataRepository: <URL of the experience index - OPTIONAL>
    h. AssessmentReportingMethod: <Method instructor uses - OPTIONAL>
5. Competencies
    a. CompetencySet: <Competency set - OPTIONAL>
    b. CompetencyFrameworkTitle: <Framework title - OPTIONAL>
    c. CompetencyFrameworkVersion: <Framework version - OPTIONAL>
    d. CompetencyFrameworkIdentifier: <Framework ID - OPTIONAL>
    e. CompetencyFrameworkDescription: <Framework Description - OPTIONAL>
    f. CompetencyFrameworkOwner: <Framework Owner - OPTIONAL>
    g. CompetencyDefinitionIdentifier: <Definition ID>
    h. CompetencyDefinitionURL: <Definition URL>
    i. CompetencyDefinitionNodeName: <Definition Node Name - OPTIONAL>
    j. CompetencyDefinitionVersion: <Definition Version - OPTIONAL>
    k. LearnerProficiencyLevel: <Proficiency Level>
    l. RequiredProficiencyLevel: <Required prof level - OPTIONAL>
6. Credentials
    a. CredentialIdentifier: <Cred ID>
    b. CredentialName: <Cred name - OPTIONAL>
    c. CredentialRegistry: <Cred registry>
    d. CredentialVersion: <Cred version - OPTIONAL>
    e. CredentialJurisdiction: <Jurisdiction - OPTIONAL>
    f. CredentialStatus: <status>

LRS Dashboards

The LRS dashboards are used to display analytics for data generated through the Sandbox. The LRS has the capability to easily create custom dashboards that can be used to study data in a wide array of views. The following six dashboards were created and enabled for the DAU Sandbox:

- Count of active users (sessions and durations) by day sorted in descending order (30 day rolling window)
- Average final score relative to average session duration per learning activity sortable on either dimension (30 day rolling window)
- Count of completions per learning activity by day sorted in descending order (30 day rolling window)
- Data table of most recent 20 activity statements
- Top 10 most frequent verbs, filterable by object/activity name
- Top 10 most frequent object/activity names, filterable by verbs

These dashboards can be viewed in either table (data only) or chart view. Figure 3 contains a sample chart view.



moodle
https://lrs.azure.usalearning.net/moodle/xapi/

14410 Statements
141 agents, 125 activities, 0 states, 0 attachments

28 MB
Storage used

18 Logs
Log entries or errors

Figure 3. Sample Dashboard Chart View. *This figure shows a sample dashboard that displays a chart view for specific data gathered from the DAU Sandbox.*

Streaming Platform

For the DAU Sandbox, Kafka receives course completion xAPI completion data from the transactional LRS. Kafka is a horizontally scalable message brokering system based on the producer-consumer model, with LRPs as the producers and LRCs as the consumers. This permits all systems and services to be developed and integrated with the expectation of time-stamp correct xAPI traffic as a guarantee. Kafka is a distributed system consisting of servers and clients that communicate via a high-performance TCP network protocol. It can be deployed on bare-metal hardware, virtual machines, and containers in both on-premise and cloud environments, and can serve as a good resource for organizations to tap into the data they collect.

Kafka combines three key capabilities so use-cases can be event streamed end-to-end with a single solution:

1. **Publishing** (write) and **subscribing to** (read) streams of events, including continuous import/export of your data from other systems.
2. **Storing** streams of events durably and reliably for as long as you want.
3. **Processing** streams of events as they occur or retrospectively.

Producers are those client applications that publish (write) events to Kafka, and consumers are those that subscribe to (read and process) these events. In Kafka, producers and consumers are fully decoupled and agnostic of each other, which is a key design element to achieve high scalability. Producers never need to wait for consumers. Kafka provides various guarantees such as the ability to process events exactly-once.

Events are organized and durably stored in topics. A topic is similar to a folder in a file system, and the events are the files in that folder. Topics in Kafka are always multi-producer and multi-subscriber: a topic can have zero, one, or many producers that write events to it, as well as zero, one, or many consumers that subscribe to these events. Events in a topic can be read as often as needed—unlike traditional messaging systems, events are not deleted after consumption.

Topics are partitioned, meaning a topic is spread over a number of "buckets" located on different Kafka brokers. This distributed placement of data is very important for scalability because it allows client applications to both read and write the data from/to many brokers at the same time. When a new event is published to a topic, it is actually appended to one of the topic's partitions. Events with the same event key (e.g., a customer or vehicle ID) are written to the same partition, and Kafka guarantees that any consumer of a given topic-partition will always read that partition's events in exactly the same order as they were written.

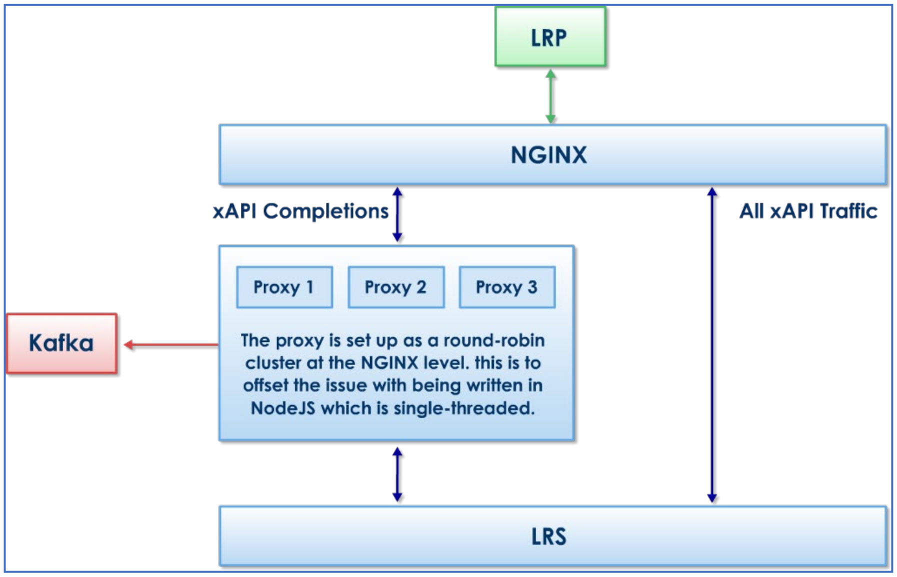

LRP
NGINX
xAPI Completions
All xAPI Traffic
Proxy 1
Proxy 2
Proxy 3
The proxy is set up as a round-robin cluster at the NGINX level. this is to offset the issue with being written in NodeJS which is single-threaded.
Kafka
LRS

Figure 4. Apache Kafka Proxy Arrangement. *This design enables a "parallel bus" of xAPI statements as described in the TLA main report.*

The communication arrangement is shown in Figure 4. To publish the LRS xAPI traffic to the Kafka cluster in real time, xAPI traffic passes through a Kafka-integrated proxy, which monitors both the original request and the LRS's response. Because the xAPI specification guarantees that the statement's MOM-relevant properties are immutable, and because the LRS response contains the assigned Universal Unique Identifiers (UUID) for each accepted statement, the proxy constructs a MOM-equivalent version of that statement without further LRS interaction. All accepted statements can then be transmitted in real-time to the Kafka cluster and subsequently to any component within the TLA network subscribing to xAPI traffic. The proxies are physically installed on the same virtual machine as the LRS.

Competency Management System

Description

A Competency Management System manages evidence of an individual's knowledge, skills, abilities, attributes, experiences, personality traits, and motivators to predict their value toward effective performance. Competencies might include skills in leadership or business, ethics, professionalism, or any number of topics related to a job. Within the Sandbox, the Activity Stream(s) stored in the Transactional LRS provide the evidence upon which competency assertions are made.

Using data from the Experience Index, the CMS sees a completion ("Person X completed Course XYZ"), makes a competency assertion ("Person X knows XYZ") and it makes another xAPI statement into the authoritative LRS ("Person X obtained XYZ competency"). The Experience Index serves as the authoritative source for course completions.

The Competency and Skills System (CaSS) platform was used to manage this evidence and infer competency against a competency framework. For the DAU Sandbox, CaSS used a competency framework for the Contractor position, mapped to content descriptors for a set of courses in the Moodle LMS to form the activity metadata.

Data Strategy

The competency data used in the Sandbox is held in a competency framework. A competency framework is used to define the elements, relationships, standards, contexts, and levels of mastery required to perform jobs and to certify the credentials used to define job placement.

The competency framework used in the Sandbox is built from the IEEE 1484.20.1 Reusable Competency Definitions (RCD). The RCD standard is the mathematical underpinning of the Sandbox's approach to competency-based learning. This standard describes the format for and relationships between the definition of a competency, the relationship to other competencies, and the alignment of evidence to help measure proficiency of the competency. The RCD standard uses linked data to define all aspects of a competency including key performance indicators, formal assessments, and other measures of proficiency.

The Contracting competency framework currently in use at DAU was ingested into CaSS for the Sandbox. This framework was mapped to 10 Contracting courses that were stored in the Moodle LMS. Under the CaSS configuration used in the Sandbox, a course completion is used as evidence that a competency has been achieved.

Learner Profile

Description

The Learner Profile is used to record learner credentials, competencies, completed courses, and other basic information. When sufficiently built out, with proper PII protections in a non-sandbox setting, these data can be shared at the enterprise level to provide a complete portrait of human performance.

The Learner Profile contains a set of RESTful APIs to create and manage user records. For the Sandbox, custom APIs were written to allow Sandbox components to manage the Learner Record. This included APIs for:

- Creating a Learner Record
- Editing a Learner Record
- Adding/Deleting a course
- Adding/Deleting a competency

A learner can be manually created and managed via the Learner Profile user interface. However, the main way a profile is updated is via data storage by CaSS.

CaSS acts on course completions it gets from the Transactional LRS, and then asserts competencies that it knows are related due to a mapping in the XI. When a course completion is received by CaSS it stores the course completion in the Learner Record and if there are competencies associated with the completed activity, the appropriate competencies are also stored in the Learner Record.

Data Strategy

The TLA Learner Profile in the Sandbox is an abbreviated version of DAU's Learner Profile.

Portal / Single Sign-On (SSO)

The portal used in the Sandbox allows role-based access to Administrator and Student tools. It provides access control of multiple related, yet independent, software systems. Users log in with a single ID and password to gain access to all TLA connected systems they are permitted to access. Keycloak is used for this purpose and has been integrated to protect learner information and to provide information security for learners, content, and organizations. This capability has been integrated to protect Personally Identifiable Information (PII) and to provide Operational Security for learners, content, and organizations if the system is ever duplicated outside of the Sandbox environment.

The SSO capability is integrated with the access portal, and the permissions are relayed to all components, preventing the user from having to log in several times. Logically, SSO is part of the back-end services (Identity Management), and the portal is an edge system, but they are a single installation in the Sandbox.

DAU / TLA Testing Report

Two different types of testing were conducted for the DAU Sandbox. The first testing type was directed at testing the functionality of the system, while the second tested specific performance requirements.

Functional Testing

The functional test was performed to determine if key functions were successfully performed once data flowed through the system. In this test, activity providers at the perimeter are intended to flow data to competency processing engines which award competencies and store that data in a representative Learner Record. Three principal functions were tested:

- Competency Awarding
- User Accessibility
- Data Integrity Verification

Testing procedures and the outcomes of each test are described below.

Testing Procedure

An activity provider script was utilized to send completion events, scored events, and view events to the Noisy LRS. This action represented the start of the data "pipeline." It was anticipated that after feeding data points to the LRS, competencies should be awarded in the expected manner.

The defined data set for the functional testing included:

- Completion events for 3 Test Courses
- Quiz Grade events for 3 Test Quizzes
- 5000 view events
- Misconfigured xAPI statements:
  - Missing object IRI Data Component
  - Invalid Timestamp Data Component
  - Missing mailto: Actor Identifier
  - Malformed JSON String
  - Extra Data Attribute

This data set was sent for test users in a variety of time frames, with delays in data, and all at once.

The defined competency configuration is contained in Table 3.

Table 3. Competency Configuration. *Object IRI and Competency links.*

Course	Object IRI	Competency Link
TC01	https://lms.azure.usalearning.net/mod/scorm/view.php?id=31	https://adlCaSS.usalearning.net/api/data/schema.CaSSproject.org.0.4.Competency/f773c304-f5ae-4a44-be97-0649d163a8c2
TC02	https://lms.azure.usalearning.net/mod/scorm/view.php?id=32	https://adlCaSS.usalearning.net/api/data/schema.CaSSproject.org.0.4.Competency/6826e249-e84c-4cc1-ae2c-ce2c3853ad42
TC03	https://lms.azure.usalearning.net/mod/scorm/view.php?id=33	https://adlCaSS.usalearning.net/api/data/schema.CaSSproject.org.0.4.Competency/ca5ac3aa-9ab1-4d94-8aaf-1eba38c3622f
Parent Competency		https://adlCaSS.usalearning.net/api/data/schema.CaSSproject.org.0.4.Competency/a5d8f3c1-2a9c-4408-a271-1fbcc11e190e

Defined Tests

Table 4 contains a listing of the tests performed and details about what the test entailed.

***Table 4. Defined tests.** List of tests conducted during the test period.*

Test Name	Test Details
Completions Sent	Send a Course Completion xAPI object to the Noisy LRS for identified Test Courses.
Competencies Awarded	Competencies for the identified Test Competencies should be awarded by the CaSS system based on the Course Completion event and check against the Experience Index.
Parent Competencies Awarded	In the CaSS system the Test Competencies should be part of a larger Parent Competency. Upon awarding of all sub-Competencies, the Parent Competency should also be awarded to the Test User.
Learner Record Updated	Awarded Competencies and Course Completions should be transmitted by the CaSS system to the Learner Record.
Scores Filtered Out	Send a Score xAPI object along with a score result. These are expected to be filtered out by the MOM Interpreter Script.
Views Filtered Out	Send a View xAPI object along with a viewed asset. These are expected to be filtered out by the MOM Interpreter Script.
Competitions Filtered In	Send a Course Completion xAPI object to the Noisy LRS. These are expected to be filtered downwards into other TLA components for processing.
Missing Object IRI	In the simulated xAPI statements, the "object"."id" data component definition (referred to as the Object IRI) is blank.
Missing mailto: Actor Identifier	In the simulated xAPI statements, the "actor"."mbox" data component definition is blank.
Malformed xAPI JSON	In the simulated xAPI statements, there is a "}" missing from the xAPI statement schema.
Extra Data Attribute	In the simulated xAPI statements, a spurious "type" attribute was added to the "object" data component. This goes against xAPI specifications.
Invalid Timestamp	In the simulated xAPI statements, the timestamp sent in is in the Unix timestamp format rather than the xAPI specification.

Results

Table 5 contains a list of tests that were conducted, the expected outcome of the test, the realized test outcome, and where the outcome resulted in a pass or fail designation.

***Table 5. Results.** Tests taken for each process with expected and test outcomes and pass/fail status.*

Test	Expected Outcome	Test Outcome	Pass/Fail
Data Processing			
Completions Sent	Completions ingested and processed by the Noisy LRS.	LRS ingested and processed completions.	Pass
Competencies Awarded	Completions pass through the system and result in the CaSS system checking against the XI system	Competencies were awarded correctly based on input data.	Pass

	to award a competency for a given completion.		
Parent Competencies Awarded	Test Competencies should be part of a Parent Competency. When sub-competencies are all awarded the Parent Competency should also be awarded.	Parent competency was awarded after all sub-competencies had been achieved.	Pass
Learner Record Updated	Awarded competencies, and course completions, should be transmitted to the Learner Record for viewing.	All competency assertions and course completions were viewable in the Learner Record.	Pass
Data Filtration			
Scores Filtered Out	Scores ingested and filtered out by the MOM Interpreter Script.	LRS ingested and filtered scored events to not be sent to other TLA components.	Pass
Views Filtered Out	Page views ingested and filtered out by the MOM Interpreter Script.	LRS ingested and filtered view events to not be sent to other TLA components.	Pass
Completions Filtered In	Completion events ingested and sent further into the TLA architecture for processing by the MOM Interpreter Script.	Completion events ingested and forwarded to the other TLA components.	Pass
Error Screening			
Missing Object IRI	Noisy LRS should reject and screen out xAPI statements with missing Object IRIs.	Noisy LRS rejected and screened out xAPI statements with missing Object IRIs.	Pass
Missing mailto: Actor Identifier	Noisy LRS should reject and screen out xAPI statements with missing mailto: Actor Identifiers.	Noisy LRS rejected and screened out xAPI statements with missing mailto: Actor Identifiers.	Pass
Malformed xAPI JSON	Noisy LRS should reject and screen out xAPI statements with malformed JSON.	Noisy LRS rejected and screened out xAPI statements with extra malformed JSON.	Pass

Extra Data Attribute	Noisy LRS should reject and screen out xAPI statements with extra unknown Data Attributes.	Noisy LRS rejected and screened out xAPI statements with extra unknown Data Attributes.	Pass
Invalid Timestamp	Noisy LRS should reject and screen out xAPI statements with invalid timestamps.	Noisy LRS rejected and screened out xAPI statements with invalid timestamps.	Pass

Results Analysis

All components functioned correctly and as anticipated. Completions were translated into competencies correctly, the parent competency was asserted after sub-competency completion was achieved, and all data were properly transmitted to the Learner Record.

Noisy LRS collection points and ingestion routines are functioning as intended and according to xAPI specifications: all incorrect xAPI transmissions were rejected; all data objects that were intended to be screened out as non-relevant to competency awarding were filtered out correctly; and course completions were ingested correctly and filtered down to the competency processing layer correctly.

Though the pool of actionable items is currently limited to recognition and acting on course completion events, all of the core pieces are in place and functioning correctly for a rapid expansion of the actionable items. The results of testing indicate that the foundation is strong, the pipeline is working and functioning as expected, and the system plan is solid.

LRS and LMS instances are highly scalable and support generation and ingestion of thousands of xAPI statements per second. During testing, the LRS with minimal configuration was able to ingest over 1,000 xAPI statements a second, which has shown to be scalable as resources are increased.

Note: When a user is first seen by the CaSS system there can be an up to 10-minute delay before that user is fully processed and sent over to the Learner Record. This currently seems to be a hard built-in limitation of CaSS related to data processing timeframes, rather than resource utilization. The delay may possibly be resolved with additional development.

Requirements Testing

Requirements testing was conducted using the accepted Requirements Traceability and Verification Matrix. A copy of this matrix is contained in Appendix A, Requirements Traceability and Verification Matrix. The copy in Appendix A includes the requirements tested for the DAU Sandbox as well as additional requirements that could be addressed in future iterations. These additional requirements are highlighted with a gray background in the tables in the Appendix.

Requirements testing was conducted in the following areas:

- Learning Records Stores
- Error Trapping
- xAPI
- Update Learner Competency CaSS

- Learner Profile
- Search Function CaSS
- Provide Config Control CaSS
- Compatibility Translation CaSS
- Credential Management
- Identity Management
- Portal
- Moodle LMS

The paragraphs that follow list the items tested, the testing procedure, the outcome of the test, and implications of each result.

LRS

These tests pertained to the various LRS instances that are utilized throughout the system for processing data.

Test ID #1:

Testing:

Log into the USALearning (USAL) LMS and generate training activity. Then log into the Noisy LRS associated with theUSAL LMS and verify content tracks have been sent with the Statement Viewer. This will indicate persistence due to the nature of the LRS database data storage.

Result: ***Pass***

Results Implications:

The USAL LMS is the only activity provider at the moment, and it has been verified to send all training related tracks to the Noisy LRS via the Logstore module.

Test ID #2:

Testing:

Log into the USAL LMS and generate training activity. Then log into the Noisy LRS associated with the USAL LMS and verify content tracks have been sent with the Statement Viewer.

Result: ***Pass***

Results Implications:

The USAL LMS is the only activity provider at the moment, and it has been verified to send all training related tracks to the Noisy LRS via the Logstore module.

Test ID #3:

Testing:

Simulated xAPI statements were sent to the Noisy LRS that had the following problems:

- Missing Object IRI Data Component
- Invalid Timestamp Data Component
- Malformed JSON String

- Missing mailto: Actor Identifier
- Extra Data Attribute

Result: ***Pass***

Results Implications:

The LRS successfully rejects all xAPI statements that do not meet the correct technical specification. The LRS can be counted on to reliably screen malformed data.

Test ID #4:

Testing:

Simulated xAPI statements were sent to the Noisy LRS that utilized an “actor” with an account object containing a UUI in the “name” field”. Learner Profile was checked for expected completion records indicating proper pipeline processing.

Result: ***Partial Pass***

Results Implications:

The LRS is able to successfully ingest xAPI statements with this configuration. The statements were correctly processed by the Noisy LRS and the Transactional LRS, however upon delivery to the CaSS the competency assertion and completion record transmission to the Authoritative LRS and Learner Profile. It appears as though the CaSS system is unable to process this type of statement.

Test ID #5:

Testing:

Log into the USAL LMS and generate training activity. Then log into the Noisy LRS associated with the USAL LMS and verify that the tracks are searchable with via the “Actor” in the Statement Viewer.

Result: ***Pass***

Results Implications:

The LRS is capable of filtering data based on data points such as “actor” or “verb”.

Test ID #6:

Testing:

Ensure that MOM statements, which are created by the MOM Interpreter Script by parsing xAPI statements, are reaching and being stored in the Transactional LRS.

Result: ***Pass***

Results Implications:

The Transactional LRS is capable of storing MOM statements for further processing and viewing.

Test ID #7:

Testing:

Ensure that competency and course completion statements from CaSS are transmitted to the Authoritative LRS.

Result: ***Pass***

Results Implications:

CaSS is able to successfully store competency events within the Authoritative LRS. The type of events being stored can be expanded with programmed CaSS business rules.

Test ID #8:

Testing:

Simulated xAPI statements were sent to the Noisy LRS and monitored at the other LRS's as the data went through the pipeline.

Result: ***Pass***

Results Implications:

The LRS provides tracks that allow a traceback from competency assertion to evidence. There are no qualifications or conferrals in this particular environment, but the environment does have traceability through all components back to the evidence.

Test ID #9:

Testing:

Ensure that statements from the USAL LMS are reaching its accompanying Noisy LRS. Each activity provider component should have its own Noisy LRS, but right now only the USAL LMS exists.

Result: ***Pass***

Results Implications:

The activity provider / Noisy LRS pairing architecture choice is acceptable for needs and works as intended.

Error Trapping

The following tests pertained to rejecting malformed statements and screening errors.

Test ID #1:

Testing:

Simulated xAPI statements were sent to the Noisy LRS that had the following problems:

- Missing mailto: Actor Identifier

Result: ***Pass***

Results Implications:

The LRS successfully rejects all xAPI statements that do not meet the correct technical specification. The LRS can be counted on to reliably screen malformed data. In this case the only non-valid user is a statement that does not have an attributed user at all.

Test ID #2:

Testing:

Simulated xAPI statements were sent to the Noisy LRS that had the following problems:

- Missing Object IRI Data Component
- Invalid Timestamp Data Component
- Malformed JSON String
- Missing mailto: Actor Identifier
- Extra Data Attribute

Result: ***Pass***

Results Implications:

The LRS successfully rejects all xAPI statements that do not meet the correct technical specification. The LRS can be counted on to reliably screen malformed data.

Test ID #3:

Testing:

Simulated xAPI statements were sent to the Noisy LRS endpoint that did not have LRS connection credentials.

Result: ***Pass***

Results Implications:

The LRS successfully rejected all messages that were not sent with a defined device username/password. Administrators are capable of registering a user for components to use to be able to send data into the system. Without this user access a component's transmitted data is rejected. This allows for Administrators to strictly control which devices are able to integrate with the system.

xAPI

These tests pertained to the implementation of xAPI data tracks and transmissions throughout the system.

Test ID #1:

Testing:

Log into the Transactional and Authoritative LRS and utilize the Statement Viewer to view tracks sent to confirm MOM xAPI profile conformant statements.

Result: ***Pass***

Results Implications:

Both the transactional and authoritative LRSs contain MOM xAPI profile-conformant statements.

Test ID #2:

Testing:

Log into the Authoritative LRS and utilize the Statement Viewer to view tracks sent by the CaSS system asserting competence.

Result: ***Pass***

Results Implications:

The Authoritative LRS receives tracks from the CaSS related to competency assertion. These tracks are stored in the LRS complete with timestamp and other identification information. The various LRS's throughout the TLA contain and preserve all of the data tracks that result in competency assertions.

Test ID #3:

Testing:

Log into the USAL LMS and complete a sample assessment built utilizing Moodle Assessments. Verify the information that arrives in the Noisy LRS via the Statement Viewer to verify it contains information about missed items.

Result: ***Fail***

Results Implications:

The default Logstore xAPI plugin does track the cmi.interaction data (questions like fill-in, matching, multiple choice, etc.). However, there are a few issues:

1. The way the plugin applies a score in the interactions has a math issue. It has to do with the conversion of raw to scaled score. The result is everyone fails all assessments no matter how they actually perform.
2. This is not a complete showstopper, but the xAPI representation of interaction data does not adhere to the most recent best practices or xAPI profiles. For example, the SCORM profile for xAPI says to use the "responded" verb when answering questions, but the default plugin uses "answered". As a result, any system rendering graphs or filtering for cmi.interactions would not find any due to the verb mismatch.
3. Some interaction types were not supported -- like "drag and drop", this results in missing or incomplete data if those question types are utilized.
4. There was a bug where in some cases the wrong data type was used for the learner_response. This resulted in the statement being invalid and rejected/not store. That also creates a situation where there is incomplete data. This applied to multiple choice and numerical question types.

A modified custom version of this plugin exists for Navy, which could be considered and developed as a potential replacement for the open source community plugin.

Experience Index

These tests pertained to the Experience Index component.

Test ID #1:

Testing:

Log into the Experience Index and attempt to create a new course experience object with a content URL of a test SCORM package. Verify that it can be associated with additional competencies.

Result: ***Pass***

Results Implications:

The Experience Index allows for anything with a URL to be linked to the experience object. As long as the data tracks are submitted with a URL associated with an experience object, data from items like simulators, LMSs, readers, or mobile devices would be able to be processed into competencies.

Test ID #2:

Testing:

Log into the Experience Index and attempt to create a new course experience object. Verify that "Description" is a metadata item that can be attributed to this object.

Result: ***Pass***

Results Implications:

The Experience Index does allow for the definition of "Description" metadata for an experience object. This can be utilized to describe its educational purpose as intended.

Test ID #3:

Testing:

Log into the Experience Index and attempt to create a new course experience object. Verify that "Content URL" is a metadata item that can be attributed to this object.

Result: ***Pass***

Results Implications:

The Experience Index does allow for the definition of "Content URL" metadata for an experience object. This is what is meant to be utilized to link to xAPI object handles.

Test ID #4:

Testing:

Log into the Experience Index and attempt to create a new course experience object with a content URL of a test SCORM package. Verify that it can be associated with additional competencies.

Result: ***Pass***

Results Implications:

The Experience Index allows for anything with a URL to be linked to the experience object. In this case a single SCORM is able to be linked to an array of associated competencies.

Test ID #5:

Testing:

Log into the Experience Index and attempt to create a new course experience object with a content URL of a decomposable test SCORM package for one of the uniquely launchable portions of the experience. Create a new object for each uniquely launchable portion of the course. Verify each can be associated with competencies.

Result: ***Pass***

Results Implications:

The Experience Index allows for anything with a URL to be linked to the experience object. In this case a decomposable SCORM is able to be created as a series of experience objects, which can each be associated with an array of competencies.

Update Learner Competency CaSS

These tests pertained to CaSS assigning learners competency.

Test ID #1:

Testing:

Simulated completion events will be sent into to the Noisy LRS for processing by the pipeline. This should result in the CaSS determining competencies via Experience Index lookup and transmitting an award of competency to the Authoritative LRS and Learner Profile.

Result: ***Pass***

Results Implications:

The CaSS system determines competence based on business rules that contain assertion processing code. Currently only two business rules exist to assert competency, checking completion xAPI objects against the Experience Index and “Parent Competency” rollup (when a user completes all sub-competencies). Both business rules for determining minimum competence to receive award worked successfully. Additional business rules should be considered.

Test ID #2:

Testing:

Simulated completion events will be sent into to the Noisy LRS for processing by the pipeline. This should result in the CaSS determining competencies via Experience Index lookup and transmitting an award of competency to the Authoritative LRS and Learner Profile.

Result: ***Pass***

Results Implications:

The CaSS system determines competence based on business rules that contain assertion processing code. Currently only two business rules exist to assert competency, checking against the Experience Index, and “Parent Competency” rollup (when a user completes all sub-competencies). Both business rules for determining minimum competence to receive award worked successfully. Additional business rules should be considered.

Test ID #3:

Testing:

Simulated completion events will be sent into to the Noisy LRS for processing by the pipeline. This should result in the CaSS determining competencies via Experience Index lookup and transmitting an award of competency to the Authoritative LRS and Learner Profile.

Result: ***Pass***

Results Implications:

The CaSS system determines competence based on business rules that contain assertion processing code. Currently only two business rules exist to assert competency, checking against the Experience Index, and “Parent Competency” rollup (when a user completes all sub-competencies). Both business rules for determining minimum competence to receive award worked successfully. Additional business rules should be considered.

Learner Profile

These tests pertained to the Learner Record component. In the Sandbox, the Learner Record meets all the requirements of the Learner Profile.

Test ID #1:

Testing:

Manually inspect the data in the Learner Profile to ensure it meets the agreed upon subset of the Learner Profile draft spec.

Result: ***Pass***

Results Implications:

The Learner Profile data is modeled after the IEEE 1484.2 draft specification (spreadsheet) provided by ADL. The Learner Profile is able to store all data in this draft specification via the APIs and has user interfaces to edit a subset of this data specified by ADL.

Test ID #2:

Testing:

Log into the Learner Record component. Verify that user data is being added with an anonymization token in the Definition Identifier.

Result: ***Pass***

Results Implications:

The definition identifier data element is the anonymized token for each item sent to the Learner Record.

Test ID #3:

Testing:

Send simulated completion data into the competency processing pipeline via the Noisy LRS. Afterwards, log into Learner Record and view the record for the simulated user. Competencies that have been asserted should appear.

Result: ***Pass***

Results Implications:

The CaSS system does have a user access layer which allows a user to log in and edit, read, update, or delete data elements of a competency framework. Creating a competency while logged in locks the competency from being edited by other users. CaSS does support User Groups which allows for some degree of group configuration management and enabling groups of people to edit items. However, any competencies created by anonymous users are fully editable by any user (including other anonymous ones). Anonymous usage inherently weakens any configuration controls within CaSS.

Test ID #4:

Testing:

Send additional metadata to the Learner Record via the Learner Record API. Verify that this information has been stored in the database.

Result: ***Pass***

Results Implications:

The Learner Record API allows for the storing of all metadata outlined in the agreed upon Learner Profile specification. Not every one of these elements is viewable on the UI however, but all can be stored in the learner record backend.

Test ID #5:

Testing:

Send simulated completion data into the competency processing pipeline via the Noisy LRS. Afterwards, log into Learner Record and view the record for the simulated user. Competencies that have been asserted should appear. Competencies that have been voided should appear as “not held”.

Result: ***Pass***

Results Implications:

For each asserted competency the Learner Record states its current learner state (allowing for things like revocation of competency).

Test ID #6:

Testing:

Log into the Learner Record component. Verify that users are able to be created via the “Add New Learner Record” button. Verify that users are able to be deleted via the “Delete” button. Verify that competency assertions are editable or deletable via the buttons in the user record dropdown.

Result: ***Pass***

Results Implications:

Logged-in users are able to successfully add learner records manually, as well as edit and delete both full learner records and individual competencies. This UI functionality is basic and not scalable to large amounts of data due to current development lifecycle of component.

Search Function CaSS

These tests pertained to the search functionality of CaSS.

Test ID #1:

Testing:

Log into CaSS and see if you can search for competencies that have been tagged for a specific job.

Result: ***Pass***

Results Implications:

The CaSS system does allow for adding tags to competencies, and for searching on those tags. Using this functionality, you can tag competencies for a specific role, and then search on it. This may not be as robust a search functionality as needs to eventually exist.

Test ID #2:

Testing:

Log into CaSS and see if you can search for competencies that have been tagged for a parent competency.

Result: ***Pass***

Results Implications:

The CaSS system does allow for adding tags to competencies, and for searching on those tags. Using this functionality, you can tag sub-competencies with their parent competency, and then search on it. This may not be as robust a search functionality as needs to eventually exist.

Test ID #3:

Testing:

Log into CaSS and see if you can search for competencies that have been tagged for a specific mastery level.

Result: ***Pass***

Results Implications:

The CaSS system does allow for adding tags to competencies, and for searching on those tags. Using this functionality, you can tag competencies with their mastery level, and then search on it. Additionally, through the use of building different mastery frameworks made up of competencies for that mastery level, you can separate the mastery levels into searchable parent competencies. This may not be as robust a search functionality as needs to eventually exist.

Provide Config Control CaSS

These tests pertained to the accessibility of editing, creating, and modifying competency frameworks within the CaSS.

Test ID #1:

Testing:

Log into CaSS and see if you can edit, read, update, delete data elements of a competency framework.

Result: ***Pass*** *(with slight reservations)*

Results Implications:

The CaSS system does have a user access layer which allows a user to log in and edit, read, update, or delete data elements of a competency framework. Creating a competency while logged in locks the competency from being edited by other users. CaSS does support User Groups which allows for some degree of group configuration management and enabling groups of people to edit items. However, any competencies created by anonymous users are fully editable by any user (including other anonymous ones). Anonymous usage inherently weakens any configuration controls within CaSS.

Compatibility Translation CaSS

These tests pertained to framework creation features of the CaSS component.

Test ID #1:

Testing:

Log into CaSS and attempt to utilize the “Import a Framework” feature to import a sample Contracting competency framework. There is currently no “Export a Framework” functionality.

Result: ***Partial Pass***

Results Implications:

The sample framework imported successfully. However, there is no “Export a Framework” functionality, which means this test item is only half fulfilled.

Credential Management CaSS

These tests pertained to overall TLA component compliance to credential policies.

Test ID #1:

Testing:

All TLA activity providers should send data into the pipeline to be processed into updates for the Learner Profile.

Result: ***Pass***

Results Implications:

Simulated training within the USAL LMS shows that the Learner Profile is reactively updated by the CaSS in response to ingested data. Additional activity providers will need to be added in the future.

Decision Support Management

These tests pertained to the user experience.

Test ID #1:

Testing:

Verify that each component in the enclave is able to link and integrate with the Keycloak Single Sign On service. Keycloak supports various connection methods such as OIDC, SAML2, and OAUTH.

*Result: **Partial Pass***

Results Implications:

The following components were able to integrate with Keycloak:

- Portal
- LMS
- LRS
- Learner Record
- Experience Index

The following components were not able to integrate with Keycloak:

- CaSS
- Kafka (no user interface / GUI)

Test ID #2:

Testing:

Go to the portal SSO login page and visually inspect that it contains a classification text.

*Result: **Pass***

Results Implications:

The login page contains the information that the user is accessing a “U.S. Government (USG) Information System (IS) that is provided for USG-authorized use only.”

Test ID #3:

Testing:

Go to the portal SSO login page and visually inspect that it contains text consenting to be monitored.

*Result: **Pass***

Results Implications:

The login page contains the standard DISA STIG warning about accessing a US government system and how rights are reserved to capture and inspect all traffic and communications.

Test ID #4:

Testing:

Sign into the Portal and attempt to change user roles via the UI.

Result: ***Fail***

Results Implications:

This is not currently possible. The Portal was not designed with this functionality, and may represent a future development item. Users' roles are designated upon login. Changes to roles require the user to re-login.

Test ID #5:

Testing:

Attempt to assign a user role other than Administrator to and administrative user. It should be denied.

Result: ***Fail***

Results Implications:

It is possible for an Administrative user to hold other roles, and thus the potential for "not unique" administrative accounts to exist. However, this is most frequently a designated "client" access control responsibility. It is on the client utilizing the system to ensure their administrative users are given distinct user/admin accounts.

Test ID #6:

Testing:

Go to the Portal and log in as a test student user. Then log in as a test administrator. The Portal should filter the available service components by the logged in user role. This should entail making components inaccessible to the student, but fully accessible to the administrator.

Result: ***Pass***

Results Implications:

The Portal successfully limited the available components to the student user to just the LMS and Learner Record. The administrator user was able to see all components. This equates to a limitation of access of Sandbox system resources related to user permission level.

Test ID #7:

Testing:

Go to the Portal and log in as a test student user. Then log in as a test administrator. The Portal should filter the available service components by the logged in user role. This should entail making components inaccessible to the student, but fully accessible to the administrator.

Result: ***Pass***

Results Implications:

The Portal successfully limited the available components to the student user to just the LMS and Learner Record. The administrator user was able to see all components.

Test ID #8:

Testing:

Go to the Portal and log in as a test student user. Then log in as a test administrator. The Portal should filter the available service components by the logged in user role. This should entail making components inaccessible to the student, but fully accessible to the administrator.

Result: ***Pass***

Results Implications:

The Portal successfully limited the available components to the student user to just the LMS and Learner Record. The administrator user was able to see all components. This results in limiting access to the data, as the Portal does not directly touch any data stored in other components.

Test ID #9:

Testing:

Go to the Portal and log in as a test student user. Then verify the portal displays a username or user information associated with the SSO token account.

Result: ***Pass***

Results Implications:

When logged in the Portal displays the first and last name associated with the SSO token. This could be changed to be any data associated with the SSO account.

Identity Management

These tests pertained to the role and permissions within the TLA system. In some cases, the results are inconsistent as not every component supports the full range of SSO and role/responsibility inheritance.

Test ID #1:

Testing:

Log into the TLA Portal with a user that has been defined as the “Administrator” role in Keycloak. This user should be able to see additional components.

Result: ***Pass***

Results Implications:

The Keycloak Single Sign On provides a Role structure capability. For the purposes of the TLA Sandbox, only Student and Administrator roles have been created or assigned. Role inheritance in other components is also not fully integrated; several components, including the Learner Record, CaSS, and Experience Index, are not mature enough to support true role-based permissions. However, the foundations for Role centralization exist and function.

Test ID #2:

Testing:

Log into the TLA Portal with a user that has been defined as the "Student" role in Keycloak. This user should be able to see standard student components.

Result: ***Pass***

Results Implications:

The Keycloak Single Sign On provides a Role structure capability. For the purposes of the TLA Sandbox, only Student and Administrator roles have been created or assigned. Role inheritance in other components is also not fully integrated; several components, including the Learner Record, CaSS, and Experience Index, are not mature enough to support true role-based permissions. However, the foundations for Role centralization exist and function.

Test ID #3:
Testing:

Include a way to select or search courses in components where courses are able to be scheduled.

Result: ***Fail***

Results Implications:

The USAL LMS does not support scheduled date searching; neither the Experience Index nor CaSS support any date related information at all.

Test ID #4:
Testing:

Launch both the Experience Index and the USAL LMS from the learner perspective. Verify the learner can launch experiences and training.

Result: ***Pass***

Results Implications:

The USAL LMS and Experience Index represent forward facing ways of a user discovering training. The Experience Index filtering is non-existent. The ability to curate lists needs to be expanded upon.

Test ID #5:
Testing:

From the learner perspective, launch the Learner Profile and verify that a learner can view their information.

Result: ***Pass***

Results Implications:

The data filtering on the Learner Profile should be developed to deliver a more "user personal" experience that makes their records clearer.

Test ID #6:
Testing:

Log into the TLA Portal with the Single Sign On, then attempt to access the Experience Index. If the user is a defined Administrator, they should see the Experience Index button and be able to auto log in.

Result: ***Pass***

Results Implications:

An “Experience Manager” role should be defined in Keycloak. This role should be able to see the Experience Index component in the Portal. The Experience Index needs a stronger role based system to be able to restrict the management of experiences.

Portal

These tests pertained to the TLA Portal, which was the centralized Single Sign On point. The Portal provides access to all TLA components capable of integrating with Keycloak.

Test ID #1:

Testing:

The TLA Portal provides the front face for the User Login. The User Login is powered by Keycloak. Any component that can integrate with Keycloak is capable of being integrated into the TLA Portal. Currently the following items are able to integrate with the TLA Portal and are accessible via the Main page in the Portal:

- USAL LMS
- USAL LRS
- Experience Index
- Learner Record

The following items are linked to on the TLA Portal home page, but do not support integration with Keycloak Single Sign On:

- Kafka (no User Interface)
- CaSS

Test ID #2:

Testing:

Keycloak is the chosen Single Sign On provider. Currently the following items are able to integrate with Keycloak:

- USAL LMS
- USAL LRS
- Experience Index
- Learner Record

The following items do not support integration with Keycloak Single Sign On:

- Kafka (no User Interface)
- CaSS

Results: ***Partial Pass***

Results Implications:

The majority of the components are able to integrate with Keycloak. Kafka does not have a User Interface and so has no way of integrating or reason for doing so. CaSS should be developed further to support OIDC, OAuth, or SAML login and integration to user accounts.

Test ID #3:

Testing:

The TLA system currently resides in a sandbox without security, and must only meet minimal security concerns related to data transmission and storage security.

Results: ***Pass***

Results Implications:

The core underlying critical security elements are in place. The system is capable of ensuring all data is encrypted in transit and storage. However, there are key failings for attempting to achieve a higher accreditation level (such as FedRAMP or DoD Impact Level). Several components are lacking key security features such as auditing of user/administrator actions.

Moodle

These tests pertained to the USAL LMS activity provider.

Test ID #1:

Testing:

Simulated xAPI statements were sent from the USAL LMS server during the test, as well as natural live data of a user navigating through courseware in the USAL LMS. Data from both sources reached and was processed by the Noisy LRS into the competency processing pipeline.

Results: ***Pass***

Results Implications:

USAL LMS with Logstore xAPI module is capable of being a perimeter activity provider. Any item that theUSAL LMS can host can be turned into something that can contribute to competency.

Test ID #2:

Testing:

The USAL LMS was hooked to the TLA Portal via Keycloak Single Sign On. Once a user logs into the TLA Portal the LMS button is available to click on and take the user to the LMS.

Results: ***Pass***

Results Implications:

USAL LMS is capable of integrating via OAuth, OIDC, or SAML to the Keycloak SSO. The USAL LMS can be a fully integrated component in the TLA architecture.

Operations in a Heterogeneous Environment or Multiple Enclaves

The Sandbox currently functions in an Azure environment. However, the architecture of the TLA lends itself well to the idea of a multiple enclave environment – for example, PERLS on the Amazon Web Services environment could be integrated at some point. Data generators are at the perimeter; in theory, they could be placed anywhere in any enclave and run by any organization. As long as they can speak the common language of xAPI these edge LRPs should be able to communicate to one of the leaf node Noisy LRSs.

The conversion by the LRS of xAPI statements into compatible MOM statements means that any activity provider able to transmit xAPI can have its output converted into MOM statements for analysis by other components. This is critical in that it allows content and systems to behave as intended using their appropriate xAPI profile (e.g. xAPI Video Profile). It can be difficult for content developers to modify default xAPI components to support the MOM Profile directly. As a result, a series of MOM Interpreter Scripts may be required for each Noisy LRS that supports a different xAPI profile. This moves the complexity of conversion to MOM away from the content developers to the system creators and architects.

Further, to support multiple enclaves, consistent object/activity and actor identification is critical. Although it may be possible to map identifiers in system components (e.g., mapping xAPI object IDs to GUIDs in the XI), this process can be complicated, time consuming, and error prone. During implementation of this single TLA enclave, the team determined that a single identifier should be used to identify a unique object as several IDs for the same user or object can significantly increase the complexity of a TLA environment. This approach can be more easily instrumented via xAPI profiles that detail the actor format and a catalog of object/activity IDs that are part of the profile itself. If ID mapping is expected in a future TLA environment, the mapping process, roles and responsibilities, and software product location(s) of the mapping must be documented in detail with sufficient examples, use cases, GUI tooling and error handling for end users.

PII and Security Concerns

The ADL Innovation Sandbox was created in an isolated environment, preventing any PII from entering the data stream from external sources. By its designation alone as a sandbox, Keycloak was implemented as a single sign-on resource, which affords protecting for PII in the future if/when it is introduced if the system is taken out of the sandbox environment.

CaSS presents a security concern, as all competency profiles not created by a logged in user are available for editing by any users to the system (even anonymous users). CaSS can be hardened to prevent unauthorized access to these profiles, but hardening was not done for this version of CaSS. DAU would need to harden CaSS if it were to be used in their environment to replace their CRMS, particularly in regard to user roles/responsibilities, auditing, and ability to integrate with their SSO solution.

The IEEE P1484 Learner Record standard PII security classification is likely Moderate. In order to avoid elevating to a High classification DAU should avoid transmitting and storing data that is extremely sensitive or highly confidential. This is generally not a concern with training related data, however training metadata related to extremely sensitive training may fall under this category.

Data Governance

Currently, users are able to be assigned to Student or Administrator roles within Keycloak. This governs which components they are able to access (so long as the component can be integrated into Keycloak). Current component integration is:

Component	Integrated to Keycloak Single Sign On
Portal	Yes
LMS	Yes
LRS (All instances)	Yes
Experience Index	Yes
Learner Record	Yes
Kafka	No
CaSS	No

Further capability to set boundaries for what users can and cannot do would be achieved through the creation of additional roles with established permissions specific to each role. This capability would be worth developing in client agencies that have their own role structure and structure for who should be able to access what component.

The largest obstacle to proper data governance within the Sandbox stems from shortcomings in CaSS and security oversight of the competency frameworks.

DAU Context

Technically, DAU does not own competencies. Their competencies are derived from the Human Capital Initiative, as well as functional area/career filed functional/governance groups. DAU maintains the competencies as well as the repositories used to maintain them. DAU does not share competencies with other services, but there are representatives from each service who are involved in the functional working groups that have a copy.

DAU limits access to competencies in CRMS through role-based read and edit capability in certain segments of the CRMS. Roles include administrator, center director, learning director, learning asset managers, product managers, etc. Normally the linkage of objectives (TLOS) to sub competencies can be done at learning director or learning asset center director level but uploading, revising, editing the competencies is restricted to admin roles or center directors. DAU authenticates using VPN to the DAU network so the user is automatically authenticated in, without having to plug in an additional username or password.

Data Labeling for Authoritative Sources

The system currently utilizes the user's email address as the unique identifier. This address is trackable throughout the system, and ties the various data elements together across various components. One chief concern with this approach is the frequency which people change emails either due to job shift, agency change, or name change. All data-generating components are linked into the system via access keys. These access keys allow all statements within the system to be linked back to which activity provider sent them. While the Sandbox operates using a SSO via Keycloak (which integrates with SAML, OIDC, etc. allowing for several integration methods) the real goal is to integrate it with a military Common Access Card (CAC) provider or client agency identity provider.

Recommendations Specific to Data Labeling for Authoritative Sources

During discussions with DAU they indicated that they use CAC Electronic Data Interchange Personnel Identifier (EDIPI) in their data warehouse to collate data elements. ADL should seriously consider how to achieve CAC integration. After CAC has been integrated, ADL should also consider storing/leveraging/making the authoritative data label out of the EDIPI, and specifically the first 10 characters of the EDIPI, as the last 6 characters are prone to changing. This guidance should be created for several components, but most critical, the Actor information should use an xAPI account and be included in xAPI profiles to be used by any TLA systems. A similar approach is taken by the Navy in the Naval Education and Training Command (NETC) xAPI profiles in draft at https://netc.usalearning.net/xapi-library/developer.html.

It is also important to recognize that content developers and typical system users may not understand the large number of details necessary to correctly identify users. To mitigate that issue, details on content launch should be included for at least:

1. Web-based content launched from an LMS,
2. Web-based content launched outside of an LMS, and
3. Non-web-based content

A launch mechanism allows content to be created without having to know how to form the user's identity, removing potential error from content development. This same technique can also be used for additional initialization of content data including the LRS endpoint to track to, the activity that is being tracked, additional context activities for the xAPI statement, and even user-defined extension values.

ADS is a web accessed (http://ads.msrr.dmso.mil) set of libraries (DMSO, Army, Navy, Air Force, MEL) of metadata on modeling and simulation (M&S) data and knowledge source. It provides general description, quality, and access information for each source. [DMSO 1039, AF 690, Army 755, Navy 1164]

TLA LRSs are the authoritative data sources for experiential data (evidence), MOM activity data, and competency assertions. LRSs receive data from Learning Record Providers (LRPs) via the standardized xAPI endpoints. However, in the DAU Sandbox environment, data can only be sent to LRSs from LRPs that have the appropriate access keys. Access keys are managed by the LRS administrator and are shared, out of band, with LRPs. Access keys are also associated with an xAPI authority. All xAPI data points have an associated authority, so it is possible to identify the LRP that sent the data or even a particular part of type of data from the LRP pending that specific configuration. For example, the test activity provider in the DAU Sandbox has its own access keys associated with an authority, so all test data can be identified, filtered, and visualized based on the authority. Any number of access keys can be created and distributed as necessary in a TLA environment.

In the DAU Sandbox, LRS scripts forward data to other LRS and components. For example, Noisy LRSs send "completions" to the Transactional LRS, and the Transaction LRS forwards this data to CASS. Then CASS sends "assertions" to the Authoritative LRS. All of the data forwarding is done via the standard xAPI endpoints so access keys and associated authorities are tied to these transactions. Thus, the xAPI data for evidence, MOM activity data, and competencies is authoritative and traceable to a particular source.

Further, xAPI data is immutable so a record of changes and voids over time persists for the lifecycleof the system.

Other DAU TLA components use the data from these LRSs, but the LRS is the primary source of truth. For example, the Learner Record includes a snapshot of the current state of a learner's competencies. However, this is a subset of the competency information available in the Authoritative LRS.

DAU Context

Users have access to many systems across DoD, so it is difficult for DAU to determine the most authoritative source, particularly since users register for these systems at different points in their career. A large challenge DAU has is determining the best source for authoritative data. For example, if they were to take a registration record from a student taking instructor-led training (ILT) and virtual instructor-led training (VILT), they would look at the source of ATRS registrations; conversely, if they are going to look at registration records for online training they would look at what was collected from the Cornerstone environment. The same sort of decision tree happens when collecting survey data from Qualtrics.

DAU looks at each new data source as it is presented and determines whether or not it has a good set of attributes that could be shared. As a general example, when a new system comes on board, generally each system has date component; DAU uses a common date dimension to commonize the data. A specific example of this situation would be DAU receiving data from ATRS or Cornerstone -- the date associated with demographic data from those sources determine if that data can be augmented to the student dimension or component dimension (army, navy, air force), or subcomponents (depending on data granularity). This process begins by looking at the level of granularity, and determining if it is going to be contradictory to other data sources that might be reporting in a very similar data structure on a student. Depending on whether or not students are allowed to update their data in that particular source drives whether it becomes more authoritative than a previous source. Each source needs to be decomposed to determine if it becomes more authoritative.

Hypothetically, DAU’s Single Sign On environment might allow for individuals to update their email address, so there will be more than one authoritative source. This means that it will have to be tracked by location and date, because an individual might have an opportunity to update their data from one or more authoritative places – DAU will downstream and feed that data from that secondary authoritative source to keep data in sync. It can be compounded if there are some sources that you don’t own and some sources that you do.

When DAU rolled out Okta in 2020, they decided to only migrate active users that have been to DAU within the past 2 years; this included 500,000 accounts that were rolled into DAU’s new SSO environment from its legacy environment. Around 20% of users did not receive an activation link to the new sign on, because in two years, over 100,000s email addresses changed. Users without an activation link cannot log in and report their changes. This means they will have to either call the help desk or, if they are a DoD member, use their CAC and fill out new form to change their email address.

Business Rules, Pattern Matching and Constraints

Business rules and pattern matching drive competency awarding and Learner Record updates. Business rules are written in CaSS for how to process competency assertions. At the time of testing, two business rules have been defined and implemented.

- **Experience Index Lookup** – The CaSS system checks an xAPI Object Internationalized Resource Identifier (IRI) in a completed statement against the Experience Index. If the Experience Index has a corresponding Object IRI, CaSS asserts competency for the competency that is tied to the object/activity in the Experience Index. At this time the Experience Index only supports defining a course completion competency criterion.
- **Parent Competency Rollup** – When a Parent Competency has been defined in CaSS that consists of sub-competencies, the CaSS system should award the Parent Competency when a user has completed all sub-competencies. This behavior is controlled through the Rollup property.

Currently the Experience Index does not contain any type of equivalency functionality.

DAU Context

DAU does not maintain on how equivalencies are defined, validated, and asserted – much the same circumstance that exists in the current sandbox. DAU lists approved experience items for 13 career fields, not including auditing, on the DAU iCatalog page: icatalog.dau.edu. Maintenance of the "people piece," though, is done strictly at the services level through the Director for Acquisition and Career Management (DACM) office or Director for Talent Management at the NAVY using their own system that validates if the person met the experience education and training piece. These services can determine equivalency relatively easily where training records are available. However, determining equivalency for experience is more of a gray area as the DACM must review a person's resume and justifications, in addition to jobs held before to meet those experience requirements. Experience requirements can be generic (for example, four years in a contracting position) and related systems have a career record brief so they can scan through and total the months. However, if there is an experience requirement that is not so straightforward (for example, the person should be in a program management office working on a three-year program performing specific duties) then a manual scan through a resume is required to determine if the person meets those requirements.

Experience requirements are usually vetted against functional governance teams and signed off by functional leaders responsible for career fields. The staff that approves the competencies are the same people who develop the experience requirements. The requirements do not necessarily tie to competencies; the courses link up and meet competencies but experience requirements do not necessarily line up correctly.

The current capabilities of the Experience Index currently do not meet the DAU requirements for being able to administer this type of pre-existing qualification requirement, or other physical evidence. The Experience Index could be improved to update both user functionality (students upload examples of qualification for certification by administrative staff), as well as administrative functionality (admins define, review, and approve user uploaded qualification evidence).

DAU does not award certificates themselves; they are awarded at the service level by DACM and DATM. This means that these two organizations will need to be integrated in some capacity into the future evidence review pipeline. One possible integration route may be DACM/DATM administrative access into whatever component is developed to support these technical requirements (likely the Experience Index or Learner Record). This solution is a more localized and secure solution as all data stays inside the established enclave. Another possible integration approach might be having the component provide an automated export functionality to be taken to a DAU component that DACM/DATM oversee. This solution is a more client-friendly solution as it allows keeping existing organizational policies/procedures in place, but poses potential data sharing approval challenges.

Lifecycle Maintenance

A thorough understanding of the entity responsible for any system's components is the primary driver of how that system is maintained. In terms of the DAU Sandbox, there are mature versions of vendor software, alpha versions of other components, open source components, non-standard items that were customized. There is no defined life cycle for the system as a whole. The individual components have a lifecycle, the Sandbox does not.

Maintenance of the TLA system should be approached from both the overall system perspective, as well as the "per component" perspective. In general, per-component maintenance will fall into three categories:

1. Package Management Update Routines: The underlying Operating Systems for each of the components will be handled via these routines. Either Windows Update or Linux Package manager should oversee the routine OS updates that host the component applications. These updates will be automatically managed by the package system and come from official vendor sources.
2. Configuration Managed Update Routines: These updates should be deployed via a configuration management tool such as Git. Typically, these will be code deployment in nature.
3. Container Managed Update Routines: These updates will be done via a container or executable deployment. Typically, these updates will replace the entire component (minus the database) and establish the new patched version in place of the old version.

Below is a chart describing how each component likely would be maintained.

Component	Likely Maintenance Methodology
Portal	Configuration Management – Git
LMS	Configuration Management – Git
LRS (All instances)	Container Managed – Executable
Learner Record	Configuration Management – Git
Experience Index	Container Managed – Docker Image
Kafka	Container Managed – Docker Image
CaSS	Container Managed – Docker Image
Keycloak	Configuration Management - Git
Operating Systems Underlying Applications	Package Management – Windows Update or Linux Package Manager

Analytics Visualizations

One critical component of a TLA environment is the visualization of system data. The TLA includes several systems based on standard (or to-be standard) data formats. Moving data and translating between the components promotes interoperability between these systems, but a larger impact of consistent data is through analytics and visualizations. Analytics and visualizations can be used by several planned roles in a TLA environment. These include, but are not limited to:

1. Student – Providing data about what the student has done, what needs to be done, comparing to the progress of other students, etc.
2. Instructor – Providing a view of a class to determine students falling behind, peer tutoring matches, etc.
3. Curriculum Designer – Providing information on the use and effectiveness of content for refinement over time.
4. Decision Maker – Providing a comprehensive view of a learning environment including the best performing schoolhouse, areas where an organization may have training gaps, areas to focus on future learning investments, etc.

In the DAU Sandbox, several systems include visualizations of the data format(s) they support. For example, CaSS has tools to view competency structures and the Learner Records includes a UI to view data elements from the 1484.2 Integrated Learner Record specification draft.

However, detailed analytics and visualizations are some of the main drivers of a TLA environment. In addition, many components in the TLA environment are backed with a specific xAPI data format, so the team moved forward with xAPI-base dashboards for the DAU Sandbox. Early in this project the Data Analytics and Visualization Environment (DAVE) software product was evaluated for potential fit. This included cross referencing ideas DAU had verses the capabilities of DAVE.

In several meetings with ADL it was determined that DAVE was missing key features and was not at a mature enough point in its lifecycle to support the DAU environment. This decision was reaffirmed by technical ADL staff.

At this time, several demonstrations of the USAL LRS dashboarding platform were performed to evaluate if this could be a sufficient alternative. After evaluating the following features, it was determined that the USAL LRS dashboards would serve as the analytics and visualization component for this phase of the DAU Sandbox:

1. Statement viewer and report generator
2. Report CSV download for additional analysis
3. Default overall dashboards
4. Dashboard editor including the ability to create bar charts, pie charts, time series, notifications, and lists
5. Dashboard and visualization embedding on other sites
6. Advanced analytics language for very specific analytics and dashboards not supported by default of in the editor UI
7. Plugin architecture to perform any other analytics or integrations required

The USAL LRS dashboard platform can also work at scale performing quick visualizations and analytics over hundreds of millions of records. Recently, the team performed a scale test at 1 billion statements.

LRS Setup Lessons Learned

The following sections describe lessons learned during the initial LRS setup in the Azure Innovation Sandbox TLA environment.

Activity Provider to MOM Data Conversion

Several of the components to push data system-to-system in the TLA did not exist or were prototypes. To handle this data manipulation, new scripts were created. This included the conversion of activity provider data to associated MOM statements and forwarding these statements to the Transactional LRS/Kafka.

Activity providers will likely not use the MOM profile by default and will likely vary in their xAPI profile usage (if they use a profile at all). So, custom logic for conversion may need to be applied for every activity provider or at least for every Noisy LRS. Depending on the complexity of the activity provider and the associated xAPI data, a detailed evaluation of the xAPI data may need to be performed in order to determine which activity provider events correlate with MOM events.

Moodle xAPI Configuration

Typically, organizations start with the Logstore xAPI Moodle plug-in to get xAPI data out of traditional LMS-housed content. The quality of xAPI data from Moodle varies based on the content and the content types and may not have a significant impact on associated MOM statements.

TLA implementation relying on Logstore xAPI may be better off to instrument individual content objects or updated Moodle content plugins for additional xAPI data resolution. Logstore xAPI is an open-source product and is often customized to create higher quality, more verbose xAPI data.

With regard to Activity Provider to MOM Data Conversion, it would save considerable time and effort if there was a set of profiles accepted for TLA activity providers. Standard scripts and/or documentation could be written to map activity provider events to MOM. This would ease TLA integrations and provide additional interoperability of like content types, even outside of a TLA environment. Further, analytics and visualizations based on Noisy LRS data would be easier to standardize and reuse.

Standard xAPI Profile Mapping Definitions

Also with Activity Provider to MOM Data Conversions, it would save considerable time and effort if there was a set of profiles accepted for TLA activity providers. Standard scripts and/or documentation could be written to map activity provider events to MOM. This would ease TLA integrations and provide additional interoperability of like content types, even outside of a TLA environment. Further, analytics and visualizations based on Noisy LRS data would be easier to standardize and reuse.

Additional Lessons Learned

The following sections describe lessons learned for remaining components as they were incorporated in the DAU Sandbox.

Portal

The template was easy to manipulate and straightforward. Building the portal as whole was straightforward, with implementing OAuth (Open Authentication) being the biggest hurdle. Integrating OAuth2.0 into anothersimilar template would be achieved with very little additional burden.

CaSS

CaSS is not a hardened and secure product. Any user can enter the CaSS URL in a web browser and navigate through the GUI. This allows anyone to create and manipulate both frameworks and the competencies within, without being logged in as an authenticated user. CaSS by default allows public access to the instance, which provides--along with concurrent editing--an easy place for a group to author frameworks. If a framework is created without being logged into a user account, the framework is public and may be edited by anyone. If the framework is created while logged into a user account, it belongs to that user and may be shared with additional users. Any logged-in user can (and the appropriate party "should") take ownership of a public framework once it is complete. Once ownership is taken, the framework may also remain publicly accessible (but not publicly editable) or private. If it is private, it would then only be accessible by privileged systems and users.

PII and Security

The implementation of the Keycloak Single Sign On approach allows for components to link back to a centralized account system. Wherever possible, components should establish a means to link to this SSO point to leverage its roles.

Internal communication channels should always be configured for HTTPS transmission to ensure security.

Lifecycle Maintenance

Prior to any accreditation effort a supply chain security review should be conducted on each component to review any external dependencies it may use in its internal development. An example of a bad finding on a supply chain security review might be finding that a software uses an abandoned or outdated dependency.

As part of the supply chain review, a much more comprehensive responsibilities matrix should be created outlining per-component vendor responsibilities and contact information. At the moment there is little specification about exactly who is responsible for each component, especially if unexpected issues arise. Also, each external vendor should be analyzed to make sure that their organizational interests are in line with the planned security posture, and that they are not owned by a foreign national organization.

For the overall system we recommend identifying monitoring approaches for each component that will allow a monitoring agent to determine if the component is online and healthy. Due to the integrated nature of the system, a breakdown of any one component potentially impacts the entire system. This may not be entirely true for perimeter activity providers, but for core internal competency assignment components an unexpected offline situation could break the entire pipeline. For example, USAL deploys a PING service for USAL LRS instances that both checks that the web server is running and that the xAPI endpoints are responding as appropriate. If either of these checks fail, support staff is immediately notified.

It is also advisable that any additions or updates undergo a Foreign-Owned Controlling Interest (FOCI) Review to prevent security breaches from unfriendly sources.

Recommended Next Steps

One of the greatest items holding the overall TLA system back from a performance and security perspective is the relative developmental immaturity of several of its components. We believe that the following items deserve specific attention during the next development cycle. Note that additional context-sensitive recommendations for DAU Sandbox components have been included in many of the sections above.

Horizontal Scalability and Federated Data Structures

Currently, the scalability of the system is mixed. Certain components are highly scalable (LMS, LRS, Keycloak, Portal), and certain components are not developed enough to be easily scalable (CaSS, Experience Index, Learner Record). Unfortunately, this leads to bottlenecks within the system that can only be partially alleviated through the use of the robust components.

The LRS is capable of filtering, storing, forwarding, and processing thousands of messages a second, and is highly scalable. Throughout the DAU Sandbox there are LRS instances which provide the ability to filter and gate data to the less scalable components, which provides some ability to scale the pipes of the system. However, the fact remains that critical core competency assertion components remain difficult to scale.

As this is a sandbox system, with limited system resources, and full load impact testing requires both more robust hardware as well as high usage costs, the exact limits of each component have yet to be determined. It would be worth a potential future round of load-specific testing to attempt to determine the limits of each component given a pre-defined set of "high/strong" system resources (ex: high CPU count, fast RAM, etc.). Determining and writing a detailed scalability plan for each component, and the overall system architecture would be a wise idea as well.

Current Key Functionality Expansion Points

The system as it exists now has a limited capability scope, but executes it completely and correctly. In essence, it provides a firm foundation for a learning architecture framework, and has several critical points that, if expanded, will vastly allow many more use-cases and improve the applicability of the system. Critical items that could see further expansion include:

- Experience Index – In its current state, the Experience Index is limited to representing course completion objects as competency requirements. This capability should be expanded to fill any types of "competency validation" checks that can be envisioned (ex: scored events with minimum grade settings).
- Noisy LRS Data Screens – Currently the Noisy LRS screens out all data that are not "course completion" events. This data screening is done because of the limitations of what the Experience Index can support from a competency validation perspective. However, as the system develops, this screen should be opened to support all the data types that allow interactions. Each activity provider component has its own Noisy LRS, which will allow individual screens to be established.
- CaSS Business Rules -- CaSS currently only two business rules to award competency exist: check against the Experience Index, or "rollup" Parent Competency assertion (user has completed all Sub-Competencies). This could be expanded to check against other systems or perform other business logic.

CaSS

- CaSS has weak data governance capabilities, which means that anonymous public users have access to competency profiles, including the ability to modify, delete, and add competencies. At a minimum this presents both data purging and denial of service attack vectors. User access restrictions should be put in place and enforced robustly.

- Business rules that govern competency awarding could be made more easily accessible. Currently raw code must be written in JavaScript files within the container image, with no accessibility to non-technically capable individuals.
- Business rules currently only include checks against the Experience Index, which only supports metadata competency mapping to Course Completion. These metadata mappings could be expanded to include at a minimum, score-related competency assertions, and custom in-person experiences.
- A strong role-based account functionality would allow CaSS to be more easily integrated with the Single Sign On portal architecture.
- CaSS should trim white spaces from the front of entered competency links to prevent matching errors due to copy/paste issues.

Learner Record

- The Learner Record should reach full maturity before it can be considered easily scalable.
- The Learner Record would benefit from having a strong role-based structuring capable of distinguishing between user level and administrative level users. ***Note***: This was a planned and approved limitation due to time constraints after the AFLSE learner record component was not available for this test instance.
- The Learner Record could be updated to include well designed user interfaces to display large sets of competencies or course completions.
- The Learner Record includes a set of information, specified by ADL, from the draft 1484.2 Integrated Learner Record specification. After initial views of the data are created, there are likely additional important elements that could be included in the Learner Record data setand potential updates to 1484.2 (e.g., a date of course completion would be valuable information in the record).

Experience Index

- The Experience Index would benefit from having strong role-based structuring capable of distinguishing between user level and administrative level users.
- The Experience Index could be updated to have the capability to display experiences other than course completions, as well as other critical items being identified via the iCatalog such as "Training/Experience" where the users are required to upload proof of training/experience to be approved by an approval officer, and grade related items.
- The Experience Index could be expanded to address the concept of "equivalency" and the ability to link experiences together.

Portal

- Based on conversations with DAU, the learner's education level could be a data point collected during user account creation.
- The Portal could be updated to include additional links to the dashboards.
- Developers could consider adding user-friendly ways to add components.

- Because the Portal was a custom-developed product for the DAU Sandbox, an installation document should be created that describes the method for adding new components in the Portal, and the technical requirements they must meet.

Dashboards

Basic dashboards could be created that hide verbs and only describes the information being presented (e.g., number of dropouts = launch - completions for an activity).

Additional Business Rules

The following business rules will enhance the utility of data stored within the DAU Sandbox.

- Score Competency – The Experience Index could be developed to support Grade/Score based objects. This would likely include a Required Scaled Score (0-1, inclusive) data element which CaSS could validate against if it receives a Grade event.
- Previous Experience / Required Training – The iCatalog utilized by DAU currently has competency requirements that allow a user to upload a piece of physical evidence to be validated by an approving officer. Example: Competency needs a bachelor's degree; user can upload a scan of their degree for approval. This business rule likely would need CaSS to check against the future component where this evidence is uploaded/approved, rather than the Experience Index.

Overall System

- An overall component architecture document should be written that clearly expresses the capabilities of each component, the intended usage, how it is managed/maintained, and the reasons for its inclusion in the architecture.
- An "Activity Provider Integration" document should be written that entails necessary steps to integrate a new activity provider component into the TLA.
- Several components do not audit adequately. The architecture will benefit if all components are capable of, at a minimum, auditing user and administrator access and actions, and outputting those audit logs to an audit file for collection and processing.
- Standards used for the four pillars in the Sandbox are in varying stages of formalization. System components cannot be versioned and maintained without breaking and then fixing connections because there are no finalized and approved standards. Standardized communication and techniques between components would further enable monitoring how these components communicate.

Final Thoughts

As the environment moves from a sandbox and development configuration into a more mature production ready posture, additional attention should be paid to security matters that impact running within an accredited enclave. Security measures such as strong role-based permissions, detailed auditing, and secure transmission and storage are vital to the system's ability to be leveraged among a wider secure community.

The establishment of a unified standard for activity providers, as well as a data ingestion and processing pipeline, means essentially that "if you can dream it you can build it." The limitations of the framework are only what actions the business rules are programmed to take in response to data that is passing through the pipeline.

The TLA Sandbox represents a robust and solid foundation that can be expanded in numerous ways to meet a huge array of learning and competency oversight needs. The word “Total” in Total Learning Architecture is not an understatement in this case. At its core, this system represents an “all encompassing” architected solution that, if expanded upon at critical joints, will allow the consumption and distillation of vast quantities of training data into insightful and meaningful responses.

Appendix A: Requirements Traceability and Verification Matrix

LRS

Test ID	Item	Currently Testable?	Test method	Test result
1	TLA compliant systems shall maintain a persistent storage of learning activity records (i.e. LRS)	YES	P/F	PASS
2	TLA compliant systems shall capture all xAPI statements generated from learning record providers	YES	P/F	PASS
3	TLA compliant systems shall ensure that xAPI statements are complete and well formed	YES	P/F	PASS
	TLA compliant systems shall provide a mechanism for administrators to purge old xAPI records	NO		
	TLA compliant systems shall maintain a record of purges to show that data has been altered	NO		
	TLA compliant systems shall provide a mechanism to ensure the integrity of xAPI data stored	NO		
4	TLA compliant systems shall allow storage of xAPI statements for the current UUID stored as actor	YES	P/F	PARTIAL PASS
5	TLA compliant systems shall allow use of filters on retrieving xAPI data by Actor (user, user interest group), date/time, activity type (object), verb, user specified extension field values	YES	P/F	PASS
	TLA compliant systems LRS shall support federated data storage, search, and retrieval between the noisy, transactional, and authoritative LRS	NO		
	The Authoritative LRS shall be able to federate data from transactional LRS located in multiple enclaves	NO		
	TLA compliant systems transactional LRS shall be sized to support a 10-year digital data retention store of all evidence	NO		
6	TLA compliant systems shall include a transactional LRS as part of core data that stores only data generated according to the TLA MOM profile.	YES	P/F	PASS
7	TLA compliant systems shall have an Authoritative LRS that stores digitally signed xAPI statements of “conferral”, “qualification” and “certification” for competency assertions.	YES	P/F	PASS

8	TLA compliant systems shall preserve the traceability between evidence, assertions, qualification/certification/conferrals and globally discoverable digital badges for credentials	YES	P/F	PASS
9	TLA compliant systems shall use "noisy" LRS to segregate data for device specific profiles	YES	P/F	PASS
	Noisy LRS profiles shall comply with IEEE P9274.2.1	NO		
	TLA compliant systems shall identify a "boundary" learning record provider that conforms to the TLA MOM for all edge devices generating learning evidence (operational data sources or learning activities)	NO		
	TLA compliant systems shall identify an Authoritative LRS for storage of conferred user credentials	NO		
10	The LRS shall comply with the server-side component of the xAPI specification (IEEE P9274)	YES	P/F	PASS

Learning Path Logic

Item	Currently Testable?
Learning Event Management service shall support courses of a single content resource, or multiple resources	NO
Learning Event Management service shall support default paths through multi-asset course or content set	NO
Learning Event Management service shall support user selected paths through a multi-asset course or content set	NO
Learning Event Management service shall be able to capture registered learning device that load or launch content	NO
Learning Event Management service shall generate a "captured" xAPI message when unscheduled experiences or courses are launched	NO

Learning Event Management service shall generate an "augmented" xAPI message if a selected goal is already a demonstrated competency	NO
Learning Event Management service shall be able to verify that an activity has closed out with completed, abandoned, or terminated	NO
Learning Event Management service shall generate the "abandoned" xAPI message after activity timeout	NO

Error Trapping

Test ID	Item	Currently Testable?	Test method	Test result
1	Learning Event Management service shall be able to identify incoming xAPI statement with an actor that is not a valid user, registered component, or identity group	YES	P/F	PASS
2	Learning Event Management service shall be able to identify if an incoming xAPI statement is not well formed	YES	P/F	PASS
3	Learning Event Management service shall be able to identify that an incoming xAPI statement is not from a registered device	YES	P/F	PASS
	Learning Event Management service shall be able to identify that an incoming xAPI statements references an invalid catalog item			
	Learning Event Management service shall generate an administrator alert if invalid xAPI statement is received			

Master Object Model

Test ID	Item	Currently Testable?	Test method	Test result
	TLA compliant enclave and federation shall be able to process learner state IAW the TLA MOM as received from edge devices			
	TLA compliant enclave and federation shall be able to process learner state IAW the TLA MOM as received from a user interface			
	TLA compliant enclave and federation shall be able to process learner state IAW the TLA MOM as detected from interaction with TLA data resources and services			
1	TLA compliant core services and data shall process performance evidence from actionable information IAW the TLA MOM	YES	P/F	PASS

xAPI

Test ID	Item	Currently Testable?	Test method	Test result
	The xAPI profiles of TLA compliant edge systems shall include templates for all learning content, activity, and experience types applicable to the federate instance	NO		
	The xAPI profiles of TLA compliant edge systems shall include a complete object life cycle (from requirement, to selection, launch, work, and closeout) for each training technology type	NO		
	TLA compliant edge systems shall use validated xAPI profiles	NO		
1	TLA compliant core systems shall use a validated TLA MOM xAPI profile	YES	P/F	PASS
2	TLA compliant systems xAPI profile shall include data elements required to audit evidence of assertions of competence	YES	P/F	PASS
	TLA compliant systems xAPI profile shall include data elements to specify context under which an assessment was evaluated	NO		
3	TLA compliant systems xAPI profile shall include data elements to specify areas not achieved during exams (i.e. grade<100%, what was missed?)	YES	P/F	FAIL

	TLA compliant systems shall include verification against the profile conformance suite as part of enclave deployment or update	NO		

Resource Validation

Item	Currently Testable?
Learning Event Management system shall be able to verify availability of resources prior to launching event	NO
Resources listed in the Experience Index shall include valid URL and available resources for web content, whether internal or external to the enclave	NO

Experience Index

Test ID	Item	Currently Testable?	Test method	Test result
	The Experience Index shall distinguish between data that is owned by an instance/enclave (authoritative source) and that which is copied to an instance/enclave from the authoritative source	NO		
	The Experience Index shall distinguish between formal course, supporting content (assets for course) and ancillary activities and content (not associated with a course)	NO		
1	The Experience Index shall include activities that are digitally instrumented contexts under which learning content or in situ tasks can be experienced (e.g. simulators, LMS, readers, mobile devices)	YES	P/F	PASS
	The Experience Index shall include content and its associated activity or activities in the form of digital assets that support the experience (e.g. eBooks, scenarios, SCORM and cmi5 packages, Portable learning device corpus)	NO		
	The Experience Index shall list applicable or allowable activities for use of content	NO		
2	The Experience Index shall include metadata for each activity, content and experience that describes its educational purpose as intended	YES	P/F	PASS

	The Experience Index shall include metadata for each activity, content and experience that describes its provenance and authority, its creation and version information, and its nomenclature	NO		
3	The Experience Index shall include metadata for each activity, content and experience that describes an object handle for use in xAPI statements	YES	P/F	PASS
	The Experience Index shall include metadata for each activity, content and experience that describes details regarding its modality, instructional style, and impact on learner cognitive or physical attributes such that two experiences otherwise labeled identically can be evaluated and prioritized for an individual learner	NO		
4	The Experience Index shall allow for the listing of a single SCORM/cmi package as a collection of associated competencies and metadata for a single experience	YES	P/F	PASS
5	The Experience Index shall allow for the listing of a decomposable SCORM/cmi package as a collection of associated competencies and metadata for each uniquely launchable portion of the experience	YES	P/F	PASS
	The Experience Index shall allow for the creation of a hierarchical course from any allowable combination of activities and content which have not been packaged using SCORM or cmi5	NO		
	The Experience Index shall also register applicable OJT/work experiences as activity types	NO		
	The Experience Index shall be able to list one or more other resources for an activity	NO		
	The Experience Index shall include ordered sets of subordinate activities and content as a course	NO		
	The Experience Index shall include ordered sets of subordinate activities and content as a user curated list	NO		
	OICS may create curated lists and direct all or a subset of their learners to complete the list	NO		
	The search function shall allow filtering and search of elements in the Experience Index by job, credential, competency defined at any level, level of mastery, activity type, authority	NO		

Activity Registry

Item	Currently Testable?
The Activity Registry shall be able to register a content management system internal to the TLA enclave as an experience	NO
The Activity Registry shall allow delisting of experiences from the Experience Index	NO
The Activity Registry shall allow content and Competency Management Service s to update metadata associated with content, activity, and experiences	NO

Update Learner Competency CaSS

Test ID	Item	Currently Testable?	Test method	Test result
1	The Competency Management Service shall generate assertions of competence based on evidence of mastery	YES	P/F	PASS
	The Competency Management Service shall maintain an evidentiary history of local training events/exercises attempted and completed, as well as scoring data	NO		
	Evidence of mastery shall include feeds from any instrumented digital learning device that can generate	NO		
	xAPI	NO		
	LRS shall federate data at TLA compliant core boundary by using TLA MOM verbs for Learner record provider state (equivalent to cmi5 states)	NO		
	The Competency Management Service shall process cascading evidence chains through associated competency frameworks (showing all competencies demonstrated by the evidence)	NO		
	The Competency Management Service shall be able to calculate progress toward a related credential as a sequence of demonstrated competencies	NO		

	The Competency Management Service shall be able to import learner Career state	NO		
	The Competency Management Service shall be able to calculate progress toward competencies not associated with a credential	NO		
2	The Competency Management Service shall determine when minimum evidentiary thresholds for demonstration/assertion of competency are achieved	YES	P/F	PASS
	The Competency Management Service shall ensure that achievement of a credential requires review and approval by an authorized approval authority	NO		
	The Competency Management Service shall evaluate the trust in evidence based on the life cycle defined in the TLA MOM (IEEE P9274.3.1)	NO		
3	The Competency Management Service shall update the Learner Profile on learner competency states		P/F	PASS
	The Competency Management Service shall generate an "assessed" xAPI message if a test activity is completed	NO		
	The Competency Management Service shall generate a "verified" xAPI message if an untrusted piece of evidence	NO		
	is separately approved by a trusted agent	NO		
	The competency system shall continuously update the state of assigned goals	NO		

Device Registration

Item	**Currently Testable?**
Registered devices shall include operational data sources or middleware systems (i.e. anything that will generate xAPI statements for the transactional LRS)	NO
Registered devices shall include content repositories or middleware (i.e. anything that will generate xAPI statements for the transactional LRS)	NO
Registered devices shall include learning management servers (i.e. anything that will generate xAPI statements for the transactional LRS)	NO

Registered devices shall include any other client device, computer, or middleware application that that will generate xAPI statements for the transactional LRS.	NO
Device registration shall include synchronization of device content to Experience Index object handles appropriate	NO
Connected devices shall support remote launching	NO

Learner Profile

Test ID	Item	Currently Testable?	Test method	Test result
1	The local Learner Profile shall be developed consistent with the TLA Learner Profile metamodel (Spec TBD)	YES	P/F	PASS
	The Learner Profile shall link back to an authoritative identity management service for PPI (personal data: name, rank, SSN, address, phone, UIC)	NO		
2	The Learner Profile shall use the internally generated anonymization token for storing user data	YES	P/F	PASS
3	The Learner Profile shall maintain a list of asserted competencies	YES	P/F	PASS
	The Learner Profile shall maintain a list of conferred credentials, CEU state, and effective dates	NO		
	The Learner Profile shall maintain a list of authorized access roles	NO		
4	The Learner Profile shall be able to store user specified attribute data defining learner preferences	YES	P/F	PASS
	The Learner Profile shall maintain a change log of updates to the profile	NO		
5	The Learner Profile shall store current learner state	YES	P/F	PASS
6	The Learner Profile shall allow for the creation, retrieval, update, and deletion of learner records	YES	P/F	PASS
	Deleted learner records shall be recoverable/auditable	NO		
	Individual Learner Profile records shall enable an administrator to conduct a full record purge after a specified period	NO		
	The Learner Profile shall maintain a mechanism to prevent hacking/loss of data integrity	NO		

	The Learner Profile shall integrate with the competency and Credential Management Services	NO		
	The Learner Profile shall maintain an auditable log of changes	NO		
	The Learner Profile shall allow for deletion of records from searches by an administrator	NO		
	The Learner Profile shall allow for "hiding" (non- permanent deletions) of user data from searches and displays by an administrator	NO		

Competency Framework CaSS

Test ID	Item	Currently Testable?	Test method	Test result
	Competency Frameworks shall be developed IAW IEEE 1484.20.1 RCD model	NO		
	The Competency Management Service shall store the knowledge, skills, abilities, and other (KSAO) behaviors required to perform a job or duty	NO		
	Each KSAO shall include relationships between competency definition objects and associated context/conditions and standards	NO		
	The context and standards under which competencies were acquired shall support determining fitness of the person for a specific job or employment	NO		
	The Competency Management Service shall define related competency objects (cognitive, psychomotor, affective, social, and metacognitive domains, standards, and context/conditions) at multiple levels of mastery	NO		
1	The Competency Management Service shall specify the competencies and level of mastery required for each job/duty	YES	P/F	PASS
2	The Competency Management Service shall be able to distinguish between qualification, proficiency, and mastery	YES	P/F	PASS
3	Credentials defined for a job/duty shall link to competency objects required to perform a job/duty	YES	P/F	PARTIAL PASS

Search Function CaSS

Test ID	Item	Currently Testable?	Test method	Test result
	The Competency Management Service shall allow a learner to search on all credentials associated with a job	NO		
1	The Competency Management Service shall allow a learner to search on all competencies associated with a job	YES	P/F	PASS
	The Competency Management Service shall allow a learner to search on all competencies associated with a credential	NO		
2	The Competency Management Service shall allow a learner to search on all sub-competencies associated with a competency	YES	P/F	PASS
	The Competency Management Service shall allow a learner to search on changes to competency framework by date	NO		
3	The Competency Management Service shall allow a learner to search by competencies with different levels of mastery (i.e. what jobs are associated with each level, and what standards and context applies)	YES	P/F	PASS
	The Competency Management Service shall generate a "clarified" xAPI message if a competency is selected that reinforces a recently completed experience that is not on the current goal-activity plan			
	The Competency Management Service shall generate a "augmented" xAPI message if a competency is selected that reinforces a recently completed competency that is			
	not on the current goal-activity plan			
	The Competency Management Service shall allow a learner to search on competency owner			
	The Competency Manager Service shall export search results as a serialized array of competency objects			
	The TLA Competency Management Service shall allow for searches of competency objects based on job, credential or as part of an unassociated top-level competency			

	The TLA competency search function shall return all lower level competency definition objects from a selected competency or credential			
	The TLA competency search function shall display the directed acyclic graph of relationships between competency definition objects			
	The TLA competency search function shall display supporting details for selected competency graphs			

Provide Config Control CaSS

Test ID	Item	Currently Testable?	Test method	Test result
1	The Competency Management Service shall allow authorized users to create, read, update, and delete elements of a competency framework	YES	P/F	PASS
	The Competency Management Service shall generate an alert when an element has been modified			
	The Competency Management Service shall maintain a record of changes (user, authority, name-value pairs)			

Compatibility Translation CaSS

Test ID	Item	Currently Testable?	Test method	Test result
	The Competency Management Service shall provide a mechanism to allow mapping of one competency framework to another			
	The Competency Management Service shall provide a mechanism to allow mapping of one credential framework to an equivalent credential			
	The Competency Management Service shall provide a mechanism to allow mapping of one credential framework to an equivalent credential with assigned experiences to close any gaps			

1	The Competency Management Service shall provide for import and export of competency framework data whole or in part	YES	P/F	PARTIAL PASS

Credential Management CaSS

Test ID	Item	Currently Testable?	Test method	Test result
	The Credential Management Service shall maintain an auditable log of trust and/or evidence that led to the credential			
	The Credential Management Service shall preserve a digitally signed badge showing the credential achieved, active date, conferral authority, and conferees name and service number			
	The Credential Management Service shall provide a validated digital export of the digitally signed badge			
	The Credential Management Service shall be able to assign user specified business rules for validating credentials to a user interest group (beyond assessments) to include source agency, military record, time in rate/job, assignment, multiple signature authorities			
	The Credential Management Service shall be able to generate non-repudiable alerts to OICS role users to establish required conferral and validation signatures			
	The Credential Management Service shall monitor achievement of CEU/PDU requirements and issue de- credentials or updates as necessary			
	The Credential Management Service shall provide a user configurable name for digital badges (e.g. diploma, certificate, badge)			
1	TLA compliant systems shall update the Learner Profile with all completed and in progress credentials for users	YES	P/F	PASS
	TLA compliant systems shall validate credentials required for a user acting in an OICS role for access, observation, or assessment			
	TLA compliant systems shall provide a secure digital badge for showing a credential has been conferred			

	TLA compliant systems shall provide an administrator configurable type for naming type of credential: (e.g. degree/diploma, badge, license, certificate, and professional rating)			
	The TLA Credential Management Service shall be able to export credentials using OpenBadge3			
	The TLA Credential Management Service shall preserve the chain of evidence between globally discoverable credentials, local copies of credentials, the assertions of underlying competencies, and the evidence gathered for the assertion.			
	The TLA credential chain of evidence shall be severable for purpose of transport or data federation (e.g. assertions sent without evidence in message payload, but still preserving discoverable links)			
	The underlying competencies that each credential represents will be defined using Credential Transparency Description Language (CTDL) and will reference the specific RCDs that each credential represents			

Decision Support Management

Test ID		Item	Currently Testable?	Test method	Test result
	General Requirements				
	General Requirements	The TLA User Interface shall provide decision support view of the collected experience data	NO	P/F	
	General Requirements	The decision support service shall enable search and filtering of data	NO	P/F	
	General Requirements	The decision support service shall enable retrieval across multiple transactional LRS (i.e. enterprise analytics)	NO	P/F	
	General Requirements	The decision support service shall be able to reconcile user identity across enclaves	NO	P/F	
	Instructor Review				

	Instructor Review	The decision support service shall enable analysis of efficacy of curriculum	NO	P/F	
	Instructor Review	The decision support service shall enable analysis of efficacy of assessments	NO	P/F	
	Instructor Review	The decision support service shall enable an analysis of learner performance distribution	NO	P/F	
	Instructor Review	The decision support service shall enable achievement velocity analysis by user interest group for OICS			
	Content Manager Review				
	Content Manager Review	The decision support service shall enable analysis of efficacy of supporting materials			
	Content Manager Review	The decision support service shall enable analysis of cost effectiveness of activities, content, and resources			
	Content Manager Review	The decision support service shall enable analysis of media suitability for training to a competency			
	Competency Management Review				
	Competency Management Review	The decision support service shall enable analysis of competency frameworks suitability for assigned jobs			
	Competency Management Review	The decision support service shall enable analysis of effectiveness of proficiency requirements for credentials			
	Competency Management Review	The decision support service shall enable analysis of robustness of credentialing processes			
	Personnel Manager Review				
	Personnel Manager Review	TLA compliant systems decision support shall enable analysis of workforce proficiency			
	Personnel	TLA compliant systems decision support shall enable			
	Manager Review	analysis of manning levels for projected job requirements			
	Personnel Manager Review	TLA compliant systems decision support shall enable analysis of facility and OIC manpower efficacy			

	Personnel Manager Review	TLA compliant systems decision support shall enable analysis of learner velocity through training pipeline			
	Personnel Manager Review	TLA compliant systems decision support shall enable analysis of proficiency duty cycle			
	Learner Decision Support				
	Learner Decision Support	TLA compliant systems decision support shall enable individual learning progression planning for current class/event			
	Learner Decision Support	TLA compliant systems decision support shall enable individual learning progression planning for current competency/badge/certificate/diploma goal			
	Learner Decision Support	TLA compliant systems decision support shall enable individual learning progression planning for next assignment goal			
	Learner Decision Support	TLA compliant systems decision support shall enable individual learning progression planning through current career arc			
	Learner Decision Support	TLA compliant systems decision support shall enable individual learning progression plans for service transition or change of career			
	Common Portal				
1	Common Portal	The portal shall employ single -sign on for all connected enclaved services	YES	P/F	PARTIAL PASS
2	Common Portal	The portal shall display an appropriate classification	YES	P/F	PASS
3	Common Portal	The portal shall display a consent to monitoring banner	YES	P/F	PASS
4	Common Portal	The portal shall allow a user to user to switch between allowable roles	YES	P/F	FAIL
5	Common Portal	The portal shall require a unique login for a user to act in the administrator role	YES	P/F	FAIL
	Common Portal	The portal shall support access to data and services at lower enclaves when MLS cross domain access is provided			

6	Common Portal	The portal shall enable access to Sandbox system resources applicable to user permission level	YES	P/F	PASS
	Common Portal	TLA system resources other than portal will only allow access by administrators			
7	Common Portal	The common portal shall filter all service access by user permission level	YES	P/F	PASS
8	Common Portal	The common portal shall filter all data access by user permission level and identity	YES	P/F	PASS
	Common Portal	The common portal shall allow interface with the alert and notification system			
	Common Portal	The common portal shall allow a user to select decision support dashboards			
	Common Portal	The common portal shall allow a user to select Learning Goal Management			
	Common Portal	Portal Learning Goal Management shall include goal selection and prioritization			
	Common Portal	Portal Learning Goal Management shall include goal and sub-goals path planning			
	Common Portal	The common portal shall allow a user to select Learning Task Management			
	Common Portal	Portal Learning Task Management shall include selection of pending and assigned tasks			
	Common Portal	Portal Learning Task Management shall include selection of assigned, shared, or created content set lists			
	Common Portal	The common portal shall allow a user to select Learning Event Planning			
	Common Portal	Portal User Management shall include group membership			
	Common Portal	Portal User Management shall include and CRUD functions for unprotected user data			

9	Common Portal	The common portal shall display username without maintaining an association to SSO token persistently in the local context	YES	P/F	PASS
	Common Portal	The common portal shall display current goals, tasks, suspense dates job, competency in work state, credential state, and identity group memberships			
	Common Portal	Identity and configuration settings shall pass to the client context without requiring reentry			
	Common Portal	The client interface and TLA planning interface shall exist as decoupled services			

Identity Management

Test ID		Item	Currently Testable?	Test method	Test result
	Roles and Permissions				
1	Roles and Permissions	TLA compliant systems shall enable login with administrator level privileges	YES	P/F	PASS
	Roles and Permissions	Administrator level permissions shall be able to access and modify user, content, service configuration, activity, resource, and competency service data	NO		
	Roles and Permissions	Administrator level permissions shall be able to assign Experience ownership to an OICS (for filtering purposes)	NO		
	Roles and Permissions	Administrator level permissions shall be able to assign competency frameworks or framework segments to a Competency Management Service	NO		
	Roles and Permissions	Administrator level permissions shall be able to create protected user identity groups with assigned users and assign access to these to OICS, competency, or content managers	NO		
	Roles and Permissions	Administrator privileges shall include CRUD permissions by segment for each of the data stores (Experience Index, LRS, Learner Profile)	NO		

2	Roles and Permissions	TLA compliant systems shall enable login with learner level privileges	YES	P/F	PASS
	Roles and Permissions	The learner access shall be able to select, deselect and prioritize goals (Jobs, credentials, or competencies)	NO		
3	Roles and Permissions	The learner access shall be able to select current scheduled courses	YES	P/F	FAIL
	Roles and Permissions	The learner access shall be able to manage (CRUD) curated experience lists	NO		
4	Roles and Permissions	The learner access shall allow for launching of current selected experiences, curated lists, or assigned courses	YES	P/F	PASS
	Roles and Permissions	The learner shall be able to search the Course Catalog	NO		
	Roles and Permissions	The learner shall be able to filter and search the entire local experience list	NO		
5	Roles and Permissions	The learner shall be able to view learner state information from the leaner profile	YES	P/F	PASS
	Roles and Permissions	The learner shall be able to review their personal performance data	NO		
	Roles and Permissions	TLA compliant systems shall enable login with OICS level privileges	NO		
	Roles and Permissions	OICS level permissions shall allow for logging observed practical exercises for assigned learners as complete-satisfactory, attempted, complete-unsatisfactory	NO		
	Roles and Permissions	OICS level permissions shall allow for reviewing progress toward goal, current grades, and state for assigned learners	NO		
	Roles and Permissions	OICS level permissions shall allow for review of assigned learner performance on assigned activities	NO		
	Roles and Permissions	OICS level permission shall allow for review of alerts and notifications sent to assigned learners	NO		
	Roles and Permissions	Sandbox systems shall enable login with Competency Management Service level privileges	NO		
	Roles and Permissions	The Competency Management Service shall be able to create, read, update, delete competency definition objects and relationships for assigned competency frameworks	NO		
	Roles and Permissions	The Competency Management Service shall be able to create, read, update, delete links between competency	NO		

		definition objects from the competency framework for each credential			
	Roles and Permissions	The Competency Management Service shall be able to create, read, update, delete job, duty, gigs, and competency frameworks	NO		
6	Roles and Permissions	Sandbox systems shall enable login with Experience manager level privileges	YES	P/F	PASS
	Roles and Permissions	User permission profiles shall be exportable to another federate instance of TLA compliant systems	NO		
	PPI/PII Protection/Privacy				
	PPI/PII	TLA compliant systems shall be able to create a locally unique anonymized identity reference	NO		
	Protection/ Privacy		NO		
	PPI/PII	The anonymized identity token shall be used to label "user" for all locally stored data	NO		
	Protection/ Privacy		NO		
	PPI/PII	TLA compliant systems shall otherwise use the anonymized reference when transmitting data referenced to users to another enclave	NO		
	Protection/ Privacy		NO		
	PPI/PII	UUID and anonymized reference keys shall be encrypted using FIPS	NO		
	Protection/ Privacy	140.2 compliant encryption or higher, as appropriate to classification level	NO		
	PPI/PII	Sensitive personal data (i.e. PPI) shall be only stored within or transmitted from the back-end identity management service	NO		
	Protection/		NO		
	Privacy		NO		
	PPI/PII	The portal shall utilize a FIPS 140.2 approved encryption of username to be displayed when received from Identity management services	NO		
	Protection/ Privacy		NO		
	PPI/PII	TLA compliant systems shall employ globally unique Identities for third party identification verification (UUID)	NO		
	Protection/ Privacy		NO		
	PPI/PII	The portal shall have mechanisms to prevent human readable linkage of username and UUID	NO		
	Protection/		NO		

	Privacy		NO		
	PPI/PII	TLA compliant systems shall be able to reconcile internal identity references with UUID	NO		
	Protection/ Privacy		NO		
	PPI/PII	The portal shall only display current name when used in the learner, admin, experience manager or Competency Management Service role	NO		
	Protection/ Privacy		NO		
	PPI/PII	The portal shall only display names for associated learners when used in the OICS role	NO		
	Protection/ Privacy		NO		
	PPI/PII	TLA compliant systems Shall be able to reconcile anonymized tokens in federated data structures (between organizations and between enclaves)	NO		
	Protection/ Privacy		NO		
	PPI/PII	TLA compliant systems shall enable configurable privacy settings at the individual datum value level	NO		
	Protection/		NO		
	Privacy		NO		
	PPI/PII	TLA compliant systems shall have a mechanism to filter data exports or visualization based on privacy settings	NO		
	Protection/ Privacy		NO		
	User Data				
	User Data	Identity management services shall be able to assign personal attribute data	NO		
	User Data	Identity management services shall be able to assign personas to a user	NO		
	User Data	Identity management services shall be able to assign privacy data to	NO		
		user records	NO		
	User Data	Identity management services shall be able to reconcile UUID to person identity in back-end services	NO		
	User Data	Identity management services shall be able to reconcile identity across enclaves (i.e. between different anonymization tokens)	NO		
	User Data	Identity management services shall be able to export a user record audit	NO		
	User Data	Identity management services shall be able to implement dynamic multi-factor authentication	NO		

	User Data	Identity management services shall be able to resolve internal identity tokens to a globally unique identity	NO		
	User Data	Identity management services shall integrate with privacy controls to prevent access to data based on locally managed policies	NO		
	User Data	The access policy manager shall include local, regional, and global business rules for data access	NO		
	User Data	User data shall incorporate proper encryption/decryption for identity tokens and personal data	NO		
	User Data	User data shall be resolvable between individuals and identity groups, and between multiple local identity tokens	NO		

Virtualization Services

Item	**Currently Testable?**
TLA compliant components shall utilize back-end services for dynamic endpoint management between components, data, and services.	NO
TLA compliant systems shall enable federated data services between enclaves	NO
TLA compliant systems shall leverage trusts between back-end identity management services	NO
TLA compliant systems shall have a configuration capability that registers service and data providers that operate within the enclave, to include back-end services and data portability between adjacent ecoservices.	NO
TLA compliant systems portal shall use a RESTful implementation to connect to enclave and federated data services	NO
TLA compliant systems shall provide a registration service for all enclave and federated data sources to manage URI blocks, permission holders, and path name/URL/IP for resources	NO

TLA compliant systems shall utilize mechanisms to dynamically track and update network and physical hosting of virtual private networks, computational resources, and containers in a	NO
contracted Platform as a service (cloud) environment	NO
TLA compliant systems shall verify core data and services (competency and Learner Profile, LRS/Learning event, management, experience catalog, Competency Management Service, competency framework, Learner Profile) are available to conduct training session	NO
TLA compliant systems shall have sufficient load balancing, failover, and redundancy to maintain Ao >98%	NO
TLA compliant systems shall have data backups to prevent loss of data even in event of core data or service failure	NO
TLA compliant systems shall have sufficient memory and storage resources to maintain 10 years of credential trust audit trail (i.e. preservation of reviews and awards)	NO
TLA compliant systems shall have sufficient memory and storage resources to maintain evidentiary records for competency in accordance with local regulations	NO
TLA compliant systems shall have sufficient memory and computational power for 90% peak duty cycle for 120% of projected user base	NO
TLA compliant systems shall have a security audit system that logs server down time, VM load shifting, attempted communication time outs, unauthorized users, or devices, and rejected xAPI statements	NO
TLA compliant systems shall implement NIST 800 controls for identity, access, zero trust device management, behavioral controls, and authentication.	NO

Portal

Test ID	Item	Currently Testable?	Test method	Test result
1	The portal shall enable single sign on for all subordinate services accessed through the portal	YES	P/F	PASS
2	TLA compliant systems shall use existing back-end services (e.g. LDAP/Active Directory) for identity management	YES	P/F	PARTIAL PASS
3	TLA compliant systems shall comply with cybersecurity policies for the installed enclave	YES	P/F	PASS

Moodle

Test ID	Item	Currently Testable?	Test method	Test result
1	Testing the LMS to LRS/Kafka stream pipeline	YES	P/F	PASS
2	Testing SSO for LMS via the Portal	YES	P/F	PASS