Europeana DSI 2– Access to Digital Resources of European Heritage

MILESTONE

MS7.2: INFRASTRUCTURE LAYER

Revision	final
Date of submission	17.08.2017
Author(s)	Krystian Adamski, Tarek Alkhaeir, Marcin Heliński, Aleksandra Nowak, Marcin Werla, Paweł Woźniak (PSNC)
Dissemination Level	Public

Co-financed by the European Union
Connecting Europe Facility

REVISION HISTORY AND STATEMENT OF ORIGINALITY

Revision History

Revision No.	Date	Author	Organisation	Description
1	26.07.2017	Aleksandra Nowak	PSNC	Initial version, based on deliverable from MS7.1
2	26.07.2017	Marcin Werla	PSNC	Review and remarks
3	27.07.2017	Aleksandra Nowak	PSNC	Final version

Statement of originality:

This milestone contains original unpublished work except where clearly indicated otherwise.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.

‘Europeana DSI is co-financed by the European Union's Connecting Europe Facility’

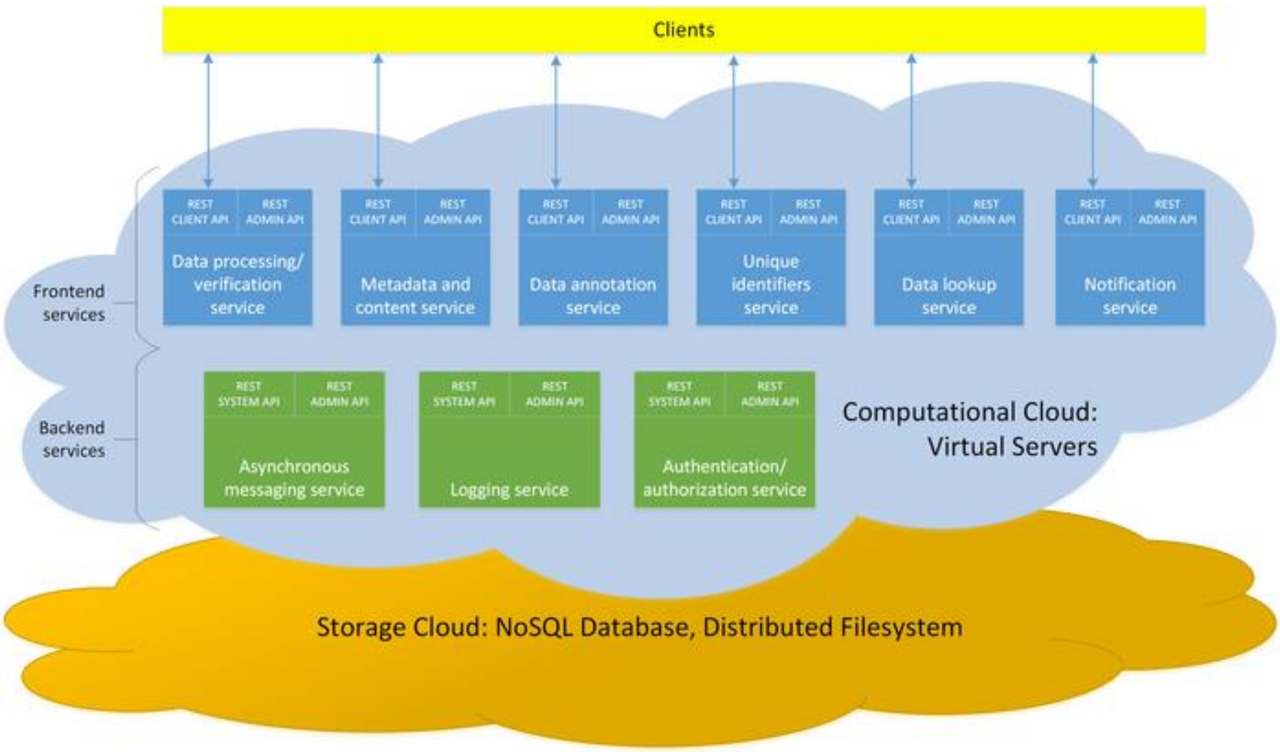
Table of Contents

Introduction

This document describes the current state of the infrastructural part of Europeana Shared Services, initially created during Europeana Cloud project and further developed during Europeana DSI-1 and Europeana DSI-2 projects. Further in this document, current logical and physical architecture of the services is described together with the functionality outline (with emphasis on new developments related to tags and revisions) and with information about monitoring mechanisms, services performance and availability.
This document is based on the MS7.1 deliverable and contains updated information where applicable.

Logical architecture of infrastructure services[1]

The architecture of the Europeana Cloud was designed following the process of gathering requirements which is described in detail in the previous Europeana Cloud deliverable D2.2. Figure 1 from that deliverable, shown here for convenience, shows the main parts of this architecture.



SaaS) system and can be used similarly to other SaaS services available on the Internet today.
Figure 1: Europeana Cloud Service Architecture

To a client, the entire system appears as a Software-as-a-Service.

A client can use several services, each one providing an API for its own functionality. Services are designed to be stateless in order to allow horizontal scalability. They also follow the standard REST service design approach.

---

1 This section is based on the MS7.1 deliverable and contains updated information where applicable

On resources level, the system consists of two types of cloud:

- **Computational Cloud**
  to provide computing capacity for services executed by clients and other services
- **Storage Cloud** to provide storage capacity for services deployed in the Computational Cloud

The **Computational Cloud** hosts all the services which can be used by clients and therefore it is the one which is explicitly contacted by them (frontend services, see below). Also backend services are hosted there (e.g. logging service). The **Storage Cloud** operates behind the scenes on behalf of the clients because all the services use it.

The services in the **Computational Cloud** are arranged in two layers:

- **Frontend Services** which are directly available for the clients of the system. They are also called functional services (blue) because they cater specific functionality used by clients.
- **Backend Services** which are internal and are not available directly for end users, but are used by other services for administrative purposes. They are also called system services (green).

Frontend Services The functionality of the Frontend Services can be divided into several groups of responsibilities:

- **Managing records identifiers** - to provide globally unique identifiers for cultural data records from diverse sources
- **Managing of metadata and content records** - To provide storage and access capabilities for cultural data records, consisting of data and metadata streams in many formats and versions To provide annotation capabilities for cultural data
- **Processing of the data** - to provide flexible, scalable and customizable cultural data processing capabilities
- **Presentation of the data** - to provide easy and reach access to high resolution images stored in Europeana Cloud.

Access to the data

Files uploaded to Europeana Cloud Storage can be accessed using REST API of Metadata and Content Service. Huge part of the cultural heritage data are high resolution images. To fully use their potential and provide better user experience an additional service was developed - Europeana Cloud Image Service. The service provides access to images stored in Europeana Cloud. It is a compatible with International Image Interoperability Framework which provides a standardized method of describing and delivering images over the web. It supports images in JPEG2000 format. Users can display images from the service using compatible viewers on their websites or use an embedded viewer. Both options give high user experience (e.g. scroll to show details, manipulate fragments of the image, compare images or their fragments, etc.)

Processing of the data

Data Processing Service is a part of Europeana Cloud dedicated to provide data processing facility. It consists of a set of build-in plugins providing different ways of data processing. Plugins available right now are useful to perform XML transformations (XSLT) and image transformations. The later plugin might be used to transform images to format compatible with Europeana Cloud Image Service described above.

Main functional improvements of the infrastructure services: revisions and tags support[2,3]

One of the major clients for Europeana Cloud is Metis, a distributed multi-tier services framework for ingestion and delivery of content to Europeana. Prior to commencing the development of Metis, Europeana analyzed the solution proposed by Europeana Cloud with regards to its functional and technical usability and identified various aspects of the system that needed to be improved both in terms of functionality. These changes affect the way that the content is represented within the Europeana Cloud system.

Europeana Cloud data model was designed in the beginning of the original project and can be found on the documentation pages[4]. While the model is descriptive enough for the use of storing and retrieving the content within Europeana Cloud, certain suggestions to improve efficiency have been made by the Metis development team. In short, in its previous iteration, the model exposed all the necessary hierarchies and relations required for the successful management of the stored records. However it imposed an overhead on the client applications that want to utilize Europeana Cloud, as extensive housekeeping operations needed to be performed on the Europeana Cloud clients in order to keep track of modifications in records and datasets, making the integration of the system complicated and reducing its efficiency. To ameliorate this issue stemming from the data modelling selected in Europeana Cloud the following specification was established collaboratively between the Metis team and the Europeana Cloud team. The latter approach removes the overhead of housekeeping data from the client side, making it efficient enough to build a system that can cater for both the needs of ingestion and presentation from third party services.

The new approach introduces a new level in Europeana Cloud record structure hierarchy - revisions and tags:

- A representation version may have one or more revisions
- A revision has an unique id, an owner, a creation timestamp and, optionally, tags
- Versions of records with the same revision have been processed by the same task execution in METIS workflow
- Tags - an additional information stored within revision (PUBLISHED, ACCEPTANCE, DELETED)

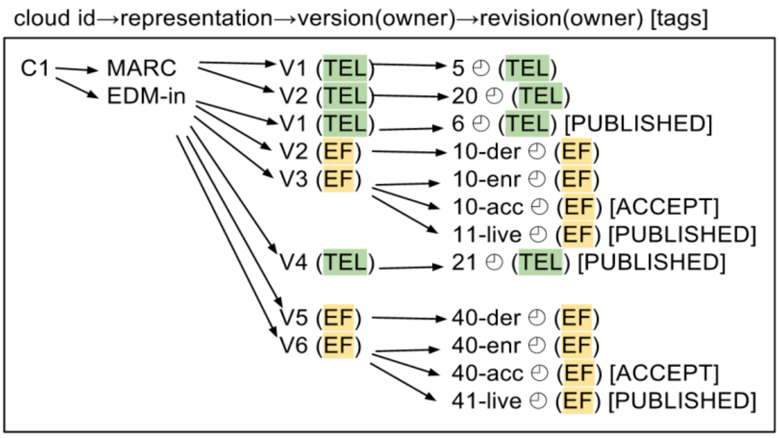


Figure 2: record with two representations: MARC and EDM-in. Both of them have several versions. With every version there is a revision associated

---

2 Developed on the basis of Metis Technical Design Plan

3 This section is based on the MS7.1 deliverable and contains updated information where applicable

4 https://confluence.man.poznan.pl/community/display/ECLOUD/Europeana+Cloud+User+Documentation#EuropeanaCloudUserDocumentation-DataModel

Physical architecture of the infrastructure services[5]

The installation of Europeana Cloud at PSNC consists of several groups of servers acting as an integrated system, called clusters, on which services of the system or 3rd party software are installed. Efficient clustering is critical for building a successful distributed computing system. On the one hand, clusters are able to execute resource-intensive tasks, such as a heavy query or a batch update, quickly because all cluster's machines work on the task altogether. On the other hand, the right balance between clusters should be created. Because resources are not unlimited, systems consisting of several clusters are to be built to avoid bottlenecks between clusters and skewed distribution of resources.

There are two types of machines used for the clusters:

1. Physical machines which provide the best of performance but do not allow for virtualization (designated by rectangles)
2. Virtual machines with networked storage (designated by ellipses)

The infrastructure of Europeana Cloud consists of several independent components that act together to achieve high performance. We followed these principles to achieve a plausible result:

- The storage part of the system, the Storage Cloud, has to be performant as well as reliable. Therefore for it we used physical machines with a lot of processing power and significant number of hard drives.
  - The database cluster can be easily scaled out using the out-of-the-box ability of the system to scale when a new node is added.
  - The storage cluster cannot be scaled as easily because new nodes have to be configured. If scaling will be needed often, a standard scaling procedure will be prepared.
- The messaging system acts as intermediary between the services and will experience a lot of traffic. Thus it is designed an independent cluster built on physical machines. The messaging software has a built-in scaling mechanism so that this cluster can be enhanced when needed.
- The search index cluster consists of two machines, one of which is used for indexing new records and the other for searching. This in order to allow quick search operations while there is a massive indexing operation going on.
- Application services reside on their own cluster for quick identification of a performance bottleneck in one of them. Horizontal scaling mechanism should be designed to allow quick resolution of such bottlenecks, load balancing and failover. For the moment there is no natural clustering mechanism to group several physical services into one logical. It will be developed in the future and rely on Apache Zookeeper. This is on our list.

[5] This section is based on the MS7.1 deliverable and contains updated information where applicable

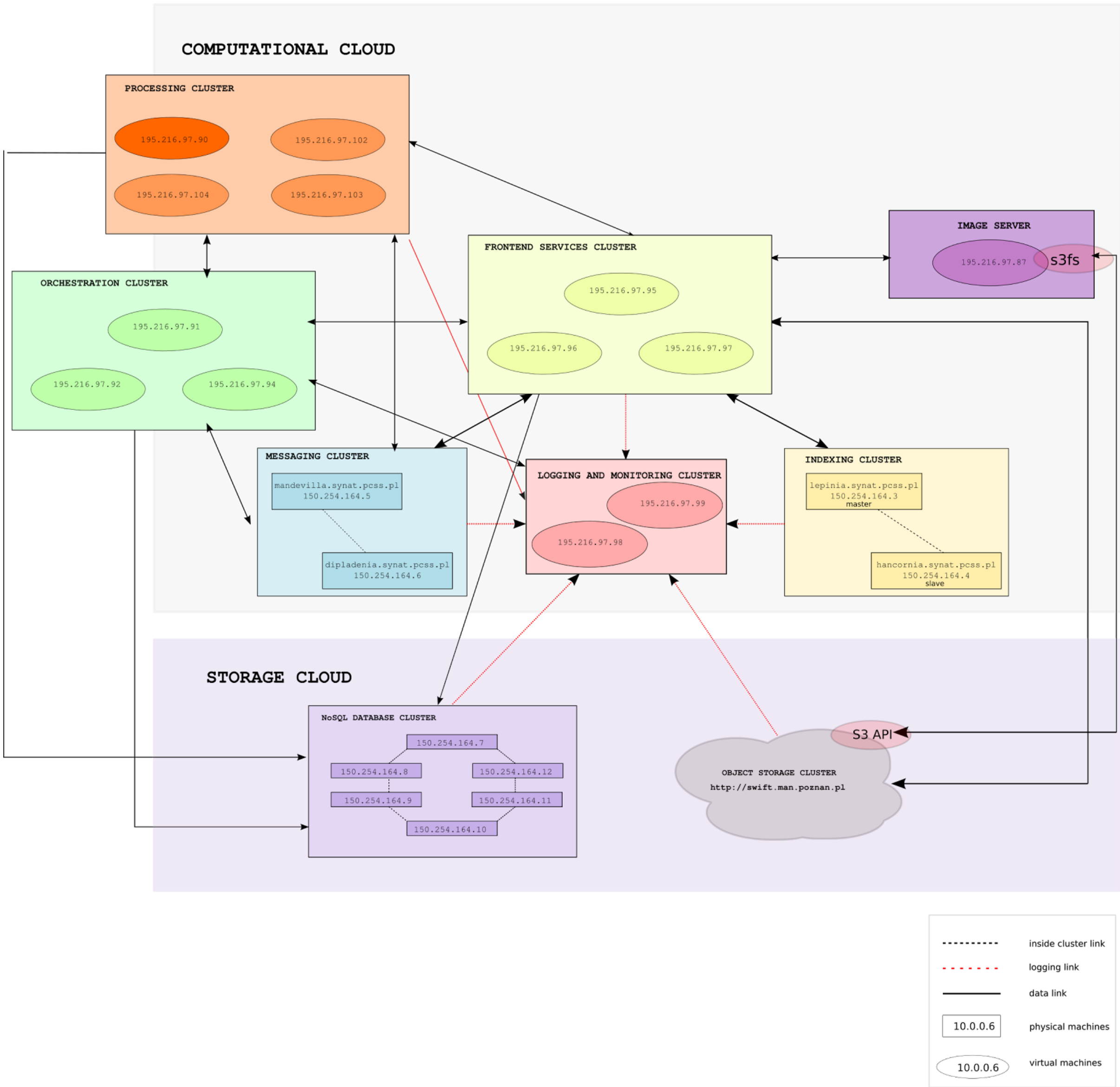


Figure 3: Europeana Cloud Deployment Scheme

Services monitoring, performance and availability[6]

To assure the highest availability of Europeana Cloud infrastructure and services are monitored by:

Ganglia Monitoring System - is a scalable distributed monitoring system for high-performance computing systems. The software is used to view live or recorded statistics covering metrics such as CPU load averages or network utilization for many nodes. One can also define and implement custom statistics. All the machines in the Europeana Cloud cluster are connected to the tool, that's why the team can easily monitor the health of the system and the usage of the resources. Ganglia reports are only available for the maintenance team.

The image below shows is an example report. It shows an aggregated load for all the application machines for the last hour.

[6] This section is based on the MS7.1 deliverable and contains updated information where applicable

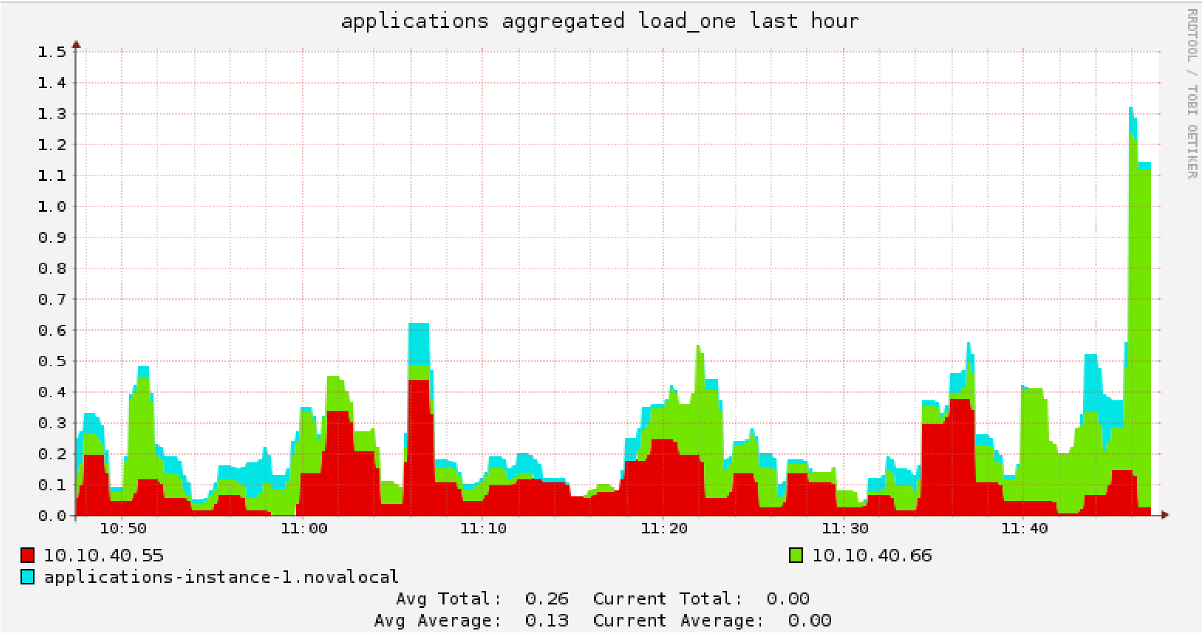


Figure 4. An example of Ganglia statistics - aggregated load for application machines

Nagios - is a tool that periodically runs health checks defined for every machine and service. In case the check failed a notification is send to support team, so they can take verify it and take the action needed to fix the problem. Based on the results of the checks availability statistics can be prepared. Nagios status and notifications are also available for the maintenance team only.
Figure below shows availability report for application machines for the period of last month.

Figures below present availability of the frontend services since the beginning of the year:

Host 'prod_applications_instance_1'

01-01-2017 00:00:00 to 27-07-2017 10:31:12
Duration: 207d 9h 31m 12s

Host State Breakdowns:

State	Type / Reason	Time	% Total Time	% Known Time
UP	Unscheduled	207d 6h 23m 20s	99.937%	99.937%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	207d 6h 23m 20s	99.937%	99.937%
DOWN	Unscheduled	0d 3h 7m 52s	0.063%	0.063%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 3h 7m 52s	0.063%	0.063%
UNREACHABLE	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
Undetermined	Nagios Not Running	0d 0h 0m 0s	0.000%	
	Insufficient Data	0d 0h 0m 0s	0.000%	
	Total	0d 0h 0m 0s	0.000%	
All	Total	207d 9h 31m 12s	100.000%	100.000%

Figure 5: availability report for applications machine 1

Host 'prod_applications_instance_2'

01-01-2017 00:00:00 to 27-07-2017 10:33:56
Duration: 207d 9h 33m 56s

Host State Breakdowns:

State	Type / Reason	Time	% Total Time	% Known Time
UP	Unscheduled	207d 8h 10m 6s	99.972%	99.972%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	207d 8h 10m 6s	99.972%	99.972%
DOWN	Unscheduled	0d 1h 23m 50s	0.028%	0.028%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 1h 23m 50s	0.028%	0.028%
UNREACHABLE	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
Undetermined	Nagios Not Running	0d 0h 0m 0s	0.000%	
	Insufficient Data	0d 0h 0m 0s	0.000%	
	Total	0d 0h 0m 0s	0.000%	
All	Total	207d 9h 33m 56s	100.000%	100.000%

Figure 6: availability report for applications machine 2

Host 'prod_applications_instance_3'

01-01-2017 00:00:00 to 27-07-2017 10:34:01
Duration: 207d 9h 34m 1s

Host State Breakdowns:

State	Type / Reason	Time	% Total Time	% Known Time
UP	Unscheduled	207d 7h 47m 59s	99.964%	99.964%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	207d 7h 47m 59s	99.964%	99.964%
DOWN	Unscheduled	0d 1h 46m 2s	0.036%	0.036%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 1h 46m 2s	0.036%	0.036%
UNREACHABLE	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
Undetermined	Nagios Not Running	0d 0h 0m 0s	0.000%	
	Insufficient Data	0d 0h 0m 0s	0.000%	
	Total	0d 0h 0m 0s	0.000%	
All	Total	207d 9h 34m 1s	100.000%	100.000%

Figure 7: availability report for applications machine 3

Host 'prod_image_server'

01-01-2017 00:00:00 to 27-07-2017 10:34:08
Duration: 207d 9h 34m 8s

Host State Breakdowns:

State	Type / Reason	Time	% Total Time	% Known Time
UP	Unscheduled	207d 7h 47m 2s	99.964%	99.964%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	207d 7h 47m 2s	99.964%	99.964%
DOWN	Unscheduled	0d 1h 47m 6s	0.036%	0.036%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 1h 47m 6s	0.036%	0.036%
UNREACHABLE	Unscheduled	0d 0h 0m 0s	0.000%	0.000%
	Scheduled	0d 0h 0m 0s	0.000%	0.000%
	Total	0d 0h 0m 0s	0.000%	0.000%
Undetermined	Nagios Not Running	0d 0h 0m 0s	0.000%	
	Insufficient Data	0d 0h 0m 0s	0.000%	
	Total	0d 0h 0m 0s	0.000%	
All	Total	207d 9h 34m 8s	100.000%	100.000%

Figure 8: availability report for Image Service machine

Pingdom - can also perform defined health checks and send notifications. Health checks are defined in Europeana Foundation's Pingdom instance, which can be accessed publicly. Right now Pingdom only reports about the health of the Europeana Cloud Image Service. Integration with the rest of services is planned.

europeana

07/27/2017 10:38:34AM (GMT +02:00)
The shown time zone is the same as yours

Europeana Cloud IIIF (history)

Europeana Cloud IIIF | Recent | History

Name ▾	Uptime	Response Time
July 2017	97.21%	409 ms
June 2017	99.97%	394 ms
May 2017	99.78%	400 ms
April 2017	100%	308 ms
March 2017	99.79%	325 ms
February 2017	99.92%	371 ms

Figure 9: Pingdom uptime status for Image Service

Logstash/Banana - is a powerful and configurable set of tools that are able to extract statistics of the services performance using logs reported by the software. It is possible to detect number and type of received requests, number of errors, client based statistics etc. Figure X presents an example board for Europeana Cloud application machines.

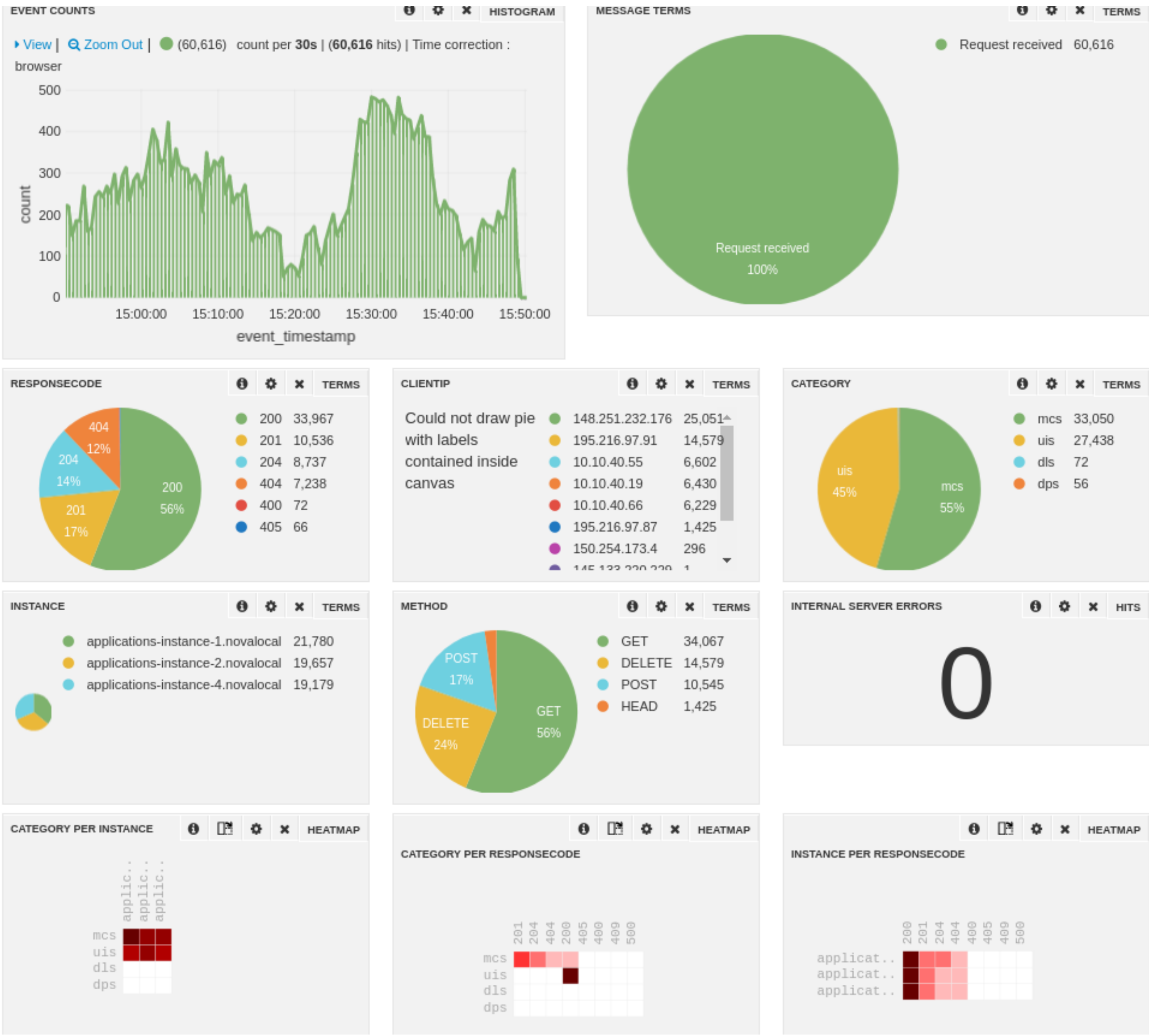


Figure 10: Banana board for Europeana Cloud

Major changes Milestone MS7.1

Since the previous Milestone Report (MS7.1) PSNC team focus was mostly on migration of the production environment and OAI-PMH harvester.

Data modelling

During our work on the deployment of “revision and tags” functionality we discovered some performance issues. Some of the requests in Metadata and Content service were not successful because of database’s timeouts. During execution of those requests database (Apache Cassandra) cluster was not stable - some of the nodes where crashing randomly.
Further investigation led us to realize that there are several problems in modeling of database tables. Apache Cassandra is a NoSQL database and a correct data modeling can heavily affect the performance. Modeling issues in one table can affect the performance of the whole database cluster. This is why we decided to analyze not only the tables with detected problems but all the tables used by Europeana Cloud.

We found out that couples of the tables are following the same problem that can result in bad distribution of data among the nodes. We decided to remodel those tables.
Moreover an external consulting company was hired to help with the diagnosis. The consultant agreed that the aforementioned modeling issues can lead to the described problems and also reviewed the modeling changes that we proposed. During this audit no additional major issues were found.
To proceed with the migration of the production environment these modelling changes should be applied.

OAI-PMH Harvester

The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) is a protocol widely used by cultural heritage institutions. It was developed for harvesting metadata descriptions of records from archives or content repositories.
OAI-PMH is used by the majority of institutions that provide data to Europeana. That’s why it was important for Europeana Cloud system to support this way of uploading metadata records to the system.
OAI-PMH harvester was developed as a new plugin for Data Processing Service (DPS). Because of that the harvester can be run in a configurable, multithreaded environment of Apache Storm as well as utilize the same REST interface as the other DPS plugins. It is important because of the planned integration with METIS.
METIS will expose a user interface where users will specify repositories to harvest and some harvesting schedules. METIS will use OAI Harvester plugin via its REST API to initiate records harvesting. This plugin will write records to Europeana Cloud’s storage and assign appropriate revisions so they can be later used by other steps of METIS workflow.
REST API also allows user to invoke harvesting process directly, without METIS interface.
As the Harvester is a part of Data Processing Service and uses Apache Storm underneath it is possible to configure the parallelism with which the harvest of a source will be executed. That’s why it will be possible to use more processing resources to harvest data from high performant and stable repositories and significantly decrease execution time of this process. In the other case, fewer resources will be used to prevent denial of service of the repository.
At the time of writing this document the implementation of the core functionalities is almost completed. When it is ready some tests will be performed mostly to check what parallelism parameters are the most suitable.

Further development

The team was able to provide a ready to use functionality that will enable users to easily put data from their repositories into Europeana Cloud storage. However, to fully integrate this Harvester with METIS further work is needed. At least 3 more functionalities are planned: tracking progress of harvest process (e.g. percentage of already downloaded records), a possibility to cancel the process and support for repositories with restricted access.