US DoD xAPI Profile Server Recommendations

31 August 2018

Megan Bowe & Aaron E. Silvers,

Data Interoperability Standards Consortium

Under contract #W911QY-16-C-0109 by

Advanced Distributed Learning

Learning Record Provider Professional Certification Recommendations

Executive Summary

This report offers objectives, requirements and recommendations related to the authoring, validation, governance and propagation of xAPI Profiles for US DoD. xAPI Profiles require a common way to be exchanged in order for semantic interoperability to work on a large scale. The exchange and management of profile data should be done through a federated network of profile servers. These xAPI Profile servers make it possible to author profiles, exchange public profile data between servers, and validate profiles to the profile specification. A learning record provider professional (LRPP) needs to be certified in the use of xAPI Profiles, both in the use of profile servers and the use of profile data in learning experiences, to enable plug-and-play interoperability at scale.

In order for a federated network of xAPI Profile servers to serve their purpose, there needs to be a way for an organization or group to create an xAPI Profile and share the information they authored without risk of losing data or outside groups changing the given xAPI Profile. This requires a governance model where a central authority informs profile servers of who should have access to edit which profiles. Such an authoritative body also needs to identify which profiles are the most appropriate in a given context, verifying the quality and substance of an xAPI Profile. This is required for tools reading profiles to know which profiles are most high priority to ingest and use.

Overview

The earliest draft of the Experience API (xAPI) specification included an embedded vocabulary. The embedded vocabulary consisted largely of verbs related to the Sharable Content Object Reference Model (SCORM), as the goals of xAPI were to address a myriad of challenges leveraging SCORM in modern technical architectures. In 2012, the community supporting xAPI decided it was necessary to remove such specific vocabulary from the specification and handle vocabulary outside the xAPI specification. This change enabled xAPI to support use cases across different industries and tools without modifying the core specification.

To accommodate this, vocabulary registries were built to allow access to terms. This allowed people to find vocabulary, but such registries have not provided sufficient supports to enable plug-and-play interoperability amongst xAPI tools. As implemented, ADL's Vocabulary Registry, the Rustici Software Vocabulary Registry and others require manual updates, and manual processes to keep the information synced across different sources. As well, there were no machine interfaces to incorporate the information these registries held. They were the right idea at the right time for the state of the xAPI Community in 2013, but these registries were not designed or developed to scale shared vocabulary usage, let alone evolve as a federated semantic platform.

To address the issues related to encouraging quality data at scale, the xAPI Profiles specification was completed by DISC on behalf of the Advanced Distributed Learning Initiative (ADL) in July 2017. xAPI Profiles provide both 1) a way to identify vocabulary and 2) guidance towards vocabulary usage in specific performance contexts. The xAPI Profile specification lays out structure where 'concepts' have statement templates which identify the required vocabulary for a statement describing a particular activity as well as patterns of statements which have particular meaning in the context of this concept.

The specification also includes a set of high-level requirements for a profile server. The specification requires that a profile server have the functionality to validate statement templates and statement patterns. Through interviews with US DoD stakeholders and requests from the larger xAPI Community, more business requirements have been identified that go beyond the technical requirements included in the xAPI Profiles specification.

This document addresses the current state of ADL's vocabulary server (Vocab Server). It defines a platform that supports US DoD and broader xAPI community needs related to xAPI Profiles. This paper also identifies what is needed, at a minimum, for a viable solution and offers a critical path to achieve the following goals.

- **Encourage a federated network of xAPI Profiles.** By developing and releasing an open source xAPI Profile Server that can network with other Profile Servers, along with needed governance, process, policy and support, ADL will provide the means for semantic interoperability while avoiding any one single point of failure which might harm xAPI's interoperability, especially for the US DoD.

- **Create a profile authoring and publishing system** which encourages profile creation by subject matter experts and professional organizations. With appropriate governance, process, policy and support, this enables trade groups and other vetted organizations to produce *verified* xAPI Profiles.

- **Enable Profile reuse** rather than writing a new profile for every need which arises.

- **Accelerate xAPI's interoperability to plug-and-play**, where the purchaser doesn't have to know all the details of xAPI to use it. Certifying the conformance of Learning Record Providers (LRPs), Learning Record Stores (LRSs), and Learning Record Consumers (LRCs) in their support of xAPI Profiles conformance establishes seamless semantic interoperability across products which can be easily identified as interoperable. This need was heard in conversations with Learning Record Provider Professionals, suppliers and vendors who have considered implementing xAPI Profiles but found there too be too little documentation and too much work to do by hand, making the barrier to entry too high for them to use xAPI Profiles in the immediate.

Current State

Vocab Server

Currently, xAPI Profiles are *somewhat* expressed on its Vocab Server[1], but as there are differences between "vocabulary" as previously described by ADL[2] and xAPI Profiles. On the back-end, the approaches to publishing and using linked data are the same, but there are some distinct differences.

xAPI Profiles have more requirements and metadata than the vocabulary currently available through the Vocab Server, the interface to ADL's ontology set. For example, ADL has a vocabulary *and* it also has an xAPI Profile for SCORM. The SCORM profile can use the ADL vocabulary and other vocabularies in addition to custom data elements, statement templates/patterns defined within the profile. Another example is Medbiquitous. They have a core vocabulary, but plan to build several different profiles based on their various medical scenarios or use cases. They have already built one profile called Virtual Patient[3].

[1] http://xapi.vocab.pub
[2] https://adl.gitbooks.io/companion-specification-for-xapi-vocabularies/content
[3] https://github.com/adlnet/xapi-authored-profiles/tree/master/virtual-patient

As it stands, the Vocab Server enables a user to look up vocabulary which has been defined, identified and used in xAPI statements. The Vocab Server supports the following use cases:

- Browse Vocabulary
- Search Vocabulary
- Read instructions how to write and contribute xAPI Profiles
- Link out to existing profiles and associated documentation.

xAPI Profile Specification

The xAPI Profile specification has two main parts. One section describes how a profile should be documented with metadata describing an xAPI Concept. Multiple xAPI Concepts make up one profile. This section of the xAPI Profile specification offers guidance on how vocabulary should linked together with metadata across different profiles.

The other section of the xAPI Profile specification details conformance requirements for minimal functionality from a profile server. This section requires that a profile server be able support:

- Statement and statement template validation[4]
- Statement pattern validation[5]
- SPARQL queries to learn[6]
    - Which profiles the server knows of?
    - What is in the profiles?
    - What vocabulary is available?

Future State

Community Requirements

Through a series of interviews with stakeholders from across the xAPI Community, requirements for an xAPI Profile server were gathered. This group included Learning Record Providers, Learning Record Stores, Learning Record Consumers and people working on xAPI implementations inside of large companies and universities. Throughout the interviews, stakeholders are aligned on how they currently work with xAPI Profiles:

1. People (a) go to the Vocab Server to search for terms and (b) discover if given terms have been defined. Upon determining if terms have been defined or not, people no longer

[4] https://github.com/adlnet/xapi-profiles/blob/master/xapi-profiles-communication.md#21-statement-template-validation
[5] https://github.com/adlnet/xapi-profiles/blob/master/xapi-profiles-communication.md#22-pattern-validation
[6] https://github.com/adlnet/xapi-profiles/blob/master/xapi-profiles-communication.md#13-example-sparql-queries

have use for the Vocab Server and leave.

2. People don't go to the Vocab Server to search for profiles which have been documented in GitHub. Either people already know about the profile in GitHub or they know one doesn't exist.

3. People, outside of the original authors, don't regularly use the profiles documented in GitHub because there is neither an easy way to see what they contain nor an automated way to make use of them.

Existing xAPI Profiles are handwritten into Github, which is prone to errors. xAPI Profiles are being created outside of the list of profiles on the Vocab Server, but there is no value for the profile creators in translating their profiles into xAPI Profiles' specified format; let alone making the xAPI Profile available on the Vocab Server. This is because:

a. **There is no validation.** Profile authors want feedback about how well their profile conforms to the spec. Examples:

    i. Riptide's HTML Courseware profile https://github.com/TryxAPI/xAPI-eLearning-Courseware-Profile

    ii. Internally developed profiles which contain information under NDA (multiple companies referenced these)

b. **The profile was authored before the xAPI Profile specification.** Without a way to share profiles in an automated way, xAPI Profile authors don't have a reason to put in the work to conform to the xAPI Profile specification until there is a profile server. Examples of this include:

    i. JISC's Learning Analytics service profiles: https://github.com/jiscdev/xapi

    ii. Open University of the Netherland's profiles: https://github.com/TrustedLA/xAPI-Dutch-Spec

    iii. The Connected Learning Analytics Toolkit from UTS Sydney: https://github.com/kirstykitto/CLAtoolkit

The target groups for creating xAPI Profiles are professional associations and product groups. These groups need an easy way to create and maintain xAPI Profiles which describe the skill development in their profession or the contextual usage of a given tool. Writing xAPI Profiles manually and managing them in Github present significant barriers to their participation.

Further, the xAPI Community lacks a review system for xAPI Profiles where professional association-created profiles are denoted in a way which allows tools to choose to use an "official" or "verified" profile over a similarly named xAPI Profile created by an amatuer or person outside an association. The lack of vetted profiles that are machine-recognizable as distinct from other contributed profiles is yet another barrier to adoption by vendors and solution providers.

As a result, the xAPI Community and xAPI's US DoD stakeholders are quite vocal and aligned about its needs for a profile server with a robust feature set addressing the following operational and implementation challenges:

1. *xAPI Profile Servers SHOULD be federated*, as opposed to a single, centralized approach. A single point of failure would put all of the tools leveraging xAPI Profiles at risk.

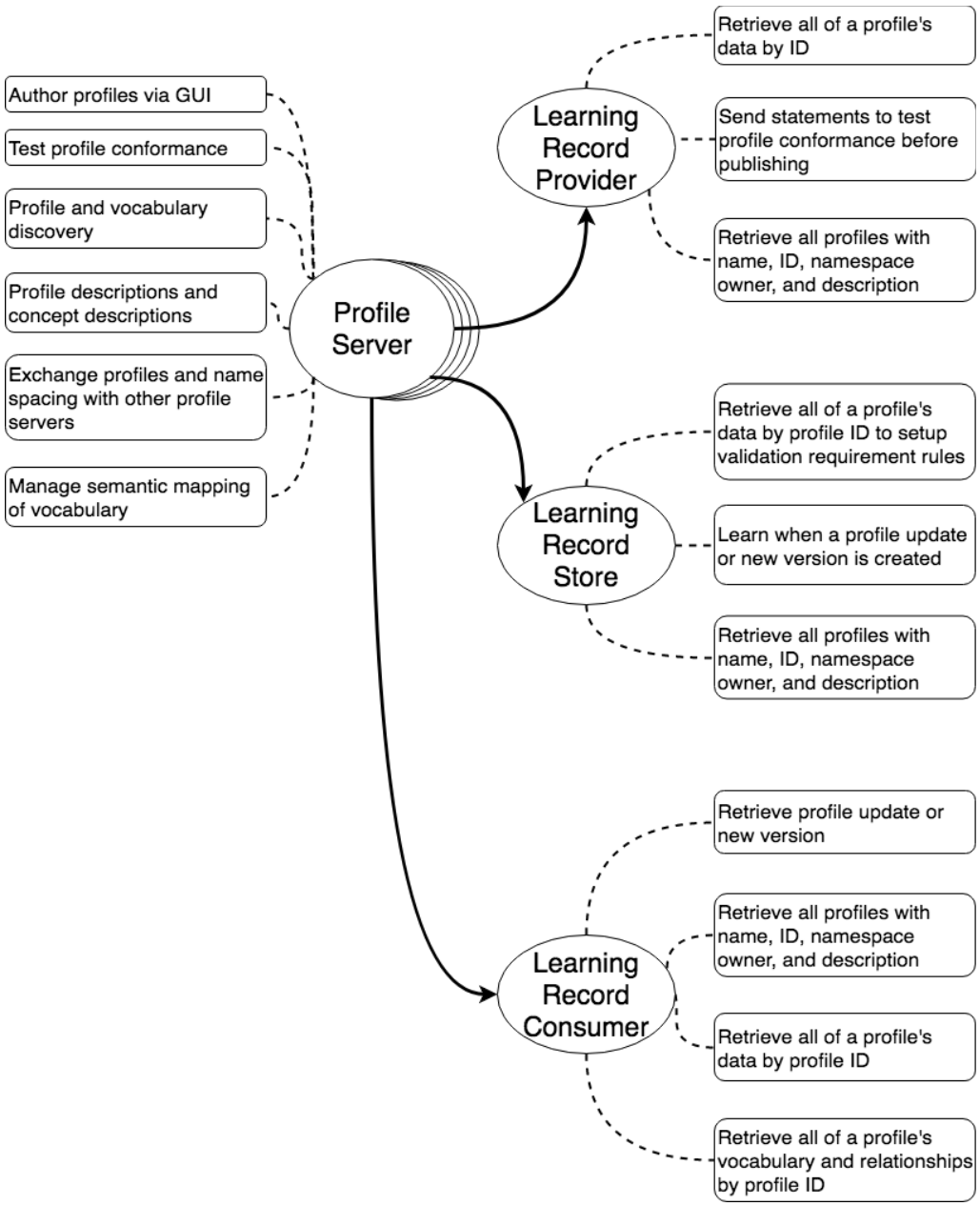
2. xAPI Profiles should not need to be manually loaded and maintained in different tools. To scale, *xAPI Profiles MUST be accessible by a tool at any time to enable automatic updates*.

3. *Learning Record Providing tools and products SHOULD be certified to xAPI Profiles*, which would signal plug-and-play interoperability to consumers.

4. *xAPI Profile Servers SHOULD support:*
    a. Code-free xAPI Profile authoring
    b. Delegation of xAPI Profile management authority with a central group maintaining permissions
    c. Open source code which enables exchange amongst a federated network of xAPI Profile Servers
    d. xAPI Profile and vocabulary discovery
    e. Semantic mapping of vocabulary
    f. Incoming statement validation to support testing of LRPs
    g. xAPI Profile management, versioning and maintenance of previous versions
    h. Creating additional context around xAPI Profiles to allow non-xAPI experts to make improved implementation decisions
    i. Backwards xAPI Profile construction (e.g. ingesting a statement stream and populating a profile based on what is in the statement stream)
    j. xAPI Profile verification (e.g. assigning an xAPI Profile as the defacto standard for a particular area).

Initial Proof Of Concept Structure

The following conceptual diagram describes how the above requirements can be satisfied in a minimally viable xAPI Profile Server application:



Design Requirements

XPS-1. A profile server shall store all data which is identified in the xAPI Profile specification.

XPS-2. A profile server shall capture profile data from user interface input and through data ingestion from other servers.

XPS-3. A profile server shall store links which identify any additional requirements of an xAPI profile which go beyond the profile specification.

XPS-4. A profile server shall store descriptive information about a profile.

XPS-5. A profile server shall grant users with namespace permissions to a profile access to edit and update all profiles which below to their namespace

XPS-6. A profile server shall check a central namespace registration authority before granting permissions for each session.

XPS-7. A profile server shall allow creation of new profiles

XPS-8. A profile server shall provide ways for users to understand a profile's intent and content from a high level perspective and from a detailed perspective.

XPS-9. A profile server shall identify profiles as public, private, public within a firewall (or other organizational boundary), draft, and submitted for verification.

XPS-10. A profile server shall identify profiles as verified or unverified through user input from the governing organization.

XPS-11. A profile server shall grant admin privileges to the governing organization allowing them to access to edit profile visibility and verification status.

Design Objectives

XPS-12. A profile server will provide interfaces and logic to support profile search and discovery of similar and related profiles.

XPS-13. A profile server will provide interfaces and logic to support vocabulary search and discovery of similar and related vocabulary.

XPS-14. A profile server will store competency ID data in relation to profiles, statement templates, and statement patterns.

XPS-15. A profile server will identify profile data from a set of statements and create statement templates and patterns.

XPS-16. A profile server will provide profile usage metrics about profiles to end users.

Functional Requirements

XPS-18. A profile server shall support SPARQL query language per the SPARQL specification https://www.w3.org/TR/sparql11-query/

XPS-19. A profile server shall provide a method of machine communication which transmits and receives public profile data and private profile data which is shared within a secure infrastructure.

XPS-20. A profile server shall provide a method of machine communication which allows tools to retrieve profile data from the server.
    1. A profile server shall provide a method of machine communication which allows tools to retrieve all known profiles on the server with descriptive information and metrics.
    2. A profile server shall provide a method of machine communication which allows tools to retrieve data from a specific profile by profile ID.
    3. A profile server shall provide a method of machine communication which allows tools to retrieve all known profiles and all of their data from the profile server.

XPS-21. A profile server shall provide a method of machine communication which pushes profile updates to tools which use profile data.

XPS-22. A profile server shall provide a method of machine communication which ingests profile data from an non-profile server source.

XPS-23. A profile server shall provide a method of machine communication which ingests vocabulary data and establishes relationships between vocabulary identified in the xAPI Profile Specification.

XPS-24. A profile server shall validate statement templates per the xAPI Profile Specification.

XPS-25. A profile server shall validate statement patterns per the xAPI Profile Specification.

XPS-26. A profile server shall validate complete profiles for conformance to the xAPI Profile Specification.

Functional Objectives

XPS-27. A profile server will provide a method of machine communication which ingests statement sets and parses the data into profiles.

XPS-28. A profile server will evaluate overall profiles for completeness and depth.

XPS-29. A profile server will track data to analyze profile usage.
    1. A profile server will track the date, time, and server where a profile was created
    2. A profile server will track the date, time, server, and user who made any update to a profile with identification of the update made and any data changed
    3. A profile server will track search activity on the profile server by user session
    4. A profile server will track browsing activity on the profile server by user session
    5. A profile server will track profile access activity on the profile server by user session.
    6. A profile server will track profile activity received from other profile servers including data, time, server, and activity.
    7. A profile server will track profile retrieval activity including data, time, tool, and type of retrieval.
    8. A profile server will track date, time, server and user who made any vocabulary additions or updates with identification of the update/addition made and any data changed

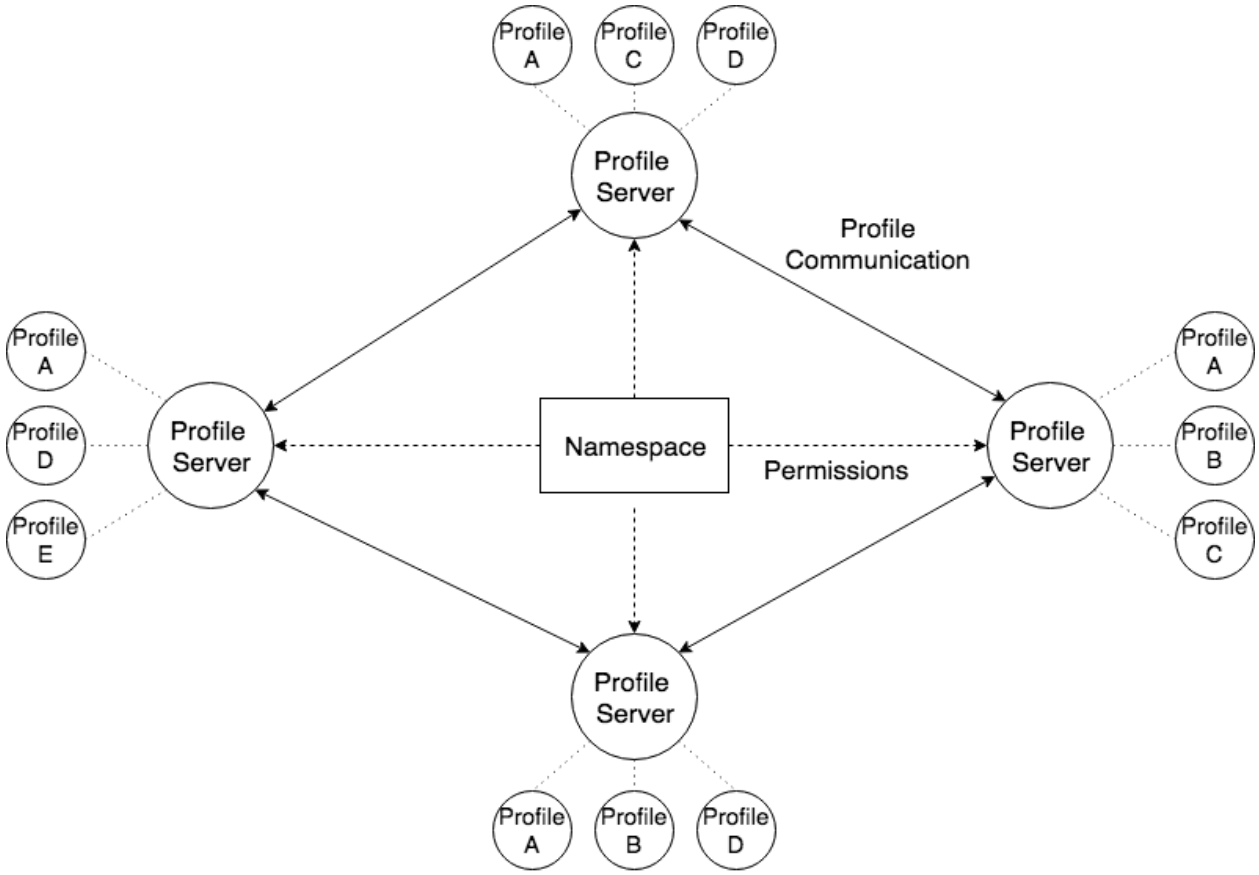
Advanced Functionality Objectives

XPS-30. A profile server will suggest vocabulary and profiles based on user activity and inputs
XPS-31. A profile server will visually display similarities and differences between a set of profiles or vocabulary
XPS-32. A profile server will suggest vocabulary to a user based on user activity and profile data inputted.
XPS-33. A profile server will support profile creation with automated feedback about profile conformance requirements and quality while the user is creating a profile
    1. A profile server will allow profiles to be created through direct entry or through a planned workflow which walks the user through the process with prompts.
XPS-34. A profile server will suggest similar statement patterns and templates which exist in other profiles
XPS-35. A profile server will use learn about connections between vocabulary items and automate matching for human approval.
XPS-36. A profile server will provide a look up for products certified to specific profiles.

Non-Functional Needs

Governance

The profile server network is intended to be built so that any public profile can be accessed by any profile server. A public profile must be owned by one organization. In order to give access to the right people on any given profile server a governing body needs to maintain an authoritative list of which profiles belong to which organization. In order to do this, an organization should register a namespace with the governing body and associate all of the profiles they create to that namespace. When a person from the organization wants to edit a profile, they log into any profile server with the credentials for their namespace and are granted access to the appropriate profiles. One community suggestion is that IEEE might be an appropriate governing body to host the namespace credentials which the profile servers reference. This was suggested because it is similar to IEEE currently running the servers where MAC addresses are stored.



Profile
_

A profile is a collection of JSON-LD data which describes which data should be in an xAPI statement or series of statements to properly describe and activity in the context which the profile addresses.

Profile Server

A profile server is a piece of software which stores, validates and shares information about a profile to tools which need to make use of the data in the profile.

Namespace

A namespace is a way to organize and identify those who have ownership of a profile's data. A group will need to govern namespacing in order to communicate permissions to profile servers.

Profile Verification

The profile server network is intended to be open to the public, which means that anyone can create a profile. There will be professional organizations and SMEs creating profiles as well as amateurs. In order for vendors to be able to manage the load of work which profiles will bring, they need a way to know which profiles are most important to focus on. A governing body must be available to indicate which profile is the most official source of data when there are multiple profiles covering similar topics. The profiles servers also need to be able to communicate which whether a profile is verified or not.

Verified profiles are the first step towards certifying tools as xAPI Profile conformant. An LRP, LRC, and LRS should be able to be certified to a particular profile (CMI5, for example). This enables DoD and private sector organizations to buy tools off the shelf which will offer plug and play interoperability when used together to create and deliver content with CMI5 (or any other verified profile).

Deltas

Current Vocab Server Versus xAPI Profile Server

The Vocab Server currently running at http://xapi.vocab.pub is an implementation of a standard SPARQL server. Of the thirty-six xAPI Profile Server (XPS) denoted design and functional objectives and requirements above, only the following are supported by the current Vocab Server.

Objectives

XPS-13. A profile server will provide interfaces and logic to support vocabulary search and discovery of similar and related vocabulary.

Requirements

XPS-18. A profile server shall support SPARQL query language per the SPARQL specification https://www.w3.org/TR/sparql11-query/

XPS-29.8 A profile server will track date, time, server and user who made any vocabulary additions or updates with identification of the update/addition made and any data changed

All other **XPS-** objectives and requirements identified earlier in this document are unique to an xAPI Profile Server. While the open-source code employed by the Vocab Server may well serve as a basis for an xAPI Profile Server, DISC recommends further design and development are necessary to meet the sum total of objectives and requirements in support of US DoD and the greater xAPI community.

Prototyping

Several groups (including JISC, HT2, Yet Analytics, and Riptide Software) have offered to host profile server prototypes for testing. When a prototype is ready, DISC recommends these groups be engaged to test the federated system and work out any issues before a formal release. There

are surely more people who would participate in this and a request should be put out at the proper time for additional volunteers.

Risk Mitigation

While the xAPI Profile specification was released in the first half of 2017 with the help of an engaged xAPI Community, the steadily growing momentum behind xAPI Profiles was interrupted through the second half of 2017. As a result, progress on the adoption and proliferation of xAPI Profiles stalled, suggesting multiple risk mitigation approaches are necessary corresponding to the development of an xAPI Profile Server to ensure the success of xAPI Profiles as an enterprise-class solution for scaling semantic interoperability of xAPI for the US DoD.

Lack of Adoption

If xAPI Profile servers are not adopted by a strong majority of xAPI users, the federated network will not be successful. The key to success indicator here is profile and vocabulary reuse. When terms, statement templates, and statement patterns are reused they will be more widely understood by tools across the community. If they are not reused and shared through profile servers we face the same challenges with semantic interoperability we face today. To mitigate the risk, the following strategies are recommended:

Marketing. The goal of this activity is to explain to people why they should be using the profile server to increase engagement.

- Explaining to organizations and government what a profile server is and why it matters.
- Making people aware that the profile server is an open source piece of technology which can help to make their use of xAPI more efficient.
- Publishing articles, producing webinars and other forms of media to make people aware of profiles and the availability of the xAPI Profile server tool.

Training. The goal of this activity is to explain to people how they should be using the profile server to increased usage. This makes it easier for people to work with profiles. When people can't use the tool or use it incorrectly, adoption of profiles will be stifled.

- Making it easy for a person to start using profile servers.
- Giving new xAPI Profile Server users the basics on how to set up, access, manage and find a profile.

- Making training available in the form of demos, step-by-step tutorials, establishing office hours for US DoD and other targeted audiences to accelerate people getting started.

Lack of Profiles

Profiles are intended to be very specific to the context of the learning experiences they are used within. In order to have a success network of profile servers, there needs to be a large number of profiles available so people can find profiles which match the context they are working within. If there isn't a matching profile, a person may choose to build their xAPI statement data free form and not set up a profile at all. This will not help to improve xAPI's semantic interoperability from where it is today. To mitigate the risk, the following strategies are recommended:

Profile Setup. The goal of this activity is to make the profile servers more attractive to the community because of the volume of data available. This initial setup is recommended to create a "starting lineup" of strong profiles for vendors to adopt which, in turn, further increases adoption of xAPI Profiles overall.

- Creating profiles which meet the needs of major government stakeholders and industry.

Profile Support. The goal of this activity is to get more profiles from standards bodies and professional organizations most capable of governing and maintaining a given profile.

- Supporting profiles created by the community would be necessary until the profile network is strong enough to be self sustaining. This would include supporting namespace set-ups and profile access.

- Answering questions, working with profile authors to create and manage profiles, hosting office hours, etc. This would allow the amount of profiles to grow without having to do all of the work to create them.

Recommendations for US DoD

DISC recommends the definition of a common xAPI Profile operating model for US DoD, used for system integrations via xAPI. The creation of specific policies and associated operational guidance would serve to entrench best practice as standard operating procedure (SOP).

Additionally, a National Initiative for Cybersecurity Careers and Studies (NICCS) xAPI Profile could support the Total Learning Architecture (TLA) and bring in xAPI Profile servers to leverage the semantic strengths of xAPI Profiles for the project.

Key Performance Indicators

Key Performance Indicators (KPIs) are necessary to gauge the success of a given initiative. With regard to xAPI Profile servers, DISC recommends the following KPIs to establish benchmarks. DISC recommends quarterly review of associated KPIs to allow for inclusion of new KPIs and the sunsetting of measures no longer of value to US DoD stakeholders.

When the following KPIs are met, technical confusion between parties who are at the level of the LRPP will be reduced. Less friction among teams working with xAPI will make it easier to efficiently and effectively leverage xAPI data at scale to drive improved learning and performance outcomes.

Publishing

- **Basic**
    - Profiles Created Over Time - count of profiles created over a specified time period compared to a prior time period. This will evidence growth in profile and profile server usage.
    - Time to Create a Complete Profile - a measurement of time from creation to completion in time actively spent working on the profile compared to time started and time completed. This will evidence usability of the profile server.

- **Advanced**
    - “Profile Quality” - this is a combination of measures including size and depth, as well as counts of count of new vocabulary versus connections to existing vocab and profiles. This will be a marker to describe the factors which indicate quality.

Usage

- Access and Retrieval per Profile Version - a count of the times a profile has been viewed for a significant amount of time and retrieved through and API or manually. This indicates how often a profile is used, which is a metric industry said they would look for in a profile before integrating it into their tools.

Server

- **Basic**
    - Time from Profile Creation to Discovery - a measurement of the time it takes for a profile to be found through search or navigation by a profile server user once it is created. This will show how well the server finds connections between profile data.

    - Profile Creation to Profile Views - a count of profile views over time from its creation. This indicates how a profile is accessed over time.

- **Advanced**
    - Connections to Other Servers - a count of how many servers a server talks to and how often profiles are exchanged. This indicates the health of the federated network.
    - Profile Update Speed - a measurement of time from when a profile is updated to when the updates are communicated to other servers. This indicates the speed of the network.

Appendix

xAPI Specification Definitions

Activity: A type of Object making up the "this" in "I did this"; it is something with which an Actor interacted. It can be a unit of instruction, experience, or performance that is to be tracked in meaningful combination with a Verb. Interpretation of Activity is broad, meaning that Activities can even be tangible objects such as a chair (real or virtual). In the Statement "Anna tried a cake recipe", the recipe constitutes the Activity in terms of the xAPI Statement. Other examples of Activities include a book, an e-learning course, a hike, or a meeting.

Activity Provider (AP): Now referred to as a Learning Record Provider. This change differentiates that the activity itself is not always the responsibility of software, rather just the tracking portion is.

Actor: An individual or group representation tracked using Statements performing an action within an Activity. Is the "I" in "I did this".

Application Programming Interface (API): A set of rules and standards created to allow access into a software application or tool.

Authentication: The concept of verifying identity. Authentication allows interactions between two "trusted" parties.

Authorization: The affordance of permissions based on role; the process of making one party "trusted" by another.

Client: Refers to any entity that might interact through requests. Some examples could be a Learning Record Provider, a Learning Record Consumer, a Learning Record Store (LRS), or a Learning Management System (LMS).

Community of Practice (CoP): A group of practitioners connected by a common cause, role or purpose, which operates in a common modality. CoPs are focused on implementing xAPI within a specific knowledge domain or use case. CoPs or independent developers, can create domain-specific vocabularies, profiles, and recipes. These practices usually involve work around defining use cases and curating the various vocabulary terms, synonyms, and other related metadata that might be

preferred within a CoP. They can also reuse existing vocabularies, profiles, and recipes already published by other CoPs or participants of the xAPI community.

Document Profile Resource: A construct where information about the learner or activity is kept, typically in name/document pairs that have meaning to an instructional system component. Not to be confused with Profile.

Endpoint: An entry point in a service-oriented-architecture. xAPI mirrors this approach with resources by defining the IRI from which communication takes place as an endpoint.

Experience API (xAPI): The collection of rules articulated in this document which determines how learning experiences are defined, formatted, and exchanged so that independent software programs can exchange and make use of this information.

Immutable: Adjective used to describe things which cannot be changed. With some exceptions, Statements in the xAPI are immutable. This ensures that when Statements are shared between LRSs, multiple copies of the Statement remain the same.

Internationalized Resource Identifier (IRI): A unique identifier which could be an IRL. Used to identify an object such as a verb, activity or activity type. Unlike URIs, IRIs can contain some characters outside of the ASCII character set in order to support international languages.

IRIs always include a scheme. This is not a requirement of this standard, but part of the definition of IRIs, per RFC 3987. What are sometimes called "relative IRIs" are not IRIs.

Internationalized Resource Locator (IRL): In the context of this document, an IRL is an IRI that when translated into a URI (per the IRI to URI rules), is a URL.

Inverse Functional Identifier: An identifier which is unique to a particular persona or Group.

Learning Experience: An event associated with learning. It is highly diverse as far as what it can be.

Examples include reading a book, taking an online course, going on a field trip, engaging in self-directed research, or receiving a certificate for a completed course.

Learning Management System (LMS): "A software package used to administer one or more courses to one or more learners. An LMS is typically a web-based system that allows learners to authenticate themselves, register for courses, complete courses and

take assessments" (Learning Systems Architecture Lab definition). An LMS in this document is used as an example of how a user is identified as "trusted" within a system and able to access its Learning Experiences.

Learning Record: An account of a learning experience that is formatted according to the rules of xAPI. A Learning Record takes on many forms, including Statements, documents, and their parts. This definition is intended to be all-inclusive.

Learning Record Consumer (LRC): An xAPI Client that accesses data from Learning Record Store(s) with the intent of processing the data, including interpretation, analysis, translation, dissemination, and aggregation.

Learning Record Provider (LRP): An xAPI Client that sends data to Learning Record Store(s). Often, the Learning Record Provider will create Learning Records while monitoring a learner as a part of a Learning Experience.

Learning Record Store (LRS): A server (i.e. system capable of receiving and processing web requests) that is responsible for receiving, storing, and providing access to Learning Records.

Metadata Consumer: A person, organization, software program or other thing that seeks to determine the meaning represented by an IRI used within this specification and/or retrieves metadata about an IRI. An LRS might or might not be a metadata consumer.

Metadata Provider: A person, organization, software program or other thing that coins IRIs to be used within this specification and/or hosts metadata about an IRI.

Persona: A set of one or more representations which defines an Actor uniquely. Conceptually, this is like having a "home email" and a "work email". Both are the same person, but have different data, associations, etc.

Profile: A specific set of rules and documentation for implementing xAPI in a particular context. Profiles generally provide a particular vocabulary of terms, some created specifically for the profile, and some are referenced from other vocabularies. Sometimes a profile might provide multiple vocabularies for different situations, and sometimes someone might curate a vocabulary from multiple sources without creating a profile. Not to be confused with Document Profile Resource.

Registration: An instance of an Actor experiencing a particular Activity.

Representational State Transfer (REST): An architecture for designing networked web services. It relies on HTTP methods and uses current web best practices.

Service: A software component responsible for one or more aspects of the distributed learning process. An LMS typically combines many services to design a complete learning experience.

Statement: A data structure showing evidence for any sort of experience or event which is to be tracked in xAPI as a Learning Record. A set of several Statements, each representing an event in time, might be used to track complete details about a learning experience.

Tin Can API (TCAPI): The previous name of the API defined in this document, often used in informal references to the Experience API.

Verb: Is the action being done by the Actor within the Activity within a Statement. A Verb represents the "did" in "I did this".

Vocabulary: A list or collection of the terms that are used by a COP for labeling or categorizing information in a particular domain. The use of a vocabulary ensures that everyone is using the same word to mean the same thing. For more information on vocabularies, see the xAPI Vocabulary Companion Specification.

xAPI Profile Specification Definitions

Absolute IRI: an IRI. Used in the JSON-LD specification to contrast with compact IRIs and relative IRIs.

Activity: an Experience API Activity. This specification helps Profile Authors mint canonical Activities.

Activity Definition: an Experience API Activity Definition.

Activity Type: an Experience API Activity Type. This specification helps Profile Authors provide additional metadata for Activity Types they control.

Attachment Usage Type: an Experience API Attachment Usage Type. This specification helps Profile Authors provide additional metadata for Attachment Usage Types they control.

Compact IRI: A shortened IRI in the form `prefix:name` that becomes expanded on processing to an absolute IRI. Described in the JSON-LD specification.

Concept: In SKOS, any unit of thought. In this specification, any of a particular list of possible things a Profile might describe.

Context: In xAPI this refers to a part of a Statement, but in this specification it usually means a JSON-LD `@context`, which is a way of mapping JSON onto semantic terms and RDF.

Document Resource: An Experience API Document Resource. This specification helps Profile Authors describe what Document Resources in particular locations need to look like.

Experience API (xAPI): The Experience API Specification. For this specification, any 1.0.x version is relevant. Describes data structures for describing experiences and APIs for communicating them.

Extension: An Experience API Extension. This specification helps Profile Authors describe what Extensions with specific identifiers need to look like.

IRI: An Internationalized Resource Identifier. Like a URL, but more general. A distributed, structured, persistent identifier.

JSON: JavaScript Object Notation. A simple way to represent data structures for computers that humans don't have too hard a time writing or reading. The way Profiles are represented in this specification.

JSON Schema: JSON Schema are a way to describe and constrain the form of JSON documents.

JSON-LD: JSON-LD turns JSON into Linked Data, making it easy to use with Linked Data tools and integrate with other datasets.

JSONPath: JSONPath provides a way to address parts of JSON documents using a `JSONPath expression`.

Language Map: A Language Map expresses multiple language-specific values at once. While used in the xAPI specification essentially identically, the controlling specification is JSON-LD Language Maps.

Learning Record Provider: As in the Experience API specification, anything creating Learning Records (xAPI Statements and Document Resources).

Media Type: A media type is a simple, structured way to refer to particular types of content. Also known as MIME Type or Content Type.

Pattern: One way a series of Statements following a Profile could look. Defined by this specification.

Profile: A profile is the human and/or machine-readable documentation of application-specific concepts, statement patterns, extensions, and statement templates used when implementing xAPI in a particular context. A profile is a way to talk about Concepts, Statement Templates, and Patterns for Experience API data in a particular context, and in particular to describe them so machines can do some processing automatically.

Profile Author: Some person or group writing a Profile.

Profile Server: A place to find, browse, and query Profiles, vocabulary concepts, and other profile metadata.

Profile Validator: Any person or machine attempting to verify if the rules in a Profile are followed for a particular set of data.

Profile Version: A Profile at a particular point in time.

PROV: PROV models the provenance of things. In this specification, PROV is used to provide rich versioning support.

RDF: The Resource Description Framework standardizes the exchange of semantic data by describing an information model of subject, predicate, and object.

Registration: An Experience API Registration. This specification uses registration to connect Statements that are following the same Pattern or Patterns.

SKOS: The Simple Knowledge Organization System provides the building blocks to describe and relate Concepts.

SPARQL: SPARQL is a query language for RDF data.

Statement: An Experience API Statement. The core unit of recorded data in the Experience API.

Statement Template: A set of rules for how Statements using certain Concepts should look. Defined by this specification.

StatementRef: An Experience API Statement Reference. Used for pointing at a second Statement from a first.

Subregistration: When multiple Patterns are being followed within a registration, subregistration is an extension specific to this specification to distinguish between them.

Verb: An Experience API Verb. This specification helps Profile Authors provide additional metadata about verbs they control.

xAPI Profile Processor Library: A programming library implementing the algorithms described in this specification.