

Möglichkeiten der Verbindung von ownCloud und Lernwerkzeugen mittels LTI

Diplomarbeit

zur Erlangung des Grades eines Diplom-Informatikers
im Studiengang Informatik

vorgelegt von
Stefan Bäcker

Erstgutachter: Prof. Dr.-Ing. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)

Zweitgutachter: Gerrit Lochmann, M.Sc.
(Institut für Computervisualistik, AG Computergraphik)

Koblenz, im Februar 2016

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden.

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.

.....
(Ort, Datum)

.....
(Unterschrift)

Zusammenfassung

In einer Welt, in der mittlerweile „Die Cloud“ als Lösung für nahezu alles angepriesen wird, stellt sich immer häufiger die Frage, ob man seine persönlichen Daten einem Fremden anvertrauen möchte, oder sie doch lieber unter der eigenen Kontrolle behält. Für die Befürworter der letzten Option steht „ownCloud“ als freies Softwarepaket zur Verfügung, um eine eigene Cloud aufzusetzen und ihre Inhalte mit Anderen zu teilen.

Um das Teilen von Lernwerkzeugen zu vereinheitlichen und damit zu vereinfachen, wurde von IMS GLOBAL die „Learning Tools Interoperability“ Spezifikation – kurz LTI – entwickelt. Diese wird inzwischen von einer zunehmenden Anzahl von Lernmanagementsystemen und Lernressourcen unterstützt. Eine interessante Herausforderung ist daher, zu untersuchen, ob und wie man ownCloud mit verschiedenen bestehenden Lernwerkzeugen mittels LTI verbinden und daraus Nutzen ziehen kann.

Ziel dieser Arbeit ist es, ein Plugin für ownCloud zu konzeptionieren und zu entwickeln, das die Kommunikation mit Lernwerkzeugen per LTI ermöglicht. Dabei soll sowohl die Consumer- als auch die Providerseite mit einem Proof of Concept berücksichtigt werden, um jeweils die Möglichkeiten und Grenzen dieser Verbindungen zu untersuchen.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	3
2.1	Learning Tools Interoperability	3
2.1.1	Basic LTI Version 1.0	3
2.1.2	LTI Version 1.1.1	6
2.1.3	LTI Versionen 2.0 und 1.2	8
2.2	ownCloud	9
2.2.1	Benutzer	9
2.2.2	Persönlich	9
2.2.3	Administration	10
2.2.4	Dateien	10
2.2.5	Weitere Apps	10
2.2.6	Smartphone App	10
2.2.7	API	10
3	Konzeption	12
3.1	Tool Provider	12
3.1.1	Login per LTI	12
3.1.2	Direktzugriff auf Ressourcen	14
3.2	Tool Consumer	14
3.2.1	Einbindung	16
4	Implementierung	18
4.1	IMS GLOBAL Referenzimplementation	18
4.1.1	Klasse „BLTI“	18
4.1.2	postLaunchHTML	18
4.1.3	replaceResultRequest	18
4.2	ownCloud	18
4.2.1	Struktur einer App	19
4.2.2	Verzeichnis: appinfo	19
4.2.3	Verzeichnis: controller	21
4.2.4	Verzeichnisse: js, img, css	21
4.2.5	Verzeichnis: db	21
4.2.6	Verzeichnis: lib	21
4.2.7	Verzeichnis: service	21
4.2.8	Verzeichnis: templates	21
4.2.9	Verzeichnis: tests	22
4.3	Konkrete Implementierung	22
4.3.1	Login per LTI	22
4.3.2	Direktzugriff auf Ressourcen	23
4.3.3	Tool Consumer	25

5	Ergebnisse und Bewertung	27
5.1	Auswahl der externen Tool Consumer	27
5.1.1	OpenOLAT	28
5.1.2	Ilias	28
5.1.3	Moodle	31
5.2	Test der ownCloud Tool Provider	33
5.2.1	Login per LTI	33
5.2.2	Direktzugriff auf Ressourcen	34
5.3	Auswahl der externen Tool Provider	36
5.3.1	Videosammlung	36
5.3.2	Etherpad Lite	37
5.3.3	IMathAS	37
5.4	Test des ownCloud Tool Consumers	39
5.5	Bewertung	40
6	Fazit und Ausblick	42
6.1	Fazit	42
6.2	Ausblick	42

Abbildungsverzeichnis

1	Bildschirmfoto: Standardansicht der Dateien App in ownCloud mit aufgeklapptem Menü	9
2	Struktureller Aufbau der drei Teilbereiche	22
3	Bildschirmfoto: LTI spezifischer Teil des administrativen Bereichs	23
4	Bildschirmfoto: Erweiterter „Teilen“-Dialog	24
5	Zusammenspiel von Tool Consumern und Tool Providern im Test	27
6	Bildschirmfoto: Einstellungen des LTI Kursbausteins in OpenOLAT	29
7	Bildschirmfoto: Einstellungen des External Contents in Ilias	30
8	Bildschirmfoto: Einstellungen in der Aktivität „External Tool“ in Moodle	32
9	Bildschirmfoto: Videosammlung in der erweiterten Dozentenansicht	36
10	Bildschirmfoto: Etherpad Lite mit aufgeklappter Autorenübersicht	37
11	Bildschirmfoto: Assessment in IMathAS	38

1 Einleitung

Im Kontext des online gestützten Lehren und Lernens gibt es eine schier unüberschaubare Anzahl von Plattformen, Werkzeugen und Bibliotheken. Alle haben ihre Stärken und Schwächen und welche von ihnen die Beste ist, liegt stets im Auge des Betrachters. Nicht selten dürfte die beste Lösung jedoch in der Kombination der Stärken verschiedener Systeme liegen.

Um diese verschiedenen Systeme nicht einfach nur nebeneinander zu betreiben, sondern zu integrieren, sind in der Regel Anpassungen an den beteiligten Komponenten vorzunehmen. Während man bei Open Source Software im Zweifelsfall selbst Hand anlegen kann, ist man bei proprietärer Software immer auf die Hilfe und Unterstützung des Herstellers angewiesen. Unabhängig davon ist eine direkte Integration immer auf eine bestimmte Paarung von Komponenten beschränkt. Der entwickelte Code, respektive die Plugins, sind meist nur sehr eingeschränkt wiederverwendbar.

An diesem Problem setzt die Learning Tools Interoperability (LTI) Spezifikation von IMS GLOBAL an. Sie beschreibt eine vielseitig nutzbare, generelle Schnittstelle für die Integration verschiedenster Systeme. Neben dem Aufbau einer Vertrauensstellung zwischen den beteiligten Partnern, definiert sie noch einen Parametersatz, der vielfältige Informationen über das aufrufende System, den eingeloggten Nutzer sowie den aktuellen Kontext enthalten kann.

Durch eine Implementation dieser Spezifikation kann somit auf einfache Weise eine breite Unterstützung für andere Systeme aus dem E-Learning Bereich erreicht werden.

Das an der Universität Koblenz-Landau vorwiegend genutzte Lernmanagementsystem ist OpenOLAT. Dieses wird am „Virtuellen Campus Rheinland-Pfalz“ (VCRP) in Kaiserslautern für das gesamte Bundesland gehostet. Während die Vorteile für die Universität durch die zentrale Bereitstellung auf der Hand liegen, gehen auch einige Nachteile damit einher. Hervorzuheben sind hier, vor allem im Kontext der Videounterstützung, der nur begrenzt zur Verfügung gestellte Plattenplatz sowie die beschränkte Netzwerkkapazität zwischen den beiden Standorten.

Während man für öffentlich zugängliche Videos auch eine Bereitstellung auf einer Plattform wie YouTube in Erwägung ziehen mag, verbietet sich dieses spätestens dann, wenn aus Datenschutz- oder Copyrightgründen der Zugriff auf einen geschlossenen Benutzerkreis eingeschränkt werden muss. Eine gängige Praxis an der Universität Koblenz-Landau ist daher die Ablage der Videos auf Servern der Universität und deren Einbindung als Lerneinheit in OpenOLAT per LTI.

Derzeit wird für diese Einbindung ein eigenständiges Werkzeug verwendet. Für jeden Anwender sind vom Administrator auf den Servern der Universität ein eigenes Verzeichnis sowie Zugangsdaten dazu anzulegen. Der An-

wender muss seine Videodateien in sein Verzeichnis kopieren und anschließend mit dem LTI-Baustein in dem Lernmanagementsystem verknüpfen.

Daneben wird jedoch von der Universität ein ownCloud-Server zur Datenablage gehostet, auf den jeder Universitätsangehörige mit seiner Universitätskennung Zugang hat. Dieser Server zeichnet sich durch vielfältige Zugriffs- und Synchronisationsmöglichkeiten aus, was ihm jedoch fehlt ist eine LTI-Schnittstelle.

Ziel dieser Arbeit ist die Entwicklung einer solchen LTI-Schnittstelle für ownCloud.

Im zweiten Kapitel sollen zunächst die Grundlagen für die Arbeit mit LTI und ownCloud gelegt werden. Um den für diese Arbeit relevanten Funktionsumfang von LTI zu verstehen, werden zuerst verschiedene Versionen der LTI Spezifikation beschrieben. Anschließend folgt eine Einführung in ownCloud, um mögliche Ansatzpunkte für eine Erweiterung zu identifizieren.

Im dritten Kapitel sollen die Erweiterungen für ownCloud konzipiert werden. Dazu gehört neben der Bereitstellung von Diensten durch ownCloud über LTI auch die Einbindung fremder Ressourcen per LTI in ownCloud.

Im vierten Kapitel wird die Implementierung beschrieben. Nach einer kurzen Einführung in bestehende Bibliotheken zu LTI und den allgemeinen Aufbau einer ownCloud App wird die Umsetzung der einzelnen Programmteile im Detail aufgezeigt.

Im fünften Kapitel werden die entwickelten Erweiterungen für ownCloud auf den Prüfstand gestellt. Neben Tests der Integration mit OpenOLAT [Ope15] und dem Mathematikassessmentsystem IMathAS [Lip16] sollen noch weitere LTI-fähige Werkzeuge und Lernmanagementsysteme ausgewählt und beispielhaft getestet werden. Anschließend wird anhand der Ergebnisse bewertet, in wie weit die Integration gelungen ist und an welchen Stellen noch Nachholbedarf besteht.

Das sechste Kapitel soll schließlich ein Fazit ziehen und einen Ausblick auf weitergehende Entwicklungsmöglichkeiten, die den Rahmen dieser Arbeit sprengen würden, geben.

Disclaimer: Im Folgenden wird rein aus Gründen der Übersichtlichkeit bei allen Bezeichnern von Personen stets nur die männliche Form verwendet. Die weibliche Form ist dabei ohne Einschränkung stets mit einzubeziehen.

2 Grundlagen

2.1 Learning Tools Interoperability

Das „IMS GLOBAL Learning Consortium“ entwickelt die „Learning Tools Interoperability“ (LTI) Spezifikation, um die Einbindung von ausgelagerten Werkzeugen und Inhalten in Lernmanagementsysteme zu ermöglichen. Diese ausgelagerten Ressourcen werden dabei als „Tool Provider“ bezeichnet; sie stellen die Inhalte bereit. Die Lernmanagementsysteme werden „Tool Consumer“ genannt, denn sie binden die Ressourcen ein und stellen sie ihren Nutzern bereit.

Das eigentliche Ziel von IMS ist die „Full LTI Spezifikation“, die einen formalen, aushandelnden Startprozess beschreibt, bei dem Tool Consumer und Tool Provider sich über diverse Parameter einigen. Zu diesen Parametern gehören sowohl die Laufzeitdienste, über die eine enge Integration zwischen den Systemen erreicht wird, und die Sicherheitsmechanismen als auch die Werkzeuge, die ein Tool Consumer innerhalb eines Tool Providers starten kann.

Zunächst wurde jedoch als Vereinfachung lediglich die Basic LTI [SMN10] Spezifikation entwickelt. Diese ermöglicht nur ein Ziel innerhalb eines Tool Providers und nur einen Sicherheitsmechanismus. Ein Zugriff auf die Laufzeitdienste des Tool Consumers ist nicht vorgesehen.

2.1.1 Basic LTI Version 1.0

Die grundlegende Funktionsweise von Basic LTI ist die folgende: Der Administrator eines Tool Providers stellt ein Werkzeug für Tool Consumer bereit. Dazu wird neben der URL des Dienstes noch ein Schlüssel und ein geteiltes Geheimnis benötigt.

Dieser dreiteilige Parametersatz wird vom Administrator oder Dozenten eines Tool Consumers verwendet, um auf den Tool Provider zuzugreifen. Dabei kann entweder der Administrator den kompletten Satz im Tool Consumer eingeben und dem Dozenten lediglich die Platzierung des Werkzeugs innerhalb seines Kontextes ermöglichen, oder der Dozent verwendet den kompletten Satz selber im Rahmen eines generischen LTI Bausteins.

Neben URL und Schlüssel werden beim Aufruf eines Tool Providers noch weitere Informationen vom Tool Consumer bereitgestellt. Diese enthalten neben allgemeinen Daten des Tool Consumers sowie des Kontextes, in den das Werkzeug platziert wurde, noch Angaben über den aufrufenden Benutzer.

Die Validität der übertragenen Daten wird mit dem OAuth Protokoll gesichert. Dazu wird mit dem gemeinsamen Geheimnis ein Hashwert über alle Informationen gebildet und ebenfalls zum Tool Provider übertragen. Dieser kann mit Hilfe des übertragenen Schlüssels das benötigte Geheimnis bestimmen und damit ebenfalls einen Hashwert berechnen. Stimmen die beiden

Hashwerte überein, so wurde der Aufruf unverfälscht übertragen. Gegen Angriffe mittels einer wiederholten Übertragung von beim Verbindungsaufbau mitgeschnittenen Datenpaketen findet die Einbeziehung eines Zeitstempels und eines nur einmalig verwendbaren Wertes Anwendung.

Nach dem Start eines Tool Providers läuft dieser vollständig unabhängig vom Tool Consumer. Eine weitere Interaktion Tool Providers mit dem Tool Consumer ist nicht vorgesehen.

Parameterbeschreibungen Es folgt eine Beschreibung aller möglichen Parameter, die bei einem Basic LTI Aufruf vom Tool Consumer an den Tool Provider übergeben werden können. Einige davon werden zwangsläufig benötigt, andere werden nur empfohlen und wieder andere sind optional. Während die optionalen Parameter üblicherweise ohne Funktionsverlust entfallen können, sollten die empfohlenen möglichst vollständig gesetzt werden. Dies liegt darin begründet, dass sie zwar für die Funktion eines LTI Aufrufs an sich nicht zwingend erforderlich sind, sie aber für den korrekten Ablauf in einem Tool Provider durchaus nötig sein können.

lti_message_type=basic-lti-launch-request Dies zeigt an, dass es sich um einen Basic LTI Aufruf handelt. Dadurch wird es einem Tool Provider ermöglicht, unter der selben URL auf verschiedene LTI Nachrichtentypen zu reagieren. Dieser Parameter wird benötigt.

lti_version=LTI-1p0 Dies zeigt an, welche Version der Spezifikation benutzt wird. Dieser Parameter wird benötigt.

resource_link_id Dies ist ein eindeutiger Bezeichner, der vom Tool Consumer für jede Platzierung eines Tool Providers unterschiedlich vergeben wird. Dieser Parameter wird benötigt.

resource_link_title Ein Titel für die Ressource. Dies ist ein anklickbarer Text, der in dem Link zum Tool Provider erscheint. Dieser Parameter wird empfohlen.

resource_link_description Eine Beschreibung für die Ressource, die typischerweise nicht mehr als ein paar Zeilen umfasst und neben dem Link angezeigt wird. Dieser Parameter ist optional.

user_id Ein eindeutiger Bezeichner für den Nutzer. Dieser sollte keine Informationen enthalten, die einen Rückschluss auf die Identität des Nutzers erlauben. Üblicherweise sollte dieser Bezeichner ein langfristig gültiger, vom Tool Consumer generierter Primärschlüssel auf den Benutzer sein. Dieser Parameter wird empfohlen.

user_image Dieses Attribut gibt eine URI zum einem Bild des Benutzers an, das als Profilbild oder Avatar verwendet werden kann. Dieser Parameter ist optional.

roles Ein Komma separierte Liste von Rollen des Benutzers. Die vollständige Liste der möglichen Werte ist in der Learner Information Services Spezifikation von IMS GLOBAL aufgeführt. Die gebräuchlichsten Werte sind „Instructor“ für Dozenten und „Learner“ für Lernende. Dieser Parameter wird empfohlen.

Oft machen Tool Provider am Vorhandensein der „Instructor“ Rolle Schreib- und Bearbeitungsrechte an einer Ressource fest, wohingegen „Learner“ nur Leserechte erhalten.

Weitere häufiger verwendete Rollen sind „Administrator“, „TeachingAssistant“ und „Mentor“.

lis_person_name_given Der Vorname des Benutzers. Er wird empfohlen, sofern er nicht aus Datenschutzgründen unterdrückt werden muss.

lis_person_name_family Der Nachname des Benutzers. Er wird empfohlen, sofern er nicht aus Datenschutzgründen unterdrückt werden muss.

lis_person_name_full Der vollständige Name des Benutzers. Er wird empfohlen, sofern er nicht aus Datenschutzgründen unterdrückt werden muss.

lis_person_contact_email_primary Die E-Mailadresse des Benutzers. Sie wird empfohlen, sofern sie nicht aus Datenschutzgründen unterdrückt werden muss.

context_id Dies ist ein opaker Bezeichner, der eindeutig den Kontext, aus dem heraus der Link gestartet wurde, bezeichnet. Dieser Parameter wird empfohlen.

context_type Eine Komma separierte Liste von URN Werten, die den Kontexttyp angeben. Die vollständige Liste der möglichen Werte kann im LIS Vokabular eingesehen werden. Dieser Parameter ist optional.

context_title Der Titel des Kontextes, üblicherweise etwa eine Zeile lang. Dieser Parameter wird empfohlen.

context_label Ein kurzes Label für den Kontext, der in eine Spalte passen sollte. Dieser Parameter wird empfohlen.

launch_presentation_locale Die Sprache, bestehend aus Ländercode und Variante, wie sie von der Internet Engineering Task Force empfohlen wird. [DPU10]

launch_presentation_document_target Dieser Wert teilt dem Tool Provider mit, wie er vom Tool Consumer eingebettet wurde. Er sollte entweder „frame“, „iframe“ oder „window“ sein. Dieser Parameter wird empfohlen.

- launch_presentation_width** Die Breite des Anzeigefensters. Dieser Parameter wird empfohlen.
- launch_presentation_height** Die Höhe des Anzeigefensters. Dieser Parameter wird empfohlen.
- launch_presentation_return_url** Eine vollqualifizierte URL, an die der Tool Provider den Benutzer zurückverweisen kann. Diese kann verwendet werden, wenn der Tool Provider fertig ist oder eine Fehlersituation aufgetreten ist. Dieser Parameter wird empfohlen.
- tool_consumer_instance_guid** Dies ist ein eindeutiger Bezeichner für den Tool Consumer. Üblicherweise wird hier der DNS-Name der Organisation oder des Tool Providers verwendet. Dieser Parameter wird empfohlen.
- tool_consumer_instance_name** Ein kurzer Name für den Tool Consumer. Er ist für den Benutzer sichtbar und sollte in eine Spalte passen. Dieser Parameter wird empfohlen.
- tool_consumer_instance_description** Eine Beschreibung für den Tool Consumer. Er ist für den Benutzer sichtbar und sollte in eine Zeile passen. Dieser Parameter ist optional.
- tool_consumer_instance_url** Dies ist die URL des Tool Consumers. Dieser Parameter ist optional.
- tool_consumer_instance_contact_email** Eine Kontaktmailadresse des Tool Consumers. Dieser Parameter wird empfohlen.
- custom_keyname** Es können beliebige weitere Parameter hinzugefügt werden, die alle mit „custom_“ beginnen. Die Bedeutung dieser Parameter ist nicht Teil der Spezifikation und wird einzig und allein vom jeweiligen Tool Provider vorgegeben.

2.1.2 LTI Version 1.1.1

In Version 1.0 von Basic LTI lief der Tool Provider nach seinem Start völlig autark vom Tool Consumer. Eine Rückmeldung vom Tool Provider zum Tool Consumer, etwa über die erfolgreiche Bearbeitung einer Aufgabe, war nicht vorgesehen. Dieses Manko wird in Version 1.1.1 [McF+12] zumindest teilweise durch die Spezifikation eines Ergebnisdienstes behoben. Das Ergebnis kann dabei in Form einer Dezimalzahl im Bereich von 0,0 bis 1,0 vom Tool Provider beim Tool Consumer gesetzt, abgefragt oder gelöscht werden.

Zu diesem Zweck wurden zwei zusätzliche Parameter definiert:

lis_result_sourcedid Dies ist ein Bezeichner, der für jede Kombination aus context_id, resource_link_id und user_id eindeutig ist. Er kann bei jedem Start verschieden sein und der Tool Provider sollte immer nur den zuletzt übertragenen speichern und nutzen. Dieser Parameter ist optional.

lis_outcome_service_url Dies ist die URL zu einem Basic Outcomes Service Provider. Ein Tool Provider kann davon ausgehen, dass sich dieser Wert nur ändert, wenn ein Tool Consumer grundlegend umkonfiguriert wird oder seine Domäne ändert. Ansonsten kann von einer eins zu eins Beziehung zwischen oauth_consumer_key und lis_outcome_service_url ausgegangen werden. Dieser Parameter wird benötigt, wenn der Tool Consumer die Rückübertragung von Ergebniswerten akzeptiert. Der Aufruf der Services erfolgt asynchron, kann also jederzeit auch außerhalb der Nutzungssession des Benutzers erfolgen. Dies kann z.B. dann nötig sein, wenn der Dozent die Ergebnisse erst von Hand berechnen oder korrigieren muss. Dazu speichert der Tool Provider die lis_outcome_service_url und die lis_result_sourcedid und verwendet diese, sobald sie gebraucht werden.

LTI Basic Outcome Service Der „Basic Outcomes Service“ kommuniziert über Plain old XML Nachrichten, die per OAuth signiert werden. Um den Schlüssel und das gemeinsame Geheimnis nicht zu kompromittieren, sollte er als Server zu Server Anwendung realisiert werden. Er sollte auf alle wohlgeformten und validen Anfragen reagieren, und sofern er sie nicht bearbeiten kann mit einem „unsupported“ beantworten.

replaceResult Mit dem Nachrichtentyp „replaceResult“ kann der Tool Provider ein Ergebnis beim Tool Consumer hinterlegen. Während der Tool Provider stets davon ausgehen kann, dass es zu jeder lis_result_sourcedid genau ein Ergebnis gibt, ist es dem Tool Consumer freigestellt, ob und wie er die Ergebnisse versioniert.

Sollte der Tool Provider versuchen, etwas anderes als eine Dezimalzahl zwischen 0,0 und 1,0 als Ergebnis zu setzen, so antwortet der Tool Consumer mit einer Fehlermeldung.

readResult Mit dem Nachrichtentyp „readResult“ kann der Tool Provider ein Ergebnis beim Tool Consumer auslesen, sofern es dort bereits gesetzt wurde. Sollte noch kein Ergebnis bekannt sein, so liefert der Tool Consumer eine leere Antwort. Wenn der Tool Consumer die Ergebnisse versioniert, so erhält der Tool Provider stets das zuletzt eingereichte Ergebnis zurück.

deleteResult Mit dem Nachrichtentyp „deleteResult“ kann der Tool Provider ein Ergebnis beim Tool Consumer löschen. Versioniert der Tool Consumer

mer die Ergebnisse, so ist es ihm überlassen, wie er die Löschung tatsächlich ausführt. Er kann entweder nur das neueste oder direkt alle Ergebnisse löschen. Er kann aber auch lediglich eine Markierung setzen und sich so das Ergebnis als gelöscht merken. Unabhängig von der Art des Löschens sollte der Aufruf eines „readResult“ nach einem „deleteResult“ kein Ergebnis zurück liefern.

Weitere Basic Launch Parameter

role_scope_mentor Eine Komma separierte Liste von `user_ids`, für die der Benutzer die Mentorenrolle einnimmt. Typischerweise wird dies für Eltern, Erziehungsberechtigte oder Prüfer verwendet. Die genaue Implementation auf Seiten des Tool Providers ist nicht vorgegeben, aber es ist üblich, dass ein Mentor Zugriff auf Nachverfolgungsdaten und Statistiken erhält, aber nicht auf persönliche Inhalte. Dieser Parameter ist optional und sollte nur gesetzt sein, wenn gleichzeitig die Rolle „Mentor“ gesetzt ist.

launch_presentation_css_url Eine URL zum CSS des Tool Consumers. Da es derzeit noch keine Standards für die Struktur von Tool Consumern gibt, ist diese nur von begrenztem Wert. Es ist zumindest möglich, grundlegende Eigenschaften wie etwa Schriftarten und Farben zu definieren und so wenigstens das Aussehen von Tool Consumer und Tool Provider aneinander anzunähern.

tool_consumer_info_product_family_code Der Produktname des Tool Consumers. Dieser Parameter soll es dem Tool Provider ermöglichen, sich optimal in das Benutzerinterface des aufrufenden Tool Consumers ein zu passen. Dieser Parameter wird empfohlen.

tool_consumer_info_version Die Versionsnummer des Tool Consumers. Eine Hauptversion gefolgt von einem Punkt, auf den noch eine Unterversion folgen kann. Dieser Parameter wird empfohlen.

2.1.3 LTI Versionen 2.0 und 1.2

LTI Version 2.0 ist ein umfassender weiterer Ausbau der Spezifikation, die zwar auf LTI Version 1.1.1 basiert, aber teilweise auch völlig neue Wege geht. Um einen Upgradepfad für Tool Consumer und Tool Provider, die noch auf einer 1.0 oder 1.1 Version basieren, aufzuzeigen, wurde ein Jahr später die Version 1.2 von LTI spezifiziert. Ein Blick in das Produktverzeichnis von IMS GLOBAL [Con16a] zeigt jedoch, dass von insgesamt 290 nach LTI zertifizierten Anwendungen nur zwölf die Versionen 2.0 oder 1.2 umsetzen. Im Rahmen dieser Arbeit wird darum auf eine nähere Betrachtung dieser beiden Versionen verzichtet.

2.2 ownCloud

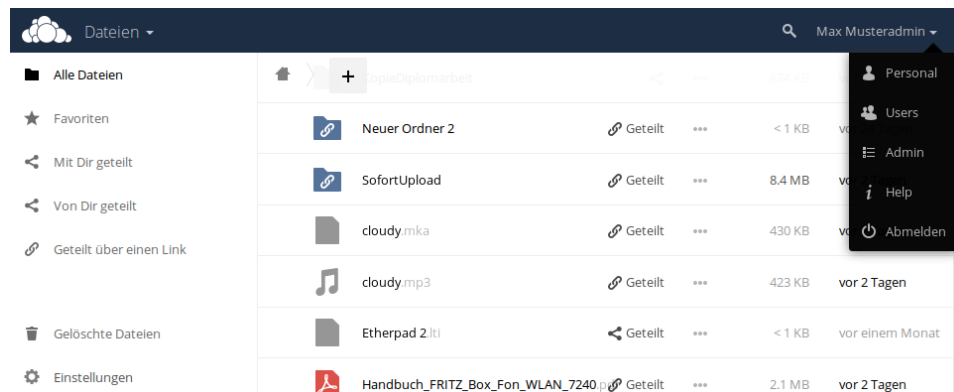


Abbildung 1: Bildschirmfoto: Standardansicht der Dateien App in ownCloud mit aufgeklapptem Menü

ownCloud [own16a] ist ein Open Source Serversystem, mit dem man Dateien hosten, synchronisieren und teilen kann. Der Zugriff auf die Daten ist über ein Webinterface, Synchronisationsclients für verschiedene Betriebssysteme oder WebDAV möglich.

2.2.1 Benutzer

Die Nutzerverwaltung und Rechtestruktur in ownCloud ist recht einfach gehalten. Ein Administrator erstellt einen Nutzeraccount, in dem er einen Nutzernamen und ein Passwort angibt und diesen Account eventuell noch Gruppen zuordnet. Bei bestehenden Nutzern kann er den vollständigen Namen und die Gruppenzugehörigkeit ändern, sowie gegebenenfalls Gruppenadministratorenrechte vergeben. Die Gruppe „admin“ hat die Sonderfunktion, dass alle ihre Mitglieder Administratoren sind, die im System sämtliche Einstellungen bearbeiten dürfen. Gruppenadministratoren hingegen dürfen nur die Mitglieder ihrer Gruppe bearbeiten.

Anstelle jeden Nutzer von Hand einzutragen, können auch verschiedene Plugins verwendet werden, die die Nutzerdaten aus anderen Quellen, etwa einem LDAP Verzeichnis, importieren oder gegen ein anderes System authentifizieren.

2.2.2 Persönlich

Jeder Nutzer kann für den eigenen Account das Passwort, den vollständigen Namen, die E-Mailadresse, das Profilbild und die Sprache ändern. Des Weiteren kann er einsehen, wie viel seiner Quotierung er benutzt und zu welchen Gruppen er gehört.

2.2.3 Administration

Im Administrationsbereich kann jeder Administrator grundlegende Einstellungen für das System, sowie eventuell vorhandene Plugins vornehmen. Dazu gehören insbesondere diverse Optionen für das Teilen von Dateien und Verzeichnissen.

2.2.4 Dateien

Die Dateien-App ist die Hauptansicht für die Dateien und Verzeichnisse eines Nutzers. Hier kann er neue, leere Dateien und Verzeichnisse anlegen oder bestehende hochladen. Er kann die Namen ändern sowie die Größe und den letzten Änderungszeitpunkt einsehen. Er kann Dateien als Favoriten markieren, herunterladen oder löschen. Sofern die Versionierung aktiviert ist, kann er ältere Versionen von Dateien wiederherstellen.

Außerdem kann er Dateien und Verzeichnisse mit anderen Nutzern teilen. Er kann dies einerseits direkt innerhalb ownClouds mit anderen Nutzern oder Gruppen. Hierbei besteht die Möglichkeit, das weitere Teilen mit anderen Nutzern sowie die Bearbeitung zu erlauben. Andererseits kann er Dateien und Verzeichnisse über einen öffentlichen Link teilen. Dieser kann zusätzlich über ein Passwort geschützt und zeitlich begrenzt eingerichtet werden.

Um die Ansicht zu filtern, ist eine Einschränkung durch den Nutzer auf die Favoriten oder die von beziehungsweise mit ihm geteilten Dateien möglich. Ebenso ist eine Suche nach beliebigen Namen verfügbar.

2.2.5 Weitere Apps

Neben der Dateien-App sind noch zahlreiche weitere Apps verfügbar, etwa für Kalender und Kontakte, Bilder und Musik oder Dokumente. Zusätzliche Apps können vom Administrator direkt aus dem Appstore [Tea16] heruntergeladen und installiert werden.

2.2.6 Smartphone App

Es existieren mobile Synchronisationsclients sowohl für Android als auch für iOS und BlackBerry. Allen gemeinsam ist, dass sie lediglich auf den Datei- und Verzeichnisteil von ownCloud ausgelegt sind. Mit ihnen kann man Daten automatisch synchronisieren und diese bearbeiten, benötigt aber zur Darstellung der Inhalte stets die passende App auf dem Smartphone.

2.2.7 API

Für die Entwicklung von Apps sind APIs vorhanden und dokumentiert. Die PHP API gliedert sich dabei in einen stabilen und einen instabilen Teil auf. Während der stabile Teil relativ zukunftssicher sein soll, ist der instabile

Teil häufigen Änderungen unterworfen und wird deshalb für die Entwicklung eigener Apps nicht empfohlen.

Eine JavaScript API ist für die Ausführung auf der Clientseite vorhanden.

3 Konzeption

Bei der späteren Implementierung sind die Seiten Tool Provider und Tool Consumer getrennt zu betrachten.

3.1 Tool Provider

Da LTI eine Vertrauensstellung zwischen Tool Consumer und Tool Provider aufbaut und einen nutzerspezifischen Datensatz überträgt, könnte man die Nutzerauthentifikation des Tool Consumers in der Art eines Single Sign-On an den Tool Provider weiterreichen, dort bei Bedarf einen neuen Account für den Benutzer anlegen und ihn einloggen. Anschließend stünde ihm das gesamte ownCloud-System zur Verfügung.

Alternativ könnte man alles, was über einen öffentlichen Link geteilt und durch ein Passwort geschützt wird, auch über einen LTI-Link anbieten. Der Vorteil dieses Verfahrens wäre, dass der Dozent das Passwort nicht preisgeben muss und die Zugriffskontrolle vollständig auf Seiten des Tool Consumers liegt.

3.1.1 Login per LTI

Um das Einloggen in ownCloud aus einem Tool Consumer per LTI zu ermöglichen, müssen zunächst ein oder mehrere Sätze aus jeweils einem Schlüssel und einem geteilten Geheimnis in ownCloud hinterlegt werden. Da nur Administratoren oder Gruppenadministratoren neue Benutzer anlegen dürfen, empfiehlt es sich, hierfür einen neuen Bereich im Admininterface einzurichten. Ob es sinnvoll ist, auch Gruppenadministratoren diese Möglichkeit einzuräumen, ist zumindest fraglich. Zum einen gibt es keinen gesonderten Bereich für Einstellungen eines Gruppenadmins, sondern dies müsste innerhalb der persönlichen Einstellungen implementiert werden. Zum anderen wäre sicherzustellen, dass neu angelegte Benutzer immer mindestens in einer der Gruppen eines Gruppenadmins verbleiben. Außerdem dürfte die zu erwartende Arbeitserleichterung für Administratoren äußerst gering ausfallen.

Wird ein neuer Benutzer angelegt, so sollten alle per LTI übertragenen persönlichen Daten auch in den neuen Account übernommen werden. Dazu gehören der vollständige Name (`lis_person_name_full`), die Emailadresse (`lis_person_contact_email_primary`), der Avatar (`user_image`) und die Sprache (`launch_presentation_locale`). Alternativ zum vollständigen Namen könnte der Name auch in Vorname (`lis_person_name_given`) und Nachname (`lis_person_name_family`) getrennt übertragen werden und müsste dann entsprechend zusammengesetzt werden. Datenschutzbedenken sollte es dabei nicht geben, denn es kann davon ausgegangen werden, dass der Nutzer der Übertragung seiner persönlichen Daten auf Seiten des Tool Consumers zugestimmt hat.

Da ownCloud als Tool Provider durchaus von mehreren verschiedenen Tool Consumern verwendet werden kann, die Nutzer IDs (`user_id`) aber stets nur innerhalb eines Tool Consumers eindeutig sind, kann man die per LTI gelieferte Nutzer ID nicht einfach übernehmen. Stattdessen sollte die Nutzer ID aus einer Kombination aus eindeutiger Tool Consumer ID (`tool_consumer_instance_guid`) und innerhalb eines Tool Consumers eindeutiger Nutzer ID bestehen. Hier würde sich eine einfache Konkatenation aus Tool Consumer ID und Nutzer ID anbieten. Da laut [HSF85] ein DNS Name nur aus Buchstaben, Ziffern, Minuszeichen und Punkten bestehen darf, würde sich beispielsweise ein Semikolon oder ein Dollarzeichen als Trennzeichen zwischen DNS Namen und Nutzer ID anbieten.

Da ein Nutzerpasswort beim Login per LTI weder übertragen noch benutzt wird, sollte dieses zufällig erzeugt werden. Auf diese Weise wird ein direktes Login in ownCloud über die Website zuverlässig ausgeschlossen. Sollte der neu angelegte Benutzer dies jedoch wünschen, so kann er jederzeit von einem Administrator oder Gruppenadministrator sein eigenes Passwort neu setzen lassen und sich damit auch ohne LTI Verbindung einloggen. Hat der Benutzer eine E-Mailadresse hinterlegt, so kann er auch über die „Passwort vergessen“ Funktion ein neues Passwort erstellen.

Dateien und Verzeichnisse kann jeder Nutzer mit anderen Nutzern teilen. Dazu benötigt er die eher kryptische und wahrscheinlich unbekannte Nutzer ID oder den vollständigen Namen des Empfängers. Alternativ kann man auch mit einer ganzen Gruppe von Personen teilen. Hier wäre es hilfreich, wenn analog zur Kontextstruktur im Tool Consumer passende Gruppen im Tool Provider bestehen, beziehungsweise erzeugt und dem Nutzer zugeordnet würden.

Im Parametersatz von LTI sind drei Ebenen vorgesehen. Der Tool Consumer (`tool_consumer_instance_...`), der Kontext (`context_...`) und die Ressource (`resource_link_...`). Sofern die entsprechenden Parameter gesetzt sind, sollten hier für alle drei Ebenen entsprechende Gruppen eingerichtet werden.

Da eine Gruppe einzig und allein über ihren Namen referenziert wird, ist ein Kompromiss zwischen Lesbarkeit und Eindeutigkeit zu finden. Während mit der `tool_consumer_instance_guid` schnell ein Kandidat für die Gruppenbezeichnung aller Tool Consumer Nutzer gefunden ist, ist dies bei den beiden tiefer liegenden Ebenen nicht so einfach. Hier gibt es jeweils die Wahl zwischen einer innerhalb eines Tool Consumers eindeutigen, aber nicht unbedingt eingängigen und leicht zu ermittelnden ID und einem bekannten, aber nicht unbedingt eindeutigen Titel. Da die einfache Verwendung eines möglicherweise mehrfach verwendeten Titels mehrere Benutzergruppen zusammenfassen würde, ist dies keine Lösung. Ebenso erscheint die schlichte Verwendung der IDs für den Benutzer unkomfortabel wenn nicht gar unpraktikabel. Die beste Möglichkeit scheint deshalb auch hier die Konkatenation von Titel und eindeutiger ID zu sein. Um die Nutzer bei der Suche nach der

richtigen Gruppe zu unterstützen, kann auch im System eingestellt werden, dass das Teilen nur innerhalb der eigenen Gruppen möglich sein soll.

Da als mögliche Rollen in ownCloud nur Administrator, Gruppenadministrator und Benutzer vorkommen, sollten alle neuen Benutzer unabhängig von ihrer Rolle im Tool Consumer nur als einfache Benutzer angelegt werden. Ein Äquivalent zur Rolle Dozent gibt es schlichtweg nicht, eine Heraufstufung zum Gruppenadmin scheint nicht gerechtfertigt. Da die per LTI zu übermittelnden Rollen oftmals vom Dozenten des Tool Consumers selbst eingestellt werden können, ist auch die Übernahme eines etwaig angegebenen Administratorenstatusses nicht angebracht.

Bei den Präsentationseinstellungen (`launch_presentation_document_target`) ist grundsätzlich ein „window“ zu wählen, denn ownCloud unterbindet aus Sicherheitsgründen die eigene Darstellung innerhalb eines Frames einer anderen Domäne. Um diese Sicherheitseinstellungen zu überwinden, ist eine Veränderung am Kern von ownCloud notwendig, die über eine App nur schwer zu realisieren ist.

Als Ergebnis des Tool Provider Aufrufs könnte man bei erfolgreichem Login eine 1,0 und eine 0,0 bei einer Fehlersituation an den Basic Outcomes Service (`lis_outcome_service_url`) zurück liefern.

3.1.2 Direktzugriff auf Ressourcen

Soll nur eine einzelne Datei oder ein Verzeichnis freigegeben werden, so empfiehlt sich hierfür die „per Link teilen“ Funktion. Diese erzeugt einen Zugriffscode für die geteilte Ressource und erlaubt das Setzen eines Passwortes.

Diese beiden Werte können auch für die Autorisierung einer LTI Verbindung in Form von Schlüssel und geteiltem Geheimnis Verwendung finden.

Persönliche Daten des Benutzers sowie Informationen über die Ressource und den Kontext sind dabei irrelevant und finden keinerlei Verwendung.

Als Ergebnis des Aufrufs könnte man auch hier analog zum Login per LTI eine 1,0 für ein erfolgreiches Abrufen des Inhaltes beziehungsweise eine 0,0 in einer Fehlersituation zurückmelden.

3.2 Tool Consumer

Um einen kompletten Parametersatz für eine LTI Verbindung zu erhalten, müssen Werte aus fünf Bereichen zusammen getragen werden.

Tool Provider Die Tool Provider URL, der Schlüssel und das geteilte Geheimnis sind spezifisch für jede Platzierung. Sie können entweder in Form eines Templates von einem Administrator hinterlegt und von einem Dozenten platziert oder komplett frei von einem Dozenten angegeben werden.

Ressource und Kontext Die Ressourcen ID (`resource_link_id`) kann automatisch erzeugt werden. Sie muss eindeutig sein, unterliegt aber ansonsten keinen weiteren Bedingungen. Wird eine Ressource lediglich innerhalb des Tool Consumers verschoben, so sollte sich die ID nicht ändern. Wird eine Ressource jedoch kopiert, so muss für die Kopie eine neue ID erzeugt werden.

Der Ressourcen Titel (`resource_link_title`) und die optionale Beschreibung (`resource_link_description`) sind vom Dozenten einzugeben.

Die Kontext ID (`context_id`) und der Kontexttitel (`context_title`) können entweder fix für die gesamte ownCloud Installation gesetzt werden oder aus einer Gruppierung, etwa einer Verzeichnisstruktur, gewonnen werden. Das Kontext Label (`context_label`) kann mit der Kontext ID gleichgesetzt werden.

Die Angabe eines Kontext Typs (`context_type`) ist zwar optional, aber das im LIS definierte „CourseSection“ für einen Kursabschnitt kann problemlos verwendet werden.

Als Einbettungsmodus (`embed_mode`) sollte eigentlich stets „iframe“ verwendet werden. Da aber manche Tool Provider aus Sicherheitsgründen die Darstellung in einem eigenen Fenster erzwingen beziehungsweise voraussetzen, sollte auch hierfür dem Dozenten eine Auswahl zwischen „iframe“ für die Darstellung in einem Iframe und „window“ zum Öffnen eines neuen Fensters ermöglicht werden.

Tool Consumer Die eindeutige ID für den Tool Consumer (`tool_consumer_instance_guid`) sowie die Tool Consumer URL (`tool_consumer_instance_url`) wird einfach aus dem Domainnamen der ownCloud Installation gewonnen.

Der Produktcode (`tool_consumer_info_product_family_code`) wird auf „ownCloud“ gesetzt, die Produktversion (`tool_consumer_info_version`) kann automatisch aus dem installierten System ermittelt werden.

ownCloud an sich kennt weder einen Namen (`tool_consumer_instance_name`) noch eine Beschreibung (`tool_consumer_instance_description`). Für diese müssen Einstellungsmöglichkeiten im administrativen Bereich geschaffen werden.

Als Kontakt-E-Mailadresse (`tool_consumer_instance_contact_email`) könnte man eine beliebige, im System hinterlegte E-Mailadresse eines Administrators benutzen. Da jedoch E-Mailadressen nicht obligatorisch sind und um auch die Verwendung von Mailinglisten zu ermöglichen, sollte auch hierfür ein Parameter im administrativen Bereich gesetzt werden können.

Benutzer Die Nutzer ID (`user_id`) kann automatisch erzeugt werden. Da diese keinen Rückschluss auf die Nutzer ID in ownCloud erlauben soll, kann man entweder einen Hashwert daraus berechnen oder in einer Datenbank

eine Zuordnung zwischen Nutzer ID und einem Zufallswert ablegen.

Die persönlichen Daten des aufrufenden Benutzers können, sofern sie in den Nutzeraccounts hinterlegt sind, von dort bezogen werden. Dazu gehören der vollständige Name (`lis_person_name_full`), die Emailadresse (`lis_person_contact_email_primary`), der Avatar (`user_image`) und die Sprache (`launch_presentation_locale`) des Nutzers. Da der Name nicht in Vor- und Nachname getrennt verfügbar ist, sollte von Versuchen, den Namen in seine Bestandteile zu zerlegen, abgesehen werden und stattdessen die Felder für Vorname (`lis_person_name_given`) und Nachname (`lis_person_name_family`) ausgelassen werden.

Um dem Datenschutz gerecht zu werden, ist eine Möglichkeit vorzusehen, wie der Benutzer die Übermittlung dieser Daten unterdrücken kann. Hierzu wäre eine allgemeine Einstellung wünschenswert, über die sich das Senden beziehungsweise nicht Senden ebenso steuern lässt wie eine Abfrage im Einzelfall. Der geeignete Ort hierfür wären die persönlichen Einstellungen eines jeden Benutzers.

In Bezug auf die Rollen (`roles`) soll nur auf die am häufigsten verwendeten Rollen eingegangen werden. Diese sind Dozent (Instructor), Lernender (Learner), Administrator (Administrator), Hilfslehrkraft (TeachingAssistant) und Mentor (Mentor). Administratoren sind eindeutig identifizierbar, Mentoren und Hilfslehrkräfte als solche hingegen in ownCloud überhaupt nicht vorgesehen. Die Rolle eines Lernenden sollte standardmäßig für alle Benutzer gesetzt werden, die Rolle eines Dozenten jeweils für den Eigentümer einer Ressource.

Interessant wird die Rollenverteilung, sobald eine Ressource mit anderen Nutzern geteilt wird. Hier würde es sich anbieten, die Zuteilung der Rolle Dozent an das Gewähren von Schreibrechten an der Ressource zu koppeln. Ein Benutzer mit Schreibrechten wäre in der Lage, den Schlüssel und insbesondere das geteilte Geheimnis einzusehen und könnte sich mit diesen Informationen eine eigene neue Ressource einrichten. Für diese selbst erstellte Ressource wäre er nun der Besitzer und hätte somit auch den Dozentenstatus.

Alternativ zu dieser Vorgehensweise könnte man die Rollen auch explizit durch den Ersteller der Ressource zuweisen lassen, das Ganze dabei getrennt nach Eigentümer, geteilt mit Schreibrechten und geteilt mit Leserechten.

Custom_fields Eine beliebige Anzahl von Custom_ Feldern muss für den Dozenten zur Verfügung stehen, selbst dann, wenn er nur ein Template platziert.

3.2.1 Einbindung

Nachdem nun geklärt wurde, welche LTI Parameter woher bezogen werden können, bleibt die Frage, wie man die Tool Provider selbst in ownCloud

platziert. Grundsätzlich bieten sich hier zwei Optionen an.

files_lti Die erste Option wäre eine Erweiterung der Standardverzeichnisansicht von ownCloud. Hierzu müsste man einen neuen Dateityp definieren, der alle nötigen Informationen über und für den Tool Provider beinhaltet oder zumindest eine Referenz auf entsprechende Datenbankeinträge enthält. Da nicht wirksam verhindert werden kann, dass ein Benutzer eine mit ihm geteilte Datei herunterlädt und in einem Dateieditor betrachtet, ist zumindest das geteilte Geheimnis durch eine Verschlüsselung vor neugierigen Blicken zu schützen.

Der neue Dateityp wäre als zusätzlicher Eintrag in das „Neu“-Menü zu integrieren und auf diese Weise jederzeit leicht zu instanzieren. Ebenso müsste jedes von einem Administrator angelegte Template in diesem Menü einen Platz finden.

Zu jeder Instanz des neuen Dateityps muss eine zusätzliche Schaltfläche zum Bearbeiten des Inhalts hinzugefügt werden, sofern der Benutzer die entsprechenden Rechte hat. Ebenso wäre eine Anzeige des über den Ergebnisdienst erhaltenen Ergebnisses denkbar, sofern es vom Tool Provider bereits für den aktuellen Benutzer übermittelt wurde.

LTI-App Als Alternative könnte man eine eigenständige App bereitstellen, die speziell auf die Einbindung von Tool Providern zugeschnitten ist. Dem Vorteil der geringeren Abhängigkeit von Änderungen an der Files-App steht ein deutlich erhöhter Entwicklungsaufwand entgegen. So wären unter anderem eine Gruppierungsfunktion sowie das Teilen von Ressourcen selbst zu implementieren.

Zudem ist die ownCloud API selbst so häufigen Änderungen unterworfen, dass auch eine Abkopplung von der Files-App eine Verlängerung der Lebenserwartung nicht schlüssig erscheinen lässt.

4 Implementierung

ownCloud ist serverseitig in PHP und clientseitig in JavaScript geschrieben und basiert auf dem Model-View-Controller Konzept. Es benötigt zur Ausführung einen Webserver und eine SQL-Datenbank. Damit sind die Entwicklungssprachen für Plugins bereits festgelegt.

Es existiert Referenzcode in „lti_util.php“ [Con12] in PHP von IMS GLOBAL für eine LTI Anbindung, sowohl als Tool Provider als auch als Tool Consumer. Dieser kapselt auch komplett das OAuth Protokoll.

4.1 IMS GLOBAL Referenzimplementation

4.1.1 Klasse „BLTI“

Die Klasse „BLTI“ erledigt das Setup eines Verbindungsaufrufs auf Tool Provider Seite und stellt Hilfsfunktionen für die Abfrage der übermittelten Parameter bereit. Um die Gültigkeit einer Kombination aus Schlüssel und geteiltem Geheimnis zu überprüfen, greift sie auf näher spezifizierbare Tabelleneinträge einer MySQL-Datenbank zurück. Dies ist der einzige Teil der Referenzimplementation, der an die Gegebenheiten von ownCloud angepasst werden muss. An Stelle der von der Referenzimplementation erwarteten bereits bestehenden Verbindung zu einer MySQL-Datenbank muss die Datenbankkonfiguration von ownCloud verwendet werden. Die MySQL-spezifischen PHP-Funktionsaufrufe müssen ebenfalls durch ihre von ownCloud vorgegebenen Pendanten ersetzt werden.

4.1.2 postLaunchHTML

Die Funktion „postLaunchHTML“ setzt ein HTML-Formular zusammen und startet dieses automatisch im Browser des Benutzers. Das Formular enthält dabei alle wichtigen Informationen für den Tool Provider und wird als Startpunkt im IFrame oder Fenster des Tool Consumers benutzt. Ein spezieller Debugmodus erlaubt dabei eine Unterbrechung des Startvorganges und die Anzeige der zu übertragenden Daten.

4.1.3 replaceResultRequest

Die Funktion „replaceResultRequest“ setzt ein Ergebnis über den Ergebnisdienst. Sie kapselt das Ergebnis in das zur Übertragung notwendige XML-Konstrukt.

4.2 ownCloud

Die Informationsbeschaffung zur Applikationsentwicklung für ownCloud gestaltet sich nicht einfach. Im Bereich der gedruckten Bücher findet sich nur

ein Werk [Pat13], das sich unter anderem auch mit diesem Thema beschäftigt. Der Umfang dieses Abschnitts liegt mit lediglich acht Seiten jedoch weit unter dem des online verfügbaren Tutorials [own16b]. Ebenso ist dieses Buch inzwischen veraltet, denn es stammt von 2013 und befasst sich deshalb noch mit ownCloud 5.

Die online verfügbare Dokumentation der ownCloud API [own16c] beschränkt sich auf die meist einzeilige Kurzbeschreibung, die automatisch aus den PHP Quelltexten generiert wird. Beispiele für die Nutzung sind darin eher selten zu finden. Darüber hinaus gibt es beim „The ”Fossies” Software Archive“ [Arc16] einen Quelltextbrowser, der auf komfortable Art bei der Durchsuchung der Quelltexte behilflich ist.

Eine aktuelle Beschreibung der JavaScript API kann man sich selber per mitgeliefertem Shellskript erstellen. Auch diese Dokumentation ist jedoch wenig umfangreich.

Was bleibt, ist also die Analyse der bestehenden Apps, sowohl der mitgelieferten als auch derer aus dem Appstore [Tea16]. Gerade bei letzteren ist jedoch erschreckend, wie viele der Apps inzwischen mit der aktuellen Version von ownCloud inkompatibel sind. Dies liegt vor allem daran, dass der stabile Teil der API nicht alle Aspekte von ownCloud umfasst und oft auf Funktionen aus dem instabilen Teil zurückgegriffen werden muss. Ob und in wie weit die App dann in der nächsten Version von ownCloud noch funktionieren wird ist fraglich. Dabei ist selbst der als stabil bezeichnete Teil der API häufigen Veränderungen unterworfen. Das Verzeichnis der Änderungen in ownCloud, das auch schon einen Teil der Anpassungen für die kommenden drei Versionen auflistet, enthält bereits zahlreiche Ersetzungen. Einige davon sind zwar lediglich Ablaufwarnungen, bei anderen sollen aber komplette Teile der API entfallen.

4.2.1 Struktur einer App

Ein Plugin - in ownCloud „App“ genannt - besteht aus einer festgelegten Datei- und Verzeichnisstruktur [dev16], die im Folgenden genauer beschrieben werden soll. Einige Bereiche sind dabei unveränderlich vorgegeben, andere dienen als Empfehlung zur Vereinheitlichung von Apps und wieder andere werden zur Bereitstellung von arbeitserleichternden Funktionen unter vorgegebenen Namen erwartet.

4.2.2 Verzeichnis: appinfo

Das Verzeichnis „appinfo“ enthält einerseits Metainformationen zur App und andererseits den Startpunkt der App, der als erstes von ownCloud aufgerufen wird und zum weiteren Setup der App dient.

info.xml Die Datei „info.xml“ enthält die Metadaten zur App. Neben der ID und dem Namen der App stehen hier Informationen zum Autor und der anzuwendenden Lizenz.

Die Angabe der Version dient einerseits zur automatischen Überprüfung, ob im Appstore von ownCloud eine neuere Version zur Verfügung steht, andererseits wird bei jedem Versionswechsel die Datenbankstruktur (s.u.) auf eventuelle Änderungen überprüft und diese gegebenenfalls eingespielt.

Ebenso können Abhängigkeiten spezifiziert werden, die alle erfüllt sein müssen, damit sich die App installieren lässt. Dazu gehören unter anderem, neben der ownCloud Version, auch die PHP-Version und der verwendete Datenbanktyp.

Außerdem kann der Typ der App angegeben werden. Dies ist insbesondere dann wichtig, wenn die App Authentifizierungs- und Filesystemaufgaben übernehmen soll.

Des Weiteren kann noch die Kategorie angegeben werden, unter der die App im Appstore gelistet werden soll.

app.php Die Datei „app.php“ ist der Startpunkt der App. Sie wird von ownCloud als erstes ausgeführt und initialisiert alle weiteren Komponenten der App. In ihr werden unter anderem die Bereiche in den administrativen und persönlichen Einstellungen, sowie die JavaScript Dateien für den Client registriert.

Es wird auf einen gültigen LTI-Kontext geprüft und, sofern dieser vorhanden ist, die Benutzerverwaltung auf LTI umgestellt.

database.xml Das Datenbankschema der App wird von dem ownCloud zu Grunde liegenden Datenbanksystem abstrahiert spezifiziert. Dazu wird eine XML-Datei verwendet, die der „MDB2 XML Schema documentation“ [Lem04] entspricht. Änderungen an dieser Datei werden automatisch beim nächsten Wechsel der Versionsnummer der App in das Datenbanksystem eingespielt.

routes.php In der Datei „routes.php“ werden Kombinationen aus URL und Zugriffsmethode mit den zugehörigen Methoden der Controller-Klassen verknüpft. Es können sowohl einzelne Routen spezifiziert werden, als auch komplette Ressourcen, eine Kurzschreibweise für die häufig verwendeten Methoden zum Anzeigen eines beziehungsweise aller Datensätze sowie zur Erzeugung, Veränderung und Löschung einzelner Datensätze. Ebenso können an dieser Stelle in die URL eingebettete Variablen für die weitere Verwendung innerhalb der App extrahiert werden.

4.2.3 Verzeichnis: controller

Die „Controller“ bearbeiten die Anfragen aus dem Web. ownCloud wählt mit Hilfe der „routes.php“ die passende Controllermethode aus und ruft sie auf. Der Controller antwortet mit einer sogenannten „Response“. Dies ist üblicherweise eine Webseite, die aus einem Template gewonnen wird, oder ein in JSON oder XML kodierter Datensatz. Es sind aber auch andere Response-typen verfügbar, etwa für Umleitungen, Zugriffsfehler oder Datenströme.

4.2.4 Verzeichnisse: js, img, css

Diese Verzeichnisse beinhalten die JavaScript Dateien, Bilder und CSS-Dateien für die Clientseite. ownCloud stellt Funktionsaufrufe zur Verfügung, die eine automatische Verknüpfung auf diese Verzeichnisse ermöglichen.

4.2.5 Verzeichnis: db

Um den Zugriff auf die Datenbank zu erleichtern, stehen die Klassen „Entity“ und „Mapper“ bereit.

Von Entity werden die Klassen für einzelne Datensätze abgeleitet. Sie abstrahieren den Datensatz von der verwendeten Datenbank und stellen automatisch Getter- und Setterfunktionen für die einzelnen Attribute bereit.

Von Mapper werden die Klassen für Tabellen abgeleitet. Sie abstrahieren die Tabelle von der Datenbank und stellen Funktionen zum Suchen, Einfügen, Ändern und Löschen von Datensätzen bereit.

4.2.6 Verzeichnis: lib

Dieses Verzeichnis enthält eventuell notwendige Funktions- und Objektbibliotheken.

4.2.7 Verzeichnis: service

Die Services entkoppeln die Bereitstellung von Datensätzen von dem verwendeten Datenspeicher. So sind etwa für den Zugriff auf Daten aus Datenbanken oder Dateisystemen verschiedene Zugriffspfade mit entsprechenden Fehlerbehandlungen vorzusehen. Für die Objekte, die diese Datensätze weiter verarbeiten, sollte es aber keine Rolle spielen, woher sie stammen.

4.2.8 Verzeichnis: templates

Die Templates bestehen im Prinzip aus einfachem PHP-Code. Alle Parameter, die vom aufrufenden Controller übergeben werden, stehen in einem speziellen Array zur Verfügung. Zur Ausgabe der Werte steht eine Funktion bereit, die alle HTML-Sonderzeichen durch ihre entsprechende Darstellung ersetzt und somit Cross-Site-Scripting Angriffe verhindert.

4.2.9 Verzeichnis: tests

In dieses Verzeichnis gehören die sogenannten „Unit Tests“, die ein automatisches Testen des entwickelten Codes erlauben.

4.3 Konkrete Implementierung

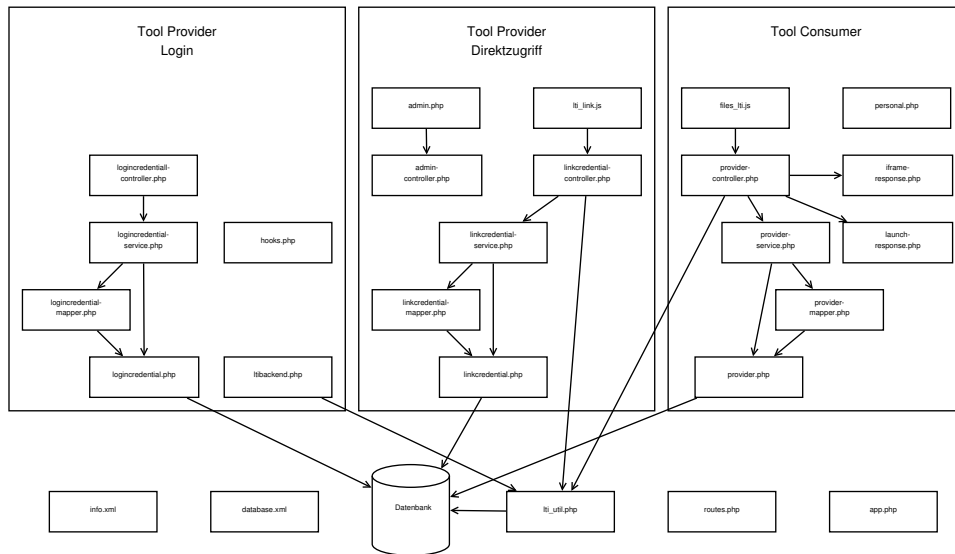


Abbildung 2: Struktureller Aufbau der drei Teilbereiche

Die Implementation lässt sich in drei Teilbereiche gliedern, die in einer gemeinsamen App zusammengefasst sind. Die Pfeile in Abbildung 2 zeigen dabei die Zugriffspfade der einzelnen Bausteine untereinander. Zugriffe auf die Kernfunktionalitäten von ownCloud wurden aus Gründen der Übersichtlichkeit bewusst nicht dargestellt.

4.3.1 Login per LTI

ownCloud bietet zwar verschiedene Möglichkeiten an, eine Kombination aus Kennung und Passwort eines Benutzers zu überprüfen, das Problem bei einer LTI-Verbindung ist aber, dass hier kein Passwort mehr übertragen wird, weil der Benutzer bereits auf Seiten des Tool Consumers verifiziert wurde. Für diesen Fall hat ownCloud das Interface „IApacheBackend“ vorgesehen. Eigentlich ist dieses für den Fall gedacht, dass der Webserver die Authentifizierung des Nutzers übernimmt und an ownCloud nur den Nutzernamen weitergibt. Der Anwendungsfall ist jedoch der gleiche: Eine als vertrauenswürdig eingestufte Instanz hat den Nutzernamen ermittelt und reicht diesen weiter.

Zu diesem Zweck ist eine Klasse zu erstellen, die als Nutzerbackend dienen kann und das IApacheBackend Interface implementiert. Sofern eine gültige LTI-Verbindung vorliegt, kann dann dieses Nutzerbackend aktiviert werden. Gleichzeitig wird ein Hook installiert, der nach einem erfolgreichen Login aufgerufen wird. Ein authentifizierter Nutzer wird darüber gegebenenfalls neu erstellt. Sofern seine persönlichen Daten übertragen wurden, werden diese dabei in seinen Account in ownCloud übernommen. Ebenso wird sein Account den Gruppen zugeordnet, die automatisch aus dem Kontext ermittelt wurden.

Um festzustellen, ob eine LTI-Verbindung gültig ist, ist eine Datenbanktabelle nötig, die die zulässigen Kombinationen aus Schlüssel und geteiltem Geheimnis beinhaltet. Dazu wird in der „database.xml“ eine Tabelle definiert, die diese Werte enthält. Die passenden Entity, Mapper und Service Klassen sowie ein Controller sind anzulegen.

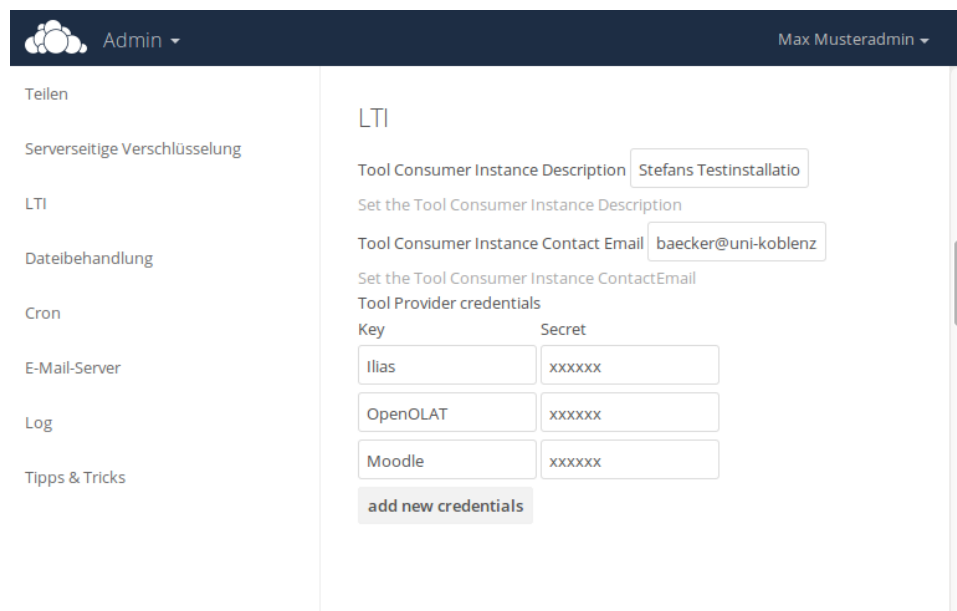


Abbildung 3: Bildschirmfoto: LTI spezifischer Teil des administrativen Bereichs

Die Inhalte der Tabelle werden im Adminbereich von ownCloud gesetzt. Dazu wird beim Start der App ein Stück Code registriert, der eine zusätzliche Sektion im Adminbereich bereitstellt. In dieser Sektion wird eine Tabelle angezeigt, deren Inhalt per Ajax über den Controller bezogen und gesetzt wird.

4.3.2 Direktzugriff auf Ressourcen

ownCloud ermöglicht es, Ressourcen per Link zu teilen und per Passwort zu schützen. Dazu erzeugt es eine zufällige Ressourcen ID, die als Schlüssel für

die LTI Verbindung genutzt werden kann. Zu diesem Schlüssel muss noch das entsprechende geteilte Geheimnis erzeugt werden und beides gemeinsam in einer Tabelle in der Datenbank abgelegt werden. Das ohnehin vorhandene Passwort lässt sich leider nicht verwenden, denn es liegt nur verschlüsselt in der Datenbank vor.

Es ist also in der „database.xml“ eine neue Tabelle anzulegen, die neben der obligatorischen ID einen Schlüssel und ein geteiltes Geheimnis als Spalten enthält. Für den Zugriff auf diese Tabelle ist ein passender Satz aus Entity, Mapper und Service zu erstellen.

Um nun auf eine Ressource per LTI zuzugreifen, ist nur noch ein Controller notwendig, der die Gültigkeit der übermittelten Daten mit Hilfe der `lti_util.php` und der Datenbanktabelle prüft. Sind diese korrekt, so setzt er den internen Authentifizierungsstatus entsprechend und leitet die Anfrage anschließend an ownCloud weiter. Da die Autorisierung dann bereits vorliegt, wird die Ressource ohne weitere Sicherheitsabfragen angezeigt.

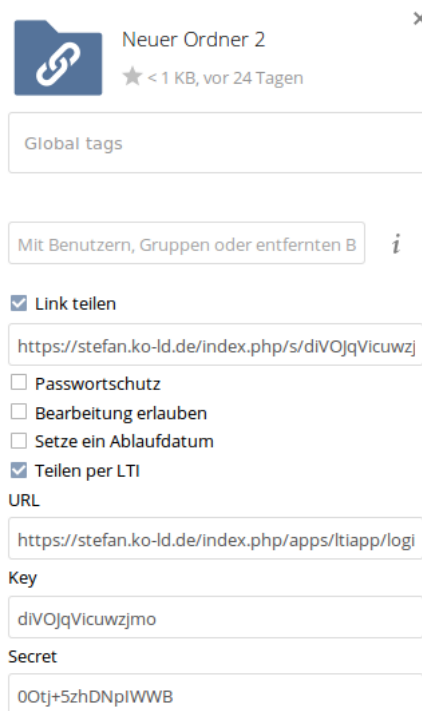


Abbildung 4: Bildschirmfoto: Erweiterter „Teilen“-Dialog

Um die Tabelle mit Inhalt zu füllen, ist neben einem passenden Controller noch etwas JavaScript nötig. Dieses muss im Browser die Funktionalität des Teilen Dialoges erweitern. Sofern die Checkbox für „Teilen per Link“ angewählt wird, muss eine zusätzliche Sektion für „Teilen per LTI“ angezeigt werden. Sofern die darin enthaltene Checkbox angewählt wird, müssen

die Zugangsdaten erzeugt und angezeigt werden. Bei einer Abwahl müssen sowohl die Zugangsdaten als auch die Anzeige wieder entfernt werden.

4.3.3 Tool Consumer

Für den Tool Consumer wurde eine Integration in den Dateibrowser implementiert. Dadurch kann das Handling von Gruppierungen in Form von Verzeichnissen und von Shares direkt von dort übernommen werden. Die Tool Provider werden dabei als Dateien vom neu anzulegenden Mimetype „application/lti“ erzeugt und in die Verzeichnisstruktur eingebunden. Die Dateien selbst haben dabei keinerlei Inhalt, lediglich der Dateiname und die interne Datei-ID finden Verwendung. Als Datenspeicher für die LTI-Verbindungsparameter dient eine Datenbanktabelle, deren Einträge mittels der Datei-ID referenziert werden. Für diese Tabelle sind wiederum Entity-, Mapper-, Service- und Controllerklassen bereitzustellen.

Ab der Version 9 von ownCloud wird es einen API-Aufruf geben, mit dem man auf einfache Art zusätzliche Einträge in das „Neu“ Menü einfügen kann. Hierüber wird ein neuer Eintrag zur Erzeugung einer LTI-Datei erstellt.

Zusätzlich sind zwei neue Dateiaktionen zu registrieren, nämlich eine zur Bearbeitung der LTI-Parameter und eine zur Anzeige des Tool Providers. Letztere wird außerdem als Standardaktion gesetzt und ersetzt damit das Herunterladen der Datei.

Die Methode zum Bearbeiten der LTI Parameter öffnet ein Overlay, lädt die Daten per Ajax aus der Datenbank und zeigt sie in einem Formular an. Ein Klick auf den „Speichern“-Schaltfläche schreibt die Daten ebenfalls per Ajax zurück in die Datenbank.

Die Methode zur Anzeige des Tool Providers öffnet ebenfalls ein Overlay und darin unterhalb des Namens und einer „Schließen“-Schaltfläche einen Iframe. Da der Dateibrowser von ownCloud nur Seiten der eigenen Website innerhalb eines Iframes zulässt, ist hierin zuerst ein weiterer Iframe zu öffnen, der auch den Tool Provider als Quelle erlaubt. Anschließend wird in dem inneren Iframe dann der eigentliche Tool Provider gestartet.

Dazu dient eine eigene Response-Klasse, die die `postLaunchHTML` Funktion der `lti_util.php` nutzt und diese mit den Daten des Tool Providers, der verwendeten Ressource, den Benutzerdaten und den Tool Consumer-Daten aufruft.

Die Daten des Tool Providers wie URL, Schlüssel und geteiltes Geheimnis kommen dabei aus der Datenbanktabelle.

Die Daten der Ressource, also die Ressourcen-ID und deren Titel, stammen aus dem Dateisystem. Da jeder Benutzer die Datei jedoch in seinem eigenen Verzeichnis sieht, kann man dieses nicht für den Kontext verwenden. Um trotzdem einen Kontext setzen zu können, werden darum erneut die Daten der Datei verwendet.

Die Benutzerdaten des eingeloggten Benutzers werden direkt aus der

Konfiguration von ownCloud ausgelesen. Dies geschieht allerdings nur, sofern dieser dies vorher explizit erlaubt hat. Dazu ist ein neuer Abschnitt in die persönlichen Einstellungen einzufügen, über den der Benutzer seine Präferenzen einstellen kann.

Die Bereitstellung der Tool Consumer-Daten erfolgt sofern vorhanden auf Basis der Systemkonfiguration. Für die nicht aus der Konfiguration ableitbaren Daten ist der schon vorhandene Abschnitt im Administrationsbereich entsprechend zu erweitern.

5 Ergebnisse und Bewertung

In diesem Kapitel soll zuerst der entwickelte Prototyp auf seine Kompatibilität mit vorhandenen Tool Consumern und Tool Providern getestet werden. Dazu wird zunächst eine Auswahl aus beiden Kategorien getroffen und die Systeme bezüglich ihrer LTI Konzepte kurz vorgestellt und getestet. Daran anschließend werden jeweils mögliche Nutzungsszenarien sowie Probleme aufgezeigt.

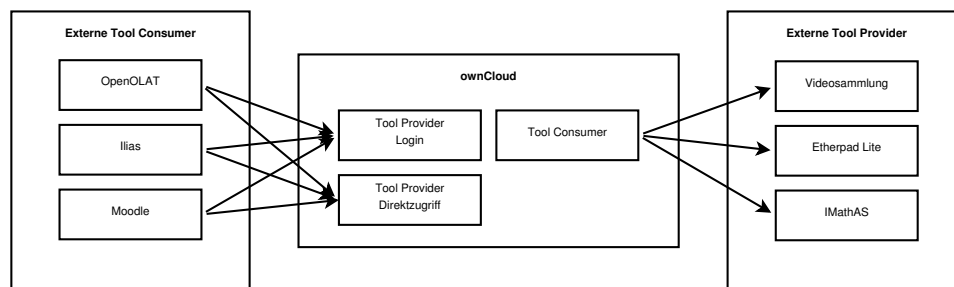


Abbildung 5: Zusammenspiel von Tool Consumern und Tool Providern im Test

Da die LTI App in ownCloud sowohl zwei Tool Provider als auch einen Tool Consumer implementiert, werden diese Teilbereiche getrennt untersucht. Die Pfeile in Abbildung 5 zeigen dabei die untersuchten Verbindungen zwischen Tool Consumern und Tool Providern an.

Eine abschließende Zertifizierung in Hinsicht auf die LTI Funktionalität wäre über IMS GLOBAL [Con16b] möglich. Im Zuge der Zertifizierung sind für die verschiedenen Stufen der LTI Standards sowohl auf Tool Consumer als auch auf Tool Provider Seite eine Vielzahl von Tests vorgesehen, die alle Aspekte des Standards abdecken sollen. Nach bestandenen Tests kann die Zertifizierung für den Tool Consumer oder den Tool Provider beantragt werden. Da sowohl die Tests als auch die Zertifizierung selbst nur zahlenden Mitgliedern des IMS GLOBAL Learning Consortiums vorbehalten ist, musste im Rahmen dieser Arbeit von der Durchführung der Tests abgesehen werden.

5.1 Auswahl der externen Tool Consumer

Wie bereits erläutert integrieren die Tool Consumer die von den Tool Providern zur Verfügung gestellten Inhalte. Im E-Learning-Umfeld kommen häufig Lernmanagementsysteme zum Einsatz, um Inhalte und Aufgaben zur Verfügung zu stellen. In dieser Arbeit sollen exemplarisch drei solcher Lernmanagementsysteme als Tool Consumer vorgestellt und genutzt werden.

„OpenOLAT“ [Ope15] ist das strategische Lernmanagementsystem der

Universität Koblenz-Landau, welches durch den Virtuellen Campus Rheinland-Pfalz (VCRP) in Kaiserslautern gehostet wird.

„Ilias“ [ILI16] wird ebenfalls an der Universität Koblenz-Landau genutzt. Allerdings ist die Nutzung hier auf einen speziellen Bereich, den der E-Klausuren, eingeschränkt. Es wird durch das Institut für Wissensmedien (IWM) nur zu diesem Zweck gehostet.

„Moodle“ [Moo16] hingegen wird nicht als offizielles Lernmanagementsystem an der Universität Koblenz-Landau eingesetzt, jedoch ebenfalls vom IWM für ein externes Projekt gehostet.

Alle drei Lernmanagementsysteme sind OpenSource und gut dokumentiert. Sie sind jeweils an verschiedenen Universitäten in Deutschland im Einsatz.

5.1.1 OpenOLAT

In OpenOLAT existiert der Kursbaustein „LTI-Seite“. Damit ist es möglich, an beliebigen Stellen innerhalb eines Kurses einen oder mehrere Tool Provider einzubinden. Neben den gemeinsamen Einstellungen aller Kursbausteine zu Titel und Beschreibung, Sichtbarkeit und Zugang, kommt hierbei noch der LTI spezifische Seiteninhalt. In diesem müssen die URL, der Schlüssel und das geteilte Geheimnis des Tool Providers als Pflichtfelder eingegeben werden. Darüber hinaus sind Einstellungen möglich, die die Übertragung von Namen und E-Mailadresse des Kursteilnehmers an den Tool Provider erlauben, sowie ein Zuordnung von Rollen in OpenOLAT zu LTI-Rollen.

Der LTI-Ergebnisdienst lässt sich einschalten und neben einem Skalierungsfaktor, mit dem man den zurückgelieferten Wert zwischen 0,0 und 1,0 an das Kursniveau anpassen kann, auch noch die für das Bestehen des Kursbausteins notwendige Punktzahl angeben.

Für die Darstellungsform kann erstens festgelegt werden, ob der Tool Provider in einem Iframe oder in einem eigenen Fenster geöffnet werden soll und zweitens die Höhe und Breite der Anzeigefläche angegeben werden.

Zu guter Letzt lässt sich noch der Debugmodus einschalten, der vor dem eigentlichen Start des Tool Providers alle Daten, die übertragen werden, anzeigt.

Die Spracheinstellung wird standardmäßig übermittelt. Die Übertragung eines Avatars ist hingegen nicht vorgesehen.

5.1.2 Ilias

Ilias selbst enthält keine LTI-Schnittstelle, kann aber über das Plugin „ExternalContent“ [Neu15] zum Tool Consumer aufgerüstet werden. Dieses Plugin stellt den Objekttyp „Externer Inhalt“ zur Verfügung. Bevor dieser Objekttyp jedoch verwendet werden kann, muss von einem Administrator mindestens ein Inhaltstyp definiert werden. Dazu ist zuerst in der Plugin-Konfiguration

LTI-ownCloud Login

Titel und Beschreibung Sichtbarkeit Zugang **Seiteninhalt**

Vorschau anzeigen

Konfiguration LTI Seite

Hilfe

*** URL**
Beispiel: http://ltiapps.net/test/tp.php

*** Schlüssel**
Beispiel: jisc.ac.uk

*** Passwort**
Beispiel: secret

Vorname/Name übertragen

E-Mailadresse übertragen

Zusätzliche Attribute custom Benutzer + -

OpenOLAT Rollen LTI Rollen

Besitzer Lerner Instruktor Administrator Assistent Lehrperson Inhaltersteller Mentor

Betreuer Lerner Instruktor Administrator Assistent Lehrperson Inhaltersteller Mentor

Teilnehmer Lerner Instruktor Administrator Assistent Lehrperson Inhaltersteller Mentor

Punkte übertragen

Skalierungsfaktor

Notwendige Punktzahl für 'bestanden'

Anzeige Eingebettet in Kurs (iFrame)
 Nur Modul anzeigen, LMS ausblenden (iFrame)
 Neues Fenster öffnen

Höhe Anzeigefläche

Breite Anzeigefläche

Alle beim Start gesendete Information anzeigen (Debug)

Speichern

Abbildung 6: Bildschirmfoto: Einstellungen des LTI Kursbausteins in Open-OLAT

SETTINGS
Save Cancel

Type LTI 1.1 (Deutsche Oberfläche)
Verwendet LTI 1.1 mit Basic Outcome Service, siehe <http://developers.imsglobal.org>

Title *
Give the object a title

Description
The description will be shown below the title

Online
This makes the object visible and usable for the users

AUFRUF-EINSTELLUNGEN

URL *
URL des externen Tools, z.B. <http://www.imsglobal.org/developers/LTI/test/v1p1/tool.php>

Schlüssel *
Schlüssel für den Zugriff, z.B. 12345

Geheimnis *
Geteiltes Geheimnis mit dem Tool-Anbieter, e.g. secret

ERSCHEINUNGSBILD

Darstellung

Eingebettet
Zeigt das Tool auf der Inhaltsseite eingebettet.

Aktuelles Fenster
Öffnet das Tool im aktuellen Fenster. Es kann einen Rücksprung-Link anzeigen.

Neues Fenster
Öffnet das Tool in einem neuen Fenster

Datenüberprüfung
Zeigt einen Link zum Betrachten der übermittelten Daten

* Required
Save Cancel

Abbildung 7: Bildschirmfoto: Einstellungen des External Contents in Ilias

ein Typname einzugeben, sowie eines der drei LTI 1.1 Typ-Modelle in der jeweils gewünschten Sprache auszuwählen. Anschließend können Einstellungen zum Datenschutz vorgenommen werden. Es kann ausgewählt werden, wie der LTI-Parameter „user_id“ instantiiert wird. Neben der Einstellung, überhaupt keine Kennung zu senden, sind hier noch ein eindeutiger Code sowie die interne Nutzer-ID beziehungsweise der Loginname möglich. Weiterhin kann hier festgelegt werden, welche persönlichen Daten wie Name, E-Mailadresse und Bild des Nutzers an den Tool Provider gesendet werden.

Nachdem von einem Administrator ein neuer Inhalts-Typ erstellt wurde, kann ein Autor diesen über den Objekt-Typ „Externer Inhalt“ platzieren. Er gibt dem Objekt einen Titel und wählt einen der zuvor definierten Inhaltstypen aus. Anschließend muss er noch die URL, den Schlüssel sowie das geteilte Geheimnis für die LTI-Verbindung eingeben. Des weiteren kann er festlegen,

ob der Tool Provider eingebettet oder in einem eigenen Fenster dargestellt werden soll. Zusätzlich kann er noch den Debugmodus zum Überprüfen der übertragenen Daten einschalten.

Ein Mapping von Rollen in Ilias zu Rollen im Tool Provider ist ebenso wenig möglich, wie die Angabe von Custom Parametern.

5.1.3 Moodle

Im Lernmanagementsystem Moodle [Moo16] ist der LTI Tool Consumer bereits in Form der Aktivität „Externes Tool“ vorhanden. Neben den für alle Aktivitäten benötigten Angaben wie Name und Beschreibung ist dabei zunächst zu wählen, ob ein vorgefertigter Kurs-Tooltyp verwendet werden soll, oder ob der Dozent alle LTI-spezifischen Angaben individuell konfigurieren will. Soll ein Kurs-Tooltyp verwendet werden, so ist dieser vorher von einem Dozenten oder Administrator zu definieren.

Für diese Definition sind lediglich ein Name und die Basis URL des Tool Providers Pflichtfelder, alle weiteren Angaben sind optional. Einige Werte sind dabei durch den Dozenten bei der späteren Platzierung nicht änderbar, andere sind nur Vorgaben, die er jederzeit anpassen kann.

Zu den nicht änderbaren Werten gehört neben der Basis URL des Tool Providers der Schlüssel, das geteilte Geheimnis und die URLs zu alternativen Icons, sofern die Aktivitäten eine eigene Darstellungsweise erhalten sollen.

Des weiteren können Custom Parameter angegeben und die Darstellung innerhalb eines Iframes oder eines Fensters gewählt werden.

Bezüglich des Datenschutzes kann eingestellt werden, ob der Name und die E-Mailadresse des Nutzer übertragen werden sollen und ob der Ergebnisdienst eingeschaltet werden soll. Zu jedem dieser drei Punkte besteht jeweils die Möglichkeit, sie grundsätzlich zu erlauben oder zu verbieten, beziehungsweise die Entscheidung dem Dozenten zu überlassen. Darüber hinaus kann noch eine SSL-verschlüsselte Verbindung zum Tool Provider erzwungen werden.

Sofern der Dozent einen vordefinierten Kurs-Tooltyp auswählt, kann er nur noch die Darstellungsform und die Custom Parameter eingeben, sowie die ihm zugestandenene Einstellungen zum Datenschutz vornehmen. Andernfalls kann er die Automatikeneinstellung wählen, die versucht, die Grundeinstellungen anhand der angegebenen Tool Provider-URL aus den konfigurierten Kurs-Tooltypen zu ermitteln, dem Dozenten aber trotzdem die einzelnen Einstellungen offen lässt.

Sollte der Empfang von Ergebnissen eingeschaltet sein, so ist noch die Interpretation der Bewertung zu konfigurieren.

Grundeinträge

Name der Aktivität* LTI-ownCloud Login

Beschreibung der Aktivität*

Beschreibung im Kurs zeigen
 Aktivitätenname bei Start anzeigen
 Beschreibung bei Start anzeigen

Typ des externen Tools* LTI Login + ✎ ✕

Start URL*

Sichere Start-URL*

Startcontainer* Standardwert

Anwenderschlüssel*

Öffentliches Kennwort* Klartext

Angepasste Parameter*

Icon URL*

Sichere Icon-URL*

[Weniger anzeigen...](#)

Datenschutz

Anwendername an Tool übergeben
 E-Mail des Anwenders an Tool übergeben
 Bewertungen des Tools akzeptieren

Bewertung

Bewertung* Typ Punkt

Skala Anwendung von Einzelfakten oder ganzheitliche Wissensnutzung

Maximalpunkte 100

Bewertungskategorie* Nicht kategorisiert

Bewertung zum Bestehen* 0,00

Weitere Einstellungen

Pflichtfelder*

Abbildung 8: Bildschirmfoto: Einstellungen in der Aktivität „External Tool“ in Moodle

Die Spracheinstellung wird bei jedem LTI Aufruf an den Tool Provider übergeben. Der Avatar des Benutzers fehlt jedoch, ebenso wie eine Möglichkeit, einen Debugmodus zu aktivieren, der die zu übertragenden Parameter vor der tatsächlichen Ausführung anzeigt.

5.2 Test der ownCloud Tool Provider

Ziel dieser Tests ist es, herauszufinden, ob die drei ausgewählten Tool Consumer die Tool Provider mit allen nötigen Informationen versorgen können, um diese zu starten. Hierzu werden in allen drei Systemen die entsprechenden LTI-Bausteine platziert und entsprechend der Vorgaben konfiguriert. Anschließend wird überprüft, ob die übertragenen Daten ihren, der Konzeption entsprechenden, Platz in ownCloud gefunden haben.

5.2.1 Login per LTI

Für das Login per LTI ist als URL die Startseite von ownCloud inklusive dem anhängenden „index.php“ zu verwenden, da die komplette URL in die Berechnung der OAuth Prüfsumme eingeht. Als Schlüssel und geteiltes Geheimnis kann ein beliebiges, der im Administrationsbereich von ownCloud hinterlegten Paare verwendet werden.

Sofern das angegebene Paar validiert werden kann, wird der Benutzer in ownCloud eingeloggt. Sollte der Benutzer noch keine Kennung in ownCloud besitzen haben, wird diese während des Einloggens für ihn erstellt.

Sofern keine gültigen Parameter übertragen wurden, wird der Nutzer zur normalen Loginseite von ownCloud weitergeleitet.

Falls der Nutzernamen oder dessen E-Mailadresse übertragen werden, werden diese in den eingeloggt Account übernommen.

Ebenso werden bei Bedarf die Gruppen angelegt, die sich aus dem Namen der Ressource, dem Namen des Kontextes, der Instanz-ID des Tool Consumers sowie des Schlüssels ergeben, und dem Account zugeordnet.

Die Darstellung erfolgt sowohl in einem Iframe als auch in einem eigenen Fenster in allen drei untersuchten Systemen einwandfrei.

Lediglich bei Ilias ist darauf zu achten, dass ein Inhalts-Typ verwendet wird, für den der Administrator die Übertragung einer User-ID erlaubt hat. Andernfalls weist ownCloud zu Recht auf die fehlende Identifikation hin. Ebenso überträgt das Plugin keinen Kontext, so dass dementsprechend auch keine Gruppe dafür angelegt wird.

Nutzungsmöglichkeiten Da ein ownCloud Server nur einen Webserver mit PHP Unterstützung und eine Datenbank benötigt, ist dieser schnell aufgesetzt. Ob dies in der eigenen Organisation, zu Hause oder bei einem beliebigen Provider geschieht, spielt dabei kaum eine Rolle. Was eine ausgewogene Lösung ist, die sowohl die Datenhoheit als auch die Kosten und die

verfügbare Bandbreite berücksichtigt, muss allerdings jeder Anwender selbst entscheiden. Sobald der eigene ownCloud Server vorhanden ist, muss nur noch die App darin installiert werden, um den eigenen Datenspeicher per LTI in das Lernmanagementsystem der Wahl einzubinden.

In diesem Datenspeicher können anschließend die Lernenden einerseits auf die Materialsammlungen zugreifen, die der Dozent mit ihnen geteilt hat oder sogar selbst welche anlegen und mit anderen Kursteilnehmern teilen, etwa im Rahmen einer Gruppenarbeit. Eine Materialsammlung ist dabei nicht nur auf Dateien beschränkt, sondern kann alle möglichen Apps von ownCloud beinhalten, so zum Beispiel auch Kontakte oder Termine.

Die verschiedenen Synchronisationsmöglichkeiten von ownCloud erleichtern dabei die Bereitstellung und den Abruf der Daten sehr.

Probleme Das grundlegende Problem dieses Ansatzes liegt darin begründet, dass ein Tool Consumer zwar neue Nutzer anmelden und somit im Tool Provider erzeugen kann, es in LTI aber keinen Mechanismus gibt, mit dem ein Tool Consumer einen Tool Provider über die Löschung eines Benutzeraccounts informieren kann. Wird ein Account also in einem Lernmanagementsystem gelöscht, etwa wegen Inaktivität oder Ausscheidens, so wird der korrespondierende Nutzeraccount in ownCloud verwaisen und die von ihm gespeicherten Daten früher oder später zu ungenutztem Datenmüll. Gerade im Umfeld der Aus- und Weiterbildung, in dem die Lernenden in der Regel nur relativ kurze Zeit verbleiben, ist diese Fluktuation nicht zu unterschätzen.

Wirksame Methoden zur Erkennung und anschließenden Beseitigung von Datenmüll fehlen in ownCloud, so dass diese noch zu implementieren wären. Eine Möglichkeit, beim ursprünglichen Tool Consumer eine Information über den Zustand des Nutzeraccounts zu erhalten, sollte einem rein zeitgesteuerten Verfahren zur Erkennung ungenutzter Inhalte und Nutzeraccounts dabei weit überlegen sein. Darüber hinaus wäre noch zu bedenken, dass Inhalte eventuell von anderen Nutzern benötigt werden könnten, selbst wenn der Ersteller nicht mehr im System aktiv ist.

Ein ähnliches Problem ergibt sich aus der automatischen Erzeugung und Zuordnung der Gruppen. Auch hier besteht für den Tool Consumer weder die Möglichkeit noch die Veranlassung den Tool Provider von der Löschung oder Umbenennung von Ressourcen oder Kontexten zu informieren. Somit werden sich im Laufe der Zeit immer mehr ungenutzte Gruppen ansammeln, die für Verwirrung der Nutzer sorgen könnten.

5.2.2 Direktzugriff auf Ressourcen

Für den Direktzugriff auf Ressourcen sind für URL, Schlüssel und geteiltes Geheimnis die Werte aus der „Teilen per LTI“-Box zu verwenden. Von den restlichen Einstellungsmöglichkeiten sind nur die für die Darstellung in einem Iframe oder neuen Fenster relevant, da die sonstigen Daten vom Tool

Provider ignoriert werden.

Sofern das Paar aus Schlüssel und geteiltem Geheimnis validiert werden kann, wird die öffentliche Linkseite von ownCloud angezeigt und es stehen dem Benutzer alle vorgesehenen Möglichkeiten offen. Verschiedene Mediendateien wie Texte, Bilder oder Videos können darauf direkt angezeigt werden, alle anderen werden zum Herunterladen angeboten.

Die Einbindung funktioniert mit allen drei untersuchten Tool Consumern einwandfrei.

Nutzungsmöglichkeiten Mit Hilfe dieses Tool Providers ist es einem Dozenten sehr einfach möglich, die in seinen Kursen benötigten Arbeitsmaterialien selbst bereitzustellen und zu verwalten. Durch die Verwaltung außerhalb des Lernmanagementsystems ist er dabei weder von einer Quotierung noch von einer Rechtevergabe durch dessen Betreiber abhängig. Er hat zudem die Möglichkeit, den Zugriff auf einzelne Dateien sehr feingranular zu steuern.

Ebenso kann es sein, dass eingeschränkte Nutzungsrechte ein Kopieren von Materialien auf ein extern betriebenes Lernmanagementsystem komplett verbieten. Auch in diesem Fall kann ownCloud eine Alternative bieten.

Auch für nur kurzzeitig angestellte Dozenten könnte es durchaus interessant sein, die eigene Materialsammlung auf einer eigenen ownCloud vor zu halten und diese so auch nach einem Wechsel des Arbeitgebers weiterhin zur Verfügung zu haben.

Des Weiteren ist es über diesen Tool Provider möglich, von einem gegebenen Lernmanagementsystem nicht direkt unterstützte Medientypen, wie etwa Videos, direkt im Browser des Benutzers wiederzugeben.

Probleme Das Problem dieses Tool Providers ist, dass er sich nur auf die Datei App von ownCloud beschränkt und somit auch nur in dieser Ansicht zur Verfügung steht. Es existieren zwar noch andere Apps, die ebenfalls ein Teilen ihrer Inhalte erlauben, da es jedoch noch keinen Standard für den „Teilen Dialog“ gibt, müsste dieser für jede dieser App einzeln angepasst werden.

Drüber hinaus ist die Unterstützung von Dateiformaten zur direkten Anzeige und Interaktion im per Link geteilten Fenster noch ausbaufähig. Außer ein paar Bild- und Videoformaten werden alle anderen lediglich zum Herunterladen angeboten.

Zwar gibt es einige Apps, die zusätzliche Dateiformate auch innerhalb der Dateien App zur Ansicht bringen können. Da es dafür aber keinerlei Richtlinien oder standardisierte Methoden innerhalb ownClouds gibt, geschieht dies stets auf eine eigene Weise. Dabei wird der Teil des per Link Teilens aber meist vernachlässigt.

Hier wäre es wünschenswert, wenn es ein einheitliches Verfahren seitens ownCloud gäbe, wie Dateiinhalte zur Anzeige gebracht werden können - so

wohl in der Dateien App als auch auf der per Link geteilten Seite.

5.3 Auswahl der externen Tool Provider

Als Tool Provider sollen an erster Stelle diejenigen untersucht werden, die bereits an der Universität Koblenz-Landau im Kontext von OpenOLAT Verwendung finden. Dies sind eine Videosammlung mit integriertem Abspieler, ein Wrapper für „Etherpad Lite“ und das Mathematik Assessmentsystem „IMathAS“. Diese bilden einen kleinen Querschnitt über verschiedene Methoden von LTI Umsetzungen und eignen sich damit in ihrer Kombination für ein Testbett.

5.3.1 Videosammlung

Bei der Videosammlung handelt es sich um eine kleine Anwendung, die einen geschützten Zugriff auf einzelne Videos erlaubt.



Abbildung 9: Bildschirmfoto: Videosammlung in der erweiterten Dozentenansicht

Als Benutzer mit der LTI-Rolle „Learner“ erhält man dabei lediglich den Zugriff auf einen Videoplayer, der das mit der LTI-Ressourcen-ID verknüpfte Video abspielen kann. Als Benutzer mit der LTI-Rolle „Instructor“ erhält man zusätzlich die Möglichkeit, neue Videos in die mit dem LTI-Schlüssel verknüpfte Sammlung hoch zu laden, sowie in der Sammlung vorhandene Videos mit der Ressourcen-ID zu verknüpfen oder zu löschen.

Die LTI-Rolle „Instructor“ ist im ownCloud Plugin an die Bearbeitungsrechte an der Datei gekoppelt. Wer die Verbindungsdaten bearbeiten und somit einsehen kann, könnte diese auch in eine eigene Datei kopieren.

5.3.2 Etherpad Lite

Etherpad Lite [Fou16] ist eine servergestützte Browseranwendung, bei der mehrere Personen gleichzeitig an einem Textdokument arbeiten können. Sie unterstützt dabei die Zuordnung einzelner Textpassagen zu ihren jeweiligen Autoren, sofern diese ihren Namen hinterlegt haben. Ebenso ist es möglich, den Zugriff auf Gruppen von Dokumenten durch einen Loginmechanismus einzuschränken.

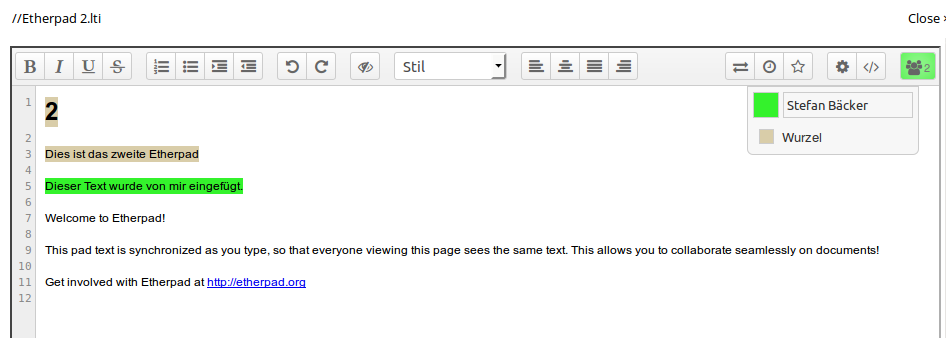


Abbildung 10: Bildschirmfoto: Etherpad Lite mit aufgeklappter Autorenübersicht

Da die Entwicklung von Etherpad Lite auf den eingebetteten Einsatz ausgerichtet ist und eine entsprechende API mitgeliefert wird, war die Implementation einer Tool Provider Schnittstelle ohne großen Aufwand möglich. Sie verwendet die ID des Nutzers und dessen Namen, um in Etherpad Lite einen Autor anzulegen beziehungsweise diesen wieder einzuloggen. Da ein Zugriffsschutz nur für Gruppen von Dokumenten, nicht aber für einzelne Dokumente möglich ist, wird zuerst eine Gruppe aus der Ressourcen ID erzeugt und anschließend in dieser Gruppe ein Textdokument mit der gleichen ID. Daraufhin wird die Sitzung für die Kombination aus Gruppe und Autor freigeschaltet und zum eigentlichen Textdokument weitergeleitet.

5.3.3 IMathAS

IMathAS [Lip16] ist ein Assessmentsystem, das speziell auf Fragen aus dem Bereich der Mathematik abzielt. Es kann allein stehend betrieben werden, erlaubt aber auch die Einbindung von Kursen und Assessments in andere Lernumgebungen per LTI. Dazu ist vom Kursautor ein geteiltes Geheimnis für den Kurs und die darin enthaltenen Assessments zu vergeben.

Anschließend kann er über verschiedene vorgegebene Schlüssel entweder direkt den kompletten Kurs oder nur ein einzelnes der Assessments einbinden. Ebenso ist es möglich, über einen weiteren Schlüssel einem Dozenten der einbindenden Lernumgebung die Auswahl des zu verwendenden Assessments innerhalb des Kurses zu erlauben.

test assessment

[Show Intro/Instructions](#)

Fragen

▶ [Q 1 \(0/10\)](#)

▶ [Q 2 \(0/10\)](#)

Note:0/20

[Druckversion](#)

Skizziere den Graphen von $y = x^2 - 3$

[Alles leeren](#) Zeichnen: [Linie](#)

Antwort 1: Geben Sie Ihre Antwort ein, indem Sie zeichnen.

Punkte möglich: 10 [License](#)
Beliebig viele Versuche.

[Absenden](#)

Abbildung 11: Bildschirmfoto: Assessment in IMathAS

Jeden dieser Schlüssel gibt es zusätzlich in jeweils zwei Ausprägungen. Einmal wird für den Studenten automatisch in IMathAS eine Kennung erstellt, zu der er nur über die aufrufende Lernumgebung Zugriff hat. Bei der anderen Variante muss der Student bei der ersten Verwendung zunächst seine Kennung aus der aufrufenden Lernumgebung mit einer bereits bestehenden Kennung in IMathAS verknüpfen.

Zusätzlich zu der Einbindung eines Kurses oder Assessments ist es zudem möglich, dass vom Administrator von IMathAS Paare von Schlüsseln und geteilten Geheimnissen angelegt werden, über die sich eine Single Sign-On Lösung bewerkstelligen lässt. Mit Hilfe dieser Paare kann man also ein Login von einem Lernmanagementsystem an IMathAS weiterreichen. Auch hierbei gibt es wieder die zwei Optionen, automatisch eine Kennung zu erstellen oder eine bereits vorhandene zu verknüpfen.

5.4 Test des ownCloud Tool Consumers

Zum Test werden für die ausgewählten Tool Provider Dateien erzeugt und entsprechend der Vorgaben aus den Tool Providern konfiguriert. Anschließend wird versucht, sie zu starten und auf korrekte Übertragung aller benötigten Parameter überprüft. Zusätzlich soll untersucht werden, wie gut sich der neue Dateityp in die Dateien App integriert, sprich welche der üblichen Dateioperationen auf ihm ausgeführt werden können, ohne die Funktionalität zu beeinträchtigen.

Das Anlegen neuer Tool Provider-Verbindungen, Editieren der Verbindungsdaten und Anzeigen aller drei Tool Provider aus ownCloud heraus funktioniert einwandfrei. Alle benötigten Parameter werden beim Start der Verbindung bereitgestellt und korrekt übertragen.

Die Filterung der Dateien mittels Favoriten beziehungsweise der „Geteilt“-ansichten hat keinen störenden Einfluss auf die Funktion. Das Umbenennen der Dateien erhält die Funktion, sofern die Dateierweiterung und damit der ihr zugeordnete neue Mimetype erhalten bleibt.

Das Verschieben der Dateien in andere Verzeichnisse sowie Löschen und anschließendes Wiederherstellen erhalten die File-ID und somit die Verknüpfung zum Tool Provider.

Das Teilen mit anderen Benutzern und Gruppen funktioniert. Das Teilen per Link hingegen erlaubt nur den Download der inhaltslosen Datei, hält aber keine Funktion zum Start des Tool Providers bereit.

Nutzungsmöglichkeiten Mit der Bereitstellung der universellen Tool Consumer Schnittstelle für ownCloud erhält der Benutzer die Zugriffsmöglichkeit auf eine Vielzahl von Tool Providern. Allein das „EduAppsCenter“ [INS16] listet 222 Angebote aus verschiedensten Bereichen auf.

Eine Materialsammlung kann damit, neben den schon in ownCloud zahlreich vorhandenen Apps, eine Vielzahl weiterer externer Anwendungen einbinden und nutzen. Für den Nutzer verschwimmen dabei die Grenzen der einzelnen Systeme, denn er muss sich in der Regel bei den externen Anwendungen nicht erneut einloggen, sondern wird über die LTI-Verbindung authentifiziert.

Studierenden, die in der Regel aus politischen Gründen in den Lernmanagementsystemen keine Autorenrechte erhalten, ist es so auch möglich, selbst Verbindungen zu Tool Providern zu konfigurieren und mit ihren Kommilitonen zu teilen.

Probleme Ein Problem aus technischer Sicht ist die fehlende Anzeigemöglichkeit auf der „per Link geteilten“ Seite. Ein Weiterreichen eines Tool Providers mittels öffentlichem Link scheidet daher aus.

Ein weiteres Problem ist, dass es keine Gruppierung einzelner Ressourcen durch einen gemeinsamen Kontext gibt. Diesen aus dem Verzeichnis,

in dem sich die Datei mit dem Link zum Tool Provider befindet, abzuleiten scheidet aus, denn jeder ownCloud Benutzer sieht die Datei in seinem eigenen Verzeichnis und würde somit den Tool Provider mit einem unterschiedlichen Kontext aufrufen. Das Verzeichnis, in dem die geteilte Datei des ursprünglichen Erstellers liegt, anhand der Datei-ID zu ermitteln würde mangels passendem API-Aufruf in einer Suche durch seinen gesamten Dateibaum ausarten und scheidet daher aus Performanzgründen ebenso aus.

Das dritte Problem ist, dass ownCloud eben kein Lernmanagementsystem ist und mit den eventuell vom Tool Provider zurückgelieferten Ergebnissen nichts anzufangen weiß. ownCloud kennt keine Abhängigkeiten, die eine Bearbeitung oder Sichtbarkeit eines Verzeichnisses oder einer Datei etwa erst nach einem bestandenen Test erlaubt.

5.5 Bewertung

Jede der drei Komponenten bietet für sich genommen schon diverse Nutzungsmöglichkeiten und somit einen sinnvollen Mehrwert für ownCloud und die damit verbundenen Systeme. Durch die Umsetzung der universellen LTI Schnittstelle ist dabei kein Nischenprodukt für eine bestimmte Anwendung entstanden, sondern eine vielseitig einsetzbare Lösung. Da LTI von immer mehr Lernwerkzeugen unterstützt wird, ist auch in Zukunft ein Nutzen gesichert.

Im Rahmen dieser Arbeit wurde das Hauptaugenmerk auf die Umsetzung des LTI 1.0 Standards gesetzt, da dort der meiste Nutzen für den Anwender in Verbindung mit ownCloud gefunden wurde. Mögliche Fortführungen wären die Umsetzung des Ergebnisdienstes nach LTI 1.1 oder eines Tool Consumer Profiles nach LTI 1.2 beziehungsweise LTI 2.0.

Bei der Entwicklung des Prototypen kam es vor allem auf eine schnelle Erreichung der vorgesehenen Funktion an. Dafür wurden oft Einbußen an anderen Stellen hingenommen. Von diesen sollen im Folgenden einige kurz angerissen werden.

Für die Entwicklung von Apps in ownCloud sind Kodierungsrichtlinien vorhanden. Einige davon betreffen die Sicherheit der Anwendung, andere die Lesbarkeit des produzierten Quelltextes. Unabhängig davon, in welche Kategorie die Richtlinien fallen, ist deren Einhaltung im gesamten Quelltext zu überprüfen. Darüber hinaus sind die im Zuge der Entwicklung irrelevant gewordenen Quelltextteile zu entfernen und der verbliebene Rest ordentlich zu dokumentieren.

Zu Beginn der Entwicklung der App lag ownCloud gerade in Version 8.1 vor. Zum Ende der Entwicklung wurde bereits die Vorabversion von ownCloud 9.0 verwendet. Die gesamten Quelltexte sind daher auf inzwischen oder bereits für die Zukunft als veraltet markierte API-Aufrufe zu untersuchen und diese durch ihre neuen Gegenstücke zu ersetzen. Gleichzeitig ist zu eruieren, ob API-Aufrufe aus dem nicht stabilen Teil inzwischen eine gleichwertige Funktion im stabilen Teil erhalten haben.

Ein Prototyp ist vor allem auf eine korrekte Bedienung und ein reibungslos funktionierendes Gesamtsystem ausgelegt. Dementsprechend besteht teilweise Nachholbedarf bei der Fehlerbehandlung. Sowohl bei Systemfehlern als auch bei Eingabefehlern sollte stets eine aufschlussreiche Rückmeldung an den Bediener erfolgen.

Die Grundlagen für mehrsprachige Apps sind in ownCloud bereits vorhanden. Alle Ausgabertexte sind durch ihr englisches Pendant zu ersetzen, eine deutsche Sprachdatei anzulegen und die Textausgaben selbst auf die automatische Übersetzungsfunktion anzupassen.

Unit-Tests sollen sicherstellen, dass bei der Entwicklung von Software die einzelnen Komponenten stets ihre Aufgaben erledigen und auch im Zusammenspiel miteinander funktionieren. ownCloud hält Tests für PHP mittels „PHPUnit“ und JavaScript mittels „Karma“ bereit. Dementsprechend sind Unit-Tests für alle Komponenten der App zu entwickeln und einzusetzen.

6 Fazit und Ausblick

6.1 Fazit

Nachdem in dieser Arbeit zunächst die Grundlagen mit der Beschreibung von LTI und ownCloud gelegt wurden, wurden anschließend in der Konzeption die möglichen Ansatzpunkte für eine Integration von LTI in ownCloud dargestellt. Dabei fanden sich zwei Möglichkeiten, Dienste in Form eines Tool Providers anzubieten. Ebenso wurden verschiedene Ansätze für die Integration eines Tool Consumers aufgezeigt.

In der darauf folgenden Implementierung wurden zunächst der Referenzcode des IMS GLOBAL Learning Consortiums sowie die grundsätzliche Struktur einer ownCloud App vorgestellt. Darauf aufbauend erfolgte die prototypische Umsetzung der zuvor erstellten Konzepte.

Anschließend folgte die beispielhafte Auswahl und Vorstellung von jeweils drei Tool Consumern und Tool Providern. Dabei wurde Wert darauf gelegt, dass sie die Möglichkeiten der LTI Spezifikation auf ihre jeweils eigene Weise nutzen und umsetzen, und somit ein breites Spektrum an Testobjekten abgaben.

Mit allen ausgewählten Testobjekten konnten erfolgreich Verbindungen zu beziehungsweise von ownCloud hergestellt werden und somit die technische Machbarkeit gezeigt werden. Ob die App auch die strengen Tests von IMS GLOBAL - und somit auch eine Zertifizierung - bestehen würde, konnte aus Kostengründen leider nicht überprüft werden.

Auf Basis des funktionstüchtigen Prototypen konnten zwar zahlreiche Nutzungsmöglichkeiten aufgezeigt werden, diese jedoch noch nicht durch einen Praxiseinsatz untermauert werden.

Gleichzeitig zeigten sich Schwachstellen im Konzept und der Implementierung von ownCloud, die sowohl die Nachhaltigkeit als auch die universelle Einsetzbarkeit der entwickelten App beeinträchtigen. Das größte Problem dürften dabei die ständigen Änderungen an der API von ownCloud sein, die teilweise bei jedem Versionsprung Anpassungen an der App notwendig machen.

6.2 Ausblick

Die Weiterentwicklung der LTI Spezifikationen ist noch lange nicht abgeschlossen. Bereits heute gibt es in der Version 2.0 zahlreiche neue Aspekte, die im Rahmen dieser Arbeit keinerlei Beachtung fanden, weil sie noch keine breite Unterstützung auf dem Markt der Lernwerkzeuge erfahren haben. Betrachtet man jedoch die vergangene Entwicklung bei der Umsetzung der Versionen 1.0 und 1.1, so deutet alles darauf hin, dass es nur noch eine Frage der Zeit sein wird, bis auch die Version 2.0 eine massive Präsenz erreichen wird. Eine weitere Untersuchung der Standards von IMS GLOBAL könnte

durchaus neue Erweiterungsmöglichkeiten für ownCloud aufzeigen und somit den Nutzen der App noch weiter steigern.

Die theoretischen Vorteile einer Verbindung von ownCloud mit Lernwerkzeugen mögen zwar offensichtlich sein, doch ob sie auch dem harten Test der Realität standhalten können, muss sich erst noch zeigen. Dies sollte mit einer ausgiebigen Benutzerstudie geschehen, bei der einerseits die Nutzung von ownCloud quantitativ im Vergleich zu anderen Materialablagemöglichkeiten bestimmt und andererseits die Zufriedenheit der Benutzer und das Verbesserungspotential der App anhand eines Fragebogens ermittelt werden kann.

Ein weiteres Problemfeld ist das automatische Anlegen von Benutzern und Gruppen sowie deren Zuordnung zueinander ohne ein funktionierendes Konzept, diese bei einem Wegfall ihrer Existenzgrundlage auch wieder zu löschen. Um diesen Wegfall überhaupt bemerken zu können, muss es für einen Tool Provider eine Möglichkeit geben, beim Tool Consumer jederzeit den aktuellen Status eines Nutzers sowie seiner Gruppenzugehörigkeiten abzufragen. Diese ist in LTI bis Version 1.1.1 jedoch nicht gegeben. Hier könnten die „Learning Information Services“ [Con16c] von IMS GLOBAL ein weiterer Ansatzpunkt zu Forschungen sein.

Ebenso wäre es vorteilhaft, wenn es in ownCloud einheitliche Programmierschnittstellen für die Darstellung von Dateiinhalten sowie der „Teilen“-Box gäbe. Ohne diese Schnittstellen sind viele Apps auf „Bastellösungen“ angewiesen, die leider oft den Sprung zur nächsten ownCloud Version nicht überstehen.

Um die weitere Entwicklung und Nutzung der App sicherzustellen, sollte diese im Appstore von ownCloud anderen Benutzern zur Verfügung gestellt werden. Dazu müsste aber zuerst der Programmcode auf ein akzeptables Maß verbessert werden. Dadurch, dass als experimentell eingestufte Apps kaum Bedingungen erfüllen müssen, um in den Appstore aufgenommen zu werden, kann man so auf einfache Art ein breites Publikum erreichen. Neben anderen Anwendern kann man so sicherlich noch weitere Entwickler für das Projekt gewinnen.

Gerne hätte ich auch die Benutzerstudie und die Weiterentwicklung der App bis zur Appstore-Reife durchgeführt, dies hätte jedoch den zeitlichen sowie inhaltlichen Rahmen dieser Diplomarbeit gesprengt.

Glossar

Dozent Unabhängig von der Bezeichnung im jeweiligen Lernmanagementsystem ist mit Dozent eine Person mit Erstellungs- oder Bearbeitungsrechten an Lernressourcen gemeint. Mögliche Synonyme sind beispielsweise Autor, Tutor, Trainer oder Instrukteur.

Geteiltes Geheimnis Das geteilte Geheimnis ist beiden Seiten der Kommunikation, also sowohl dem Tool Consumer als auch dem Tool Provider, bekannt. Es wird verwendet, um damit eine Prüfsumme über den versendeten Datenblock zu berechnen. Nur in dem gesetzten Fall, dass der Tool Provider die gleiche Summe errechnen kann, hat der Tool Consumer das richtige Geheimnis verwendet.

IMS GLOBAL Learning Consortium Das erklärte Ziel von IMS GLOBAL ist es, Technologien vorwärts zu bringen, die auf erschwingliche Weise die Teilhabe an Lerninhalten und die Erreichung von Lernzielen ermöglichen. Zu den Mitgliedern gehören Bildungseinrichtungen, Softwarehersteller und Regierungsorganisationen.

Kontext Ein Kontext ist die Umgebung, in die ein Tool Provider eingebettet wird. Bei einem Lernmanagementsystem ist dies typischerweise ein Kurs. Der Begriff Kontext wurde absichtlich gewählt, um die Unabhängigkeit von einer Lernplattform zu unterstreichen.

OAuth OAuth ist eine Erweiterung zum klassischen Client-Server Modell, bei der die Authentifikation, Autorisierung und Identifikation vom Server weg auf eine vertrauenswürdige dritte Instanz verlegt wird.

Schlüssel Mit jedem Schlüssel ist genau ein geteiltes Geheimnis verknüpft. Verschiedene Schlüssel können so etwa verschiedenen Lernressourcen, Autoren oder Tool Consumern zugeordnet werden.

Student Ein Student ist ein Nutzer einer von einem Dozenten erstellten Lernressource. Mögliche Synonyme sind beispielsweise Lernender oder Kursteilnehmer. Eine studentische Hilfskraft, die mit der Kursbetreuung beauftragt wurde, wird zu den Dozenten gezählt.

Tool Consumer Ein Tool Consumer nutzt Werkzeuge oder Inhalte, die ihm von einem Tool Provider zur Verfügung gestellt werden. Typische Beispiele für Tool Consumer sind Lernmanagementsysteme wie beispielsweise OpenOLAT, Ilias oder Moodle.

Tool Provider Ein Tool Provider stellt Werkzeuge oder Inhalte für Tool Consumer zur Verfügung. Zu den ersten bedeutenden Anwendungen gehörten E-Book-Bibliotheken.

Tool Provider URL Dies ist der Einsprungspunkt, über den ein Tool Provider von einem Tool Consumer gestartet wird. Üblicherweise ist dieser für alle Aufrufe des Tool Providers gleich. Unterschiedliche Ziele innerhalb des Tool Providers werden in der Regel über unterschiedliche Schlüssel ausgewählt.

Literatur

- [Arc16] The "Fossies" Software Archive. *owncloud 8.2.2*. 2016. URL: <http://fossies.org/dox/owncloud-8.2.2/>.
- [Con12] IMS GLOBAL Learning Consortium. *lti.zip*. 2012. URL: <http://www.imsglobal.org/developers/LTI/test/v1p1/lti.zip>.
- [Con16a] IMS GLOBAL Learning Consortium. *IMS Certified Product Directory*. 2016. URL: <https://www.imsglobal.org/cc/statuschart.cfm>.
- [Con16b] IMS GLOBAL Learning Consortium. *Learning Tools Interoperability® Certification Suite*. 2016. URL: <https://www.imsglobal.org/lti-certification-suite>.
- [Con16c] IMS GLOBAL Learning Consortium. *OneRoster and Learning Information Services*. 2016. URL: <https://www.imsglobal.org/activity/onerosterlis>.
- [dev16] The ownCloud developers. *ownCloud Developer Manual*. 2016. URL: <https://doc.owncloud.org/server/9.0/ownCloudDeveloperManual.pdf>.
- [DPU10] M. Davis, A. Phillips und Y. Umaoka. "*BCP 47 Extension U*", *RFC 6067*. 2010. URL: <https://www.rfc-editor.org/rfc/rfc6067.txt>.
- [Fou16] The Etherpad Foundation. *Etherpad*. 2016. URL: <http://etherpad.org/>.
- [HSF85] K. Harrenstien, M. Stahl und E. Feinler. *DOD INTERNET HOST TABLE SPECIFICATION*. 1985. URL: <https://www.ietf.org/rfc/rfc952.txt>.
- [ILI16] ILIAS. *ILIAS E-Learning*. 2016. URL: <http://www.ilias.de/>.
- [INS16] INSTRUCTURE. *Edu Apps*. 2016. URL: <https://www.eduappcenter.com/>.
- [Lem04] Manuel Lemos. *MDB2 XML Schema documentation*. 2004. URL: http://www.wiltonhotel.com/_ext/pear/docs/MDB2/docs/xml_schema_documentation.html.
- [Lip16] David Lippman. *IMathAS*. 2016. URL: <http://www.imathas.com/>.
- [McF+12] Greg McFall u. a. *IMS Global Learning Tools Interoperability Implementation Guide*. 2012. URL: <http://>.
- [Moo16] Moodle. *Moodle - Open-source learning platform*. 2016. URL: <https://moodle.org>.

- [Neu15] Fred Neumann. *ILIAS E-Learning - External Content (with LTI 1.1 support)*. 2015. URL: http://www.ilias.de/docu/goto_docu_cat_3466.html?lang=de.
- [Ope15] OpenOLAT. *Lernplattform OpenOLAT - LMS OpenOLAT*. 2015. URL: <http://www.openolat.com/>.
- [own16a] ownCloud. *A safe home for all your data*. 2016. URL: <http://owncloud.org>.
- [own16b] ownCloud. *ownCloud Developer Manual*. 2016. URL: https://doc.owncloud.org/server/9.0/developer_manual/app/tutorial.html.
- [own16c] ownCloud. *ownCloud Documentation*. 2016. URL: <http://api.owncloud.org/>.
- [Pat13] Aditya Patawari. *Getting Started with ownCloud*. 2013.
- [SMN10] Charles Severance, Greg McFall und Lance Neumann. *IMS Global Learning Tools Interoperability Basic LTI Implementation Guide Final v1.0*. 2010. URL: <http://www.imsglobal.org/lti/index.html#lti1.0>.
- [Tea16] apps.ownCloud.com Team. *ownCloud Apps*. 2016. URL: <https://apps.owncloud.com/>.