Enable Richer Content Interactions with OLSA On-demand Communication White Paper for SCORM 2.0 (.6)

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) On-demand Communication is introduced as a solution for enabling richer content interactions between a Content Service Provider and a Learning Management System.

We propose to have OLSA On-demand Communication adopted as part of SCORM 2.0.

References

The reference below contains the latest specification of the OLSA On-demand Communication Web Service.

[1]

http://www.letsi.org/letsi/download/attachments/327693/Ondemand+Communication+Public+Specification.pdf?version=1

The reference below is a related SCORM 2.0 white paper that describes a technology 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.

[2] *Automate E-learning Asset Management with OLSA Asset Integration - White Paper for SCORM 2.0*

Introduction

SkillSoft's Open Learning Services Architecture (OLSA) On-demand Communication is a Web Service that enables richer content interactions between a Content Service Provider and a Learning Management System (LMS).

A Content Service Provider (CSP) is a fairly recent term 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.

The On-demand Communication API has been in production for over 1 year. SumTotal and Xerceo are LMS vendors that have led in the adoption of this API. SkillSoft is currently the only Content vendor supporting this API as a Content Service Provider. These are some of the stakeholders for this technology.

The OLSA On-demand Communication 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 On-demand Communication is already based on existing standards where appropriate, some examples are:

- AICC CMI XML for the progress data format (http://www.aicc.org/xmlbindings/cmixml)
- AICC or SCORM launch protocols
- 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)

Content in this context means any e-learning asset such as **traditional** static courseware, **dynamic** content (content generated at run-time), **advanced content** (content that can launch additional content), and **embedded learning** (content accessed through a separate portal system)

Advanced Content and Embedded Learning are examples of richer content interactions enabled by On-demand Communication. In these examples, content can be launched from the CSP with the LMS being unaware of it. Existing standards like AICC or SCORM provide no mechanisms for sending the tracking data back to the LMS so that it can be the system-of-record for these events. These are some of the key motivations behind the On-demand Communication API (see also Problem Definition).

OLSA On-demand Communication is agnostic with respect to the progress data format used to describe a learner's interaction with a given asset. AICC CMI XML is the initial format supported by this API. Other progress data formats can be supported as well.

OLSA On-demand Communication is agnostic with respect to the launch protocol used for a given asset. AICC and SCORM are obvious choices to support. Other launch protocols can be supported as well.

Figure 1 describes the system architecture. Assume that a separate process has taken care of installing assets from the CSP into the LMS (for an example, see reference [2]). Learners can log into the LMS and launch various kinds of assets. In all cases, progress data is managed through the CSP system and the LMS must make On-demand Communication calls to retrieve the data. The necessity for this data flow is explained in detail in the Problem Definition.

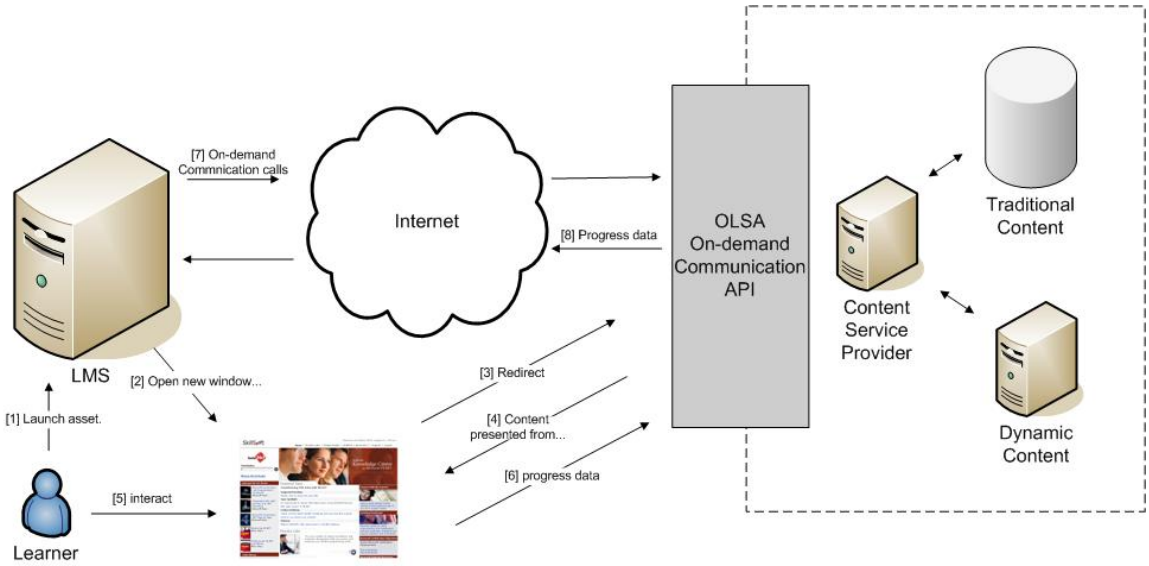

Figure 1: System Architecture

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.** (see reference [2] for more details).

Problem Definition

Advanced Content

The market puts pressure on content vendors to constantly innovate. One area of innovation is the need for content vendors to provide richer content interactions that Advanced Content allows.

On the outside, Advanced Content looks like a single AU or SCO (using AICC or SCORM terminology) to an LMS, but internally it contains its specific training material as well as access to other AUs or SCOs. An example of Advanced Content is the SkillSoft KnowledgeCenter. A KnowledgeCenter presents a domain-specific portal-like experience (e.g., Java Programming KnowledgeCenter, Program Management KnowledgeCenter, etc.). Embedded within this experience are links that allow learners to launch other domain relevant Courses, Mentoring Assets, On-line Books, etc.

Advanced Content is not intended to replace the current multi-AU or multi-SCO approach that is codified in the relevant standards. Advanced Content is a rich content interaction that is needed to complement existing approaches.

A company like SkillSoft can provide the following market-driven value-adds with Advanced Content:

1. Pre-aggregate into a single coherent package of all of the relevant training material on a specific topic as well as links to other related training content (something the customer does not have time, expertise, or resources to do themselves).

2. Ensure a specific, consistent look and feel in the KnowledgeCenter experience regardless of what kind of LMS is used to access it.

3. Provide a service to dynamically update over time the content and links that make up a given KnowledgeCenter to keep it relevant.

4. Provide customers with a way of modifying an Advanced Content structure to add or remove off-the-shelf content or include links to company-specific content.

Figure 2: A sample KnowledgeCenter main page

In figure 2, a sample KnowledgeCenter main page is shown to illustrate these features. Assume that the Referral Object for the KnowledgeCenter is installed in the LMS. The learner can initially launch the KnowledgeCenter from the LMS's catalog. This action will result in the learner landing on a screen that appears something like figure 2. From figure 2, the learner may navigate to a variety of pages covering the topic at hand. The arrows 1 and 2 indicate some of the many links that the learner can select which will launch another separate distinct trackable asset.

The challenge with Advanced Content is ensuring all of progress data generated from pre-aggregated content can be retrieved by the LMS so that it can maintain its role as the system of record for the customer.

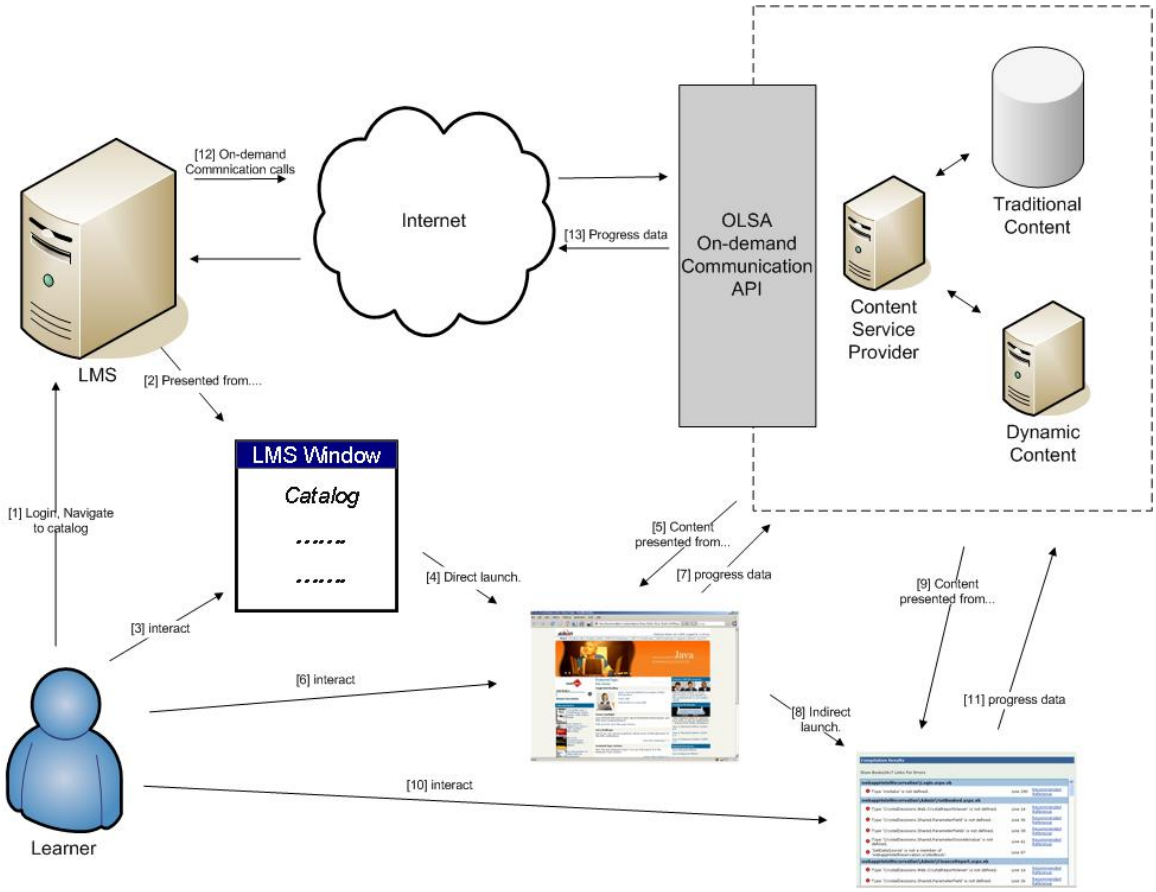

Figure 3: Data flow for Advanced Content

In Figure 3, we illustrate the On-demand Communication data flow that supports Advanced Content. Again we assume some other process occurred to install into the LMS the Referral Objects for the Advanced Content asset, as well any other assets that can be accessed from within the Advanced Content asset.

The data flow starts with the learner logging into the LMS and navigating to the Catalog. From the Catalog the Learner can *directly* launch the Advanced Content. The LMS is aware of the direct launch. Once the Advanced Content is presented, the learner can now start interacting with it. This interaction might generate progress data against the Advanced Content itself. Another outcome of the interaction might be an *indirect* launch of additional assets. The LMS is not aware of indirect launches. The learner can now interact with the indirectly launched asset. This might generate progress data against the indirectly launched asset. All progress data for direct and indirect launches can be retrieved via the On-demand Communication API in a single consistent feed.

Embedded Learning

Another area of innovation is the need for content vendors to provide richer content interactions that Embedded Learning allows.

Embedded Learning allows a content vendor to deploy training through other customer systems in addition to the LMS. For example, a customer may have in place a corporate portal as well as a corporate LMS. The customer would like access to training deployed in both systems. The customer needs to ensure that the LMS remains the system-of-record for all training regardless of how it is accessed.

An example of Embedded Learning is the SkillSoft Learning Portlet. A SkillSoft Learning Portlet allows a customer to embed a Portlet containing SkillSoft training in his corporate Portal.

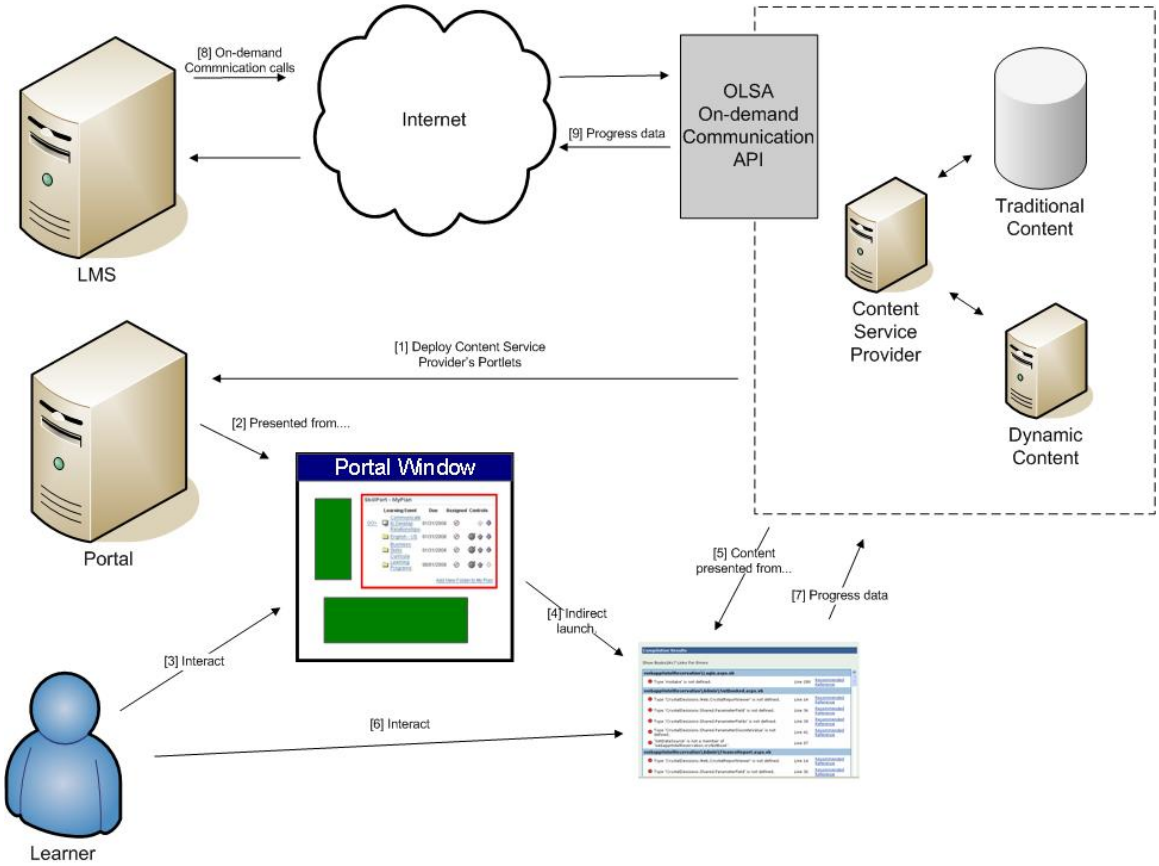

Figure 4: Data Flow with Embedded Learning

In Figure 4, we illustrate the On-demand Communication data flow that supports Embedded Learning. Again we assume some other process occurred to install into the LMS any Referral Objects for training assets accessible as Embedded Learning.

The data flow starts with the Content Service Provider's Portlets being deployed into the customer's portal (details on this are beyond the scope of this document). The Portal can now display pages containing the CSP's Portlet (indicated by the red bounded rectangle). The Learner interacts with the Portlet. One result is the indirect launch of a training asset. The Learner can now interact with the training asset, which results in progress data being generated and accumulated in the CSP. All progress data for these indirect launches can be retrieved by the LMS via the On-demand Communication API.

LMS Behind The Firewall

OLSA On-demand Communication addresses a security requirement of some customers that locally host their own LMS but will not make it visible on the public Internet. To address this situation, OLSA On-demand Communication 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.

The Cross-Domain Problem

In addition to addressing all of the issues identified above, On-demand Communication also addresses the Cross-Domain Problem. With On-demand Communication:

1. All assets will only communicate back to the Content Service Provider (the system of origin for the content).

2. The LMS must retrieve data by issuing Web Service requests to the Content Service Provider.

Given the above, cross-domain issues are no longer encountered.

Use cases

The key use-cases described in the Problem Definition are as follows.

Use-case-01: Track Advanced Content	
Description	In this use-case a content service provider needs to deploy pre-aggregated content as a single AU (or SCO) course to the LMS. This pre-aggregated content contains training material specific to itself, as well as links out to additional e-learning assets. These external e-learning assets are single AU (or SCO) courses as well. Progress data for the Advanced Content asset and any contained assets must be sent back to the LMS.
Actors	LMS and Content Service Provider
Assumptions	The CSP is visible to the LMS. The LMS is not necessarily visible to the CSP. Through a separate process, all necessary assets (the Advanced Content asset and any contained assets) are installed as Referral Objects in the LMS.
Steps	1. Learner logs into the LMS 2. Learner navigates to an LMS catalog screen listing the Advanced Content asset as available training 3. Learner launches the Advanced Content asset C1 (direct launch).

	4. Learner interacts with C1. Some of these interactions might result in progress data for C1 being communicated back to the CSP. 5. Learner launches an asset C2 contained in C1 (indirect launch). 6. Learner interacts with C2. Some of these interactions might result in progress data for C2 being communicated back to the CSP. 7. Learner may terminate sessions with C1 and C2 in any order. 8. LMS eventually retrieves progress data for C1 and C2 9. Learner may now view progress data results from within the LMS.

Use-case-02: Track Embedded Learning	
Description	In this use-case a content service provider has a customer that would like training deployed to other non-LMS systems, like a corporate portal, in addition to the LMS. Progress data for assets launched from any non-LMS system must be sent back to the LMS.
Actors	LMS and Content Service Provider
Assumptions	The CSP is visible to the LMS. The LMS is not necessarily visible to the CSP. Through a separate process, all necessary assets are installed as Referral Objects in the LMS. Through a separate process, all necessary assets are available from the relevant non-LMS systems.
Steps	1. Learner logs into Corporate Portal 2. Learner navigates to a portal screen listing available training 3. Learner launches the asset C1 from the portal screen (indirect launch). 4. Learner interacts with C1. Some of these interactions might result in progress data for C1 being communicated back to the CSP. 5. Learner terminates session with C1. 6. LMS eventually retrieves progress data for C1 7. Learner may now view progress data results from within the LMS.
Variations	C1 in the above sequence could also be Advanced Content.

Given the above use-cases and the manner in which On-demand Communication addresses them, the use-cases that impact the LMS should also be examined. With On-demand Communication, the conventional flow of progress data is significantly altered

(see Figure 3 and Figure 4). This results in the following LMS use-cases that must be addressed by any API like On-demand Communication.

Use-case-03: Execute Bulk Report	
Description	In this use-case, the LMS system has an admin-level user that needs to run a report that contains results over some part of the learner population. The number of learners and the volume of associated data involved can be extremely large. This use-case typically occurs at a frequency of once a day, week, or month. A report does not have to be produced instantly. There is an expectation that the report will take on the order of several or tens of minutes to generate. A report may be out-of-date by some significant time interval. For example the data in the report may be accurate to only within the last hour or 24 hours, etc.
Actors	LMS and admin-level user
Assumptions	Learners have accessed training ~~from the LMS~~ to generate progress data. Some or all of the training is provided through the CSP, through some external process it has been installed into the LMS. Learners may have also exercised either or both use-case-01 or use-case-02 to generate progress data.
Steps	1. Admin-level user logs into their LMS. 2. The admin-level user runs some available report in the LMS 3. After N minutes the report completes and the admin-level user can now access the report. 4. The admin-level user logs out.
Variations	This admin-level user may run more then one kind of report within the session.
Issues	Depending on the time interval being used by the LMS to retrieve progress data from the Content Service Provider, the data in the report will be out-of-date by that same time interval.

Use-case-04: Execute End-User-MyStatus	
Description	In this use-case, the LMS has a learner that needs to perform a LMS action that needs a near-immediate response (e.g., display the student's current status, or provide access to the next training asset in a sequence-of-training-assets). The response must contain any unretrieved progress data of any training asset for the specific learner.

	The number of learners involved can be extremely large. A high number of this use-case may occur in bursts. The pattern will vary depending on the learner population. The data involved for a single learner is relatively small (much smaller then a bulk report). The average response time in this use-case must support the LMS system's user interface's responsiveness requirements.
Actors	LMS, CSP, Learner
Assumptions	Learners can access training from the LMS (or some other means) to generate progress data. Some or all of the training is provided through the CSP, through some external process it has been installed into the LMS
Steps	1. Learner logs into their LMS 2. Learner navigates to an LMS catalog screen listing an asset as available training. 3. Learner launches the asset C1 (direct launch). 4. Learner interacts with C1. Some of these interactions might result in progress data for C1 being communicated back to the CSP. 5. Learner terminates C1. 6. At this point, the LMS may need up-to-date progress data for C1 to present the next screen to the learner. LMS uses some mechanism to retrieve the necessary progress data from the CSP. The action is processed in N seconds. 7. The learner continues with his LMS session. 8. The learner eventually logs out.
Variations	This learner may run one or more of this kind of action during the LMS login session.

Use-case-05: Reset Status of a Learning Object	
Description	In this use-case the LMS system must be able to reset the state of an e-learning asset (to the initialized state) for a particular learner. This learner may have already generated progress data for this asset. This learner may have even already *completed* this asset.

	One example of this is in a compliance setting where a particular asset must be taken once a year, every year for some set of learners for a given customer. Each time such an asset is re-taken by the learner, the asset must be started with a blank slate. Any previous completion history collected for the learner may be preserved by the Content Service Provider and LMS.
Actors	LMS, CSP, and Learner
Assumptions	Learners can access training from the LMS to generate progress data. Some or all of the training is provided through the CSP, through some external process it has been installed into the LMS. The learner in this use-case may have previously accessed or even completed the relevant asset.
Steps	1. The learner logs into their LMS 2. The learner navigates to an LMS screen which requires him to re-take an e-learning asset 3. The learner launches the asset. The LMS informs the CSP that the asset should be launched in its initialized state (i.e., blank slate). 4. The asset presents an initialized state to the learner. 5. The learner interacts with the asset. 6. The learner exits the asset 7. The LMS can now display to the learner the up-to-date progress data (in-progress or completed) for this asset based only on the most recently executed launch session.

Summary and Recommendations

OLSA On-demand Communication is a production quality solution, gaining adopters in the corporate/enterprise E-learning market, for addressing the issues described in the Problem Definition.

SkillSoft is offering the OLSA On-demand Communication 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 On-demand Communication 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.