Elearning
Communication
Open-Data

Elearning, Communication and Open-data: Massive Mobile, Ubiquitous and Open Learning

Deliverable Title	D3.1 Validation of technical specifications of ECO modules
Deliverable Lead:	GEO
Related Work package:	WP3
Author(s):	Javier Viñuales (GEO), Stefaan Ternier (OUNL), Jorge Driesner (TLS),Aladdin El-Schallah (HUM), Alessandra Tomasini (POLIMI), Kjeld Loozen (REIMERIT), Vítor Rocio (UAB), Felix Bohuschke (HUM)
Dissemination level:	Public (PU)
Due submission date:	31 March 2014
Actual submission:	31 March 2014

Abstract	The objective of this document is to list the most important technical specifications of each MOOC platform that will be integrated in the overall ECO architecture.
Keywords	Technology, platform, module, functionalities

cip Co-funded by the European Union	*Project funded by ICT Policy Support Programme as part of the Competitiveness and Innovation framework Programme (CIP)*

Disclaimer

This document has been produced in the context of the ECO Project, which has received funding from the European Community's CIP Programme under grant agreement n° 621127.

This document contains material, which is the copyright of certain ECO consortium parties, and may not be reproduced or copied without permission.

In case of ***Public (PU)****:*
All ECO consortium parties have agreed to full publication of this document.

In case of ***Restricted to Programme (PP)****:*
All ECO consortium parties have agreed to make this document available on request to other framework programme participants.

In case of ***Restricted to Group (RE)****:*
All ECO consortium parties have agreed to full publication of this document. However this document is written for being used by <organization / other project / company etc.> as <a contribution to standardization / material for consideration in product development etc.>.

In case of ***Consortium confidential (CO)****:*
The information contained in this document is the proprietary confidential information of the ECO consortium and may not be disclosed except in accordance with the consortium agreement.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the ECO consortium as a whole, nor a certain party of the ECO consortium, warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

The user thereof uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors view.

Versioning and Contribution History

Version	Date	Main modification or revision	Contributions by
0.1	19/03/2014	Initial draft	CSEV, TLS, REIMERIT
1.0	31/03/2014	Final	GEO
1.1	03/04/2014	addition & revision of the chapter LogiAssist	HUM
1.2	31/01/2015	revision of the chapter iMOOC and LogiAssist	HUM, UAB
1.3	09-02-2015	Improved version: Better readability, incorporating changes of the last year	REIMERIT, GEO, UNED, HUM

Main contributors to the document:

Partner	Contributors
GEO	Javier Viñuales
REIMERIT	Kjeld Loozen
TLS	Jorge Driesner
HUM	Felix Bohuschke
OUNL	Stefaan Ternier
POLIMI	Alessandra Tomasini

Reviewers to the document:

Partner
EADTU, EDITRAIN,UNIOVI

Executive summary

The objective of this document is to list the most important technical specifications of each MOOC platform that will be integrated in the overall ECO architecture.

The objective of this particular task is to carry out analysis of all the technical specifications of each of the modules/platforms that are to be integrated in the global ECO platform. Once integration will be done, an Open Data exposure solution will be implemented by means of integrating an Open Data Catalog platform such as CKAN.

This task will generate D3.1 (validation of technical specifications of ECO modules) before the launching of the architectural design and platform integration so that the main technical information and functionalities may be shared with all project partners – this is important so that all HUB´s understand the available capabilities and possibilities open to them.

This deliverable is also a starting point for Task 3.2: Platform architecture. The ECO project is not about creating a new MOOC solution, its goal is to find a way of integration between the existing platforms. Therefore, a thorough understanding of what is currently possible and what is not is of the essence.

Within this task, the following platforms shall be analyzed;

- Web based MOOC solutions
    - OpenMOOC (UNED and CSEV)
    - Open edX (POLIMI)
    - weMOOC (TLS)
    - iMOOC (Universidade Aberta)
    - LogiAssist (Humance)
- Mobile based solutions
    - ARLearn (OUNL)

Table of contents

2.1. ARLearn (OUNL)

2.1.1. Module: Authoring tool

(Single) Sign-on

Technology

Data storage, retrieval and search

API

2.1.2. Module: ARLearn smartphone app

Data storage, retrieval and search

iOS

Android

1. Web based solutions

1.1. OpenMOOC (GEO)

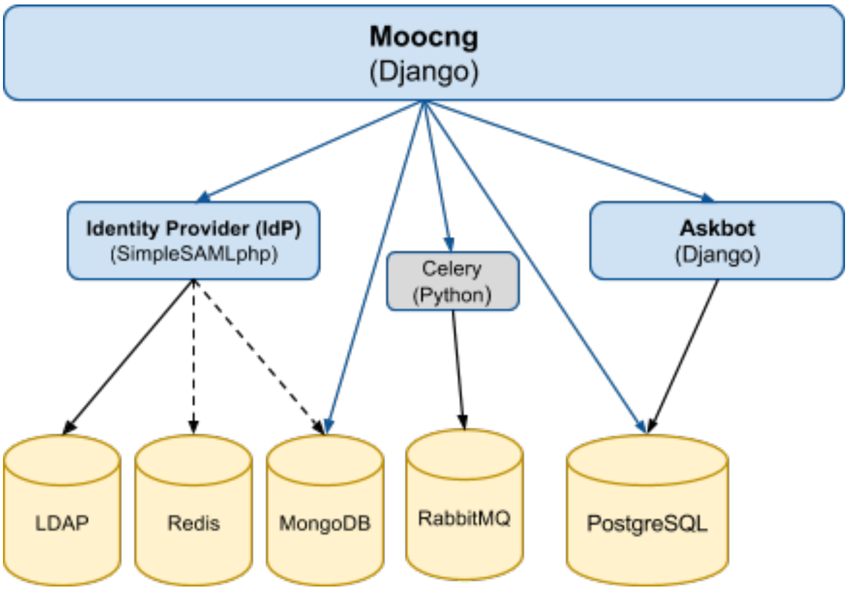
URL: *http://www.openmooc.org* | *Documentation* | *How to collaborate (community)*

Contact: *Users mailing list*

Testing environment available: http://demo.openmooc.org/ *(or demo appliance)*

The OpenMOOC Project aims to develop an open access platform based on free source components to build a connectivist environment. Its highlights are: video-based lessons combined with the use of smart discussion system and progressive learning where information is divided in a set of knowledge pills that the student can learn and consolidate step by step. The collaboration between instructors and students is the most important part and thus there is much emphasis on the usage of technologies and ideas that promote it.

OpenMOOC analyzes the different modules used on Coursera, Udacity, edX, and also other projects like Class2go, Course Builder and learning data trying to get the best out of them and integrate them as different modules into a new platform, completely made on open source code. To solve the issue of identity and access management, the different modules are enhanced to support the SAML protocol. The system integrates four components: An identity provider server (IdP) for SSO and easy integration with existing organizations' Identity Access Management Systems; The MOOC engine itself (Moocng), that uses the latest web technologies to offer a scalable and rich experience, and integrates video classes, homework and exams, reports and so on; A Q/A system (Askbot) to promote debate and discussion about the topics of the courses.



An interesting component to integrate in the future will be a tool that handle with all the data that is stored on the different components. The data mining and the data analytics can help us to improve educational methodologies, learning information for insights regarding student performance and learning approaches. Rather than rely on periodic test performance, instructors can analyze what students know and what techniques are most effective for each pupil. By focusing on data analytics, teachers can study learning in far more nuanced ways. Online tools enable evaluation of a much wider range of student actions, such as how long they devote to readings, where they get electronic resources, and how quickly they master key concepts.

1.1.2. Modulename: Moocng

URL: https://github.com/OpenMOOC/moocng

Code license: Apache 2.0

This is the platform's engine. This module allows teachers to create, manage and release courses and students to apply and follow them.

Functionality

- Video/slide integration with documents and teacher's remarks
- Self assessment progress
- Peer review exercises
- Assets management
- Social discussion forum. Follow up own/others' questions
- WYSIWYG + Latex interface for creation of questions
- Identity federation based on standard (SAML2). Just in time provisioning when accessing the components.
- Medals (badges) for assessing your social behavior in the forum and for success in a course
- Teacher admin panel to create and manage the course. Also has stats and communication features (announcements and mail)
- It supports external billing systems

(Single) Sign-on

Moocng supports not only single sing-on because it supports SAMl 2.0, a version of the SAML standard for exchanging authentication and authorization data between security domains. SAML 2.0 is part of federated access management. Federation lets access management cross organizational boundaries, and helps organizations share identities and services without giving away their identity information, or the services they provide.

Technology

Moocng is built with Django, the powerful python web development framework, used by large Internet very large web cloud solutions companies like Google, Facebook, Pinterest, and so on.

One of the problems in this kind of platform is the scalability. Many students mean many data and much memory and CPU consumption. In OpenMOOC, to mitigate the negative effects of the massification, we have made a series of design decisions: the exponential student data growth is stored in a non-relational database (MongoDB), many actions are performed by Javascript on the browser (modern Javascript tools like jQuery, Backbone.js, Mediaelement.js have been used) which reduces the load on the server, the video hosting is decoupled from the platform (currently it uses Youtube, Vimeo, Prezy or Scripdb but can be easly adapted to use other systems) and the different components can be deployed clustered in high availability.

Data storage

PostgreSQL for courses, and MongoDB and RabbitMQ for student interactions.

API

OpenMOOC has an API that exposes all the courses and its resources. It has been developed following the REST philosophy. It requires the user to be authenticated in the platform, and it offers extra resources for the teachers and other privileged users, for editing content purposes.It has resources available for each level of the data hierarchy, starting with the courses, modules, nuggets, questions and other exercises, etc. It covers a course completely, and it is possible to create the whole UI using only the API, as the OpenMOOC software already does.
It also has special private resources for editing courses. It is possible to modify and create content within a course using only the API. This API requires the user to be an administrator or to be a teacher of the course being edited.

1.1.1. Modulename: Identity provider (IdP)

This component is responsible on the identity of the users. This includes:

- User registration
- User management
- Single Sign On (SSO)

Technology

PHP (SimpleSAMLphp)

Data storage

LDAP

API

Yes, SAML

1.1.3. Modulename: Askbot

This component is a Q&A platform written in Python/Django. At present time it's the main way of communication between teachers and students.

Technology

Python (Django) xz

Data storage

PostgreSQL

API

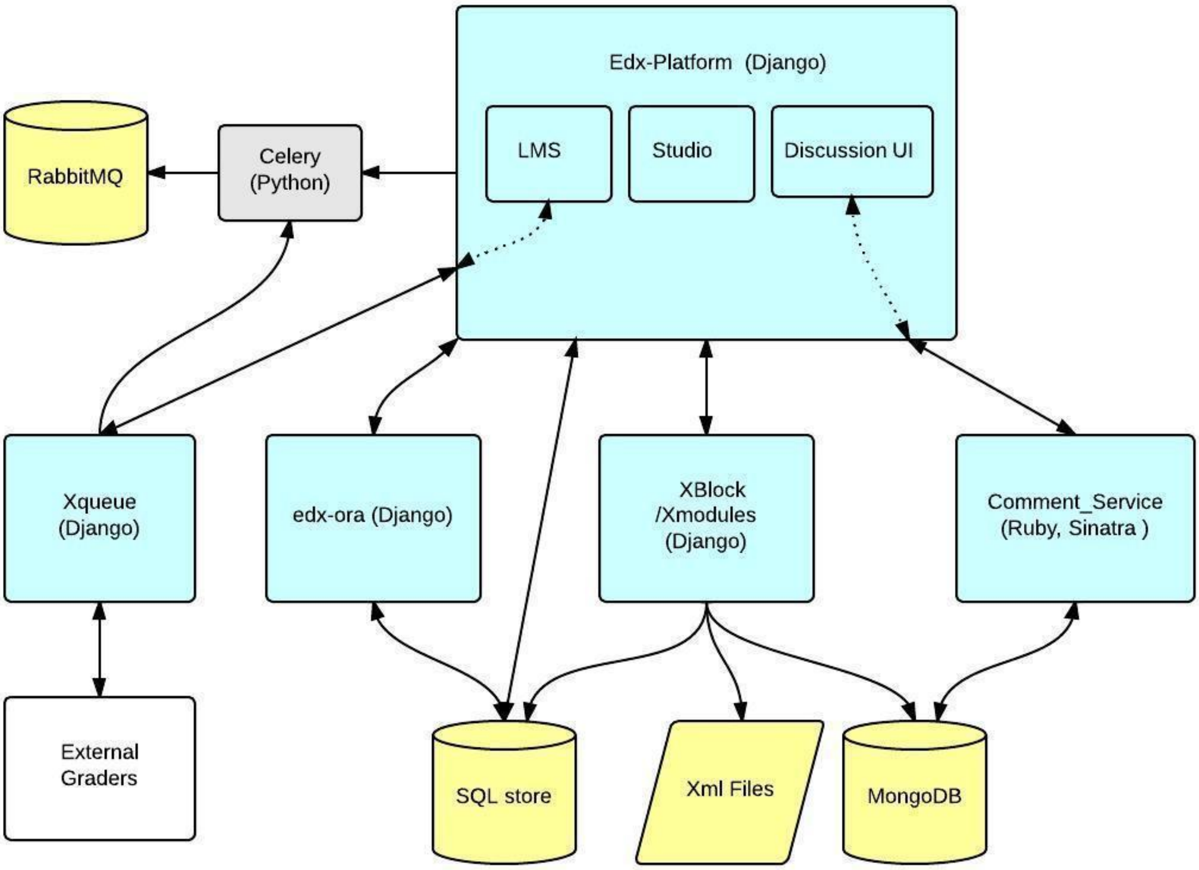
Through moocng's API

1.2. Open edX (POLIMI)

***URL**: http://code.edx.org/*

Contact:

Open edX is the open source Mooc platform (AGPL license) created by edx.org. Open edX combines the use of video, discussion forums, wiki and a grading system based on different supported problem types. The courses consist of chapters built by a set of knowledge units. Each unit can be splitted in many elements like text, video, discussions or problems. Unit can be classified and associated with a grading policy defined by teacher and with different time scheduling and deadlines. Each course has associated an intelligent discussion forum where students and teachers can discuss and collaborate on a unit. Additional material can be added as static pages or uploaded files by teacher or can be made with a collaborative wiki.



1.2.1. Module: EDX Platform

This component includes:

- LMS: platform providing courses access for students. Displays content, runs quizzes and interactive apps (like discussion forums)
- Studio: the authoring tool that allows teachers to create, manage and release courses and students to apply and follow them
- Wiki
- Xmodules : the courseware components that are being upgraded to the new XBlock architecture in the coming months
- Problems : the Studio component that allows teachers to add interactive, automatically graded exercises to course content (checkbox, dropdown, multiple choice, text input, numerical input and advanced problems)
- ***User documentation available for Studio*** *(course authoring) at this link* *http://edx.readthedocs.org/projects/ca/en/latest/* *and* *http://ca.readthedocs.org/en/latest/create_problem_component.html* (for Problems)

(Single) Sign-on

Edx platform use Django auth backend and users are allowed to create an account and login (after account validation). In our installation in Polimi we will enable users to authenticate also with an existing Polimi account provided by a Shibboleth Identity Provider**.**

Technology

EdX has been developed in Python/Django (with some Ruby and NodeJS as well)

Data storage, retrieval and search

As data store edX platform use MongoDB for courseware contents and MySQL for user-data (e.g. user tables, user state, authN, authZ, etc.). For searching edX Platform integrates Elastic Search

API

The public roadmap defines a completely redesign for RESTFul support an give access to course content to enable easier content search, content sharing, and the creation of new authoring tools.

1.2.2. Module: CS Comments Service – server side

CS Comments Service is an independent comment system which supports voting and nested comments. It also supports features including instructor endorsement for education-aimed discussion platforms. A discussion forums can be activated for each course or unit. ***Technical documentation availabe le at these links:***

https://github.com/edx/cs_comments_service/wiki/System-Overview

http://edx.readthedocs.org/projects/devdata/en/latest/internal_data_formats/discussion_data.htm

The edX platform contains all the presentation / UX for edX forums, and is where the user permissions for individual forums are managed. Core models for things like users and courses also live in the edX platform; some lifecycle events (such as creating courses or enrolling students in courses) trigger callbacks to the comments service in order to seed initial forums content.

Technology

The comments service application is an HTTP data layer for edX forums content built with Ruby /Sinatra. It provides the edX platform with RESTful access to resources representing the stuff of forums, e.g. threads, comments, and replies.

Data storage, retrieval and search

The comment service use MongoDB as data storage and Elastic Search for searching.

API

The LMS frontend call this api to visualize discussions forum in the course. The API is in REST protocol

1.2.3. Module: XBlock

XBlock is a new component architecture for building courseware in a pre-alpha release at this moment. On a high level, XBlocks is a Python language-level API, and it provides sensible defaults for things like storing data. XBlocks could be wrapped up in LTI, and one could make an LTI XBlock. Technical documentation available at this link *https://xblock.readthedocs.org/en/latest/*

This module is not fully implemented and currently is not integrated in edx-platform (officially)

Technology

Python/Django

Data storage, retrieval and search

This is a component architecture to provide a better API to resources stored in edX Platform with Mongo DB , MySql and searchable with Elastic Search (see 1.2.1)

API

API protocol: REST

1.2.4. Module: Xqueue

XQueue defines an interface for the LMS to communicate with external checker services. For example, when a student submits a problem in the LMS, it gets sent to the XQueue. The XQueue then has the problem graded by an external service and sends the response back to the LMS. *https://github.com/edx/xqueue*

This is needed for Open Response Assessor (see 1.2.5)

Technology

Python/Django

Data storage, retrieval and search

XQueue use MySQL for data storage and need a RabbitMQ installation as message broker

API

XQueue define a Rest API that external graders use to response back to the LMS after problem grading

1.2.5 Module: ORA (Open Response Assessor)

The Open Response Assessor will take a submission from an XQueue installation, pass it through machine learning grading, peer grading, and staff grading as appropriate, and return a result to LMS. This is to be used with the edx-platform and XQueue. It allows for the assessment of open response problems on the edX platform.

https://github.com/edx/edx-ora

Technology

Python/Django . A correct XQueue installation is needed

Data storage, retrieval and search

XQueue use MySQL for data storage.

API

A new version of ORA will be provided with a full API . At this moment ORA2 is a prototype (https://github.com/edx/edx-ora2)

1.3. WeMOOC (TLS)

***URL**: http://wemooc.com*

***Contact:** jorge.driesner@telefonica.com*

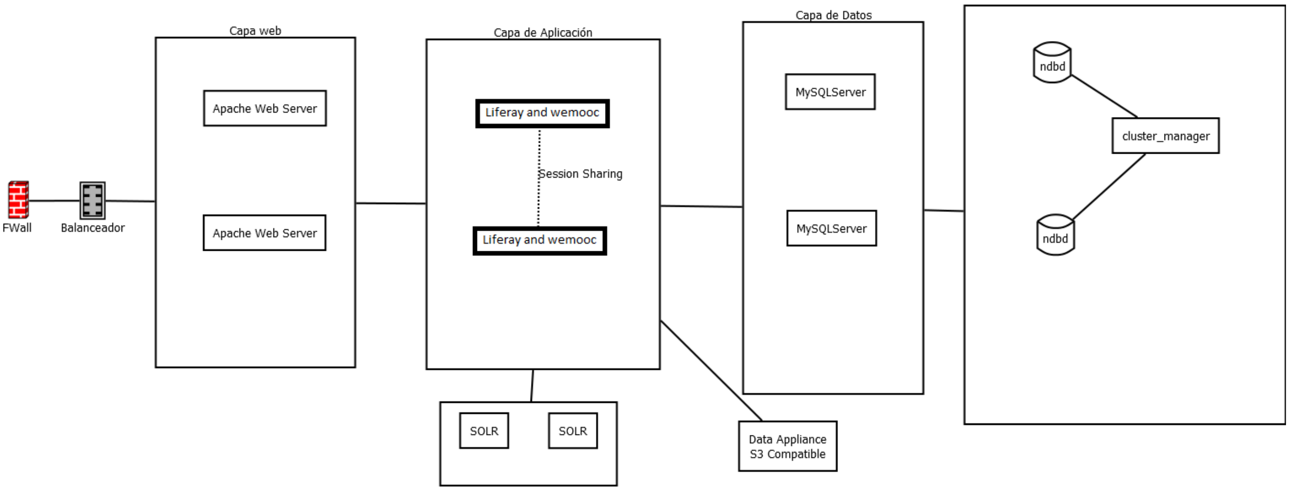
WeMOOC is a virtual environment created by TLS that puts a powerful tool at your disposal for the creation and editing of courses in MOOC format. WeMOOC offers different modules:

- Open and massive self managing training environment.
- Allows the creation of a catalogue of courses organized by subject.
- Permits the creation of many types of activities and is LTI compliant in order to integrate any LTI compliant system as a Learning Activity for students.
- Must be integrated with Liferay Portal, and use its social and collaborative features.

WeMOOC can be combined with other **optional** components developed by TLS like:

- Karma gamification tools.
- Badges and diplomas management.
- Integration with Open Badge Initiative by Mozilla.
- Question and Answers portlets.
- Superior level structure: possibility of grouping content and MOOCs in terms of different levels – for universities, schools or themed areas.

In this diagram we illustrate the high availability infrastructure to deploy a wemooc hub:



Depending on the concurrency estimations, the system must be sized accordingly to avoid any performance issue.

For more information about the Liferay deployment specifications go to http://www.liferay.com/en/products/liferay-portal/tech-specs.

1.3.1. Module: Course Manager

This module allows teachers to create, manage and release courses and offers the following features:

- Creation of a thematic catalogue of courses.
- Add and delete courses in a private environment.
- Publish courses to the catalogue.
- Assign users and profiles to courses (teachers, editors, curators, students)
- Management of the modules of courses uploading and setting up activities dates with previous prepared multimedia material.
- Management of the training plan and learning activities:
    - **Multimedia resources**: html content, video and documents. Download or deliver by streaming.
    - **Quiz/test**: html content, battery of questions, many types of questions (multiple choice, orderable, complete phrases, etc.), establish number of attempts and test pass percentage, general feedback of activity and specific question (correct or incorrect), possibility to improve grade with a set number of attempts. Ability to export results to an xml file.
    - **Peer to peer (P2P)**: html content to contextualize information, set the number of evaluations for students to perform, designed in three stages: submit task, evaluate other colleagues and view received evaluations.
    - **Questionnaire**: html editor for questions and ability to export answers.
    - **SCORM**: player without partial progress following.
    - **LTI compliant activities**: features to integrate many open educational resources available on the internet.
- Management of the enrollment models on courses:
    - Self registration and enrollment: The platform allows a student to register onto the platform in order to enroll in a course of their interest from the list of courses published in the catalogue.
    - Restricted registration and enrollment: Another option is to limit student registration both on the platform as well as during the enrollment period. Here the administrator is responsible to register (on the platform) or validate (enrollment in a course) by approving or rejecting student applications. The platform also includes a system for massive uploading of user data so that students can be associated with the correct course and learning community as outlined by the course catalogue.

Technology

WeMOOC is composed on a set of portlets developed in Java and must be deployed in a Liferay 6.1 GA2 (version 6.1 GA3 or 6.2 is not supported at this point) portal instance in order to run his LMS features and wemooc courses make extensive use of Liferay collaborative tools like blogs, forums, wikis, document library and shared calendar and we rely on the potent, well designed and trustful features of Liferay to manage user management, roles and permissions management, single sign-on features, look and feel, multiple devices support and so on. We recommend that any institution that will install and run his own hub based in wemooc, should also know how Liferay portal works (http://www.liferay.com/es/documentation/liferay-portal/6.1/user-guide)

Data storage

MySQL or any other supported by Liferay.

Retrieval and search

Apache SOLR

API

WeMOOC has an API that exposes all the courses, modules and some type of activities to any authorized client following the REST philosophy, but they can also be exposed as web services relying on the Liferay Service Builder capabilities. Also the students' progress can be read through the API.

Social media enabled: *Share course url*

1.3.2. Module: Monitoring tools

This module allows teachers follow up students' progress with these tools:

- **My Course Stats**: basic reporting module which records student activity throughout each course module and by each of the individual activities within the module.
- **My Gradebook**: summary of students and activities per module and course.
- **Student Manage**: allows the student to track his or her own progress through a simple reporting tool.

Technology

Portlets developed in Java.

Data storage

MySQL

Retrieval and search

Apache SOLR

API

API protocol: REST

1.3.3. Module: Communication tools

This module allows teachers communicate directly to all students enrolled in the course through email messages, based on templates that can be configured by the teachers and editors.

Technology

Portlets developed in Java.

Data storage

MySQL

Sending

SMTP server

No external API available.

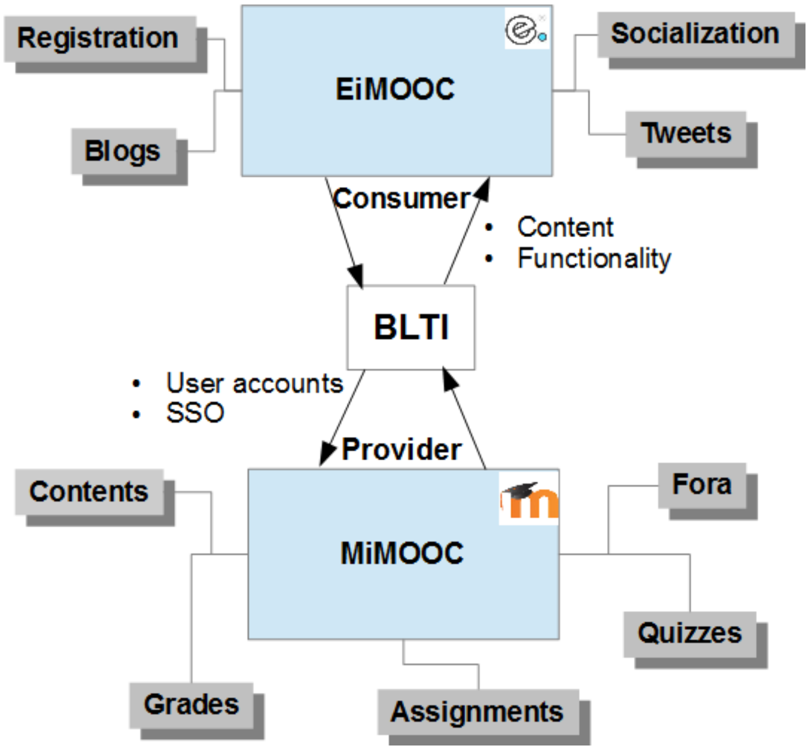
1.4. iMOOC (Universidade Aberta)

URL: *http://imooc.uab.pt*

Contact: *vitor.rocio@uab.pt*

The iMOOC platform is built from open source components, combining the features of two major web platforms, Elgg (http://elgg.org/) and Moodle (https://moodle.org/). Formal and formatted learning aspects (content access, course organization and sequencing, quizzes, assessment) are typically provided by learning management systems, like Moodle. Most elearning and online distance learning initiatives use this kind of platforms, for the development of closed, instructor-controlled courses. A MOOC is, by definition, an open course, and to provide an open environment for students to freely interact and share their opinions, thoughts, contents, web bookmarks, etc., we chose to add a social networking platform, based on the Elgg software, integrated in a global environment, as seamless as possible, with Moodle. The integration is done using the IMS specification for LTI (learning tools interoperability), in the basic version, providing users single sign-on and profile information sharing.

The following figure illustrates the architecture of iMOOC: Elgg functions as a front office (module EiMOOC), where students register and access their courses, and Moodle (module MiMOOC) is the back office where instructors build contents and manage the course. The BLTI connection is configured to treat MiMOOC as a learning tool provider, and EiMOOC as a consumer, so that students can access the platform through the front office and conduct all their activities in the same environment, navigating seamlessly between both systems.



1.4.1. Module: MiMOOC

MiMOOC is the formal learning module (LMS), containing all course contents and activities. Discussion fora, quizzes, assignments and gradebook are all active plugins. User accounts are created automatically upon navigation from EiMOOC.

Technology

Moodle (www.moodle.org)

Data storage

MySQL; PostGres SQL; SQL Server

Retrieval and search

Moodle internal search functionality; Google-indexable

API

Moodle provides a set of APIs for several aspects (https://docs.moodle.org/dev/Core_APIs), some of the most relevant for ECO being:

- Data manipulation
- Logging
- File manipulation
- Calendar
- Enrolment
- Messages
- Media embedding
- Activity completion
- Question database

API functions can be exposed through web services, using one of several protocols (SOAP, XML-RPC, REST).

1.4.2. Module: EiMOOC

EiMOOC is the Informal learning and social network module, providing the landing page. Users register in this module, and access their profile, as well as web 2.0 tools:

- blogs
- micro-blogging
- social bookmarking
- media sharing
- social connections
- groups

Technology

Elgg (http://elgg.org)

Data storage

MySQL

Retrieval and search

The search functionality in Elgg is provided through plugin hooks. The current version uses a core plugin that searches across the whole platform.

API

A web services plugin enables Elgg to expose functionality to the web, through a RESTful interface.

Social media

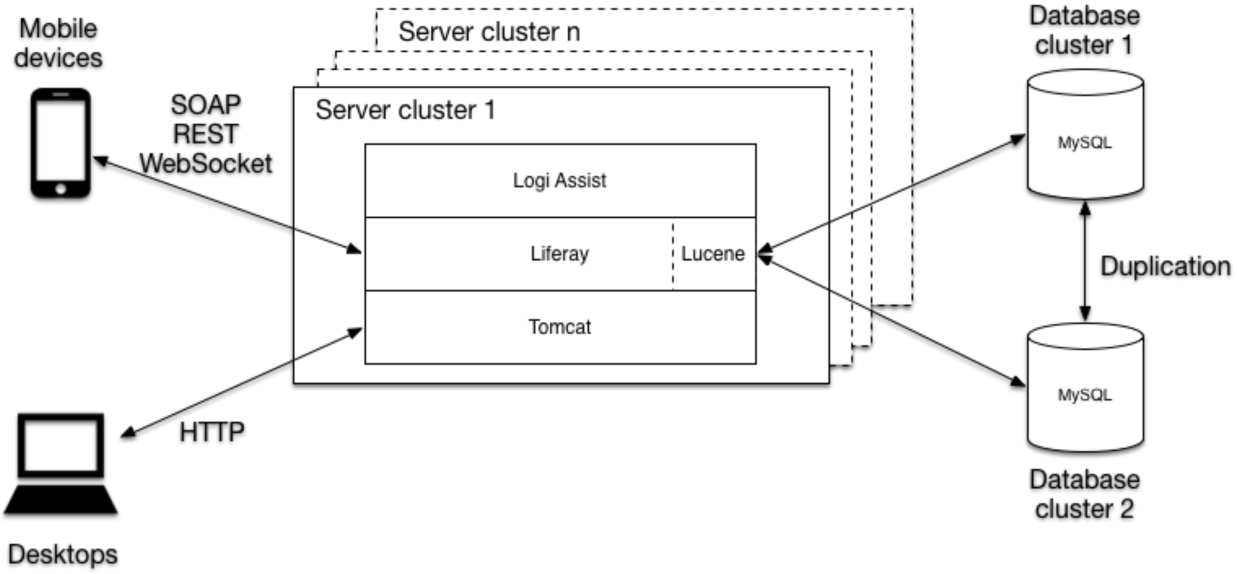
Share course url in Facebook, Twitter, Google+, etc.

1.5. Logi Assist (Humance AG)

URL: *http://eco.humance.de*

Contact: *Felix Bohuschke (fbo@humance.de)*

Logi Assist is a project in the logistic branch focused on truck drivers and hauliers. Logi Assist is accessible via web browser, smartphone and tablet and offers a set of different tools integrated in a core assistance framework (e.g. educational contents, vehicle telematics, tour planning, handling of dangerous goods).



To be able to implement all ECO relevant updates, Humance decided to create a new server instance based on the existing Logi Assist platform. This new instance was styled according to the ECO style guide and supports the requested features according to the ECO project, like e.g. Login with ECO accounts or responsive design.

1.5.1. Module: Education

This module allows teachers to create, manage and publish courses. The Education module provides teachers and learners the following features:

- Creation of courses holding multiple content types like:
  - Folders
  - PDF files
  - Texts following the portal style guide
  - Texts with custom style
  - Audio files with the associated transcript

  - Video files
  - Questionnaires with
    - Single choice questions
    - Multiple choice questions
    - Input questions
  - Surveys with
    - Single choice questions
    - Multiple choice questions
    - Input questions
- Configuration of unlockable course contents
- Course versioning
- Course review process
- Course simulation
- Configuration of visibility of courses (all users or users of a specific group)
- Assignment of users
- Configurable notification system
- Progress tracking
  - Individual course progress
  - Individual progress of each content of a course
  - Individual learner’s behavior tracking
    - Number of time a course is viewed
    - Time spent in a course
    - Number of time a content is viewed
    - Time spent in a content
    - Last unlocked contents
    - Last completed contents
- Visualization of learners progress in
  - Education dashboard
  - Courses overview

Technology

The education module of Logi Assist is a portlet based module developed in Java. All portlets are developed using the Vaadin UI Framework. The module has to be deployed in a Liferay 6.2 GA2 portal instance. Beside the features of the education module, the “out of the box” features of Liferay (like blogs, message board or wiki) are also supported. The education module uses Liferay’s user, role and permission system to manage the educational contents.

Data storage, retrieval and search

The data storage of the education module of Logi Assist is based on a MySQL database. All database indexes and queries are cached and can be searched. Currently Lucene is used as the search engine but can be replaced by any other engine (like Apache SOLR).

API

All features can be made available by using Liferay's Service Builder capabilities (SOAP or REST interfaces). In addition, Logi Assist has a WebSocket interface. With this interface, mobile clients can:

- Sign in
- Retrieve available courses
- Start and continue courses
- Track the signed in user's progress and behaviour

As there isn't any mobile client for the ECO version of Logi Assist, this interface is currently disabled.

2. Mobile based solutions

2.1. ARLearn (OUNL)

URL: *http://ou.nl/arlearn*

Contact: *stefaan.ternier@gmail.com*

ARLearn is a toolkit for organizing mobile context aware courses. With this set of tools, authors can create content and bind it to context (e.g. location, time, an artefact). Furthermore content can be scripted such that based on the actions of users will take in the course, new content will appear or disappear. ARLearn features a mobile smartphone app which enables users to play courses. This app caches content and logic on the device so that courses still continue even when a user loses network connection.

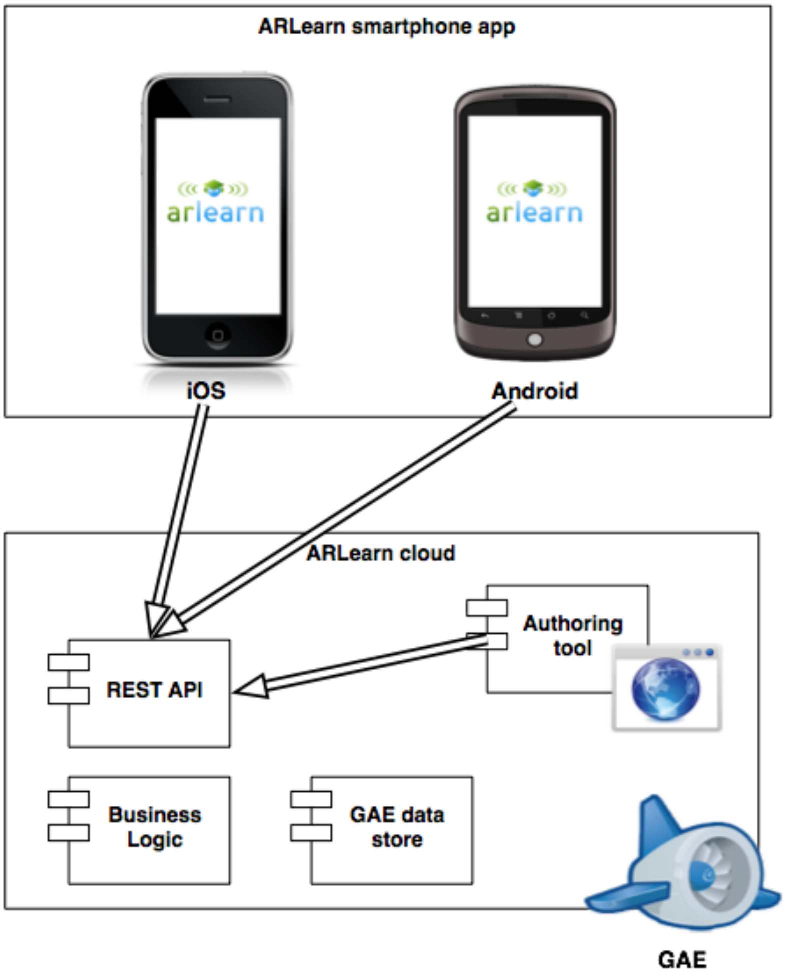


Figure 1: ARLearn smarthone app (top) communicates via a REST API with the ARLearn cloud (bottom)

2.1.1. Module: Authoring tool

URL: *http://streetlearn.appspot.com*
Code license: *GNU Lesser GPL*

The authoring tool is cloud application that enables teachers (or authors in general) to create ARLearn content. This tool enables users to create ARLearn courses including content such as multiple choice questions, audio, video, narrative content and content that requires the user to collect data (e.g. pictures, video, audio, measurements,…). Apart from authoring this tool also features a course library and a way to browse through the results of the course. I.e. an overview of the answers users have given or overview the data they have collected.

(Single) Sign-on

At this moment ARLearn enables users to authenticate with an existing account. Users can login either with a Google, Facebook, LinkedIn or Twitter account. ARlearn builds on the OAUTH protocol to implement authentication against these account providers. The list of account providers can be dynamically extended. In the weSPOT project, a weSPOT account provider was set up and integrated via OAUTH.

Technology

The ARLearn cloud applications have been implemented using JAVA servlet technology. Rather than hosting these on server like Tomcat or Jetty, it was decided to implement ARLearn as a cloud based application Google App Engine (GAE) application. GAE enables building and running applications on Google's infrastructure. This solution was chosen as there are no servers to maintain and more importantly because scalability. With GAE resources like bandwidth, CPU, server instances scale automatically. GAE employs a model that charges on the resources that are used.

Data storage, retrieval and search

ARLearn builds on three storage infrastructures available in GAE

The GAE datastore is a schemaless object datastore. This datastore is used to store all data about games, content, runs, players, etc. The high replication datastore is designed such that it work over multiple datacenters and it can sustain data center outages. App Engine's datastore is built on top of BigTable and differs in this sense from the traditional database management systems. BigTable is comparable to a distributed hashtable that is highly optimized for read operations.

For searching, a search index is maintained that enables Google-like searches over Games and Content. This functionality is used in the ARLearn game library, where users can find existing mobile courses.

The Blobstore allow storages of large data object including audio, video or pictures that are recorded by the ARLearn users.

API

URL: *http://portal.ou.nl/en/web/arlearn/development*

Both the ARLearn smartphone app as the authoring tool make use of a REST based API to communicate with the cloud. This API uses JSON to encode both data that is sent to the server (HTTP POST) and data that is retrieved (HTTP GET). Through this API all ARLearn functionality is available including e.g. creation of content, retrieval of users, registering users, etc.

A user must be authenticated prior to using the ARLearn API. After authentication (OAUTH), the client application receives a token that gives API access. This token must be provided to server with every single request using the HTTP "Authorization" header. With this token, the server can retrieve and authorize the user that issued the request. It is recommended to only use the API over a secure https connection.

2.1.2. Module: ARLearn smartphone app

URL: *https://play.google.com/store/apps/details?id=org.celstec.arlearn2.android*
Code license: *GNU Lesser GPL*

This smartphone app enables users to play courses that were created using the authoring environment. With this app, users can play content such as multiple choice questions, video, audio, scan qr/nfc codes.

(Single) Sign-on

In the same way as the authoring tool, the smartphone app requires the user to log in via the OAUTH protocol. In addition the smartphone app supports a QR login. With this type of login users can play a game in a more or less anonymous way. A teacher can print a batch of e.g. 20 QR tokens, distribute them to the learners and enable them to register to a mobile course via this token.

Technology

- The ARLearn app has been built as a native app because it heavily relies on sensor technologies and background processes.
- The Android version of this app is a JAVA application that builds on the following frameworks
- The Android SDK offers access to standard Android components including NFC scanning, Google Cloud Messaging (GCM), capturing media, GPS positioning, Google Maps, etc.
- GreenRobot implements an eventbus for gluing together activities, fragments and background threads. As mobiles often feature a dual or quad core processor, apps need a means to communicate between these various threads.
- ARLearn uses the ZXing scanner for scanning QR codes. The integration with this open source tool was realized via android intent mechanism, realizing a loosely coupled integration.
- The iOS version of ARLearn is currently under development and uses the following technologies

- The iOS Cocoa and Cocoa Touch framework offer high-level APIs that implement capturing media, GPS positioning, Apple Push Notifications and MapKit.

Data storage, retrieval and search

iOS

Both Android and iOS feature a SQLite relational database to manage data. Core data is a persistence framework that enables data to be manipulated as high level objects representing entities and their relationships. Core data offers an abstraction layer to XML, binary or SQLite store that manages the data. ARLearn builds on the Core Data framework to manage data on the mobile iOS device.

Android

GreenDAO is an open source ORM toolkit that maps Java objects to the underlying MySQL database via an easy to use API. GreenDAO in this sense is a Core Data alternative for Android. Just like Core Data it (behind the curtains) intelligently caches objects that were already loaded from the database, to minimize expensive database operations.

3 Conclusion

WP3 partners worked all together analyzing the capabilities of the modules from the platform providers, emphasizing the integration possibilities. Each platform provider had to test and describe the existing interfaces, as well as all the technology supported by their own platform. As a result of analysis of the features of each platform, architecture, modules, integration capabilities and technology used, it is concluded that we can develop a powerful and scalable architecture based on the integration of existing platforms.