Deliverable

Project Acronym:	**LoCloud**
Grant Agreement number:	**325099**
Project Title:	**Local content in a Europeana cloud**

Wikimedia Application

Revision: Version 1.0

Authors:

Dimitris Gavrilis, Costis Dallas and Dimitra-Nefeli Makri [ATHENA]

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	**Public**	**x**
C	**Confidential, only for members of the consortium and the Commission Services**	

Revision History

Revision	Date	Author	Organisation	Description
0.1	20/08/2014	Dimitris Gavrilis	ATHENA	
0.2	25/08/2014	Dimitra-Nefeli Makri	ATHENA	
0.3	01/09/2014	Dimitris Gavrilis	ATHENA	
0.4	07/09/2014	Dimitris Gavrilis, Dimitra-Nefeli Makri	ATHENA	
1.0	09/09/2014	Dimitris Gavrilis	ATHENA	

Statement of originality:

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Contents

Executive summary

This deliverable presents the Wikimedia application which will be used within the LoCloud infrastructure. The application allows the harvesting of content provided by small cultural institutions or independent experts and uploaded to Wikimedia installations, and the provision of enriched content to Europeana.

The application (or microservice) has been built as a web service (REST based) and uses the Wikimedia API in order to communicate with Wikimedia. The application's main functionalities are to harvest content form Wikimedia, parse the harvested content, and identify useful entities that can be mapped to the ESE or EDM metadata schemas.

The application is connected to the LoCloud infrastructure through its REST services. The LoCloud aggregator (MoRe) uses the services in order to allow users to initiate a harvest and get content into the aggregator. The mapped ESE / EDM records are delivered to MoRe, where they can be enriched using the various enrichment services available on the aggregator and then provided to Europeana.

The base URL of the application can be found at:
http://more.locloud.eu/wikimedia/

The API console is accessible from:
http://more.locloud.eu/wikimedia/console.php

1. Introduction

Overview of the microservice

The Wikimedia application has been built as a web service and uses a REST interface in order to communicate with Wikimedia commons and facilitate the exchange of information. On top of the REST services, an API console has been built that demonstrates its functionality. The user is required to provide the URL to the Wikimedia API installation endpoint (e.g. http://commons.wikimedia.org/w/api.php), plus the user identifier that has provided the content to be harvested (e.g. PMRMaeyaert). The application will then use the API in order to retrieve the records associated with this user, parse and extract useful information that is then displayed to the user. This information includes the following elements so far:

- author
- description
- source
- date
- image (url)

The harvested information can then be mapped to ESE or EDM (Europeana Semantic Elements , Europeana Data Model , see references for futher information) in order to be ingested into the LoCloud Metadata Aggregator (MoRe).

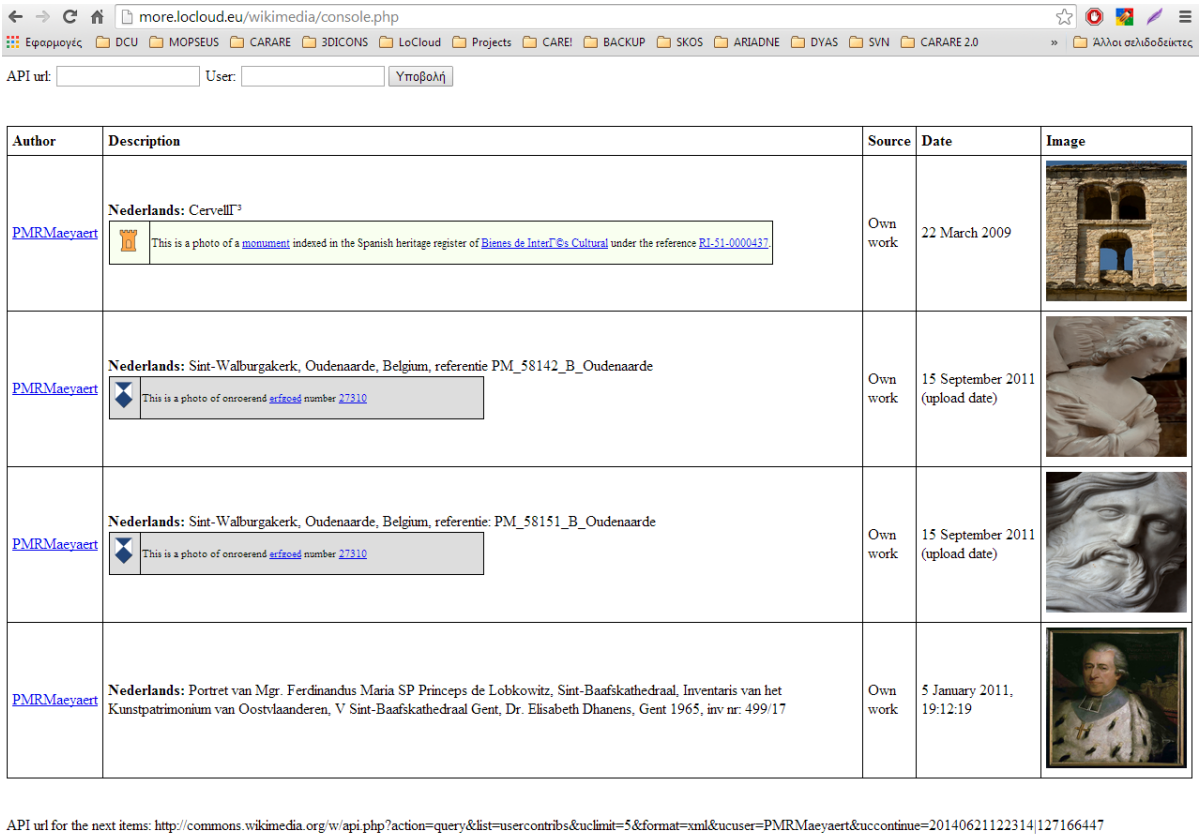


Figure 1: Wikimedia application console. List of harvested records

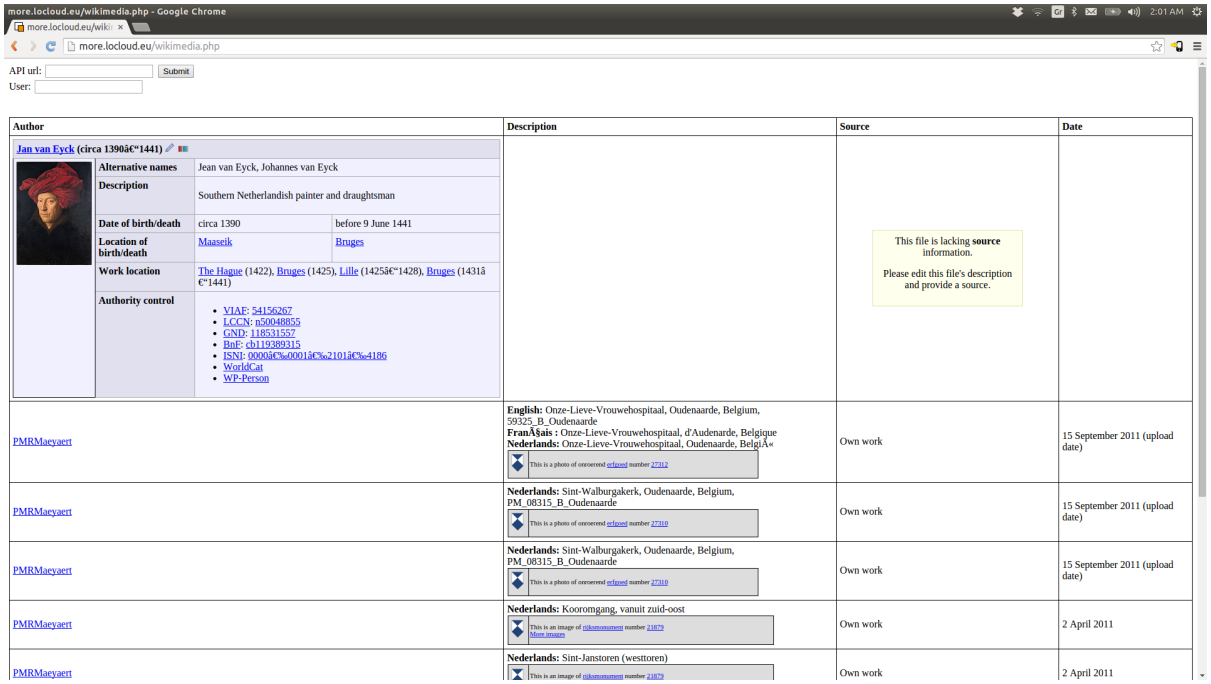


Figure 2: Wikimedia application console. List of harvested records

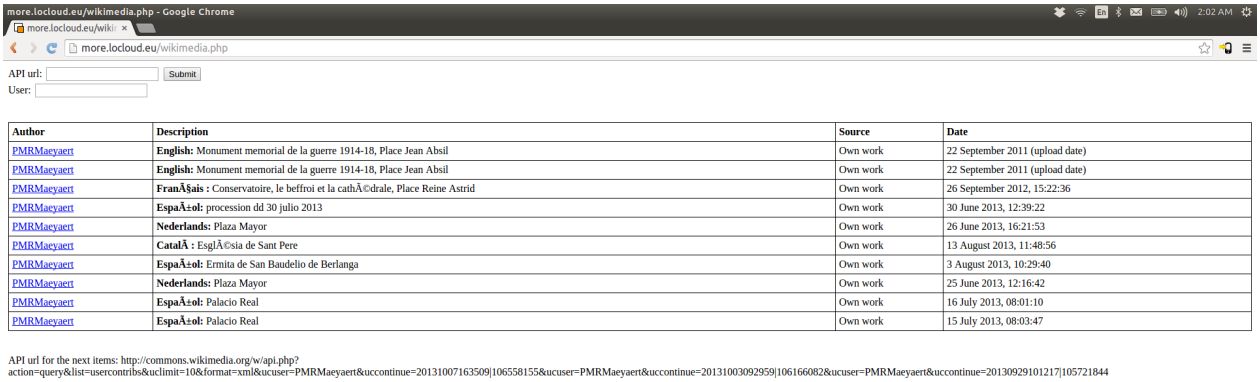
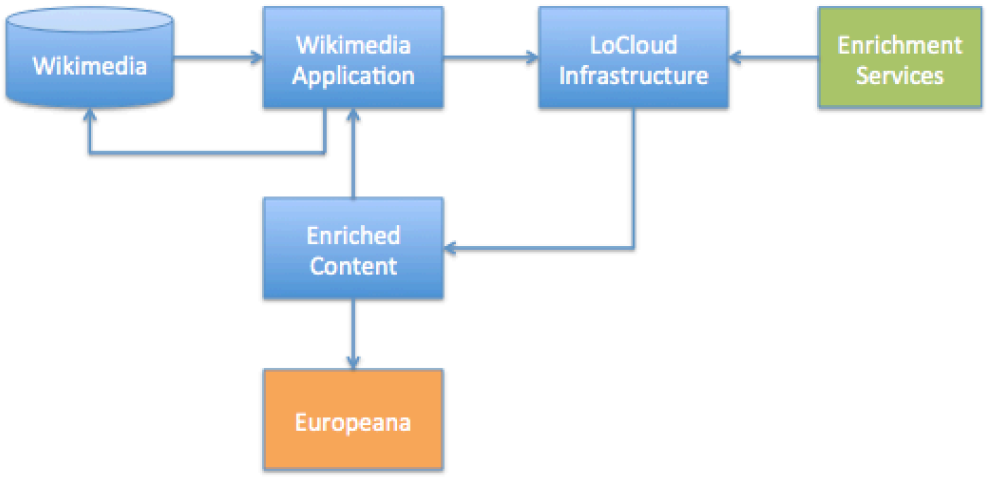


Figure 3: Wikimedia application console. List of harvested records

The harvested and transformed records will be enriched within the LoCloud MoRe aggregator and through the various enriched micro-services. Enrichment through MoRe allows the streamlining of all available enrichment micro-services into robust workflows. Typical enrichment scenarios include:

- Annotation of text using Wikipedia lemmas
- Discovery of associations of records in close proximity
- Annotation with subject terms from various thesauri



The overall workflow is depicted in the above schema where the Wikimedia application harvests content from Wikimedia and ingests it into the LoCloud infrastrure (MoRe). There, the content is enriched through the various enrichment micro-services and the enriched version is delivered to Europeana and back to the Wikimedia application for possible re-ingestion into Wikimedia as a new version.

Overview of the development methodology

The Wikimedia application has been built as as web service and the PHP language was used to develop it. Although it makes extensive use of REST services, the Wikimedia Commons API library has been used. The main challenge in developing the Wikimedia application focuses on extracting and correctly parsing content from Wikimedia, because Wiki content is not always semantically correct and little information can be extracted with a good degree of confidence. The semantic disambiguities have to do mainly with the fact that Wikis are inherently difficult in: a) capturing metadata and b) annotating parts of the wiki text. This means that few metadata per page are available and textual descriptions within the pages of the wiki are not annotated with metadata (they appear as plain text). The testing of the application involves two parts:

Testing of the application

Various API calls were tested using Chrome REST clients. The tests involved using combinations of various parameters.

Testing of the content

Testing using real content was carried out using a dummy Wikimedia installation and using the real Wikimedia Commons. The harvested content was parsed so that the appropriate entities could be extracted.

2. Getting started

Operating system

The Wikimedia application can be installed in any web server that has Apache HTTP and PHP installed. This includes the following operating systems:

- Windows server
- Linux server
- MacOS server

Server

The minimum server requirements of the Wikimedia application match those of the Apache HTTP server. Wikimedia is an extremely lightweight microservice providing it harvests no more than five simultaneous Wikimedia installations.

Terms of use (API key)

There are two access levels in place in order to ensure the proper use of the application, and safeguard the server's resources.

Access to the REST API calls

These require an API key that is provided on request. Each API call must be accompanied by this API key in order to provide proper audit and resource limitation (when needed).

Access to the API console

The access to the API console is free and does not require any keys. It was meant as a demo of the application's capabilities.

Authentication

There is no authentication required (except the use of an API key when using the REST API calls).

Base URL

The base URL of the application can be found under:
http://more.locloud.eu/wikimedia/

The API console is accessible from:
http://more.locloud.eu/wikimedia/console.php

3. API Reference

Harvest

The id of the batch is received or is created.

Request

Method	URL
[***POST or GET***]	http://more.locloud.eu/wikimedia/Harvest.php?api_key=AAAAA&harvest_id=1

Parameter	Datatype	Description
api_key	String	The API Key
url	String	The base URL oft he WIkimedia installation
contributor	String	The contributor name
harvest_id	Integer	The harvest_id of the batch. If empty, a new harvest is created and returned.

Response

Status	Response

200	**An XML document with the id of the batch to harvest is received (harvest_id).**
	
	Example response:
	`<harvest id="1">`
	`<url>`
	`http://commons.wikimedia.org/w/api.php?action=query&list=usercontribs&uclimit=5&format=xml`
	`</url>`
	`<contributor>PMRMaeyaert</contributor>`
	`</harvest>`

ListItems

A list with all the harvested items is received.

Request

Method	URL
[*GET*]	http://more.locloud.eu/wikimedia/ListItems.php?api_key=AAAAA&harvest_id=1

Parameter	Datatype	Description
api_key	String	The API Key
harvest_id	String	The id of the harvest
c_token	String	A token containing the id of the page to harvest. If c_token is empty, the first page is received

Response

Status	Response
200	**An XML document with the harvested items is received. Each item contains a record identifier.** Example response: <items> <item userid="1661583" user="PMRMaeyaert" pageid="28077239" revid="131600008" parentid="103515855" ns="2" title="User: PMRMaeyaert" timestamp="2014-08-14T20:54:06Z" top="" comment="" size="150"/> <item userid="1661583" user="PMRMaeyaert" pageid="16731736" revid="131338987" parentid="62183871" ns="6" title="File:Sant Ponç de Corbera PM 25909.jpg"timestamp="2014-08-12T09:50:25Z" top="" comment="PMRMaeyaert uploaded a new version of File:Sant Ponç de Corbera PM 25909.jpg" size="371"/> <item userid="1661583" user="PMRMaeyaert" pageid="16497745" revid="131281535" parentid="115967226" ns="6" title="File: 27310 Oudenaarde Sint-Walburgakerk 76.jpg" timestamp="2014-08-11T13:30:11Z" top="" comment="/* {{int:filedesc}} */" size="638"/> <item userid="1661583" user="PMRMaeyaert" pageid="16497751" revid="131281244" parentid="115967231" ns="6" title="File: 27310 Oudenaarde Sint-Walburgakerk 77.jpg" timestamp="2014-08-11T13:24:43Z" top="" comment="/* {{int:filedesc}} */" size="639"/> <item userid="1661583" user="PMRMaeyaert" pageid="33507675" revid="127166448" parentid="0" ns="6" title="File:Gent Sint-Baafskathedraal portret bisschop Lobkowitz B STB 578.jpg" timestamp="2014-06-

	21T12:23:15Z" new="" comment="User created page with UploadWizard" size="528"/> </items>

GetItem

The harvested item is received.

Request

Method	URL
[*GET*]	http://more.locloud.eu/wikimedia/GetItem.php?api_key=AAAAA&harvest_id=1&item_id=Gent_Sint-Baafskathedraal_portret_bisschop_Lindanus_B_STB_602.jpg

Parameter	Datatype	Description
api_key	String	The API Key
harvest_id	String	The id of the harvest
Item_id	String	The id of the item to harvest

Response

Status	Response
200	**An XML document with the harvested item is received.** Example response: <response version="0.92"> <file> <name> Gent Sint-Baafskathedraal portret bisschop Lindanus B STB 602.jpg </name> <title> File:Gent_Sint-Baafskathedraal_portret_bisschop_Lindanus_B_STB_602.jpg </title> <urls> <file> http://upload.wikimedia.org/wikipedia/commons/3/30/Gent_Sint-Baafskathedraal_portret_bisschop_Lindanus_B_STB_602.jpg </file> <description> http://commons.wikimedia.org/wiki/File:Gent_Sint-Baafskathedraal_portret_bisschop_Lindanus_B_STB_602.jpg

```
</description>
</urls>
<size>117595</size>
<width>463</width>
<height>600</height>
<uploader>PMRMaeyaert</uploader>
<upload_date>2014-06-21T12:23:14Z</upload_date>
<sha1>3e862647cad40628a0027bd828611cf1ca5f897f</sha1>
<date>
<span style="white-space:nowrap"><time class="dtstart"
datetime="2011-01-05">5 January 2011</time></span>,
19:12:21
</date>
<author>
<a
href="http://commons.wikimedia.org/wiki/User:PMRMaeyaert"
title="User:PMRMaeyaert">PMRMaeyaert</a>
</author>
<source><span class="int-own-work">Own work</span></source>
<permission/>
</file>
<description>
<language code="default">
<div class="description mw-content-ltr nl" dir="ltr"
lang="nl" style=""><span class="language nl"
title=""><b>Nederlands:</b></span> Portret van Mgr. Willem
Lindanus, Sint-Baafskathedraal Inventaris van het
Kunstpatrimonium van Oostvlaanderen, V Sint-Baafskathedraal
Gent, Dr. Elisabeth Dhanens, Gent 1965 inv nr: 499/2</div>
</language>
</description>
<categories>
<category>All media needing categories as of
2014</category>
<category>Media needing categories as of 21 June
2014</category>
<category>Uploaded with UploadWizard</category>
</categories>
<licenses selfmade="1">
<license>
<name>CC-BY-SA-3.0</name>
<full_name>Creative Commons Attribution Share-Alike
V3.0</full_name>
<attach_full_license_text>0</attach_full_license_text>
<attribute_author>1</attribute_author>
<keep_under_same_license>0</keep_under_same_license>
<keep_under_similar_license>1</keep_under_similar_license>
<license_logo_url>
http://upload.wikimedia.org/wikipedia/commons/thumb/7/79/CC
_some_rights_reserved.svg/90px-
CC_some_rights_reserved.svg.png
</license_logo_url>
<license_info_url>http://creativecommons.org/licenses/by-
sa/3.0/</license_info_url>
<license_text_url>
http://creativecommons.org/licenses/by-sa/3.0/legalcode
</license_text_url>
</license>
</licenses>
</response>
```

HTML Status Codes

All status codes are standard HTTP status codes. The ones below are used in this API.

2XX - Success of some kind
4XX - Error occurred in client's part
5XX - Error occurred in server's part

Status Code	Description
200	OK
201	Created
202	Accepted (Request accepted, and queued for execution)
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict
412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

4. How to install the microservice

The application is a web service and does not require any installation, it can be used directly through its REST interface. However, it is possible to install on a new server. The requirements for that installation are:

- Apache HTTP Server (version 2.0 and above)
- PHP Support (version 5.0 and above)
- MySQL (version 5.0 and above)

5. How the microservice is installed in LoCloud

The Wikimedia application can be connected to the LoCloud infrastructure through its REST services. The LoCloud aggregator (MoRe) can use the services in order to allow users to initiate a new harvest and get content into the aggregator, using one of the intermediate formats supported by LoCloud. Once the Wikimedia records have been delivered to MoRe, they can be enriched using the various enrichment services available. These services include the addition of vocabulary terms, the annotation with Wikipedia lemmas etc.

6. Conclusions

In conclusion, the micro-service's general approach includes harvesting of metadata on top of a RESTful architecture. The web based application and REST endpoints have been tested using real-data.

The main challenges in developing and operating such a service have to do mostly with the ambiguities of the harvested metadata. This is because of Wikimedia's lack of formalization (and enforcing of that normalization) of metadata.

Once the Wikimedia records have been harvested using the application and they are on the MoRe aggregator they can take advantage of the full range of enrichment micro-services before being published on Europeana.

7. References

ESE – Europeana Semantic Elements - http://www.europeana.eu/portal/ese_index.html

EDM – Europeana Data Model - http://pro.europeana.eu/edm-documentation

LoCloud, 2013: D2.3 Modified MoRe Prototype

Wikimedia Commons - http://commons.wikimedia.org/wiki/Main_Page

Wikimedia Commons API - http://tools.wmflabs.org/magnus-toolserver/commonsapi.php