HIGH INTEGRATION OF RESEARCH MONOGRAPHS IN THE EUROPEAN OPEN SCIENCE INFRASTRUCTURE

Deliverable D6.2
Metrics services specification

Grant Agreement number	:	731102
Project acronym	:	HIRMEOS
Project title	:	High Integration of Research Monographs in the European Open Science infrastructure
Funding Scheme	:	EINFRA-22-2016
Project coordinator Organization	:	CLEO-CNRS
E-mail address	:	pierre.mounier@openedition.org
Website	:	http://www.hirmeos.eu
WP and tasks contributing	:	WP6
WP leader	:	Ubiquity Press
Dissemination level	:	PU
Due date	:	30/06/2018
Delivery date	:	17/01/2019

Contents

I.   Overview

Ubiquity Press (UP) has developed an altmetrics service (T6.2) and a citations service (T6.3). The software that collects the altmetrics and citations runs on Google Cloud servers and it's managed by UP.

II.   Altmetrics service

The application was initially written using the Django framework in Python, but after a load test it has been decided to rewrite it using the Flask framework (Python).
The software has been released as open source on GitHub.
The application consists of a core part responsible for the most basic behaviour of the service (keeping the web service running and available); such code is available in the `core` module. Such module is also responsible for most of the user authentication and user-facing pages (see screenshots below).

Every data source available to the application has been developed as a plugin; these are available in a dedicated Python module named `plugins`. Currently, the following plugins have been published:

- `crossref_event_data`: holds the code to get metrics from the CrossRef Event Data API; this module provides the following metrics: Twitter (by DOI), Wikipedia, Hypothes.is
- `facebook`: legacy module holding the code for the Facebook integration; this integration has been discontinued (as agreed in the HIRMEOS meeting in December 2018) and will be soon removed
- `twitter`: holds the code for the integration with Twitter, in order to gather metrics of shares of a monograph by URL (not available through CrossRef Event Data)

The code to run the plugins in asynchronous fashion is run from a different module, called `processor`. Periodically, this module schedules the tasks in each plugin as a RabbitMQ task, which is executed by the workers; the application uses the Celery library (configured in `celery.py` file).

The application exposes the collected metrics using a dedicated REST API interface, defined in module with the same name.

III. Citations service

The citations service has been developed as a plugin for the application, available as `crossref_cited_by`.

IV. Statistics collection agent

Developed by OBP, have been open sourced on GitHub, having one repository for each driver:

- Unglue.it driver (still a work in progress)
- Web server access logs driver
- Open Edition driver
- World Reader driver
- Google Analytics driver

The drivers are coordinated by a broker, available in its own repository. Some accessory backend services have been developed to support the broker and the drivers:

- Identifier Translation Service
- OAI URI importer

V. Reader analytics API

The collected metrics are served by Python-based a REST API, available in its own repository.