

Máster Interuniversitario en Seguridad de las TIC (MISTIC)

TFM UOC

Estudio de la Integración del estándar OKI OSID con Learning Tools Interoperability (LTI versiones 1.1.1 y 2.0)

Convocatoria: enero

Profesor: Antoni Bertran Bellido

Curso 2013-2014

Alumno: José María de Córdoba Zea josedcz@uoc.edu

Agradecimientos

A mi familia, por apoyarme durante todo este tiempo, de manera especial a mi madre que me inculcó el interés por aprender y superarme día a día y a mi tía Elisa, por enseñarme que sólo el trabajo, el esfuerzo y la tenacidad conducen a las *metas* marcadas.

A mis compañeros y amigos, por ofrecerme su consejo y su desinteresada ayuda sin necesidad de yo pedírsela. Huelga decir que os nombraría a todos, pero la posibilidad de dejarme a alguno en el tintero me aterra, así que vosotros sabéis bien quiénes sois.

A Miriam, por haber estado pacientemente junto a mí durante toda la duración del proyecto y la redacción de esta memoria.

A mi tutor, Antoni, por su paciencia y comprensión.

A la Universitat Oberta de Catalunya y a la Universidad de Granada, por proporcionarme la formación necesaria para poder afrontar con garantías y superar este trabajo fin de máster.

RESUMEN

Resumen

El presente trabajo está centrado en el análisis de la integración del estándar OKI OSID (Open Knowledge Initiative Open Service Interface Definition) en su versión 3 con las posibilidades que proporciona LTI (Learning Tools Interoperability) para la integración de servicios de proveedores externos en plataformas de aprendizaje (LMS). De la combinación de ambos, se pretende definir un modelo que permita agregar la potencia de OSID para la interoperabilidad de herramientas heterogéneas mediante una arquitectura basada en servicios (SOA), junto con la facilidad de poder introducir cambios en las implementaciones que proporcionan esos servicios de forma ágil, con la potencia de la comunicación que se establece entre consumidor y productor de esos servicios sustentada con LTI.

Palabras Clave: *LTI OSID OKI SOA LMS interoperabilidad escalabilidad*

Abstract

This work is focused on the analysis of the integration of OKI OSID (Open Knowledge Initiative Open Service Interface Definition) version 3 with the capabilities provided by LTI (Learning Tools Interoperability). The aim is achieving the integration of external services providers with Learning Platforms (LMS). The combination of both, we define a model to add OSID power interoperability of heterogeneous tools via a service-based (SOA). In addition, some of the strengths of this union are the possibility to easily include changes to implementations that provide these services architecture and the powerful communication established between consumer and producer of these services supported with LTI.

KeyWords: *LTI OSID OKI SOA LMS interoperability scalability*

ÍNDICE

Índice

Introducción.....	9
Estado del arte.....	9
LTI 1.1.1 vs LTI 2.0.....	13
Introducción.....	13
¿Qué aporta LTI a los usuarios?.....	13
Antecedentes (LTI.1.0 y LTI.1.1).....	14
LTI.1.1.1.....	14
LTI.2.0.....	20
Comparativa LTI 1.1.1 vs LTI 2.0 (draft).....	25
Introducción.....	25
Inconvenientes LTI 1.1.1.....	25
Ventajas de LTI 2.0.....	25
OKI OSID 3.....	27
Introducción.....	27
OSID Version 3.....	29
Authentication.....	31
Competencias.....	31
Modelo de autenticación.....	31
Subpaquete: Authentication Process.....	33
Círculo de Confianza (Trust).....	33
Subpaquete: Authentication Keys.....	35
Subpaquete: Authentication Key Batch.....	35
Subpaquete: Authentication Batch.....	35
Authorization.....	35

ÍNDICE

Subpaquete: Authorization Rules.....	37
Subpaquete: Authorization Batch.....	38
Logging.....	38
Integración LTI – OKI OSID.....	41
Introducción.....	41
Servicios presentes en OSID.....	41
Servicios presentes en LTI.....	42
LTI 1.1.1.....	42
LTI 2.0.....	42
Definición del Wrapper.....	44
Implementación.....	45
Instalación de Tomcat en un Mac OS 10.9.1 Mavericks.....	45
Configuración del entorno de desarrollo	45
Creación de la Base de Datos.....	46
Archivo de Configuración del enganche con la Base de Datos:.....	49
Configuración de puertos y seguridad.....	50
Manual de la aplicación.....	51
Conclusiones.....	54
Bibliografía.....	55

ÍNDICE DE FIGURAS

Índice de Figuras

Figura 1: Catálogo de aplicaciones y herramientas certificadas por el IMS para el estándar LTI.....	11
Figura 2: Logo OSID.....	12
Figura 3: Esquema de conexión de una aplicación con un entorno de usuario final con los elementos de OSID[OSIDSpecFmwk].....	12
Figura 4: Logo IMS GLOBAL Learning Consortium.....	13
Figura 5: Esquema LTI v 1.1.1.....	18
Figura 6: Envío de mensajes mediante LIS como parte de una comunicación bidireccional entre el TP y el TC.....	19
Figura 7: Esquema LTI v 2.0.....	22
Figura 8: Logo OKI.....	27
Figura 9: Diagrama OSID v2.....	28
Figura 10: Relación entre OSID, S-OSID y L-OSID.....	30
Figura 11: Paquetes de interfaces presentes en OSID v 3.0.0rc1.....	31
Figura 12: Diagrama de autenticación OSID.....	32
Figura 13: Diagrama de autenticación Clásico.....	32
Figura 14: Leyenda Diagramas OSID.....	32
Figura 15: Diagrama del Proceso de Autenticación en OSID.....	34
Figura 16: Diagrama del Claves de Autenticación en OSID.....	35
Figura 17: Modelo de Autorización Clásico.....	36
Figura 18: Modelo de Autorización en OSID.....	37
Figura 19: Diagrama del Reglas de Autorización en OSID.....	38
Figura 20: Diagrama Clásico de Logging.....	39
Figura 21: Diagrama del Logging en OSID.....	40
Figura 22: Esquema de Comunicación con LTI y OSID.....	44

INTEGRACIÓN LTI – OKI OSID

Figura 23: Esquema de la Base de datos de LTI Tool Provider.....	49
Figura 24: Configuración Dynamic Domain Name System (DDNS) en el router.	50
Figura 25: Vista de configuración de herramientas consumidoras.....	52
Figura 26: Vista de ejecución de la herramienta en Moodle.....	52
Figura 27: Configuración de Tool Provider con Clave y Secreto en Moodle.....	53

INTRODUCCIÓN

Introducción

Con la realización de este trabajo fin de máster se pretende hacer un estudio de la integración del estándar OKI OSID versión 3 definido por MIT (<http://osid.org/specifications/>) con el estándar Learning Tools Interoperability (LTI, <http://imglobal.org/LTI/>) en sus versiones 1.1.1 y 2.0.

Para ello en primer lugar se hará una comparativa entre ambas versiones del estándar LTI diseñado para comunicar plataformas de aprendizaje (LMS) con diferentes servicios complementarios siguiendo una arquitectura productor/consumidor, enfatizando las diferencias entre ambos, principalmente en el aspecto de la seguridad de ambos estándares.

A continuación se expondrá en qué consiste la interfaz OKI OSID y cómo esta permite a diferentes aplicaciones comunicarse entre sí añadiendo una capa al modelo, independientemente de las tecnologías que soportan las aplicaciones puedan ir cambiando en el tiempo.

Para dar forma a lo anterior se pretende crear un modelo que permita comunicar la funcionalidad que proporciona el estándar LTI para la comunicación de servicios con la potencia y versatilidad del estándar OKI OSID.

Estado del arte

Cada LMS tiene unas características propias que lo diferencian del resto y que hacen decantarse por uno u otro en función del propósito específico de los contenidos y las comunidades de aprendizaje que en ellos se creen. Por otro lado tenemos aplicaciones de terceros y proveedores externos como Wimba o Elluminate que deben decidir para qué LMS desarrollan plugins y eso aparte de ser costoso, supone un desarrollo específico por cada plataforma a la que se quiera dar soporte. Por tanto los proveedores de servicios (TP del inglés ToolProvider), se ven obligados a esperar a que exista una demanda por parte de los clientes (TC del inglés Tool Consumer) para desarrollar los plugins.

LTI propone un estándar para conectar mediante plugins LMS (TC) y herramientas/aplicaciones proveedoras de contenidos (TP) del tipo que se puedan estas puedan consistir tanto en otros gestores de contenidos orientados al aprendizaje o no como pueda ser por ejemplo Wordpress o cualquier otra herramienta susceptible de tener cabida en un LMS. De esta forma un proveedor de contenidos sólo tendría que desarrollar una única integración válida para cualquier LMS que soporte el standard y tendría a su

INTRODUCCIÓN

disposición la plataforma de su elección que le dé soporte a su herramienta de forma segura sin tener que desarrollar y mantener integraciones personalizadas en cada plataforma. Para ello el estándar se propone empezar con una funcionalidad básica e ir avanzando en ella según las necesidades de los clientes TP o LMS, siempre bajo el paraguas de una organización como la IMS con su experiencia y garantía en el proceso de desarrollo y mantenimiento del estándar.

El soporte a lo anterior se articula en el estándar en base a 3 aspectos fundamentales:

1. Los contenidos: entendidos como recursos de aprendizaje que puedan ser importados/exportados por diferentes LMS y que vienen a ser cursos completos encapsulados, por ejemplo y nombrados como Common Cartridge (CC). Un CC no viene a ser más que una descripción en formato XML que define las propiedades de un recurso/cojunto de recursos en base a unas propiedades definidas [LIS]. Podemos encontrar un ejemplo en [LTIV1.1.1] y vienen a ser como una especie de paquetes de fácil distribución al estilo de un SCORM pero con los contenidos con ubicación remota servidos mediante LTI.
2. Las herramientas: son el propio estándar LTI en sí mismo
3. La arquitectura de la información: viene a ser los servicios que proporciona el estándar para poder ofrecer los contenidos mediante las herramientas presentes en el LTI.

En la actualidad existen gran cantidad de aplicaciones y productos disponibles para LMS que cuentan con alguna de las distintas certificaciones de interoperabilidad de la IMS[IMSCert]. Muchas de ellas están destacadas en el catálogo de IMS[IMSCtlg] en el que están presente diferentes LMS (Moodle, Sakai, Blackboard, Desire2Learn, etc), herramientas de eportfolio (Mahara, Chack&Wire,etc), ebook (CourseSmart, CourseLoad, etc), herramientas colaborativas(elgg, Piazza, etc), herramientas de clasificación, herramientas para creación y puesta en valor de contenidos y vídeos entre otras herramientas.

Compañías e instituciones como Moodle, Sakai, Blackboard, Desire2Learn, Wimba o la propia UOC cuentan con productos certificados para las distintas versiones de LTI publicadas (1.0, 1.1 y 1.1.1), así como para las sucesivas revisiones de Common Cartridge, Interactive WhiteBoard/ Common File Format (IWB/CFF v1.0) y el protocolo Learning Information Services and Accessible Portable Item. A continuación odemos ilustrar gran parte de los productos con soporte para el estándar en la siguiente figura:

Estudio de la Integración del estándar OKI OSID con Learning Tools Interoperability (LTI versiones 1.1.1 y 2.0)

INTRODUCCIÓN

learning tools interoperability™ catalog

Use LTI tools inside these platforms - See what they support! click on for more info

ebook tools

Bookshelf® from VitalSource®
Integrated online, downloadable and mobile access for your e-textbooks

CourseSmart
Find and access eTextbooks and digital course materials

Counseload
Delivers course materials through any web-enabled device

nook
Barnes and Noble
Use nook Study to read your ebooks and texts

assessment tools

Learning.com
Web delivered curriculum and assessments

Questionmark
Use Questionmark assessments directly from your QMG

Tri20
Assessment solutions to help foster a culture of evidence

ALEKS™
Use artificial intelligence for your assessment system

BadgesStack
Uses a game-like approach that assesses skills, recognizes learner achievement

Panther Learning
Peer review system designed to eliminate reviewer bias

Edmentum
Online curricula and assessments

Waypoint Outcomes
Learning and assessment software

WebPA
Online peer assessment tool that enables every team member to recognize individual contributions to group work

grading tools

Compass Learning
Gradesbook and content management for K-12

Turnitin
Plagiarism prevention and electronic grading

WebAssign
Online homework and grading

other tools

Florida Virtual School Global Hybrid Hosting

Rustici Software SCORM plugin

SPV Software Products
A simple ratings tool

eportfolio tools

Chalk & Wire
ePortfolio authoring, assessment system, and data analysis tools for students, faculty and professional staff in higher education and K-12

Mahara
Open source eportfolios where you can collect, reflect on and share your achievements success

Pathbrite
ePortfolios help students collect, track and showcase a lifetime of learning and achievement, and recommend pathways for continuous success

PebblePad
Users create records of learning, achievement, and aspiration

SchoolChapters
Portfolio solutions for web, mobile, and print delivered in the cloud

Taskstream
Software to support assessment processes, demonstrate learning, and promote continuous improvement

specialty course tools

Harvard Business for Educators
Access special simulations

McGraw-Hill Campus
Access specialty McGraw-Hill education resources

NBC Learn
Original videos and archival news stories from NBC News

Atomic Learning
Online training video tutorials and technology integration resources

ChemVantage
An Open Education Resource for General Chemistry

NoteFlight
Online music writing application that lets you create, view, print and hear music notation

SPACE
Spatial Activities in Course Environments provides map data and related content

Toolwire
Special purpose learning content and learning environments

MyOpenMath
Online algorithmically generated math homework with auto grading of algebraic and numerical answers, with content tied to open texts

Ucertify
Online IT certification test preparation

SAS Curriculum Pathways
Interactive, standards-based Resources for ELA, Social Studies, Math and Science

Desire2Learn Leap
Provides students with personalized learning paths using existing course content

content creation and delivery tools

LTIWordPress
Free and open source blogging tool and a dynamic content management system (CMS)

SoftChalk Cloud
Create and manage e-learning content for delivery inside or outside of a LMS

Aubrey Daniels Blitz
Develop, deliver, and administer training lessons

WordPress Multisite
A Wordpress plugin that allows multiple versions

collaboration tools

Blackboard Collaborate
Create virtual classrooms, offices and meeting spaces

Learning Objects
Social software for learning

LTI Plugin for Vanilla 2
Provides forum capability via a LTI connection

Elgg
Create your own fully-featured social networks and applications

Plaza
Helping students and instructors ask for and provide course help online

Polycom
Voice, video and content-sharing solutions

Voicethread
Collaborate with voice, video or text

Pruli
Interact with Instructors and Fellow Students

classroom capture/lecture tools

McGraw-Hill Tegrity
Records class time and course content for students to watch live, or review on-demand

BigBlueButton
Open source web conferencing platform for online learning

Echo360
Watch lectures, collaborate with peers and participate in class on smartphones, tablets and laptops

MediaSite
Lecture capture, academic webcasting and video management for education

YouSeeU
Lecture capture, collaboration, and video management for education

video

Mediasite
Media platform for blended learning

ShareStream
Online Video Platform and Streaming-Media Solution for Education

Figura 1: Catálogo de aplicaciones y herramientas certificadas por el IMS para el estándar LTI

INTRODUCCIÓN

Por otra parte se puede definir OSID como una propuesta de un conjunto de interfaces de servicios web dirigidos de forma específica a comunidades de Instituciones de Enseñanza Superior con el objetivo de proporcionar un conjunto de interfaces de servicios web, con una arquitectura basada en servicios (SOA en inglés Service-Oriented Architecture) para la construcción de sistemas de aprendizaje en base a la interoperabilidad de de aplicaciones, permitiendo la evolución de las mismas dentro de un sistema complejo.



Figura 2: Logo OSID

Con arquitecturas del tipo SOA se pretende desvincular las interfaces de sus servicios de sus implementaciones concretas, facilitando de este modo el poder hacer cambios en las implementaciones que proporcionan los servicios sin que la vista del consumidor (interfaz de la aplicación) se vea afectada.

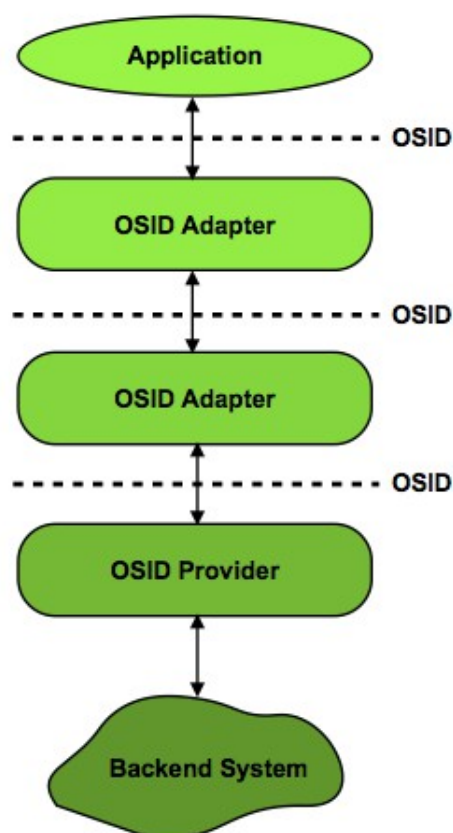


Figura 3: Esquema de conexión de una aplicación con un entorno de usuario final con los elementos de OSID[OSIDSpecFmwk]

LTI 1.1.1 vs LTI 2.0

LTI 1.1.1 vs LTI 2.0

Introducción

Antes de comenzar a detallar las diferencias existentes entre ambos estándares, se ofrece un pequeño marco de donde se sitúa el estándar LTI y cuál es su propósito. Posteriormente se analizan cada una de las versiones del estándar y se hace una tabla comparativa entre ellas

El estándar LTI (Learning Tools Interoperability) es una especificación desarrollada por el consorcio mundial IMS¹, organismo dedicado a la búsqueda y establecimiento de mecanismos para la interoperabilidad entre plataformas de elearning y sus contenidos, con la integración y definición de estándares para esos cometidos. Se trata de una organización sin ánimo de lucro, apoyada por más de 190 organizaciones [HIMS], que incluye empresas, organizaciones educativas y gubernamentales líderes en el ámbito educativo y del elearning. Bajo sus siglas se han publicado algunos de los 20 estándares más ampliamente utilizados en Tecnologías Educativas para metadatos, preguntas y respuestas, ePortfolio, definición de vocabularios o interoperabilidad o integración de herramientas remotas (como es el caso del LTI) entre otros muchos campos, la mayoría de ellos en el campo de actuación de las instituciones educativas de Enseñanza Superior.



Figura 4: Logo IMS GLOBAL Learning Consortium

¿Qué aporta LTI a los usuarios?

De cara al usuario podemos distinguir los beneficios del uso del estándar LTI en función del tipo de usuario presente en los procesos de eaprendizaje:

- Alumnos: No es necesaria una credencial para cada entorno de aprendizaje, sino que con una sola se puede acceder a recursos de terceros desde su propio curso online en el que participan.

1 El nombre formal es el IMS Global Learning Consortium, Inc. o abreviadamente el IMS GLC. El nombre original proviene de cuando comienza en 1997 el proyecto Instructional Management Systems, si bien al tiempo se ve que este término plantea más preguntas que respuestas y surgen conceptos como LMS (Learning Management System), curso o server learning entre otros.

LTI 1.1.1 vs LTI 2.0

- Profesores: pueden incluir en sus cursos aplicaciones de terceros (recursos de otros CMS o aplicaciones de blogs por ejemplo) que sean interesantes para los alumnos y que no tenían disponibles en el LMS
- Desarrolladores de aplicaciones de terceros/LMS: Con desarrollar la aplicación/contenido conforme a lo establecido en el estándar LTI y validarla (<http://validator.imsglobal.org/> y/o <http://www.imsglobal.org/developers/alliance/LTI/cert/index.php> (es necesario ser miembro y pasar la validación del TP o del TC según sea el caso mediante un proceso formal), esta estará disponible para cualquier LMS o aplicación que sea un TC y quiera utilizarla sin necesidad de realizar códigos específicos para cada una de ellas).
- Administradores de entornos virtuales de aprendizaje (VLE): Descarga de trabajo debido a no tener que realizar actualizaciones e instalación de nuevos sistemas cada vez que un cliente requiere disponer de una nueva extensión o herramienta (TP) en el LMS o aplicación para el VLE (TC). Además se aseguran las aplicaciones del Tool Provider no comprometen en parte la seguridad del sistema ya que se ejecutan en servidores diferenciados.

Antecedentes (LTI.1.0 y LTI.1.1)

Inicialmente se vino a llamar LTI Básico a la versión 1.0 de LTI que fue lanzada el 30 de mayo de 2010 y que proporciona un método sencillo pero estandarizado de establecer una conexión segura desde un TC a una herramienta externa (TP). En un primer momento la comunicación entre ambos es unidireccional y no se permite la retroalimentación de una salida desde la herramienta externa (TP) hacia el TC.

Posteriormente en marzo de 2012 se lanza la versión 1.1 en la que ya sí que se permite esa comunicación. Como resultado de esa comunicación entre TP y TC, esta característica permite nuevas características en el uso de las herramientas externas, como por ejemplo que los estudiantes puedan recibir una calificación como resultado de su interacción con la aplicación.

Finalmente existe una versión 1.1.1 del estándar en la que se entrará con detalle a continuación y que se diferencia de las anteriores por la inclusión del soporte para la figura (rol) del mentor presente en los entornos de aprendizaje clásicos como el representante del alumno (padres o tutor/es legal).

LTI.1.1.1

LTI versión 1.1.1 fue publicado originalmente el 12 de junio de 2012 teniendo su última revisión con fecha del 20 de septiembre de 2012 [LTIv1.1.1], para definir las especificaciones que determinen un estándar para permitir integrar aplicaciones remotas y contenidos en un LMS Learning (Management System).

LTI 1.1.1 vs LTI 2.0

Se trataría de un esquema básico productor-consumidor en el que la petición se realiza mediante el protocolo definido en el LTI v1.1.1. Así se tienen dos elementos:

- el consumidor “Tool Consumer (TC)”, que viene a ser lo que habitualmente se entendería como el propio LMS que es la herramienta que va a “consumir”, acceder a los datos.
- el productor o proveedor, denominado “Tool Provider (TP)”, que el servidor o sistema externo que proporciona un servicio o contenido al consumidor.

La relación entre los elementos implicados en LTI queda establecida por una serie de casos de uso que se enumeran a continuación:

- Establecimiento de las credenciales de dominio para un TP
- Establecimiento de las credenciales a nivel de enlace
- Gestión de las credenciales para enlaces importados desde un Common Cartridge (CC)
- Ejecución de un contenido enlazado autorizado desde un contexto (TC)
- Ejecución de un contenido enlazado autorizado importado desde un Common Cartridge (CC), con credenciales establecidas previamente (secreto o clave)
- Ejecución de un contenido enlazado autorizado importado desde un Common Cartridge (CC), sin credenciales establecidas
- Comunicar una puntuación decimal (0.0 y 1.0) desde la herramienta (TP) al TC (por ejemplo un LMS)

Además es necesario definir un “contexto” donde se produce esa transacción definida por LTI v1.1.1 que dentro del ámbito del elearning y los LMS puede ser traducido a un curso, colección, proyecto o colección de recursos con un conjunto de común de usuarios y roles definidos en el LMS, y aplicables en función del contexto que sea de aplicación en cada momento. Para distinguir esos contenidos o recursos dentro del contexto se utiliza el concepto de identificador del enlace al recurso o `resource_link_id`. Así se tiene que para un contexto dado (`context_id`) se tienen una serie de recursos enlazados e identificados mediante un `resource_link`. De este modo se puede especificar como se dijo antes que en función del contexto se establezcan las restricciones que se estimen para los recursos por parte de los administradores del TP.

En cualquier caso en el ámbito del LTI si bien se parte de la premisa fundamental de fomentar en la medida de lo posible el uso de usar el subconjunto más pequeño de funcionalidad de LTI para cada caso, se tienen disponibles una serie de parámetros en su mayoría opcionales (aunque hay algunos **obligatorios**) para las transacciones entre TP y TC. A continuación se muestran junto con un ejemplo de su valor

LTI 1.1.1 vs LTI 2.0

- **lti_message_type=**basic-lti-launch-request → indica que se trata de un mensaje de inicio de comunicación LTI y permite que se acepten por parte del TP sucesivos mensajes desde la url del TP) (obligatorio)
- **lti_version=**LTI-1p0 → especifica la versión del protocolo LTI que se utiliza en ese mensaje (obligatorio)
- **resource_link_id=**88391-e1919-bb3456 → se trata de un identificador opaco que garantiza la unicidad de un contenido para un TC en cada enlace que se haga. Si se hicieran varios enlaces desde un mismo TC cada uno de ellos tendría uno diferente (obligatorio)
- **resource_link_title=**My Weekly Wiki → contiene el título para el enlace del recurso enlazado por cuestiones de accesibilidad (recomendado)
- **resource_link_description=**... → contiene la descripción para el enlace del recurso enlazado (opcional)
- **user_id=**0ae836b9-7fc9-4060-006f-27b2066ac545 → identifica unívocamente al usuario para un TC determinado con un identificador disociable del mismo (recomendado)
- **user_image=**http://.... → contiene la imagen o avatar del usuario que realiza la petición con un formato de archivo de tipo imagen y relación de aspecto cuadrada (opcional)
- **roles=**Instructor → contiene una lista de valores URN para los roles. Puede encontrarse más información sobre el formato en [LTIv1.1.1] (recomendado)
- **lis_person_*** → Campos que contendrán en base a lo establecido en las especificaciones diseñadas para definir cómo los sistemas gestionan el intercambio de información en los que se describe como estructurar la información referente a personas, grupos, pertenencias, jerarquías, cursos y resultados en el contexto del aprendizaje [LIS]. No deberían de aparecer en casos en los que se requiera preservar la privacidad el envío de información que identifique al usuario que realiza un ejercicio o valora un contenido, por ejemplo (opcional)
- **role_scope_mentor=**f5b2cc6c-8c5c-24e8-75cc-fac504df920f,dc19e42c-b0fe-68b8-167e-4b1a8f2b367e → campo introducido en la versión 1.1.1 para indicar para qué usuarios el usuario actual tiene acceso como mentor. Contiene la lista de ids de usuarios mentorizados separados por comas. (opcional)
- **context_id=**8213060-006f-27b2066ac545 → identificador opaco que identifica el contexto (TC) desde el que se lanza un enlace (recomendado)
- **context_type=**CourseSection → cadena separada por comas de valores URN [LIS] que identifican el tipo de contexto (opcional)
- **context_title=**Design of Personal Environments → Texto con el nombre (una línea) del contexto por cuestiones de accesibilidad (recomendado)
- **context_label=**SI182 → etiqueta en texto del contexto (fácil legibilidad para encajar en una columna) (recomendado)

LTI 1.1.1 vs LTI 2.0

- `launch_presentation_locale=en-US` → idioma, país y juego de caracteres a utilizar, para una correcta representación en idiomas con caracteres no ASCII
- `launch_presentation_document_target=iframe` → indica al navegador si el contenido a mostrar en el TC ha de serlo en la ventana o en un iframe (recomendado)
- `launch_presentation_css_url=` → enlace a capa CSS para presentación (opcional)
- `launch_presentation_width=320` → ancho del iframe o ventana donde mostrar el contenido del TP (recomendado)
- `launch_presentation_height=240` → alto del iframe o ventana donde mostrar el contenido del TP (recomendado)
- `launch_presentation_return_url=http://lmsng.school.edu/portal/123/page/988/` URL donde el TP puede especificar que se redirija al usuario en el TC en caso de fin de uso de la herramienta, error, etc. En este tipo de casos tendría sentido mostrar el mensaje de error `lti_msg_error` que se generaría (recomendado)
- `tool_consumer_info_product_family_code=desire2learn` → identificador del tipo de TC que se está utilizando para depurar y facilitar la asistencia (recomendado)
- `tool_consumer_info_version=9.2.4` → Versión de la aplicación TC utilizada, información útil para el TP, para actualizaciones, etc. (recomendado)
- `tool_consumer_instance_guid=lmsng.school.edu` → identificador unívoco para el TC, normalmente la url. (recomendado en casos de hosts compartidos)
- `tool_consumer_instance_name=SchoolU` → nombre de la instancia del TC – se mostrará en una columna – (recomendado)
- `tool_consumer_instance_description=University of School (LMSng)` → descripción del TC (se mostrará en una línea) (opcional)
- `tool_consumer_instance_url=http://lmsng.school.edu` → URL de la instancia del TC (opcional)
- `tool_consumer_instance_contact_email=System.Admin@school.edu` → un correo para contactar con los admins del TC (recomendado)
- `custom_keyname=value` → valores personalizados a definir por el creador del TP que pudieran ser necesarios

En cuanto a la securización de los mensajes intercambiados entre el TP y el TC, se establece en el estándar LTI v1.1.1 el uso del protocolo OAuth 1.0a, siendo la clave transmitida en cada mensaje y su generación basada en una clave mediante criptografía asimétrica basada en HMAC-SHA1. En el TP se recalcula el valor de la firma para comprobar la autenticidad del mensaje enviado por el TC.

Existen 3 tipos básicos de gestión de credenciales, por parte del TC, definidos en el LTI v1.1.1 según a quién se asocian las credenciales:

LTI 1.1.1 vs LTI 2.0

1. Credenciales asociadas con el dominio del TP. Todos los TP comparten una misma clave o “secreto”
2. Credenciales asociadas a todas las URL del mismo TP. Así por cada TP se tiene una clave o “secreto”,
3. A cada enlace de un TP se le apropia su propia credencial, asociada a cada enlace particular.

Cada una de ellas tienen sus ventajas, mientras que la 1 y la 2 permiten una visión más integrada de una instancia de TC y una TP relacionadas desde la perspectiva del administrador, la tercera permite no hacer distinciones en función de la procedencia de los enlaces.

A continuación se puede ver un esquema en el que se refleja gráficamente lo que se ha introducido acerca del esquema de funcionamiento de LTI en cuanto a la relación de un TC con un TP, el intercambio de mensajes LTI (Oauth 1.0a así como las figuras del administrador del contenidos del LMS y con acceso al TP, el usuario que accede a un recurso a través del TC y el instructor que proporciona los contenidos al TP y los sirve a través del TC.

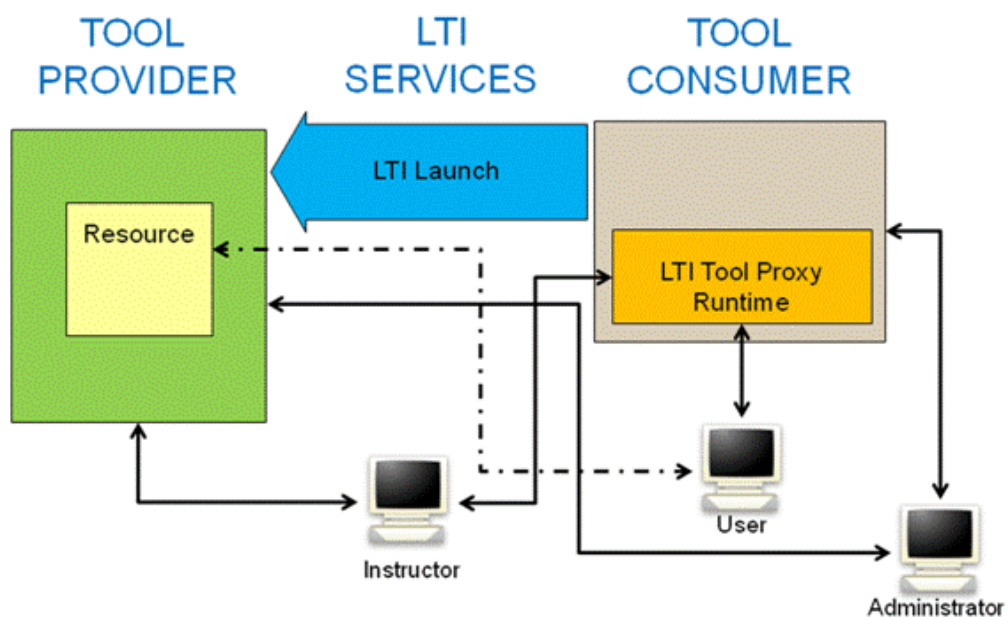


Figura 5: Esquema LTI v 1.1.1

Una vez se ha visto como el esquema de interacción entre el TC y el TP, se debe analizar como se envían mensajes entre ambos con el resultado de esa interacción.

Para ello se definen los siguientes campos de datos que dan soporte a la comunicación entre ambos para este propósito:

- `lis_result_sourcedid=83873872987329873264783687634` → campo único para cada combinación de la terna `context_id` / `resource_link_id` / `user_id` que identifica un único campo de las calificaciones presentes en un TC y que contienen la calificación de un usuario `user_id` para un recurso `resource_link_id` perteneciente a un TP instanciado concreto `context_id`. (opcional)

LTI 1.1.1 vs LTI 2.0

- `lis_outcome_service_url=` → URL a la que preguntará el TP para requerir operaciones/acciones y que responderá a las mismas en el caso de que estén definidas en el TC. Típicamente un TP sólo responderá a peticiones provenientes de un rol del tipo “Estudiante” (learner) para sus propias actividades (ver `lis_result_sourceid`), pero ya será cosa del propio TP y de cómo se establezcan las operaciones permitidas según los roles para el TC.

Existen otros parámetros adicionales para dar soporte a otras formas de comunicación:

- `lis_person_sourcedid=school.edu:user` → identifica al usuario que realiza la acción en base a valores urn definidos en [LIS]
- `lis_course_offering_sourcedid=school.edu:SI182-F08` → identifica el curso en el que se realiza la acción
- `lis_course_section_sourcedid=school.edu:SI182-001-F08` → identifica el contexto en el que el usuario realiza la acción

Además existen 3 tipos de mensajes que permiten obtener/modificar/eliminar los resultados/calificaciones para un usuario y un recurso dados. El valor de la calificación ha de estar entre 0.0 y 1.0. Se trata de mensajes en formato POX (Plain Old XML):

- `readResult` → se trata de obtener la calificación actual para un `lis_result_sourcedid` dado (recomendado). Si el campo no existiera devolverá un campo vacío en el POX y no un mensaje de error.
- `ReplaceResult` → En el caso de que haya problemas en la recepción del resultado de la operación se recibirá en el TC un `imsx_codeMajor` encapsulado en un `imsx_statusInfo` con el mensaje indicando si se ha producido un problema o todo ha sido satisfactorio.
- `deleteResult` → se trata de obtener la calificación actual para un `lis_result_sourcedid` dado (recomendado). El resultado de la operación será recogido en un mensaje en el TC `imsx_codeMajor` encapsulado en un `imsx_statusInfo` si se ha producido un problema o todo ha sido satisfactorio.

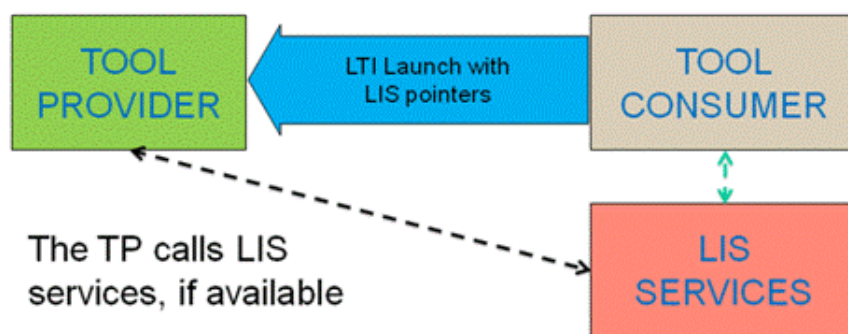


Figura 6: Envío de mensajes mediante LIS como parte de una comunicación bidireccional entre el TP y el TC

LTI 1.1.1 vs LTI 2.0

LTI.2.0

La versión 2.0 de LTI es publicada el 1 de noviembre de 2012 y revisada el 15 de Julio de 2013 [LTIv2.0]. Esta revisión del estándar introduce una nueva arquitectura para dar soporte a la funcionalidad que soportaba la versión anterior y es compatible hacia atrás para comunicarse con TC o TP que sólo sean compatibles con la versión 1 o alguna de sus revisiones.

Las principales diferencias con respecto a las características añadidas en esta nueva versión de LTI son las siguientes:

- Se desliga el concepto de herramienta o aplicación del TP dado que se entiende que se podrían tener herramientas, que si bien no cumplan con el estándar LTI, pueden ser recubiertas con una capa de comunicación que sí cumpla con el estándar y por tanto permita la comunicación mediante LTI con cualquier TC. Para ello es necesario que la herramienta defina en esa capa un gestor de recursos para procesar los mensajes que reciba del TC. El gestor de recursos es capaz de gestionar las distintas peticiones que se pueden requerir por cada instancia del tool proxy a un misma aplicación o herramienta (integrada o no en el TP), con sus diferentes parámetros y su punto de reentrada dentro del TC
- Creación diferentes instancias del proxy en el TC por cada instancia que se requiera al TP.
- Para la comunicación entre los Tool Consumer y el Provider o Tool se utiliza REST y JSON-LD.
- Definición de un marco común para describir mensajes y servicios y permitir de forma fácil el añadir nuevos tipos, en definitiva para favorecer la extensibilidad del protocolo. En definitiva se trata de capacidades o prestaciones (capabilities) como la extensión a través de variables que permite definir variables que aparece en esta versión, el TC ha de comunicar al TP las que soporta (de este modo el protocolo es compatible hacia atrás) y el TP decide cuáles desea habilitar.
- Aparición de los mensajes LTI del tipo Tool Proxy Registration Request y Tool Proxy Reregistration Request. Los dos últimos forman parte del proceso de establecimiento de la integración entre una herramienta un TP. Este proceso consta de 4 fases:
 - Solicitar acceso a la herramienta (TP)
 - Obtener el perfil del TC para garantizar que se cumplen los requerimientos mínimos para la integración
 - Registrar un Tool Proxy en el TC:
 - Requiere en el mensaje los siguientes parámetros: lti_message_type, reg_key, reg_password, tc_profile_url, launch_presentation_return_url, siendo recomendados launch_presentation_document_target, launch_presentation_height, launch_presentation_width,

LTI 1.1.1 vs LTI 2.0

user_id, roles y opcionales launch_presentation_css_url, launch_presentation_locale.

- En el momento que se quiera actualizar los datos del proxy ya desplegado se lanzaría una petición Tool Proxy Reregistration Request con los mismos parámetros y los valores de los campos actualizados, si bien es necesario que el proxy a actualizar/reregistrar exista.
- Poner el Tool Proxy disponible para recibir del TP.
- Se ha eliminado la convención de renombrar los parámetros personalizados conforme a las reglas de LTI versión 1 (eliminar mayúsculas y reemplazar los caracteres especiales y símbolos de puntuación por barras bajas), si bien si en la comunicación se detecta un TP que opera bajo LTI se sigan respetando.
- Los siguientes parámetros para mensajes se consideran obsoletos: context_title, context_label, resource_link_title, resource_link_description, lis_person_name_given, lis_person_name_family, lis_person_name_full, lis_person_contact_email_primary, user_image, lis_person_sourcedid, lis_course_offering_sourcedid, lis_course_section_sourcedid, tool_consumer_info_product_family_code, tool_consumer_info_product_family_version, tool_consumer_instance_name, tool_consumer_instance_description, tool_consumer_instance_url, tool_consumer_instance_contact_email.
- Definición de Perfiles tanto en el TC como en el TP para que la información esté disponible para ambos mediante la comunicación que proporcionan los servicios LTI. En este perfil se almacena qué versiones de LTI soporta una herramienta/TC, qué prestaciones y qué servicios web se pueden proporcionar.
- Se añade el caso de uso necesario para “actualizar un Tool Proxy perteneciente a un TC” y queda obsoleto “Comunicar una puntuación decimal (0.0 y 1.0) desde la herramienta (TP) al TC (por ejemplo un LMS)”

El funcionamiento del LTI2.0. se puede ver simplificado de forma esquemática en la siguiente figura:

LTI 1.1.1 vs LTI 2.0

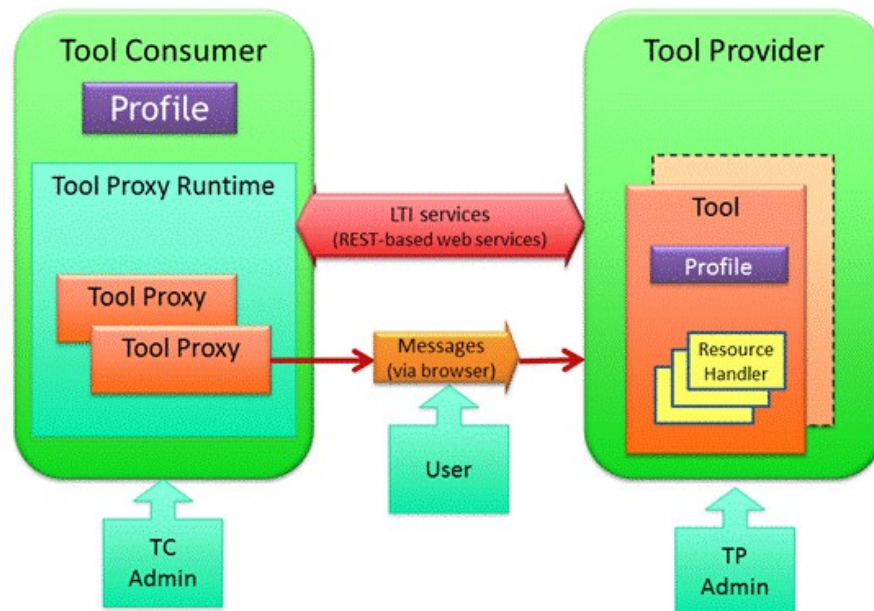


Figura 7: Esquema LTI v 2.0

A continuación se ve en más detalle cada uno de los elementos y acciones presentes en el esquema de LTI y que permiten que desde un LMS (TC) se puedan poner accesibles para sus usuarios contenidos alojados en una herramienta (Tool) que proporciona un proveedor externo (TP). Se analiza como se articula el mecanismo para lo anterior en base a un Tool Proxy, los perfiles y contratos, así como la gestión de los recursos especificando como desarrolla la comunicación entre ambas entidades.

En el lado de la plataforma consumidora TC (Tool Consumer) es necesario contar con los siguientes pasos:

- Publicar el Profile del TC en alguna URL accesible. Para que desde un LMS se pueda añadir una herramienta externa de un TP es necesario poder indicar donde se encuentra el perfil particular donde se muestra la información necesaria para que el TP conozca las capacidades y características del TC para crear el contrato de integración conocido como Tool Proxy (combinación del contrato de seguridad y el perfil de la herramienta del TC con la que se establece la integración).
- Creación de una interfaz de usuario para que se pueda pedir registrar una herramienta por parte del administrador del TC: esta interfaz ha de aceptar una url proporcionada por el administrador y entonces enviar una petición de registro del Tool Proxy a la URL especificada. Esto permitiría desde por ejemplo un LMS poder añadir a un recurso de una plataforma externa como contenido a ofrecer a los usuarios del LMS.
- Implementación del Servicio de Tool Proxy: Permite que el TP registre un Tool Proxy para poder conectarse a una herramienta (tool). Con este servicio se permite la comunicación del TP, como un LMS con el

LTI 1.1.1 vs LTI 2.0

proveedor del servicio (aplicación externa) y se separa o compartimenta esa comunicación en base a una relación entre el tool proxy con la herramienta específica dentro del tool provider (TP), permitiendo diferenciar y separar las diferentes comunicaciones y servir los recursos externos del TP en el TC de forma independiente y diferenciada, lo que permite en un momento dado habilitar unos tipos de recursos y deshabilitar otros sin tener que deshabilitar toda la comunicación TP-TC.

- Creación de una interfaz de usuario para hacer accesible el Tool Proxy registrado: Se utiliza para que el administrador pueda dar el visto bueno y mostrar las implicaciones de seguridad (definidas en los contratos de seguridad que se establecen en el registro del proxy entre el TC y la herramienta (del TP)) que conlleva poner el Tool Proxy disponible. De este modo en un LMS se ha de permitir por parte de su administrador en caso de que sea necesario que el proveedor del servicio (TP) y en el caso concreto la herramienta autorizada del mismo tenga acceso a información de tipo personal, del curso o de las calificaciones de los usuarios del TC, si bien por ejemplo en el uso de calificaciones se trata tan sólo de una buena práctica y no de algo obligatorio.
- Creación de una interfaz de usuario que permita a los creadores de contenidos dentro del TC crear los enlaces LTI asociados con cada tipo de recurso. De esta forma se pondrían a disposición de los usuarios de un LMS los contenidos albergados por el TP una vez ya se ha establecido la comunicación a través de los mensajes de tipo POST y REST.
- Servicio para ejecutar los contenidos enlazados: sirve para poner disponibles los recursos del TP para los usuarios del TC. Con esto se pondrían a disposición de los usuarios de un LMS los contenidos albergados por el TP a través de los servicios LTI basados en REST y JSON_LD.
- Soporte para gestionar la capacidad o prestación (capability) de crear un resultado (calificación) así como modificarlo (recalificarlo mediante peticiones PUT) para un tipo de recurso definido en el TC y proporcionado por el TP. Es decir, en un LMS se tendría que permitir la posibilidad de calificar las acciones dentro de un rango (0.0 y 1.0) de los usuarios del sistema (TC) al interactuar con los recursos externos proporcionados por la herramienta externa (TP).

En el lado de la herramienta externa (TP) es necesario contar los siguientes mecanismos:

- Gestor de mensajes para permitir el registro del TP: El gestor debe poder recibir las peticiones para el registro de un Tool Proxy. Para ello son necesarios los siguientes elementos:

LTI 1.1.1 vs LTI 2.0

- Perfil de la herramienta (Tool Profile): A semejanza del Profile del consumidor describe la información para que el TC conozca las capacidades y características del TP para crear el contrato de integración conocido como Tool Proxy. Como mínimo ha de contar con la siguiente gestión de registro de la actividad de los usuarios del TC con un recurso del TP:
 - Habilitar la capacidad o prestación (capability) de crear un resultado (calificación) así como modificarlo (recalificarlo mediante peticiones PUT) para un tipo de recurso proporcionado por el TP.
 - Contar con un parámetro personalizado (GUID) para identificar una calificación con la declaración de una variable Result.sourceld.
- Creación de un contrato de seguridad (security contract) que defina la integración con el TC. Este contrato ha de enumerar los servicios que del TP va a usar la herramienta (Tool) con la que se hará la comunicación con el TP y que llevará aparejada la asociación con el TP mediante la clave/secreto/url de la herramienta en el TC. Con esto se permite evitar interacciones problemáticas cuando se daban diferentes servicios desde un TP a uno o a varios TP.
- Creación del Tool Proxy propiamente dicho que combina el contrato de seguridad y el Perfil de la herramienta, que soporte las capacidades definidas en el Tool Profile.
- Registro del Tool Proxy mediante el envío al TC de una petición de registro al Servicio de Tool Proxy.
- Devolución del control al TC redirigiendo al usuario que registra el ToolProxy de vuelta al TC en base a la dirección launch_presentation_return_url especificada en la petición de registro del Tool Proxy (ToolProxyRegistrationRequest).
- Gestor de mensajes para las peticiones de ejecución de los contenidos proporcionados por la herramienta. Este gestor permite recibir las peticiones basic-lti-launch-request que solicitan un contenido. Además ha de tener soporte para en base a un GUID identificar una calificación con la declaración de una variable Result.sourceld .
- Enviar una calificación al TC así como modificarlo (recalificarlo mediante peticiones PUT) para un tipo de recurso definido en el TC y proporcionado por el TP.

LTI 1.1.1 vs LTI 2.0

Comparativa LTI 1.1.1 vs LTI 2.0 (draft)

Introducción

Ya se han visto los principales detalles de las diferentes versiones de ambas versiones del protocolo LTI. A continuación a modo de resumen los inconvenientes presentes en LTI en su versión 1.1.1 y qué soluciones se están proponiendo en la revisión del protocolo LTI 2.0 que aún es sólo un borrador (draft).

Inconvenientes LTI 1.1.1

LTI v1.1.1 y sus versiones anteriores sin bien han supuesto una revolución en cuanto al tiempo y el coste de integrar contenidos y recursos digitales en LMSs a nivel básico del orden de 10 a 100 veces [IMSODIRE] viene a ser un subconjunto simplificado y una aproximación de la implementación completa de LTI. Se tiene la necesidad de contar con una implementación de tipo “empresarial” para LTI ya que LTI en sus primeras versiones se queda corta en cuanto a su funcionalidad dados los principios de sencillez aplicados en la definición del protocolo.

Además realmente la implementación del protocolo por parte de la comunidad se veía lastrada debido a la definición básica y orientada sobre todo a las instituciones de enseñanza superior dejando un poco de lado al entorno empresarial.

Por otro lado el uso valores URN para la definición del vocabulario estándar LTI para las entidades (contextos y roles) se queda corto para permitir la extensibilidad y fueron dando lugar de hecho a la aparición de extensiones no oficiales, de las que algunas de ellas fueron añadidas formalmente en revisiones de LTI 1, como la gestión de roles o calificaciones.

Ventajas de LTI 2.0

- Compatibilidad con la versiones anteriores de LTI. Se pueden mantener todo el trabajo que se ha ido realizando anteriormente y aprovecharlo.
- Paso de un modelo de productos certificados a uno de modelo de servicios negociados, donde dos productos son capaces de comunicarse entre sí en base al soporte del protocolo (entender sus perfiles mediante LTI-Services) y configurarse en consecuencia, llevando las capacidades de plug&play a un nivel superior.
- Posibilidad de proporcionar una mayor integración y soporte, mediante la inclusión de los perfiles en las herramientas que permitan especificar donde un contenido enlazado debe aparecer y además se proporcione soporte en función del lenguaje que tengan seleccionado un usuario en el TC.

LTI 1.1.1 vs LTI 2.0

- Una arquitectura más flexible y extensible como se vio en el apartado anterior, orientada a ofrecer mayor soporte a servicios y eventos, como la mejora en la mensajería (capabilities) y comunicación entre el TP y el TC.
- Definición de entidades mediante URLs que apuntan a las mismas dentro de una ontología RDF.
- Posibilidad de disponer de autorizaciones de enlaces a contenidos gestionados desde el gestor de recursos disponible en el tool proxy del TP y que permiten la autorización de recursos del mismo tipo que agrupan enlaces dispuestos en forma masiva.
- Posibilidad mediante el uso de PUT (REST) en los mensajes entre TC y TP de volver a generar una nueva calificación (Result) para un recurso o ítem y no sólo calificarlo de nuevo cada vez, mediante una petición POST. Esto permite tener un histórico de calificaciones para un alumno-recurso y registrando los múltiples intentos de realización de un ejercicio.
- Introducción del uso de JSON (JavaScript Object Notation) en sustitución de POX (Plain Old XML) para las comunicaciones de calificaciones (results) que permite una serie de mejoras:
 - JSON proporciona estructuras de objetos y arrays
 - Descarga el procesamiento de la información en el lado del cliente mientras que con XML se tenía que hacer en el lado del servidor. Además permite el Cross Domain.
 - Es flexible y permite la extensibilidad , permitiendo la creación de estructuras de datos que dan soporte a la estructura de tool proxies, tools y profiles definida.

OKI OSID 3

OKI OSID 3

Introducción

OSID nace en 2001 como una parte de la iniciativa desarrollada bajo el amparo del MIT y de la Fundación Andrew W. Mellon, con la denominación de proyecto Open Knowledge Initiative (OKI) con el objetivo de proporcionar una arquitectura que permita la interoperabilidad entre aplicaciones desarrolladas para entornos educativos de instituciones de Enseñanza Superior. En 2006 comienza el desarrollo de OSID 3K para rediseñar las especificaciones iniciales para poder aplicar la arquitectura para un mayor número de servicios en diferentes dominios y permitir la integración de los mismos tanto cuando se tratara de pequeños sistemas o sistemas empresariales a gran escala[OSID].



Figura 8: Logo OKI

OSID proporciona como se comentaba en la introducción una arquitectura basada en servicios que proporcionan las múltiples interfaces definidas por los paquetes presentes en él. De este modo se convierte en una ayuda para el desarrollo de soluciones software en base a dividir el problema en los distintos servicios a ofrecer, muchos de ellos ya especificados en el marco de OSID. Por tanto la comunicación y coordinación entre estos también es gestionada por OSID y de este modo descargar a los desarrolladores de estas tareas para que se centren en la parte de la lógica de la aplicación.

En la especificación versión 2.0 de OSID presentaba los siguientes elementos:

- Launcher, es el encargado de crear el contexto OSID que se encargará de poner todos los datos necesarios para que el sistema pueda funcionar correctamente. Indican los token de sesión, el contexto de la aplicación, etc.
- Componentes, que se encargaran de facilitar el uso de los OSIDs ya que proveen una API muy sencilla de utilizar y se comunican con OkiBus Cliente o el Gateway.
- OkiBus Cliente, nos permiten poder acceder de manera remota desde una herramienta a un LMS, para ello transforma el mensaje OSID a una llamada webservice que será la que llegará al OkiBus Servidor.

OKI OSID 3

- OkiBus Servidor, transforman el mensaje del OkiBus Cliente a OSID para comunicarse con el Gateway.
- Gateway, son los adaptadores de OSID al lenguaje del LMS (Moodle, Sakai, etc).

Gráficamente un diagrama de lo descrito para conectar una aplicación y un LMS, enfatizando la parte de autenticación autorización y registro, quedaría como sigue:

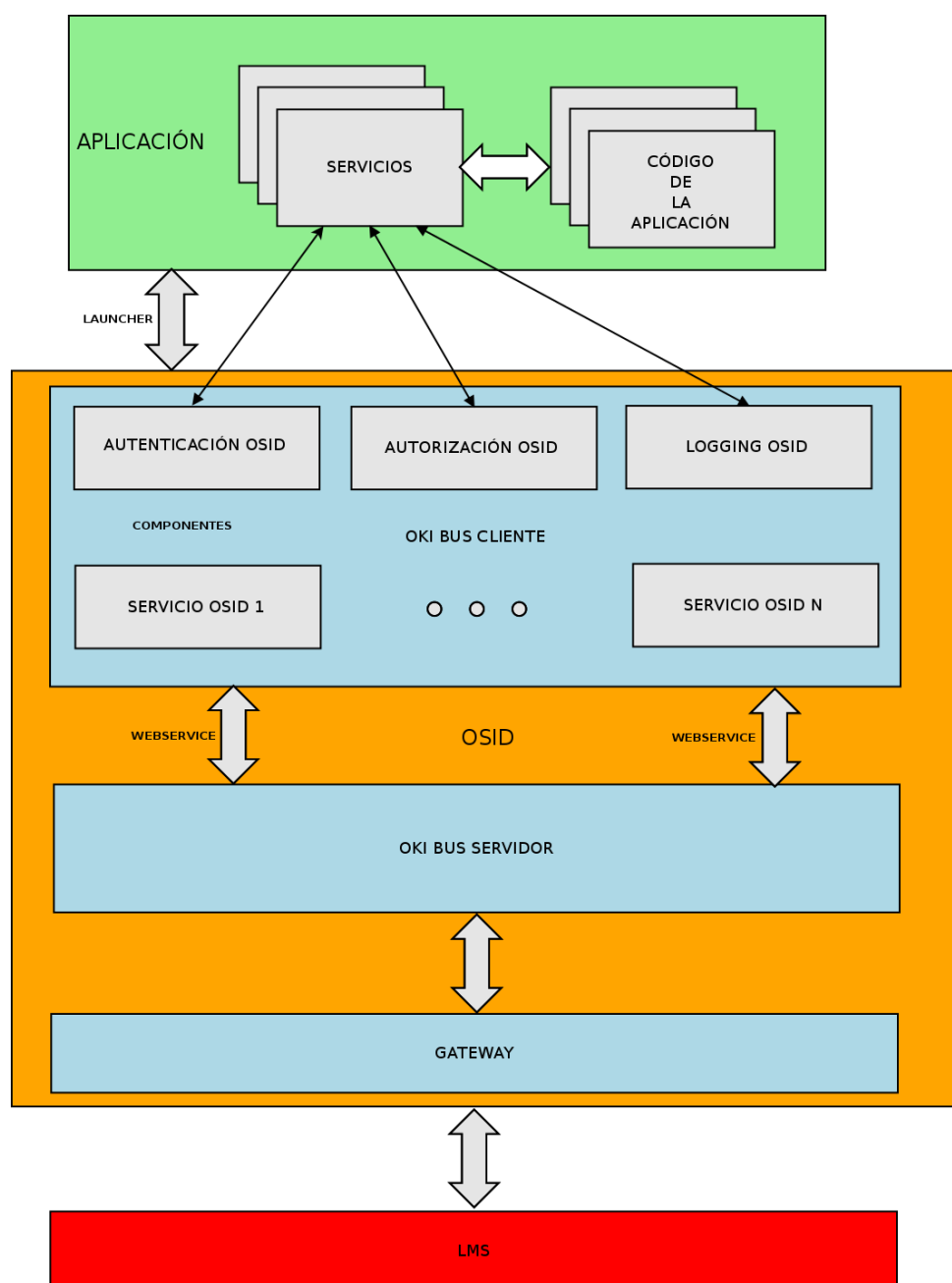


Figura 9: Diagrama OSID v2

OKI OSID 3

OSID Version 3

Las mejoras introducidas en la versión 3 en los siguientes ámbitos:

- **Interoperabilidad:** Posibilidad de intercambiar las implementaciones en caliente. Idealmente una aplicación cliente debería poder conectarse a cualquier proveedor utilizando exclusivamente los mecanismos proporcionados en OSID.
- **Usabilidad:** Se han incluido nuevas interfaces para servicios que eran demandados y no estaban cubiertos en versiones anteriores, fomentando una mayor reutilización de los códigos y proporcionando más implementaciones a bajo nivel en OSID para reducir los costes de desarrollo y proporcionar un mayor grado de flexibilidad y facilidad de integración.
- **Integración del Sistema:** en tanto que las definiciones de los interfaces garantizan la parte de las especificación, partes de la lógica dependerán de la implementación concreta y de la compatibilidad entre el consumidor con esta. Se ha mejorado para facilitar estos desarrollos la gestión de las configuraciones de las implementaciones.
- **Curva de aprendizaje:** La especificación permite a los desarrolladores que son los que al final crean el software en el marco de OSID presentarles un entorno y unos diagramas lo más cercanos al lenguaje de programación pero sin perder de vista el compromiso con la integridad de los interfaces y la interoperabilidad de OSID.
- **Incremento del número de paquetes,** rediseño de la semántica de las interfaces, mejora de los paquetes locale, search y catalog, servicios de cola (masivos) y reglas.

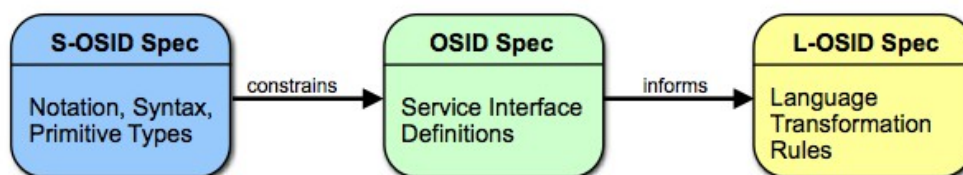
En cuanto a la terminología de OSID podemos hablar de las siguientes entidades dentro de OSID:

- **Packages (Paquetes o servicios):** Conjuntos de interfaces que definen el comportamiento de los datos y las operaciones permitidas en el ámbito de un servicio. A su vez estas interfaces se articulan en base a gestores, sesiones, objetos, relaciones, condiciones, reglas y resultados, que se instancian como métodos en Java pertenecientes a diferentes clase y que se encuentran dentro de los diferentes paquetes de OSID.
- **Consumidor OSID:** Aplicación que hace uso de las interfaces que proporciona OSID.
- **Proveedor:** Aplicación que implementa lo especificado por las interfaces OSID
- **Adaptador OSID:** aplicación que forma parte tanto del consumidor como del productor OSID y que permite la comunicación entre ambas para permitir el uso de las abstracciones definidas en el proveedor y consumidor.

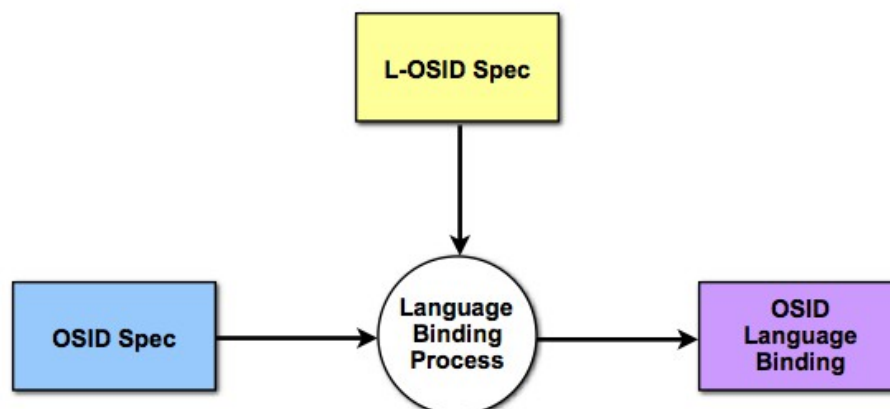
OKI OSID 3

El objetivo de la especificación de OSID con su sintaxis es proporcionar mecanismos para la definición de los elementos del software con la arquitectura basada en servicios de forma que se puedan trasladar estas definiciones a diferentes lenguajes de programación para las implementaciones concretas. Para ellos se hace uso de la notación, estructura semántica y primitivas presentes en OSID para definir las interfaces en el lenguaje natural con una sintaxis neutral utilizando los elementos de la especificación de la estructura de OSID (S-OSID).

Además para cada lenguaje de programación soportado por OSID, existen una especificación OSID para ese lenguaje en concreto (L-OSID) que permite sin perder la abstracción del medio del lenguaje hacerla compatible con el lenguaje en cuestión tanto en sintaxis como en filosofía. Gráficamente:



The Relationships among the specifications.



The OSID language binding process.

Figura 10: Relación entre OSID, S-OSID y L-OSID

OSID dispone de los siguientes paquetes de interfaces, si bien los que aparecen en rojo contienen a su vez subpaquetes con más funcionalidades (hasta 162 paquetes[OSIDrcu]):

OKI OSID 3



Figura 11: Paquetes de interfaces presentes en OSID v 3.0.0rc1

A continuación se va a realizar un análisis de los servicios de autenticación, autorización y logging que proporciona OSID, analizando las fortalezas y vulnerabilidades presentes en ellos.

Authentication

Define un servicio para que aplicaciones nativas puedan adquirir las credenciales de autenticación para el acceso a un punto remoto y define los medios para aplicaciones de servidor para validar y/o asignar las credenciales en un agente. Las credenciales pueden recibirse en cualquier formato, incluyendo datos recuperados desde una sesión HTTP.

Competencias

Hacer transparente la adquisición de las credenciales en la aplicación nativa.

Validar las credenciales recibidas en la aplicación del servidor.

Asignar las credenciales en un agente.

Gestionar las cuentas de usuario

Gestionar el anillo de claves.

Modelo de autenticación

A continuación podemos ver en contraposición el diagrama clásico de autenticación de un usuario en un sistema mediante una clave y cómo se trasladaría al modelo OSID:

OKI OSID 3

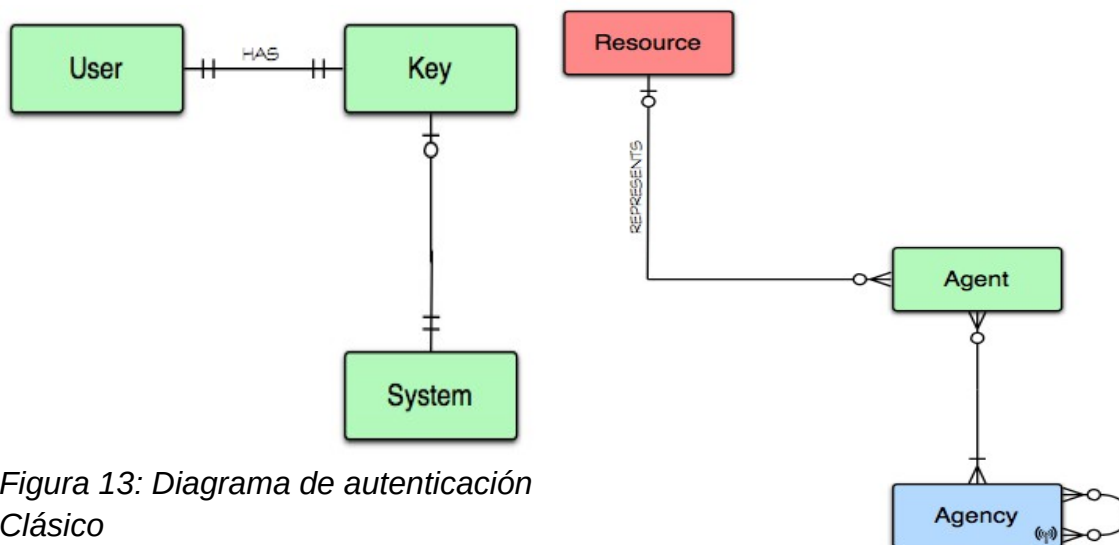


Figura 13: Diagrama de autenticación Clásico

Figura 12: Diagrama de autenticación OSID

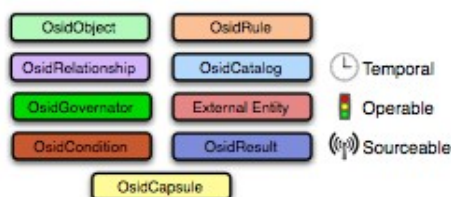


Figura 14: Leyenda Diagramas OSID

Para entender la transición de uno a otro, el primero de ellos representa a un usuario que tiene una clave que le permite el acceso a un sistema.

En OSID se define un agente como un individuo o proceso que interactúa con el sistema y forma parte de él como entidad. A su vez ese agente contará con un displayId, un Id, será de un tipo (type) y tiene una serie de propiedades (Properties). Un agente es asignado a/representa un único recurso, pero a su vez un recurso puede estar representado/asignado a múltiples agentes. Por ejemplo en el caso de una persona, esta puede utilizar varias tecnologías de autenticación, cada una de ellas con una identidad de autenticación diferente. El desacoplamiento de la identidad autenticada de la identidad de la persona, proporciona un medio de integrar múltiples servicios en los que existen diferentes identidades de autenticación para una misma persona, que afectan al tratamiento de la autorización. A su vez los agentes son organizados en catálogos (OSIDcatalog) de Agencias (Agency) federables.

Es destacable el hecho de que la autorización es un servicio diferenciado que se encarga de ver qué funciones está autorizado a realizar el agente y se verá más adelante, siendo por tanto la autenticación tan sólo responsable de la gestión de la identidad del agente.

OKI OSID 3

Cada agente de un recurso puede ser utilizado para garantizar la definición distintos niveles de seguridad, si bien también se podría obtener optar por definir un pseudo-recurso por cada agente. Estos niveles de seguridad de se pueden vincular al tipo de agente y gestionarse dentro del servicio de autorización OSID. El tipo de agente podría servir como indicador de la fortaleza de la autenticación y aunque podría ir de la mano de una tecnología de autenticación específica, un acoplamiento demasiado fuerte con una determinada tecnología también supondría limitar la flexibilidad del mecanismo de autenticación.

Para poder realizar la funcionalidad y competencias descritas el paquete authentication cuenta con varios subpaquetes o subservicios que se detalla a continuación

Subpaquete: Authentication Process

La autenticación OSID ayuda al consumidor OSID a adquirir y validar credenciales de autenticación sin tener que sin tener que gestionar los detalles de un entorno de autenticación en particular. La autenticación se hace en dos pasos:

- El usuario adquiere un conjunto de credenciales y esas credenciales se transportan a un par remoto
- El pare remoto valida esas credenciales y se determina la identidad del usuario.

Este proceso se refleja en la autenticación OSID 3 con la definición de las correspondientes dos sesiones OSID:

- AuthenticationAcquisitionSession: Una sesión para adquirir las credenciales de un usuario y enviarlas a un par remoto para su autenticación.
- AuthenticationValidationSession: Una sesión para recibir y validar las credenciales de autenticación de un par remoto que se solicita autenticarse.

El transporte de las credenciales de autenticación es responsabilidad del consumidor de la autenticación OSID, apoyada normalmente en un protocolo de aplicación existente. Existen métodos para obtener y proporcionar credenciales en cada extremo, siendo proporcionado por el proveedor de autenticación OSID con posibilidad de tener más de un formato para las credenciales, incluidos mecanismos de reto-respuesta.

Círculo de Confianza (Trust)

En la Autorización OSID, las autorizaciones pueden ser gestionadas por un conjunto de agentes relacionados con un recurso. El conjunto de agentes se pueden seleccionar basándose en el nivel de confianza del mecanismo de autenticación. Un objeto Trust (confianza) representa una categoría de agentes generados a partir de un mecanismo de autenticación que cuentan con un nivel de confianza especificado por su autenticación.

Los objetos Trust no son gestionados explícitamente en el proceso de autenticación OSID. Nos servirán para facilitar la comunicación (orquestración) entre un proveedor de autenticación de OSID y un proveedor de autorización

OKI OSID 3

OSID. Un proveedor de autorización OSID puede consultar la CircleOfTrustSession para determinar si el agente que ha recibido pertenece a un Trust especificado en uno de las autorizaciones “autorizadas”.

Por ejemplo, una autorización creada mediante la especificación de un recurso. El recurso puede ser una persona individual o un grupo de empleados. Mientras que los empleados puedan estar autorizados a leer su correo corporativo utilizando su cuenta, las peticiones en el sistema ERP debe hacerse utilizando el sistema de autenticación de empresa, e incluso tal vez un tipo específico de credenciales. Una autorización puede ser creada para un grupo de empleados, pero restringido a un Trust, donde el Trust representa a cualquier agente relacionado con el conjunto de los empleados que se haya autenticado de la forma deseada.

La multiplicidad de agentes por cada recurso, así como la alineación con un proveedor de autorización OSID, es una consideración a tener en cuenta en el diseño de un proveedor de autenticación de OSID. No solo identifican la autenticación como una entidad singular, sino que también pueden representar algo sobre el estilo de autenticación que se utiliza para realizar una autorización.

A continuación se puede ver cómo se especifica lo anterior en el modelo OSID:

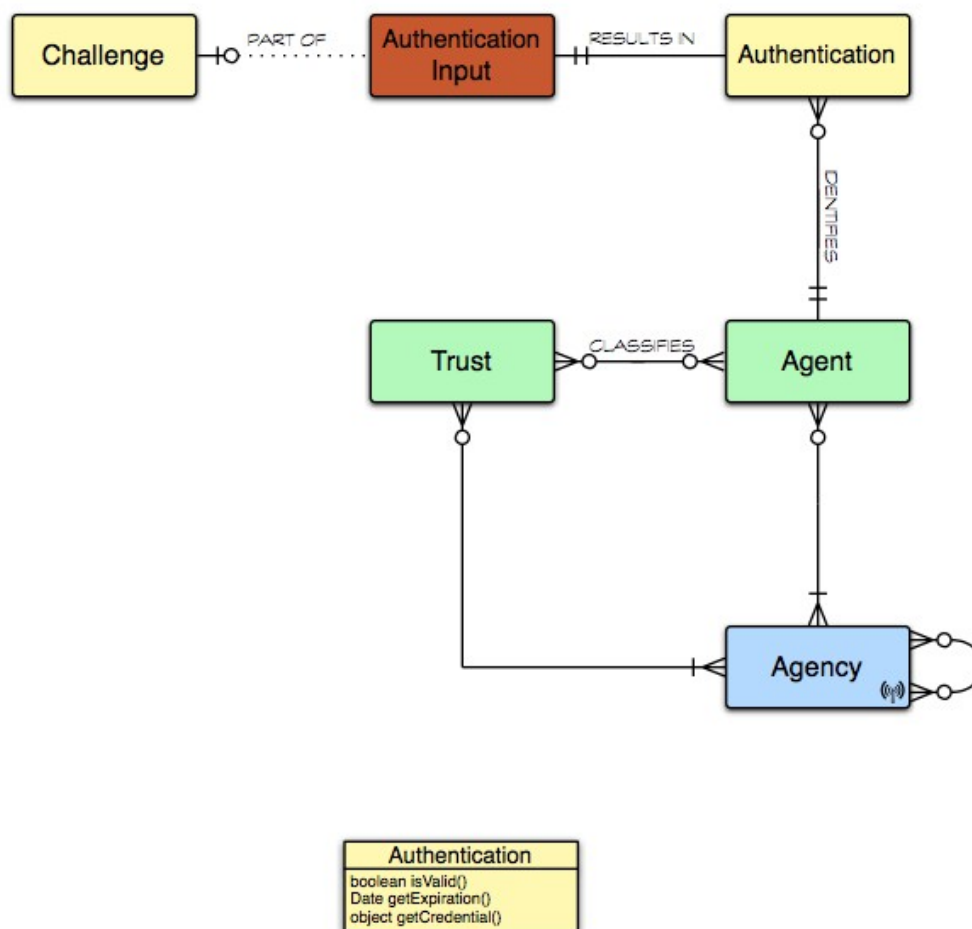


Figura 15: Diagrama del Proceso de Autenticación en OSID

OKI OSID 3

Subpaquete: Authentication Keys

La autenticación de claves define una clave asociada con un agente para la gestión de las claves. Las claves son catalogadas en agencias y asociadas a un agente (un agente puede tener varias claves, pero una clave sólo puede estar asociada con un agente).

A continuación se puede ver cómo se especifica lo anterior en el modelo OSID:

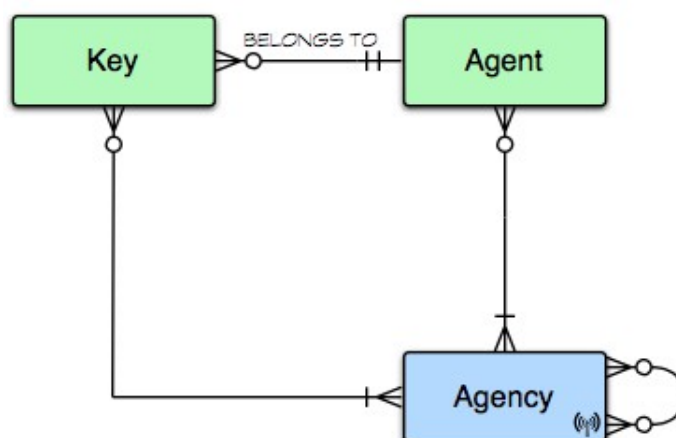


Figura 16: Diagrama del Claves de Autenticación en OSID

Subpaquete: Authentication Key Batch

Gestiona las relaciones de claves, agencias y agentes de forma masiva.

Subpaquete: Authentication Batch

Gestiona las relaciones entre agencias y agentes de forma masiva.

Authorization

Este paquete (servicio de autorización) gestiona y consulta las autorizaciones que no son más que relaciones entre elementos OSID (OSIDRelationship) que definen quién puede hacer qué. La gramática de una autorización incluye el sujeto o actor (quién), la acción o verbo (qué hacer) y el objeto o contexto (a quién). Estos tres componentes son necesarios en una autorización para que tenga un significado explícito y una instancia de los mismos conforma una autorización como la que sigue:

- Agent: el actor (ej: josedcz@uoc.edu)
- Function: la acción (ej: acceder a un test)

OKI OSID 3

- Qualifier: el objeto o contexto dentro de una función (ej: curso TFM LTI UOC dentro de Moodle)

La tupla anterior definiría un rol y el servicio básico de Autorización OSID se encarga de preguntar si un determinado agente está autorizado a realizar una determinada función o acción con un calificador, es decir comprobar que la tupla existe. El Agente se genera a partir de un servicio de autenticación (como se ha visto en el apartado anterior) y la función y el calificador son proporcionados por la aplicación consumidora (un LMS por ejemplo).

Se pueden distinguir dos tipos de autorizaciones:

- Explícitas son las gestionadas directamente como se especificaba anteriormente.
- Implícitas son aquellas que se derivan por herencia de recursos funciones y calificadores.

A su vez las autorizaciones y calificadores pueden organizarse en estructuras denominadas Vault Cataloging, que se sería algo así como un catálogo de credenciales. Se trata de estructuras jerárquicas donde cada nodo incluye los datos de autorización correspondientes a sus hijos, teniendo en el raíz de la estructura todas las autorizaciones disponibles.

Además existe un mecanismo para la notificación de cambios en el servicio soportado a través de las funciones AuthorizationNotificationSession, FunctionNotificationSession and VaultNotificationSession.

El modelo de autenticación clásico sería el siguiente:

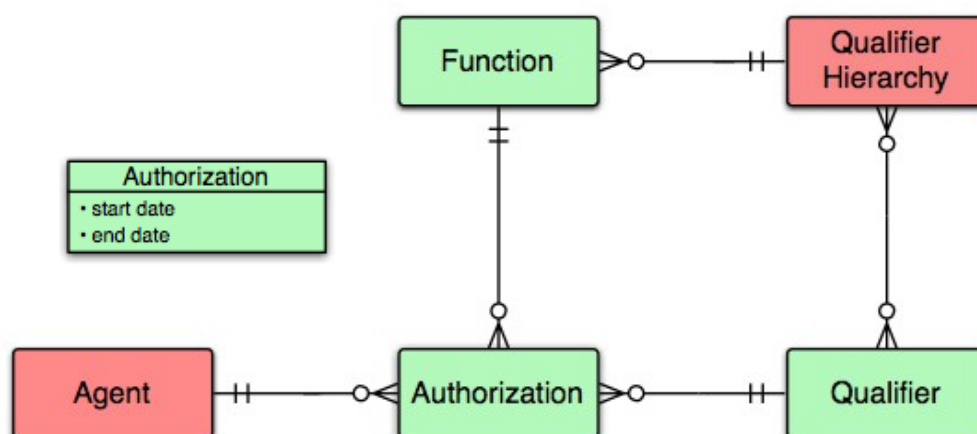


Figura 17: Modelo de Autorización Clásico

Frente al modelo de Autorización presentado por OSID:

OKI OSID 3

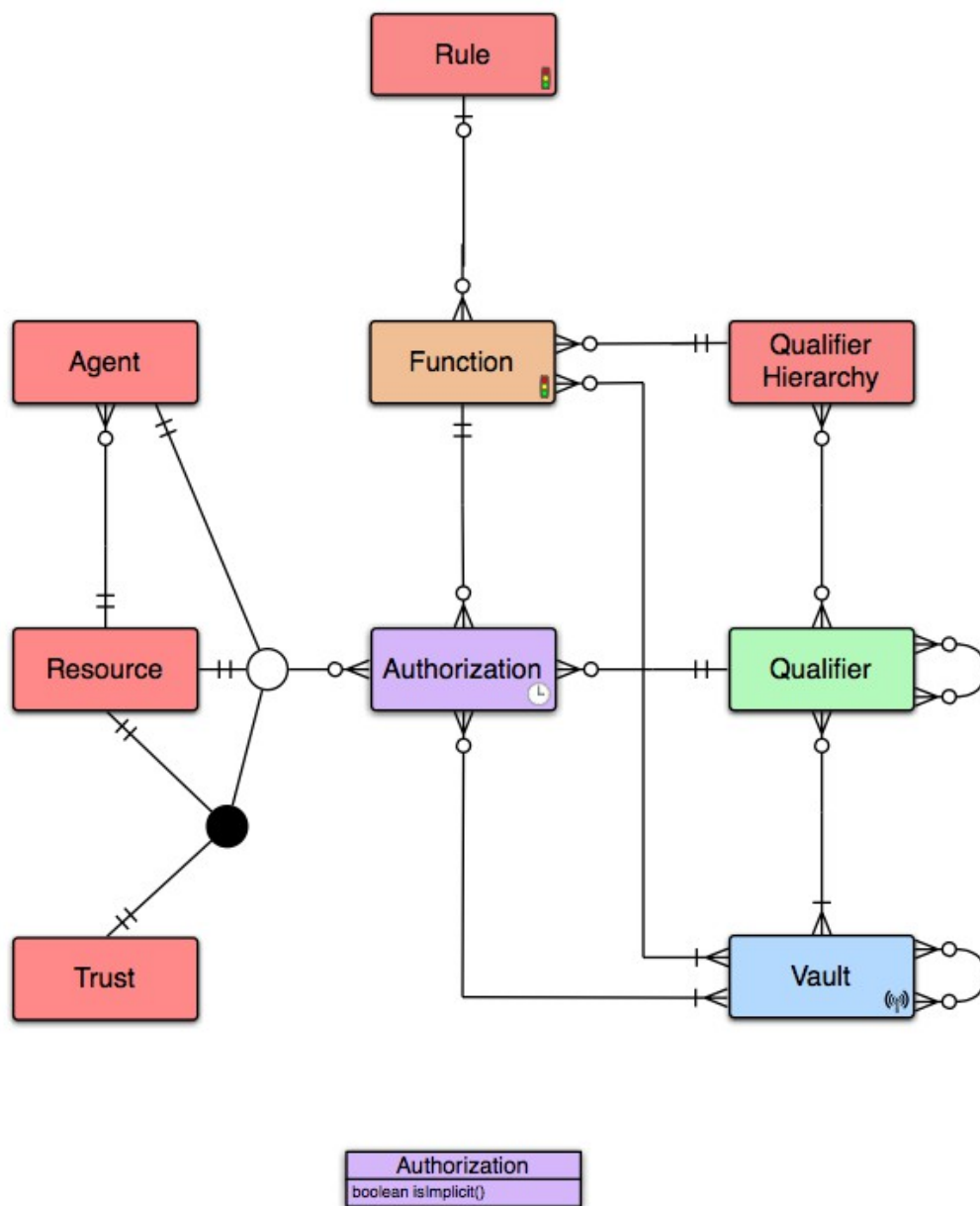


Figura 18: Modelo de Autorización en OSID

Subpaquete: Authorization Rules

Para la gestión de las autorizaciones en el tiempo este paquete permite activar/desactivar las autorizaciones en base a fechas de vigencia de las autorizaciones. Con estas reglas se especifica si se quiere extender la vigencia de una autorización, cuándo expiraría o rescatar autorizaciones ya caducadas. A su vez estos paquetes son catalogados dentro del catálogo de credenciales (Vault cataloging).

A continuación se puede ver cómo se especifica lo anterior en el modelo OSID:

OKI OSID 3

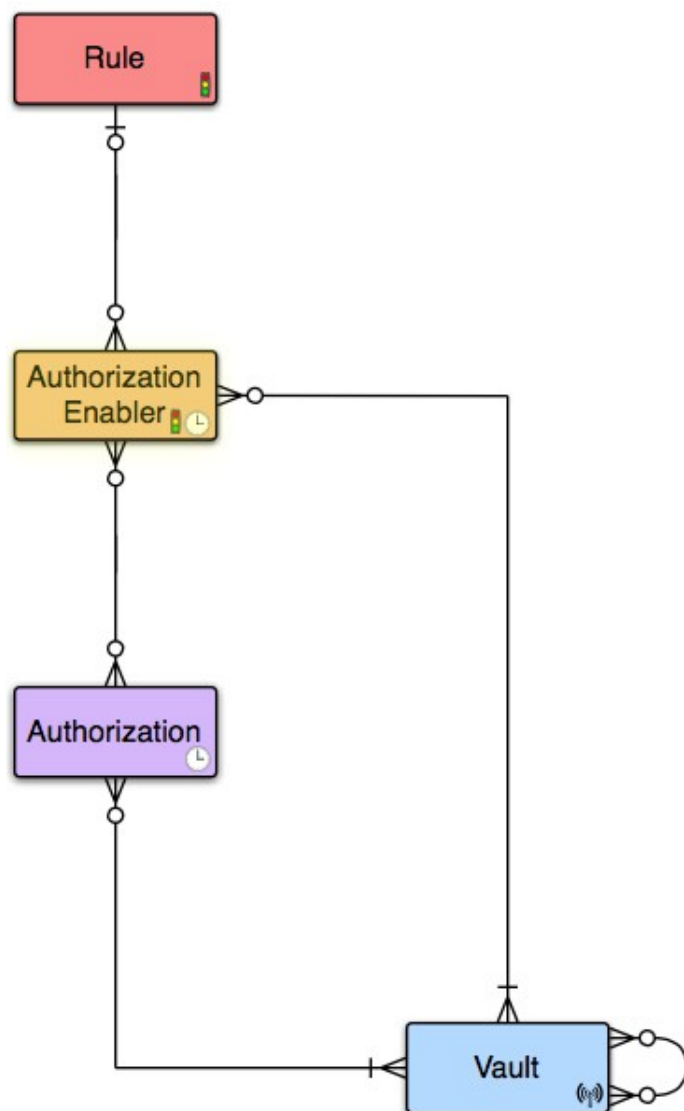


Figura 19: Diagrama del Reglas de Autorización en OSID

Subpaquete: Authorization Batch

Gestiona las relaciones entre funciones, cualificadores y autorizaciones de forma masiva.

Logging

OKI OSID 3

El servicio de logging proporciona medios para el registro de los logs acceso (lectura y escritura). Se define un log como una colección de entradas y cada entrada está compuesta por los siguientes campos:

- Type (tipo)
- timestamp (sello de tiempo)
- Agent (Agente asociado al Recurso)
- Record (registro)

A continuación se puede ver cómo se especifica lo anterior en un modelo clásico y en OSID:

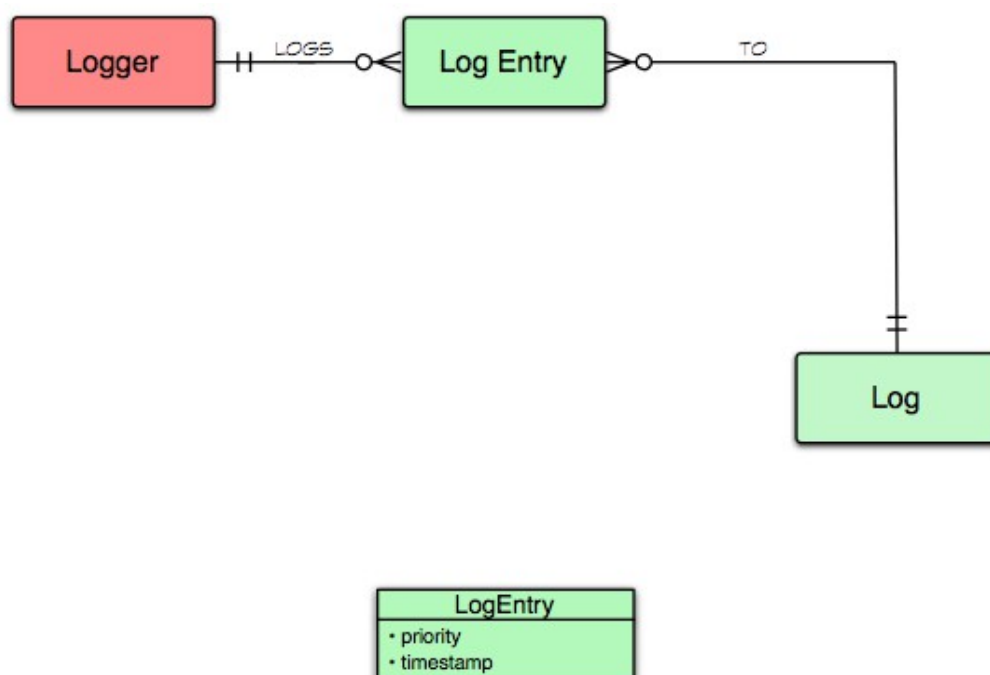


Figura 20: Diagrama Clásico de Logging

OKI OSID 3

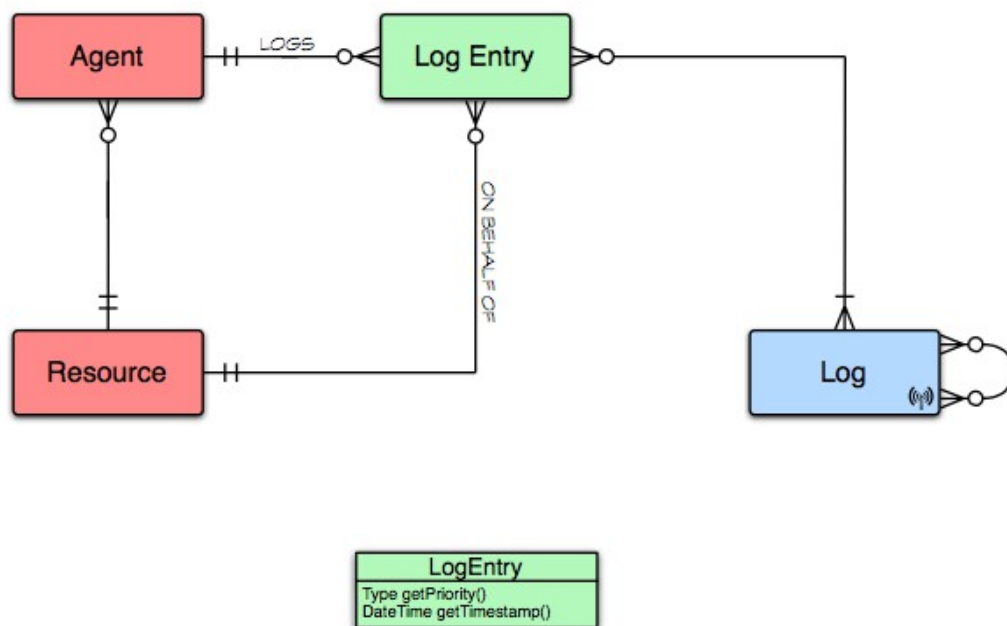


Figura 21: Diagrama del Logging en OSID

INTEGRACIÓN LTI – OKI OSID

Integración LTI – OKI OSID

Introducción

En este apartado se va a hacer un estudio de como integrar los servicios de LTI y de OKI OSID. En apartados anteriores se ha hecho una revisión del funcionamiento y versiones de LTI, así como de los paquetes de autenticación, autorización y logging de OSID.

En este punto también cabe mencionar que si bien se ha hecho el estudio de esta alternativa para la integración de herramientas externas dentro de un LMS, existen otras alternativas como puedan ser soluciones del tipo GLUE! (Group Learning Uniform Environment <http://www.gsic.uva.es/glue/>), que consiste en introducir una capa intermedia entre TC y TP que permite una integración ligera, compuesta de un núcleo y adaptadores para cada lado (herramienta externa y LSM respectivamente).

Servicios presentes en OSID

A continuación vamos a ver qué servicios de los presentes en los paquetes de OSID podrían ser de aplicación para el paradigma definido en la comunicación de herramientas definida en LTI:

- Authentication (Autenticación):
 - comprueba si el usuario que quiere usar la aplicación está registrado al sistema y si puede tener acceso.
- Authorization (Autorización):
 - Permitiría definir quién está autorizado a hacer qué y cuando. De este modo si autorizaría o no la ejecución de la acción.
- Logging (Registros):
 - comprobaría cuando (fecha y hora) un usuario accede a una aplicación concreta.
- Locale (idioma):
 - permitiría modificar cómo se muestra el contenido dentro de las aplicaciones en función del idioma y el charset definido en cada entorno.
- Gradding (Calificación):
 - si bien este servicio no ha sido desgranado en el apartado dedicado a OSID, sería necesario para la gestión de las calificaciones que proporciona LTI.
- Configuration:

INTEGRACIÓN LTI – OKI OSID

- es necesario para poder almacenar y administrar la configuración de los servicios.

Servicios presentes en LTI

LTI 1.1.1

Podemos hablar de 2 servicios principales:

- Autenticación (se encargaría de autenticar las identidades de un TP y un TC para que se establezca la comunicación entre ambos):
 - Establecimiento de las credenciales de dominio para un TP
 - Establecimiento de las credenciales a nivel de enlace
 - Gestión de las credenciales para enlaces importados desde un Common Cartridge (CC)
- Autorización (se encargaría de permitir la ejecución de los contenidos en base a la existencia de una autenticación previa y a los roles del sistema):
 - Ejecución de un contenido enlazado autorizado desde un contexto (TC)
 - Ejecución de un contenido enlazado autorizado importado desde un Common Cartridge (CC), con credenciales establecidas previamente (secreto o clave)
 - Ejecución de un contenido enlazado autorizado importado desde un Common Cartridge (CC), sin credenciales establecidas
- Calificación (calificar una actividad sobre un recurso realizada por un usuario del TC en una sesión):
 - Comunicar una puntuación decimal (0.0 y 1.0) desde la herramienta (TP) al TC (por ejemplo un LMS)
- Idioma:
 - En función a lo establecido en el parámetro de la comunicación `launch_presentation_locale` se podría seleccionar un idioma para la presentación de un recurso.

LTI 2.0

- Autenticación:
 - En el lado de la plataforma consumidora TC (Tool Consumer) es necesario contar con los siguientes pasos:
 - Publicar el Profile del TC en alguna URL accesible.

INTEGRACIÓN LTI – OKI OSID

- Creación de una interfaz de usuario para que se pueda pedir registrar una herramienta por parte del administrador del TC.
- Implementación del Servicio de Tool Proxy
- En el lado de la herramienta externa (TP) es necesario contar los siguientes mecanismos:
 - Gestor de mensajes para permitir el registro del TP:
 - Perfil de la herramienta (Tool Profile):
 - Creación de un contrato de seguridad (security contract) que defina la integración con el TC.
 - Creación del Tool Proxy
 - Registro del Tool Proxy
 - Devolución del control al TC
- Autorización
 - En el lado de la plataforma consumidora TC (Tool Consumer) es necesario contar con los siguientes pasos:
 - Creación de una interfaz de usuario para hacer accesible el Tool Proxy registrado
 - Creación de una interfaz de usuario que permita a los creadores de contenidos dentro del TC crear los enlaces LTI asociados con cada tipo de recurso
 - Servicio para ejecutar los contenidos enlazados
 - En el lado de la herramienta externa (TP) es necesario contar los siguientes mecanismos:
 - Gestor de mensajes para las peticiones de ejecución de los contenidos proporcionados por la herramienta
- Calificación
 - En el lado de la plataforma consumidora TC (Tool Consumer) es necesario contar con los siguientes pasos:
 - Soporte para gestionar la capacidad o prestación (capability) de crear un resultado (calificación) así como modificarlo (recalificarlo mediante peticiones PUT) para un tipo de recurso definido en el TC y proporcionado por el TP.
 - En el lado de la herramienta externa (TP) es necesario contar los siguientes mecanismos:
 - Enviar una calificación al TC así como modificarlo (recalificarlo mediante peticiones PUT) para un tipo de recurso definido en el TC y proporcionado por el TP.
- Locale (idioma):

INTEGRACIÓN LTI – OKI OSID

- permitiría modificar cómo se muestra el contenido dentro de las aplicaciones en función del idioma y el charset definido en cada entorno en base al `launch_presentation_locale` y en función también de lo definido en el tool proxy concreto pudiendo tener diferentes idiomas para una mismo TP y TC en función de las necesidades.

Definición del Wrapper

Gráficamente el Wrapper que integraría el soporte para la comunicación entre LTI y OSID sería como sigue:

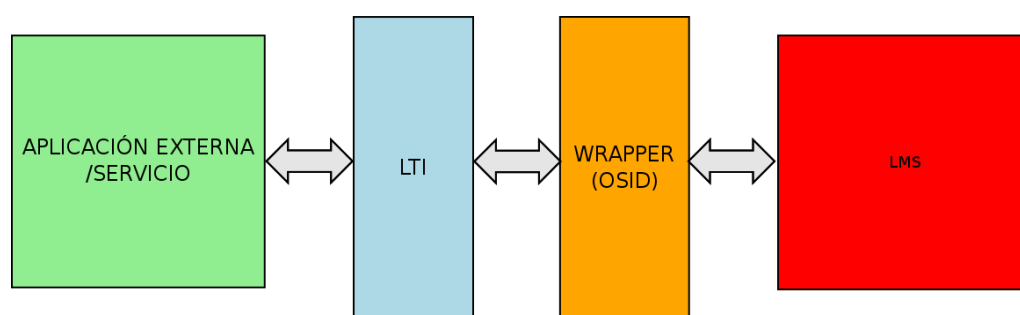


Figura 22: Esquema de Comunicación con LTI y OSID

IMPLEMENTACIÓN

Implementación

En este capítulo se describe la forma de llevar a cabo la implementación de una aplicación LTI que permita ser integrada en una aplicación que soporte el protocolo, por ejemplo un Moodle.

Instalación de Tomcat en un Mac OS 10.9.1 Mavericks

Seguir los pasos de: <http://wolfpaulus.com/journal/mac/tomcat7/> :

- Descargar Tomcat
 - get <http://tomcat.apache.org/download-70.cgi>
- Moverlo a la carpeta /usr/local
 - sudo mv apache-tomcat-7.0.50/ /usr/local/
- Crear un alias para ubicar fácilmente Tomcat
 - sudo ln -s /usr/local/apache-tomcat-7.0.50/ /Library/Tomcat
- Dar permisos al usuario actual
 - sudo chown -R jose /Library/Tomcat
- Dar permisos de ejecución usuario actual
 - sudo chmod +x /Library/Tomcat/bin/*.sh
- Lanzar Tomcat
 - /Library/Tomcat/bin/startup.sh

Configuración del entorno de desarrollo

La aplicación también se puede lanzar desde el propio de desarrollo el tomcat, que quizá sea más cómodo para el desarrollo. Sin embargo voy a explicar cómo sería:

Ahora se procede a poner disponible la aplicación LtiToolProvider-1.1 a partir de la cual se hará la implementación, añadiéndole una adaptación con la parte de gestión de consumidores disponible en la aplicación rating. De esta última sólo se utiliza la parte de admin, el css que necesita la pantalla de configuración, las imágenes de la carpeta imágenes (no todas) y las clases Config y Db, que se pueden incluir por ejemplo en el propio paquete org.oscelot.lti.tp.

Se crea en IDE de desarrollo por ejemplo NetBeans o Eclipse con soporte para JEE

IMPLEMENTACIÓN

Hay que satisfacer las siguientes dependencias de las bibliotecas (se incluye el directorio /Library/Tomcat/webapps/Rating-1.1.0/WEB-INF/lib/ en el path o se incluyen directamente en las librerías del proyecto):

- `servlet-api` (si se usa tomcat 7.0 no es necesaria)
- [commons-httpclient](#) (commons-httpclient-3.0.1.jar /Library/Tomcat/webapps/Rating-1.1.0/WEB-INF/lib/)
- [jdom](#) (cp -R jdom/*.jar /Library/Tomcat/webapps/Rating-1.1.0/WEB-INF/lib/) tiene que ser la versión 1.1.3
- [oauth-core.jar](#) (ya estaba cp oauth-core-20100601.jar /Library/Tomcat/webapps/Rating-1.1.0/WEB-INF/lib/)
- [commons-codec](#) (cp commons-codec-1.7.jar /Library/Tomcat/webapps/Rating-1.1.0/WEB-INF/lib/)
- [gson](#) (google-gson-2.2.4/*.jar /Library/Tomcat/webapps/Rating-1.1.0/WEB-INF/lib/)
- [LtiToolProvider](#) (cp LtiToolProvider-1.1.01.jar /Library/Tomcat/webapps/Rating-1.1.0/WEB-INF/lib/)
- [mysql-connector-java](#) (cp -R mysql-connector-java-5.1.28/*.jar /Library/Tomcat/webapps/Rating-1.1.0/WEB-INF/lib/)

Por otro lado habría que crear un servidor apache tomcat 7.0 con el Asistente y luego arrancarlo.

Creación de la Base de Datos

Se ha instalado una instancia de mysql (bastaría por ejemplo con tener un XAMPP). Se crea la BD, el usuario y las tablas (en mysql):

```
create database uoc;

CREATE USER 'uoc'@'localhost' IDENTIFIED BY 'SnFbL6p4crUCt9ue';

GRANT USAGE ON * . * TO 'uoc'@'localhost' IDENTIFIED BY 'SnFbL6p4crUCt9ue'
WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0
MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;

GRANT ALL PRIVILEGES ON `uoc` . * TO 'uoc'@'localhost';

CREATE TABLE lti_consumer (
  consumer_key varchar(255) NOT NULL,
  name varchar(45) NOT NULL,
  secret varchar(32) NOT NULL,
```

IMPLEMENTACIÓN

```
Iti_version varchar(12) DEFAULT NULL,  
consumer_name varchar(255) DEFAULT NULL,  
consumer_version varchar(255) DEFAULT NULL,  
consumer_guid varchar(255) DEFAULT NULL,  
css_path varchar(255) DEFAULT NULL,  
protected tinyint(1) NOT NULL,  
enabled tinyint(1) NOT NULL,  
enable_from datetime DEFAULT NULL,  
enable_until datetime DEFAULT NULL,  
last_access date DEFAULT NULL,  
created datetime NOT NULL,  
updated datetime NOT NULL,  
PRIMARY KEY (consumer_key)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
CREATE TABLE Iti_context (  
  consumer_key varchar(255) NOT NULL,  
  context_id varchar(255) NOT NULL,  
  Iti_context_id varchar(255) DEFAULT NULL,  
  Iti_resource_id varchar(255) DEFAULT NULL,  
  title varchar(255) NOT NULL,  
  settings text,  
  primary_consumer_key varchar(255) DEFAULT NULL,  
  primary_context_id varchar(255) DEFAULT NULL,  
  share_approved tinyint(1) DEFAULT NULL,  
  created datetime NOT NULL,  
  updated datetime NOT NULL,  
  PRIMARY KEY (consumer_key, context_id)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
CREATE TABLE Iti_user (  
  consumer_key varchar(255) NOT NULL,  
  context_id varchar(255) NOT NULL,  
  user_id varchar(255) NOT NULL,  
  Iti_result_sourcedid varchar(255) NOT NULL,  
  created datetime NOT NULL,  
  updated datetime NOT NULL,  
  PRIMARY KEY (consumer_key, context_id, user_id)
```

IMPLEMENTACIÓN

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE lti_nonce (  
  consumer_key varchar(255) NOT NULL,  
  value varchar(32) NOT NULL,  
  expires datetime NOT NULL,  
  PRIMARY KEY (consumer_key, value)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE lti_share_key (  
  share_key_id varchar(32) NOT NULL,  
  primary_consumer_key varchar(255) NOT NULL,  
  primary_context_id varchar(255) NOT NULL,  
  auto_approve tinyint(1) NOT NULL,  
  expires datetime NOT NULL,  
  PRIMARY KEY (share_key_id)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE lti_context  
  ADD CONSTRAINT lti_context_consumer_FK1 FOREIGN KEY (consumer_key)  
  REFERENCES lti_consumer (consumer_key);
```

```
ALTER TABLE lti_context  
  ADD CONSTRAINT lti_context_context_FK1 FOREIGN KEY  
  (primary_consumer_key, primary_context_id)  
  REFERENCES lti_context (consumer_key, context_id);
```

```
ALTER TABLE lti_user  
  ADD CONSTRAINT lti_user_context_FK1 FOREIGN KEY (consumer_key,  
  context_id)  
  REFERENCES lti_context (consumer_key, context_id);
```

```
ALTER TABLE lti_nonce  
  ADD CONSTRAINT lti_nonce_consumer_FK1 FOREIGN KEY (consumer_key)  
  REFERENCES lti_consumer (consumer_key);
```

```
ALTER TABLE lti_share_key  
  ADD CONSTRAINT lti_share_key_context_FK1 FOREIGN KEY  
  (primary_consumer_key, primary_context_id)
```


IMPLEMENTACIÓN

REFERENCES lti_context (consumer_key, context_id);

El esquema de la base de datos sería:

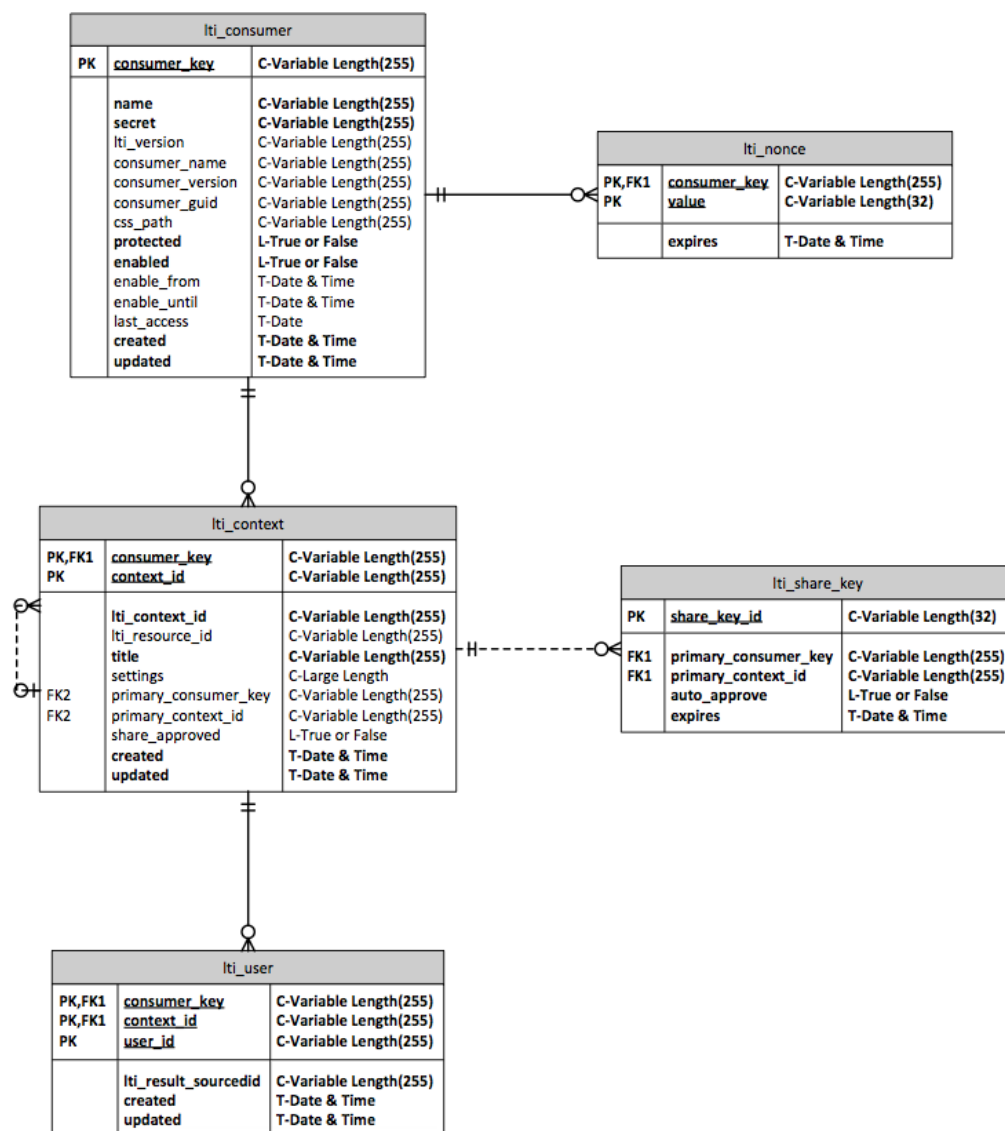


Figura 23: Esquema de la Base de datos de LTI Tool Provider

Archivo de Configuración del enganche con la Base de Datos:

Habría que configurar el archivo Config.java así:

```

    /// Application setting

    public static final String APP_NAME = "TMF_UOC";
    
```

IMPLEMENTACIÓN

```
/// Database connection settings

public static final String DB_NAME = "jdbc:mysql://localhost/uoc"; //
jdbc:mysql://localhost/MyDb"

public static final String DB_USERNAME = "uoc";

public static final String DB_PASSWORD = "SnFbL6p4crUCt9ue";

public static final String DB_TABLENAME_PREFIX = "";
```

Configuración de puertos y seguridad

Una vez habilitada la herramienta hay que ponerla disponible al exterior y si nos encontramos detrás de una infraestructura con una red local y contamos con una ip dinámica, se puede proceder así:

- Abrir puerto 80 en el servidor
 - En Mac OS bastaría con deshabilitar el firewall
- Redireccionar el puerto 8080, el que usa tomcat al 80 en el servidor
 - En el terminal con la instrucción: `ipfw add 100 fwd 127.0.0.1,8080 tcp from any to any 80 in`
- Abrir puerto 80 en el router para la ip del servidor
 - Añadir la regla a tal efecto
- Habilitar mapeo de una dirección dinámica ip con una url estática con un servicio para esta función, por ejemplo dyndns y que se ejecute el cliente para realizar la actualización cuando se cambie la ip.

Dynamic Domain Name System (DDNS)

DDNS

DDNS Service	DynDNS.org
User Name	corecorp2
Password
Host Name	tfmapp.dyndns.org
Type	Dynamic
Wildcard	<input type="checkbox"/>

Figura 24: Configuración Dynamic Domain Name System (DDNS) en el router

IMPLEMENTACIÓN

Manual de la aplicación

Para la aplicación sólo son necesarios dos ficheros jsp. El primero de ellos de configuración en la carpeta admin permite dar de alta consumidores en base a los siguientes parámetros:

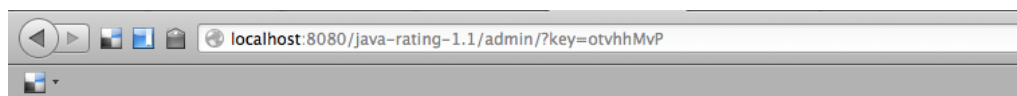
Nombre	Descripción	Obligatorio ?	Valor por defecto
Name	El nombre que se muestra en la instancia de consumidor	Sí	Ninguno
Key	Una cadena única para identificar al VLE que conecta con la aplicación	Sí	Ninguno
Secret	Secreto usado por oauth para lanzar la aplicación	Sí	Una cadena generada de forma aleatoria
Enabled?	Determina si las peticiones de ejecución de la herramienta deben ser aceptadas	No	No
Enable from	Fecha/hora DESDE la que se aceptan las peticiones de ejecución(ej. 1-Jan-2013 00:00)	No	Ninguno
Enable until	Fecha/hora HASTA la que se aceptan las peticiones de ejecución(ej. 1-Jan-2013 00:00)	No	Ninguno
Protected?	Indica si la petición de ejecución debe hacerse solo sin el Tool Consumer tiene el mismo GUID que el que se le pasa	No	No

Tabla 1: Parámetros de configuración de la aplicación

El funcionamiento de esta configuración es el definido en la aplicación rating y no se ha modificado por ser correcto. Además se ha dado de alta una aplicación consumidora.

Estudio de la Integración del estándar OKI OSID con Learning Tools Interoperability (LTI versiones 1.1.1 y 2.0)

IMPLEMENTACIÓN



Access to this page should be restricted to tool provider administrators only.

Name	Key	Version	Available?	Protected?	Last access	Options
Moodle UOC TFM	otvhhMvP		✓	✓	None	 

Update consumer

Name:*

Key:*

Secret:*

Enabled?

Enable from:

Enable until:

Protected?

* = required field

Update consumer

Cancel

Moodle UOC TFM

otvhhMvP

wyBKdBA4FRd01TFzHNJg4uAi7tGegVOt

☒

☒

Figura 25: Vista de configuración de herramientas consumidoras

El fichero del proveedor en sí mismo index.php en el raíz de la carpeta lo que hace es coger los parámetros que se pasan en la llamada desde Moodle comprueba que hay una clave para ese consumidor (si no hay lo indica por pantalla y no se mostraría nada).

Una vez comprobado que es un cliente se muestran todos los datos que se han recibido desde moodle incluidos nombre de usuario, nombre y apellidos, clave del curso, nombre del curso y rol, etc.

Ahora queda comprobar que es un cliente válido para ello se compara la firma que se puede generar con los datos almacenados en la base de datos (consumidores dados de alta en la aplicación de administración) con los datos recibidos desde Moodle y se generan 2 mensajes y se validan, indicando si es correcto no.

A continuación se ven ambas vistas de configuración en Moodle y de Ejecución:

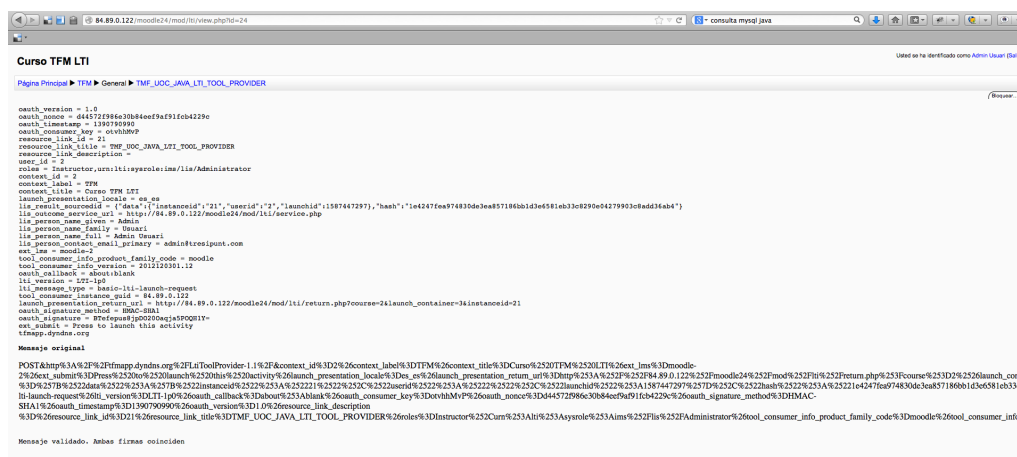


Figura 26: Vista de ejecución de la herramienta en Moodle

Estudio de la Integración del estándar OKI OSID con Learning Tools Interoperability (LTI versiones 1.1.1 y 2.0)

IMPLEMENTACIÓN

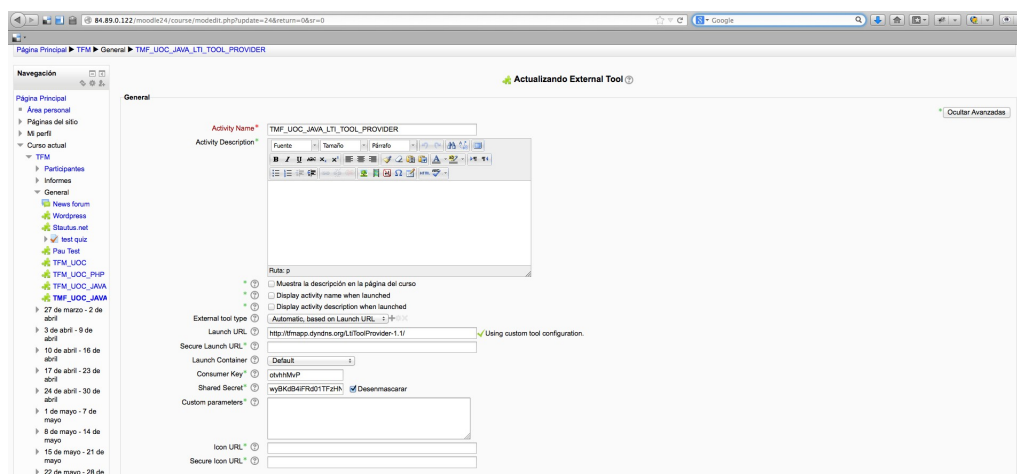


Figura 27: Configuración de Tool Provider con Clave y Secreto en Moodle

El resultado de la implementación se distribuye como un zip para trabajar con él en netbeans o como un .war para su despliegue.

CONCLUSIONES

Conclusiones

Después del estudio de herramientas de integración, si bien LTI es la mejor opción para la integración de herramientas mediante el cumplimiento del estándar que se define para la comunicación, ya que este tras sus revisiones y mejoras permite una integración sencilla y estándar de la herramienta y la posibilidad de contar con comunicación entre LSM y herramienta externa, para por ejemplo tener las calificaciones de los ejercicios que se realizan en la herramienta externa, también en el entorno LMS. Además la versión 2.0 de LTI si bien se encuentra aún en fase de borrador, promete avances en el autoaprovisionamiento (Tool Proxy) y la extensibilidad, para buscar un modelo más “empresarial” con vocabularios extensibles (ontologías RDF), con la sustitución del XML de POX por JSON y la articulación de los ToolProxy con los Perfiles y contratos, en búsqueda de un modelo plug&play, en base a como se comunican ahora las herramientas (Tools) con los LMS (TC) y descarga de parte del procesamiento al lado del cliente. Es ahí donde está la gran mejora en cuanto a la implementación a ese proceso de negociado para el establecimiento del Tool Proxy en base a unas reglas definidas de antemano y con el soporte de JSON y basado en REST y que permite la negociación entre el consumidor y el productor.

Sin embargo pese al optimismo en cuanto a las novedades tecnológicas que promete esta nueva versión también hay que mencionar, la existencia de una cierta falta de compromiso o dudas con el modelo de LTI por parte de muchas instituciones que aún no apuestan realmente por él viéndose esto reflejado por ejemplo en el número de aplicaciones certificadas para el uso de IMS LTI en su versiones 1.0, 1.1 y 1.1.1 es de 119, 74 y 46 respectivamente [IMSICC], decreciendo conforme se introduce unas mayores prestaciones y quedándose muchas tan sólo en la primera especificación sin hacer el esfuerzo de adaptar sus aplicaciones a las revisiones.

Con respecto a OSID aporta por cómo se define el modelo independencia de la implementación a las interfaces implementadas para los servicios definidos en OSID. Además proporciona escalabilidad al sistema ya que añadir nuevos servicios en base al paradigma definido en OSID se hace sencillo, así como su integración de forma modular y progresiva.. Sin embargo también supone contar con mayores dificultades para la implementación debido a que se trata de una autenticación más compleja y en principio es conveniente su implementación en Java ya que ese es lenguaje sobre el que la implementación de OSID está disponible. Esta desventaja debería verse compensada en tanto que futuras modificaciones y adaptaciones a diferentes tecnologías tendría un coste menor al mantenerse el modelo de interfaces y servicios definidos en OSID.

BIBLIOGRAFÍA

Bibliografía

LIS: IMS Global, Learning Information Services, 2013,
<http://www.imsglobal.org/lis/>

LTIv1.1.1: © 2012 IMS Global Learning Consortium, Inc., IMS Global Learning
Tools Interoperability™ Implementation Guide Final Version 1.1.1, 2012,
<http://www.imsglobal.org/LTI/v1p1p1/ltiIMGv1p1p1.html>

IMSCert: IMS, IMS Interoperability Conformance Certification Status, 2013,
http://www.imsglobal.org/cc/statuschart.cfm?show_all_nav_listCompliance3=1

IMSCtlg: IMS Global, IMS Global Developer, 2013,
<http://developers.imsglobal.org/catalog.html>

OSIDSpecFmwk: OSID, OSID Specification Framework, 2013,
<http://osid.org/library/osm/osm-2.pdf>

HIMS: © 2001-2012 IMS Global Learning Consortium, Inc. , About IMS Global
Learning Consortium, 2013, <http://www.imsglobal.org/background.html>

LTIv2.0: IMS Global, IMS Learning Tools Interoperability™ (LTI)
Implementation Guide Version 2.0 Public Draft Revision 2, 2013,
<http://www.imsglobal.org/LTI/v2p0pd2/ltiIMGv2p0pd2.html>

IMSODIRE: RobAbel Chief Executive of the IMS Global Learning Consortium,
You say you want a revolution? Well, you know, the IMS Open Digital
Innovation Revolution has begun, 2012, <http://www.imsglobal.org/blog/?p=143>

OSID: OSID, Open Service Interface Definitions, 2013, <http://osid.org>

OSIDrcu: Tom Coppeto, Release Candidate Update, 2013,
<http://osid.org/library/documents/OSIDV3RC.pdf>

IMSICCS: IMS Global, IMS Interoperability Conformance Certification Status,
2014, <http://www.imsglobal.org/cc/statuschart.cfm>