

Automate E-learning Asset Management with OLSA Asset Integration - White Paper for SCORM 2.0 (.3)

This work is licensed under the Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Submitter

Shota Aki, Software Architect, SkillSoft PLC, Nashua, NH (<http://www.skillsoft.com>) SkillSoft PLC (Nasdaq: SKIL) is a leading provider of e-learning and performance support solutions for global enterprises, government, education and small to medium-sized businesses. SkillSoft enables business organizations to maximize business performance through a combination of comprehensive e-learning content, online information resources, flexible learning technologies and support services. SkillSoft currently has over 3,000 customers with over six-million registered users worldwide.

Abstract

SkillSoft's Open Learning Services Architecture (OLSA) Asset Integration is introduced as a solution for automating the management of the initial bulk export and periodic-updating of E-learning Assets from a Content Service Provider into a Learning Management System (LMS).

We propose to have OLSA Asset Integration adopted as part of SCORM 2.0.

References

This reference contains the latest specification of the OLSA Asset Integration Web Service.

[1] <http://www.letsj.org/letsj/download/attachments/327693/Asset+Integration+Public+Specification.pdf?version=1>

Introduction

SkillSoft's Open Learning Services Architecture (OLSA) Asset Integration is a Web Service that allows a Content Service Provider to automate the management of the initial bulk export and periodic-updating of E-learning Assets into a Learning Management System.

A Content Service Provider (CSP) is fairly recent term used to describe any entity that manages the distribution of content as a *service* to an LMS. Concepts such as Service-Oriented-Architecture, and Content-As-A-Service are embodied in a CSP implementation.

This API has been in production for over 3 years. A number of LMS vendors support it, including Saba and NetDimensions. SkillSoft is currently the only Content Service Provider supporting it. At this time there are more than 100 production deployments using this API. These are some of the stakeholders for this technology.

The OLSA Asset Integration specification has been presented to the AICC. SkillSoft has offered to submit this specification to their Content Service Architecture working group. This process is currently on-going.

OLSA Asset Integration is already based on existing standards where appropriate, some examples are:

- AICC and SCORM for content package formats
- SOAP for Web Services bindings
- WS-Security for authentication (<http://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf>)

E-learning assets in this context include traditional static courseware, dynamic content (content generated at run-time), and advanced content (content that can launch additional content).

In the OLSA paradigm, the physical content is not delivered to the LMS. Instead standard content package formats like AICC course structure files or SCORM 1.2 PIFs are used to describe assets. Embedded in the content package is an absolute launch URL that *references* back to the actual content hosted by the Content Service Provider. These light-weight content packages are called **Referral Objects**.

OLSA Asset Integration is agnostic with respect to the content package formats used to describe a given asset. AICC and SCORM are obvious choices to support. Other content package formats can be supported as well.

Figure 1 describes the system architecture. The LMS starts things off by issuing Asset Integration calls to retrieve Assets that the customer has purchased from the Content Service Provider. In response Referral Objects for the Assets are returned. The Referral Objects are installed in the LMS. The learners can now access and launch these Assets. However, since the Assets are actually physically hosted by the Content Service Provider, all launch requests turn into redirects back to the Content Service Provider.

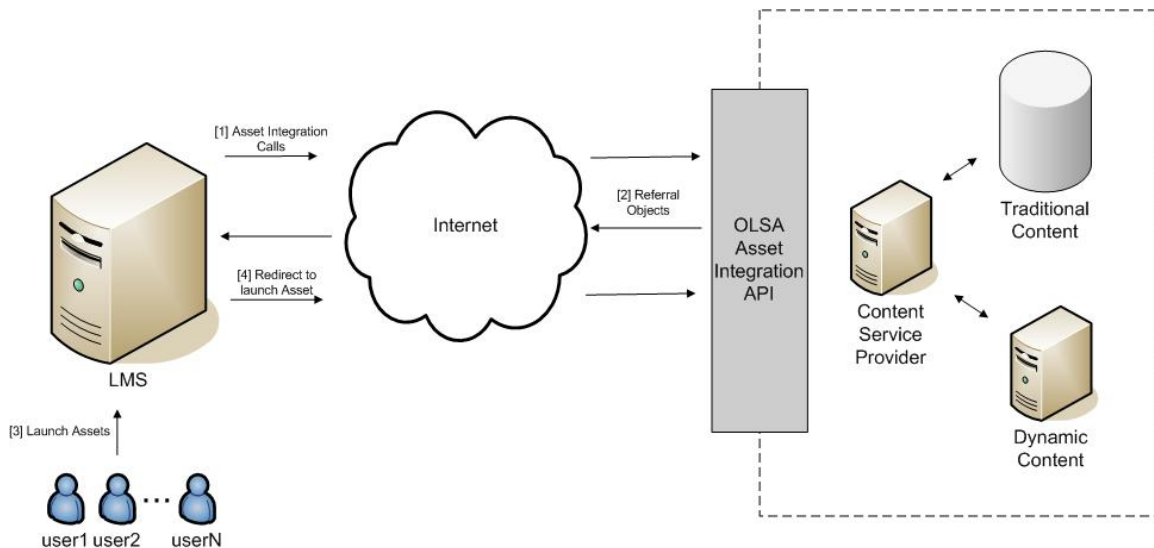


Figure 1: System Architecture

The business-process of managing the purchase of content (which can be manual or automated) is beyond the scope of OLSA Asset Integration. OLSA Asset Integration only assumes that the external business process results in a credential being made available to the customer to represent his entitlement. This **credential** encodes authentication (who the customer is) and authorization (what assets can be accessed) information. The credential will then be configured in the customer's LMS, which is then used by the LMS when issuing OLSA Web Service calls to a given Content Service Provider.

The term **entitlement** will be used to represent the assets purchased by the customer. A credential authenticates and authorizes access to an entitlement. An entitlement is not necessarily a fixed set of assets. Over the life of a business arrangement the assets making up a given entitlement may change.

Problem Definition

One of the major pain points for our industry is managing the process of exporting e-learning assets from a Content Service Provider to an LMS. Without solutions like OLSA Asset Integration, the customer must manually manage this process. This can be labor intensive and error prone. This process does not scale when 1000s of e-learning assets are involved.

The second pain point is maintenance and configuration of Players and other related technologies used to playback e-learning assets. Before OLSA Asset Integration, typically customers would physically host the content locally in their own networks. Whenever a content vendor updated a version of the player, the customer would be left with the task of correctly applying the update. If the player needed to be configured in a particular manner, again the customer would be left with performing this task. In the OLSA Asset Integration model, the content and course player(s) are hosted and managed by the Content Service Provider (i.e., content vendor). As a result, the customer is isolated from these maintenance and configuration details.

The last pain point addressed by OLSA Asset Integration is that some customers locally hosting their own LMS will not make it visible on the public Internet. To address this situation, OLSA Asset Integration is designed so that the LMS must initiate all Web Service requests. Hosting the Web Services API and making it visible to the LMS (e.g., the public Internet) should not be an issue for a Content Service Provider since it is their business goal to provide access to content-related-services to their myriad of customers.

All of the above pain points are magnified when the customer is dealing with many different content vendors.

Use cases

The use cases addressed by OLSA Asset Integration are described below.

Use-case-01: Perform Initial Bulk Load Of Assets	
Description	In this use-case, the assets purchased by a Customer from a Content Service Provider (CSP) must be exported from the CSP and loaded into the LMS. This is the initial bulk load of all purchased assets (aka entitlement) into the LMS.
Actors	LMS and Content Service Provider (CSP)
Assumptions	The customer has configured the LMS with the necessary credential to call the CSP. The LMS uses AICC, SCORM or any content package format that both parties support.
Steps	<ol style="list-style-type: none"> 1. The LMS issues a request to the CSP to retrieve all of the assets purchased by the given customer, at this given point in time. 2. The CSP processes the request and responds with a URL to a zip file containing the relevant Referral Objects. 3. The LMS processes the response by retrieving the zip file and iterating through the specified Referral Objects and installing each one. 4. The LMS acknowledges successful processing of the response to the Content Service Provider. 5. At this point, a learner may now login to the LMS and access and launch these retrieved assets.
Variations	This may not be the very first time for the LMS. The LMS may be going through this process again to resynchronize its catalog because due to some processing error on its part, it needs to start from scratch again against the CSP.

Use-case-02: Perform Periodic Updating Of Assets

Description

This use case happens after Use-case-01. After the initial bulk export, the customer would like any changes to his [entitlement](#) to be automatically reflected in his LMS over the life of the business arrangement with the Content Service Provider (CSP).

The LMS will query the CSP for any changes to the customer's entitlement. The LMS will query using a configurable interval like once a day, etc.

The type of changes that may occur are (with corresponding OLSA status values in parenthesis):

- A new asset is now available for the customer (entitled)
- An asset previously retrieved for the customer has been modified (modified)
- An asset previously retrieved for the customer is no longer available (not_entitled)

Changes will be relative to the most recently successfully executed use-case-01 or use-case-02.

The following example illustrates these concepts with events occurring at times t0, t1, etc.

t0: Use-case-01 successfully executed. In use-case-01, the LMS sees these initial status values from the CSP:

Asset C1 entitled
Asset C2 entitled
Asset C3 entitled

[After the LMS processes these changes, C1, C2 and C3 are accessible from LMS]

t1: The following changes occur on the CSP:

- New Asset C4 installed at the CSP
- existing asset C2 Modified at the CSP
- existing asset C3 Removed at the CSP

t2: Use-case-02 successfully executed. In use-case-02, the LMS sees these status values from the CSP:

Asset C4 entitled
Asset C2 modified
Asset C3 not_entitled

[After the LMS processes these changes, C1, C2 and C4 are accessible from LMS]

	<p>t3: The following changes occur on the CSP: - new Asset C5 installed at the CSP</p> <p>t4: Use-case-02 successfully executed. In use-case-02, the LMS sees these changes from the CSP: Asset C5 entitled [After the LMS processes these changes, C1, C2, C4 & C5 are accessible from LMS]</p> <p>Etc.</p>
Actors	LMS and Content Service Provider (CSP)
Assumptions	Same as use-case-01.
Steps	<ol style="list-style-type: none"> 1. The LMS issues a request to the CSP that it needs any changes to the customer's entitlement since the most recent successfully executed use-case-01 or use-case-02. 2. The CSP processes the request and responds with a URL to a zip file containing the relevant Referral Objects. Asset information is returned only for the 3 OLSA states: entitled, modified and not_entitled (as described above). If the status of an asset is not_entitled then its Referral Object is not delivered in this case. If none of the above 3 states apply then no information on an Asset is returned by the CSP. 3. The LMS processes the response by retrieving the zip file and iterating through the specified assets as follows: <ol style="list-style-type: none"> a. It installs all 'entitled' assets. b. It re-installs all 'modified' assets c. It uninstalls all 'not_entitled' assets 4. The LMS acknowledges successful processing of the response to the CSP. 5. At this point, all changes are now visible to the learner population. 6. The LMS waits for some specified interval (on the order of hours, days, etc), and repeat these steps.
Variations	If use-case-01 does not precede this use-case then the CSP will behave as if it is use-case-01 (i.e., it is the very first time and all assets returned as 'entitled').
Non-functional	Periodic updates occur at reasonable intervals after use-case-01. A reasonable interval depends on many factors but the expected order of

	magnitude is hours or days (not seconds or minutes).
--	--

Use-case-03: Selectively Retrieve Asset	
Description	<p>In this use-case the LMS needs access to an asset it has not yet retrieved via the previous use-cases.</p> <p>A hypothetical example: Let us suppose that the LMS has also integrated to a Content Service Provider's (CSP) search engine API. A learner performs a search and gets a search result for an asset that was just entitled for the customer by the CSP, but the LMS has not retrieved it yet via the previous use-cases. The learner has requested to launch this asset and the LMS needs the asset information for the asset to satisfy this request.</p>
Actors	LMS and Content Service Provider (CSP)
Assumptions	Same as use-case-01. In addition, it is assumed that somehow the LMS is able to determine the ID of the asset in question. The asset in question must be in the customer's entitlement .
Steps	<ol style="list-style-type: none"> 1. The LMS issues a selective retrieval request to the CSP for Asset Information for a specific asset. 2. The CSP returns a single Referral Object for the specified asset. 3. The LMS processes the Referral Object (installing it on-the-fly) 4. At this point, the asset is now accessible to all Learners.
Variations	

Summary and Recommendations

OLSA Asset Integration is a mature solution, in wide-spread use in the corporate/enterprise E-learning market, for addressing the issues described in the [Problem Definition](#).

SkillSoft is offering the OLSA Asset Integration specification to LETSI for SCORM 2.0 to provide a standards-sanctioned approach for addressing the issues described in the [Problem Definition](#) to the general E-learning community.

We feel that basing a standard on existing practice is the best approach for the following reasons:

1. Proof-of-concept is already established. The risk that the technology will not work is mitigated.

2. Acceptance in the market is already established. There is already growing pool of adopters that can provide critical mass for wider adoption.
3. A real-world problem is being solved. The issues described in the [Problem Definition](#) were solved by the e-learning industry for pragmatic, business reasons.

A key challenge for a new standard is achieving adoption. The previously stated reasons improve the likelihood of a new standard gaining wider adoption. Incorporating OLSA Asset Integration as part of SCORM 2.0 follows this approach and will help achieve the desired outcome for both the LETSI and OLSA communities.

The following steps outline a roadmap for standardizing this API:

1. Review the public specification [\[1\]](#) with the LETSI community
2. Demonstrate the technology to the LETSI community. This technology is already in use in production environments. It should be straight forward to demonstrate its use and interoperability.
3. Incorporate feedback and enhancements into the public specification, taking care to avoid introducing unnecessary backwards incompatibilities.
4. Approve the public specification.