

# D3.2a Final ECO Platform Architecture Design

Citation for published version (APA):

Loozen, K., Viñuales, J., & Ternier, S. (2014). *D3.2a Final ECO Platform Architecture Design*.

## Document status and date:

Published: 30/04/2014

## Document Version:

Peer reviewed version

## Document license:

CC BY

## Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

## General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

<https://www.ou.nl/taverne-agreement>

## Take down policy

If you believe that this document breaches copyright please contact us at:

[pure-support@ou.nl](mailto:pure-support@ou.nl)

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 30 Sep. 2023

Open Universiteit  
[www.ou.nl](http://www.ou.nl)






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

<b>Deliverable Title</b>	D3.2a Final ECO Platform Architecture Design
Deliverable Lead:	GEO
Related Work package:	WP3
Author(s):	Kjeld Loozen (REIMERIT), Javier Viñuales (GEO), Stefaan Ternier (OUNL)
Dissemination level:	Public (PU)
Due submission date:	30 April 2014
Actual submission:	30 April 2014

Abstract	The objective of this document is to define the final architecture of ECO Platform
Keywords	Technology, platform, module, api, web, mobile, data storage, functionalities, integration

 <small>Co-funded by the European Union</small>	<p><i>Project funded by ICT Policy Support Programme as part of the Competitiveness and Innovation framework Programme (CIP)</i></p>
---	--

## 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
v.0.1	16-04-2014	First version	UNED, REIMERIT, OUNL
v.0.2	16-04-2014	Second version integrating modifications and suggestions from partners	UNED, REIMERIT, OUNL
v 1.0	30-04-2014	Final consolidated version submitted to the EC	UNED, REIMERIT, OUNL
V1.1	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
UNED, OUNL, UVA

## Executive summary

This deliverable presents the ECO platform architecture. It reports on the outcomes of the work so far conducted under task T3.2: Overall platform architecture. It specifies all components that will constitute the ECO platform and describes the connection to each other.



## Table of contents

- 1. Introduction
- 2. ECO Architecture Components
- 3. Component details
  - 3.1 ECO IDP
    - Single Sign-On (SSO)
    - User profile
    - Technology
  - 3.2. EcoPortal
    - Features
    - Technology
  - 3.3 ECO Backend
    - ECO Core (business logic)
    - ECO Backend API
    - ECO Registry
    - ECO Data layer
    - Technology
  - 3.4 ECO MOOC Platform
    - ECO Server side API
    - Course metadata harvesting
    - Learning Analytics Services
- 4 Conclusion
- Annex 1: Glossary of Acronyms

## 1. Introduction

In the whole ECO project, the overall technical solution plays a small, but important part. ECO focusses mostly on pedagogical and educational challenges and it is not positioned as a technical development project. Therefore, WP3 cannot create from scratch a new MOOC platform that meets all the demands that may arise after the analysis of WP2.

However, WP3 will strive to create a single access point to all connected MOOC platforms and tries to deliver a mobile experience to the user. However, we realized that it's important not to exclude a large user group that is still using desktop browsers instead of mobile devices to consume MOOC courses. Since the target number of ECO users is high, it is important to make ECO available to everybody: desktop as well as mobile users.

According to this reasoning, this single access point (later to be referred to as EcoPortal) is implemented as a web application, running on both desktop and mobile browsers. At time of writing there is no known feature that specifically needs functionality that is only available on mobile devices (like camera or GPS), which further supports this decision.

We are convinced that the most important feature of the ECO architecture must be its expandability: The possibility for any MOOC platform to integrate with ECO following a rather simple path. The fact that both POLIMI and UAB agreed to connect their MOOC platform (Open edX and iMOOC respectively) to ECO, thus already increasing the number of connected platforms as stated in the DOW, proves that we are going in the right direction.

.

The new architecture will be implemented using modern technologies and standards like REST, OpenID Connect or SAML, Node.js, HTML5 and CSS.

## 2. ECO Architecture Components

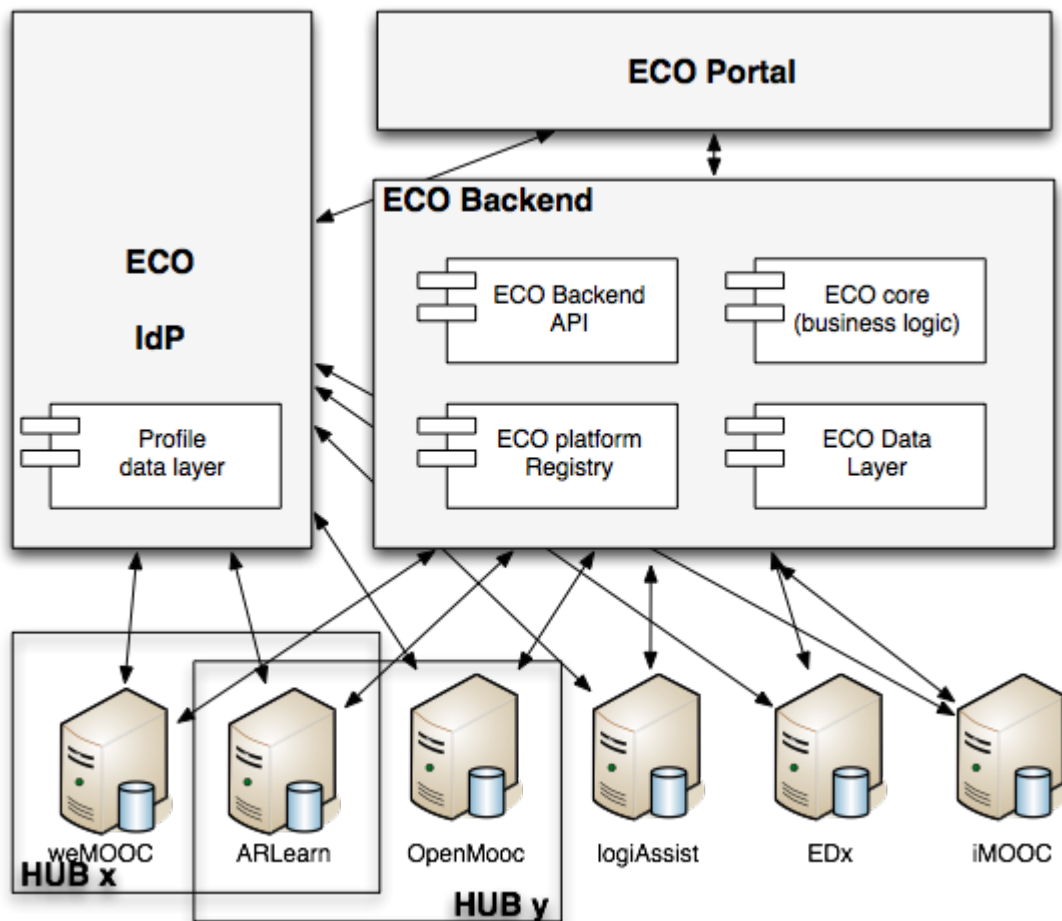


Figure 1: ECO architecture components

Figure 1 lists the ECO architecture and the interrelationships between the various components.

Each of these components are briefly introduced below.

The architecture supports different MOOC platforms connecting to ECO.

Every WP4 hub covers different topics and will pick the MOOC platforms that is best suited to organize their MOOCs. A pilot hub can be supported by more than one MOOC platform. For example, a “traditional” OpenMOOC course can be complemented with a mobile, interactive ‘gaming’ part using ARLearn.

We’ll discuss these components shortly and provide greater detail in the next chapters.



**MOOC Platform**

At the bottom of this architecture, the MOOC platform is the solution to deliver content of a specific course. With the MOOC platform, partners will realize the pedagogical models that are outlined in D2.2. ECO will initially support OpenMOOC, weMOOC, Logi Assist and ARLearn and will in a second phase be complemented by the edX platform and iMOOC, hosted by POLIMI and UAB respectively.

As ECO has a strong focus on mobile and seamless learning, every MOOC platform must be accessible using a mobile device and must provide an interface that is optimized for mobile devices. The WP3 partners responsible for the various MOOC platforms will assure that this is actually the case on every MOOC platform and will implement or enhance mobile support when necessary.

**EcoPortal**

The ECO portal is the main entry point for a user to enter ECO. This application will be developed as a 'responsive web app' meaning it runs in a user's desktop or mobile browser. The ECO portal will adapt to different screen sizes for both desktop and mobile devices. Through this app, users will have uniform access to content stored in ECO and via single sign-on they will be able to navigate from this portal to the MOOC platform hosting the actual content.

**Eco Backend**

The ECO backend groups all 'business logic' of the ECO meta services. This container consists on several components that:

1. Manage course metadata and synchronize (harvest) these with the various MOOC platforms.
2. Organize MOOC platforms via a registry
3. Manage Learning Analytics data

**ECO Identity Provider (ECO IDP)**

The ECO Identity Provider provides the user login mechanism to the ECO Platform. The ECO portal will rely on this component to authenticate ECO users as well will all the MOOC platforms.

### 3. Component details

In this chapter the various components will be further described

#### 3.1 ECO IDP

This component handles user authentication. It contains functionality to let a user identify himself, but it also remembers session state: It remembers that a user has logged in already. It offers this functionality to MOOC platforms as well as the ECO Portal app.

##### Single Sign-On (SSO)

Single Sign-On enables users to login once and have automatically access to multiple applications. Once a user is registered on ECO using the ECO portal, they can use their ECO credentials to login to both the ECO portal and to the ECO MOOC platforms.

The various MOOC platforms will be complemented (if this functionality is not already available) with a login module that connects to the ECO Identity Provider (IDP). Note that it is possible that this IDP co-exists (on a MOOC platform) with other IDP's like Google, Facebook, Twitter, etc.

At the ECO Portal level, users will be able to authenticate with their ECO credentials as well. After login, they will be able to navigate to a MOOC platform without having to re-login. For instance, from within the ECO portal, a user can browse or search courses offered by all HUBs and can enroll directly to a specific course.

When the learner uses the ECO Portal on a mobile device and wants to start an ARLearn the system will start the mobile app and automatically login the user using his ECO credentials. If the app is not installed, the ECO portal app will bring the user to the appstore where they can download the ARLearn app. This behaviour is supported by using the technology of 'custom URL schemes' that enables mobile applications to register an URL as a trigger to open the application.

##### User profile

Once a user registers on ECO he is encouraged to complement his user profile. The fields of this profile can be used to complement course data with anonymous user data, supplying useful information about the division of a course between gender, age-groups, etc. It also gives ECO a way of proposing certain courses to the user.

The ECO profile has the following properties:

- Email address (required)
- Firstname
- Lastname
- Gender
- Birthdate
- Country
- Language
- Areas of interest:
  - Educational science
  - Social sciences
  - Humanities
  - Natural sciences and Mathematics
  - Biomedical sciences
  - Technological sciences

This user information will be shared with a 'relying party' (i.e. a party that uses the ECO IDP as an authentication server, i.e. a MOOC platform).

The first time an user accesses a client application he will be informed about the kind of profile data the client application wants to have access to. The user can give consent or decline this access.

#### Technology

This component will act as the authorization server and will offer this functionality by implementing the OpenID Connect standard (OIDC), which is a simple identity layer on top of the OAuth 2.0 protocol. It allows clients to verify the identity of the end user based on the authentication performed by an authorization server, as well as to obtain basic profile information about the end user in an interoperable and REST-like manner. To put it simple: OIDC is OAuth2 enhanced with JSON Web Tokens (JWT).

## 3.2. EcoPortal

The ECO portal is the frontend web app for the end-user. It gives the user a single point of entry to the several 'ECO-compliant' MOOC platforms. However, it does not provide a universal user experience for all MOOC platforms! The specific functions a specific MOOC platform offers (e.g. handing in an assignment, wikis, watch course video's, etc) will not be replicated to the portal. EcoPortal is implemented as a 'responsive web app', meaning that it runs in a web browser and that it will adapt to different screen sizes, both desktop and mobile.

During the term of WP3, this app will be extended with several features. Although these features are visible in the app, most of the underlying technical work will be done in the backend component of ECO.

### Features

At this time the envisaged features are:

- A course catalog displaying all course offered by the MOOC platforms, providing information about course duration, course content, provided languages, etc.
- Single Sign On (SSO): A user registers on ECO and doesn't have to register again on a connected MOOC platform.
- Integrated ECO Twitter/news feeds: The ECO project delivers twitter (and news?) feeds regarding the ECO project. The content of these streams will also be visible in the web app.
- Integrated ECO events feeds: The ECO project delivers a event calendar regarding events that are ECO -related. The content of these streams will also be visible in the web app.
- Enroll on a specific course: The user can use the web app as an access point to start up a MOOC platform's user interface. Depending on that platform, a web browser is started (OpenMOOC, WeMOOC, edX, iMOOC, Logi Assist) or a mobile app is started (ARLearn).
- Course available overview: On ECO, the user can browse all courses offered by the connected MOOC platforms.
- Accumulated achievements overview: All achievements earned in the connected MOOC platforms are listed on the user's ECO-profile. WP3 will investigate the integration of standards like Mozilla OpenBadges in order to offer a widely supported and recognized standard of recognizing and verifying learning.
- Progress overview: The user's progress on a specific course is listed on his ECO-profile.
- Share your ECO-profile: Show what you're doing and what you have achieved on ECO on popular social platforms (Facebook, Twitter, LinkedIn). The level of exposure to these platforms is manageable by the user.

The order of implementation is outlined in the separate roadmap deliverable 3.2b.

### Technology

The ECO portal will be implemented as a browser-based application using HTML5/JavaScript/CSS and KendoUI/Bootstrap libraries. As a result, it will be supported by the current versions of all modern browsers. This web app will connect to the ECO core business logic via the ECO backend API.

### 3.3 ECO Backend

The ECO backend is server-based implementation that groups all ‘business logic’ of the ECO meta services. The backend will synchronize course metadata with the MOOC platforms and will feed aggregated data to the ECO portal.

#### ECO Core (business logic)

This component will contain the functionality needed for the ECO portal as well as common ECO logic, a.o.: Retrieving the user profile information in order to show it to the user, get the list of courses that a user is subscribed to, synchronize course metadata with the MOOC platforms, account creating functionality (sign up), user profile management functionality, etc.

#### ECO Backend API

This interface is implemented on the ECO backend and allows for communication with the ECO backend. This includes:

1. Functions called by the ECO portal web app. For example, a function that retrieves a user’s profile in order to show it to him.
2. Functions called by a MOOC platform (server-to-server). For example, a function that allows a MOOC platform to inform the ECO backend that a new course has been released.

This interface will be implemented as a REST API. The reason to choose REST as the protocol to implement the ECO API lies in the known characteristics of this protocol, a.o.: Simplicity, human readable and less overhead. And, very important: It’s a widely accepted standard.

The authorized access to the API is implemented in two ways:

- For server to server API calls, the ‘OAuth2 client credentials grant’ mechanism is used. This requires the web app to request an access token first at the ECO IDP and after reception uses this access token as a key to specific API functions.
- For user-related API calls intended to be consumed by the ECO Portal, the OpenID Connect ‘Implicit flow’ is used. In short, this requires the web app to request an access token first at the ECO IDP. But in contrast to the previous non-user related scenario, at this time the user is asked to supply his credentials to prove his identity (once, at login). After success, the Portal app receives an id\_token (which contains secured information about the issuer of this token (e.g. ECO) and the user himself, and also an access token that can be used as a key to unlock specific user related API functions.

Communication with the API will happen over a secure SSL connection.

There is also a large part of the ECO API that has to be implemented on each MOOC platform, thus provide services to EcoBackend. These include a.o. user progress information and course metatadata.

#### ECO Registry

The ECO Registry manages the various MOOC platforms that are connected to ECO. Each MOOC platform entry that is managed by this component captures metadata about the platform.

This registry is furthermore used by the ECO business logic to harvest course metadata. Prior to harvesting,

the business logic will query the list of MOOC platforms. Next for each platform, the business logic layer will fetch the service URLs of the OAI-PMH target and start harvesting the course updates.

The registry will offer similar functionality for Learning Analytics Data.

#### ECO Data layer

The Data layer manages various ECO data relevant at a meta level:

- ECO user profile data information that was harvested from the various MOOC providers. This data is synchronized on a daily basis.
- Learning Analytics data that captures events triggered by users within the various MOOC platforms.

The actual data will be managed by databases that scale well. MongoDB or Hadoop are examples of data stores that enable data to be stored over many server entities.

#### Technology

The proposed technology to implement the different backend components is Node.js. This allows scalability and the possibility to host the backend on a wide choice of PaaS providers as well as to host it rather simple on a self-owned server.

### 3.4 ECO MOOC Platform

The bottom layer of the architecture shown in chapter 2 lists all ECO supported MOOC platforms.

In order for a MOOC platform to become part of ECO, becoming 'ECO compliant', it must comply with the following conditions:

- Allow users to login using their ECO credentials. In other words: Use the ECO IDP for authenticating users.
- Implement the API functions needed by ECO Backend as described in the remainder of this chapter.
- Implement a url-startup mechanism which enables the ECO portal web app to launch the MOOC platform. In case of a browser based MOOC, this will be realized using an OpenID Connect 'id\_token'. In case that a MOOC platform is a mobile app, it must support 'custom url schemes' in order to be able of being started up by a 3rd party.
- Implement an OAI-PMH endpoint enabling ECO but also third party search engines to harvest MOOC metadata using this protocol.
- Incorporate a general 'ECO styling' to the platform in order to unify the user experience

The next section illustrates how the ECO MOOC platforms will offer an API to enable the ECO back-end to fetch course metadata on a regular basis.

#### ECO Server side API

The platforms will offer a REST API to expose functionality to the ECO backend. This API must be implemented on each MOOC platform in a uniform way. It must comply with the requirements that ECO sets, in order for the platform to be 'ECO compliant'. Functionality that must be implemented include a.o. functions regarding user progress and course information.

#### Course metadata harvesting

One of the final tasks of this WP is to provide course Metadata harvesting.

This enables ECO MOOC metadata to be disseminated to third party search engines such as:

- Globe
- Lacro
- Wikiwijs

ECO will build on the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) to extract course metadata from the ECO MOOC providers. OAI-PMH is a protocol developed by the Open Archives Initiative and offers a low-threshold mechanism to exchange metadata between repositories. OAI-PMH is based on a client server architecture and implements a RESTful service. Harvesters (clients) request metadata records from a repository (server) and serialized using XML. A harvester can restrict records by

- date, enabling harvesting only records that were modified in specified data range.
- sets, defined by the repository. This enables harvesting subsets (e.g. all records related to the "physics" topic).

The OAI Protocol has been adopted over the past few years by many digital libraries, institutional repositories and learning object repositories.

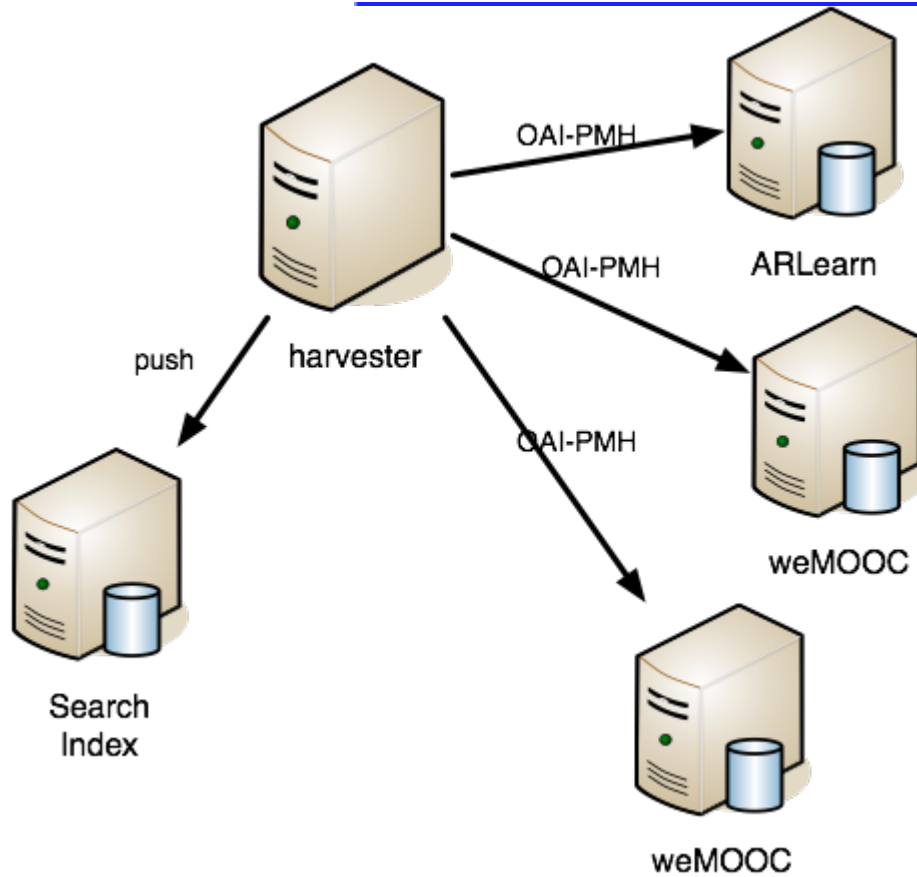


Figure 2: OAI-PMH based architecture for course metadata harvesting

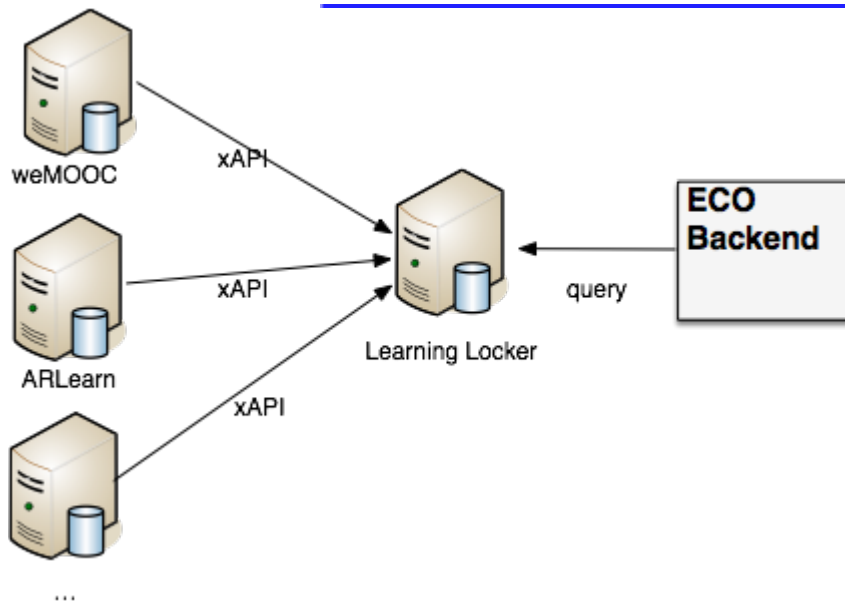
The architecture depicted in figure 2 illustrates how an intermediate repository supports searching a federation of repositories. In this architecture “ARLearn” and “weMOOC” are both repositories that expose their course metadata through an OAI-PMH target.

The harvester will on a daily basis harvest all course record updates from the various providers and update the course index in the ECO data layer.

### Learning Analytics Services

The various MOOC platforms will log user actions. These actions will be made available via an API to the ECO backend. Learning Locker was selected a store for managing Learning Analytics statements. All ECO MOOC providers will submit xAPI statements as users “experience” learning activities to the central “Learning Locker” store





**Figure 3:** Architecture for managing LA data

The Learning Locker store implements functionality that enables reporting and querying via a web based interface. The ECO backend service has web service based access to the Learning Locker store and will be able to dynamically query user profile information. E.g. how many users enrolled to a MOOC, how many pages has a user visited.

Within WP2 (task 2.3) various learning analytics statements have been defined. These statements are currently being implemented by the various MOOC platforms.

## 4 Conclusion

The design of the final ECO architecture has been focused on extensibility. We managed to design an open, lightweight architecture that offers a clear path for any MOOC platform that will connect to ECO now, or will connect in the future.



## **Annex 1: Glossary of Acronyms**

### **API - Application programming interface**

Specifies how software components should interact with each other.

### **REST - Representational state transfer**

*(Source: Wikipedia)*

is a software architectural style consisting of a coordinated set of architectural constraints applied to components, connectors, and data elements, within a distributed hypermedia system. REST ignores the details of component implementation and protocol syntax in order to focus on the roles of components, the constraints upon their interaction with other components, and their interpretation of significant data elements.

### **JSON - JavaScript Object Notation**

*(Source: Wikipedia)*

is an open standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs. It is used primarily to transmit data between a server and web application, as an alternative to XML.

### **JWT - JSON Web Token**

JWT is a simple way to send information in the clear (usually in a URL) whose contents can be verified to be trusted. It works by simply encoding a string made up of a small JSON object and hashing it using a secret shared between the two parties. The algorithm is configurable, but is usually HMAC SHA-256

### **MOOC - Massive open online course**

*(Source: Wikipedia)*

is an online course aimed at unlimited participation and open access via the web. In addition to traditional course materials such as videos, readings, and problem sets, MOOCs provide interactive user forums that help build a community for students, professors, and teaching assistants (TAs). MOOCs are a recent development in distance education.

### **SSL - Secure Sockets Layer**

*(Source: Wikipedia)*

are cryptographic protocols which are designed to provide communication security over the Internet.

### **SSO - Single sign-on**

*(Source: Wikipedia)*

is a property of access control of multiple related, but independent software systems, in case of ECO platforms. With this property a user logs in once and gains access to all systems without being prompted to log in again at each of them.

### **CSS - Cascading Style Sheets**

is a style sheet language used for describing formatting and the look & feel of a document written in a

markup language.

## **HTML5**

is a markup language.

## **OpenID Connect**

*(Source: <http://www.openid.net/connect>)*

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It allows Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.

OpenID Connect allows clients of all types, including Web-based, mobile, and JavaScript clients, to request and receive information about authenticated sessions and end-users. The specification suite is extensible, allowing participants to use optional features such as encryption of identity data, discovery of OpenID Providers, and session management, when it makes sense for them.

## **Node.js**

*(Source: [Wikipedia](#))*

is a software platform for scalable server-side and networking applications. Node.js applications are written in JavaScript.

## **HUB**

Organisational entity.

**IdP - Identity Provider** (a.k.a. Identity Assertion Provider).