D9.4 OPENAIRE LITERATURE BROKER SERVICE

31/12/2015

OpenAIRE2020

Open Access Infrastructure for Research in Europe towards 2020
Deliverable Code: D9.4 – Version (2.0) Final

This deliverable describes the OpenAIRE Literature Broker Service. The Service is designed to offer subscription and notification functionalities for institutional repositories to: (i) learn about publication objects in OpenAIRE that do not appear in their collection but may be pertinent to it, and (ii) learn about extra properties or relationships relative to publication objects in their collection.

H2020-EINFRA-2014-1
Topic: e-Infrastructure for Open Access
Research & Innovation action
Grant Agreement 643410

Document Description

D9.4 – OpenAIRE Literature Broker Service

WP9 – *Front-end services*	
WP participating organizations: **ARC**, CNR, ICM, CERN, UNIBI , UMINHO, JISC, COUPERIN	
Contractual Delivery Date: 12/2016	Actual Delivery Date: 12/2015
Nature: Report	Version: 2.0 (Final)
Public Deliverable	

Preparation Slip

	Name	Organisation	Date
Authors	Paolo Manghi Alessia Bardi Claudio Atzori Michele Artini	CNR CNR CNR CNR	30/12/2015
Edited by	Paolo Manghi	…	
Reviewed by	Sandro La Bruzzo	CNR	
Approved by		…	
For delivery	Mike Chatzopoulos	UoA	

Document Change Record

Issue	Item	Reason for Change	Author	Organization
	Draft version	First draft of the deliverable	Paolo Manghi	CNR
	Final	Updates out of reviews	Paolo Manghi	CNR

Table of Contents

Disclaimer

This document contains description of the OpenAIRE2020 project findings, work and products. Certain parts of it might be under partner Intellectual Property Right (IPR) rules so, prior to using its content please contact the consortium head for approval.

In case you believe that this document harms in any way IPR held by you as a person or as a representative of an entity, please do notify us immediately.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of the OpenAIRE2020 consortium and can in no way be taken to reflect the views of the European Union.

The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 28 Member States of the Union. It is based on the European Communities and the member states cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors. (http://europa.eu.int/)

OpenAIRE2020is a project funded by the European Union (Grant Agreement No 643410).

Acronyms

OLBS	OpenAIRE Literature Broker Service
WP	Work Package

Publishable Summary

This deliverable relates to the work carried out under task T9.5, "Open Access Publication Broker Service". The task's focus is on the design of the OpenAIRE Literature Broker Service (OLBS), presenting its underlying ideas, requirements, and architectural specification. The OLBS is designed to offer subscription and notification functionalities for institutional repositories to: (i) learn about publication objects in OpenAIRE that do not appear in their collection but may be pertinent to it, and (ii) learn about extra properties or relationships relative to publication objects in their collection. Due to the high variability of the information space the following problems may arise: (i) subscriptions may vary over time to adapt to information space evolution, (ii) repository managers need to be able to quickly test their configurations before activating them, (iii) notifications may be redundant, and (iv) notifications may be very large over time. This deliverable present the data model and software architecture of the OLBS, specifically designed to address these issues.

1 | INTRODUCTION

The OpenAIRE technological infrastructure provides services that populate the so-called OpenAIRE Information Space, a graph-like information space aggregating information about publications, datasets, organizations, persons, projects and several funders (e.g. European Commission, Wellcome Trust, Fundação para a Ciência e a Tecnologia, Australian Research Council) collected from hundreds of online data sources (e.g. publication repositories, dataset repositories, CRIS systems, journals, publishers). Data sources providing content to OpenAIRE and interested in augmenting their local collections may benefit in a number of ways from the OpenAIRE information space. This is particularly true for institutional repositories, whose mission is that of growing a complete collection of the scientific publications produced by the authors affiliated to the institution they serve. Repository managers' goal is twofold: bringing in the collection all articles produced by such authors and making sure the metadata is as complete and up-to-date as possible.

To this aim, OpenAIRE infrastructure will be equipped with the OpenAIRE Literature Broker Service (OLBS) whose general principles and ideas are described in this deliverable. The OLBS implements a subscription and notification mechanism supporting repository managers at enhancing the content of their repository taking advantage of the OpenAIRE information space. A number of initiatives started working on "brokering" approaches favoring single-deposition of publication metadata with subsequent automated delivery to other repositories. Some focused on techniques for automatic deposition into a repository (SWORD project), while others focused on the complementary aspects of how to broker publication information from publishers to relevant/interested repositories. SHARE-US and JISC/EDINA are two of such initiatives, based respectively in the U.S. and U.K. The OpenAIRE Literature Broker Service (OLBS) offers to repository managers the possibility to subscribe to special "addition" or "enrichment" events, in order to be respectively notified about: *(i)* publication objects in OpenAIRE that do not appear in their collection but may be pertinent to it, or *(ii)* properties or relationships relative to publication objects in their collection that do not appear in their local metadata. Due to the high variability of the OpenAIRE information space, where new data sources are continuously added or removed, harmonization rules, mining, and deduplication algorithms are refined, following problems may arise: *(i)* subscriptions may vary over time to adapt to information space evolution, *(ii)* repository managers need to be able to quickly test their configurations before activating them, *(iii)* the same notifications may be sent more than once, and *(iv)* notifications may be very large in number over time. This deliverable presents the data model and software architecture of the OLBS, specifically designed to address these issues.

2| REPOSITORY LITERATURE BROKERS IN THE LITERATURE

The literature deluge makes reporting and tracking of research results harder for all stakeholders in the scholarly communication. Researchers often feel to lose precious time when asked to provide detailed metadata information about their articles multiple times at different locations, e.g. institutional repository and funders. As a consequence publication metadata can be poor, subject to mistakes and found at different locations. Publishers own publication metadata information, but a direct interaction with repositories is to be considered an exception, for both technical (e.g. lack of shared author identifiers) and cost reasons. As a consequence, a number of initiatives started working on approaches favoring single-deposition of publication metadata with subsequent automated delivery to other repositories. Some approaches focused on techniques for automatic deposition into a repository (SWORD project), while others focused on the complementary aspects of how to broker publication information from publishers to relevant/interested repositories. SHARE and JISC/EDINA are two of such initiatives, based respectively in the U.S. and U.K.

SHARE (SHared Access Research Ecosystem) is a higher education and research community established in 2013 that supports preservation, access and re-use of research results across United States. The first project set up by SHARE is SHARE Notify (http://www.share-research.org/projects/share-notify/). A public beta version of the service is available since April 2015 and counts about 600,000 metadata records about articles and datasets from more than 30 providers. SHARE Notify allows interested stakeholders of the scholarly communication (e.g. researchers, repositories, funders) to subscribe for notifications about research release events such as the publication of an article in a peer-reviewed journal, the deposition of a pre-print version in an institutional repository or the deposition of a dataset. Notifications are distributed as Atom feeds, consumable via common RSS readers, containing metadata summaries about the research results matching the subscription query. The subscription query may include any metadata field of the SHARE schema, also in combination with boolean operators according to the Lucene syntax. While it is possible to subscribe for events related to one or more data sources (journals, repositories, etc.), it is not yet possible to subscribe for events related to authors' institutions, as the majority of metadata records collected by SHARE does not contain explicit authors' affiliations. A JSON API is also available to build dedicated applications consuming the content collected by SHARE.

JISC is a UK initiative that promotes ICT in education and research. Built in collaboration with EDINA, the prototype of the JISC Publications Router (http://broker.edina.ac.uk/) offers a notification system (PostCards) and an automatic mechanism based on the SWORD protocol to transfer metadata and files from one location to another. Upon subscriptions, users can select one or more repositories of interest. The Postcards system will send them emails with a list of metadata records suitable for the selected repositories. The "suitability" of a record with respect to a given repository is automatically calculated by extracting the authors' affiliations from the metadata records. Though the subscriptions criteria are static, the Postcard system of the JISC Publications Router is very flexible in terms of the format of the notifications: citations in ASCII, bibtex, endnote, Dublin Core metadata records are only a subset of the formats that a subscriber can choose to receive. Currently the service is undergoing a full revision to improve its quality and make it a production system in 2016.

The JISC/EDINA Publications Router is more mature than SHARE Notify and its capabilities of detecting authors' affiliations and of sending notifications in different formats are valuable. Metadata records collected by the router can be bulk downloaded via the standard OAI-PMH protocol. On the other hand, the "young" SHARE Notify gives to users more control on their subscription topics and the availability of JSON API allows IT-skilled users to build applications on top of the SHARE content.

3 | OPENAIRE INFORMATION SPACE

The OpenAIRE Information Space, whose data model is shown in *Figure 1* and described in Deliverable 8.1, builds on the OpenAIRE guidelines (http://guidelines.openaire.eu) and is inspired by the DataCite and CERIF initiatives. Its main entities are: *results* (datasets and publications), *persons*, *organizations*, *funders*, *funding streams*, *projects*, and *data sources*.

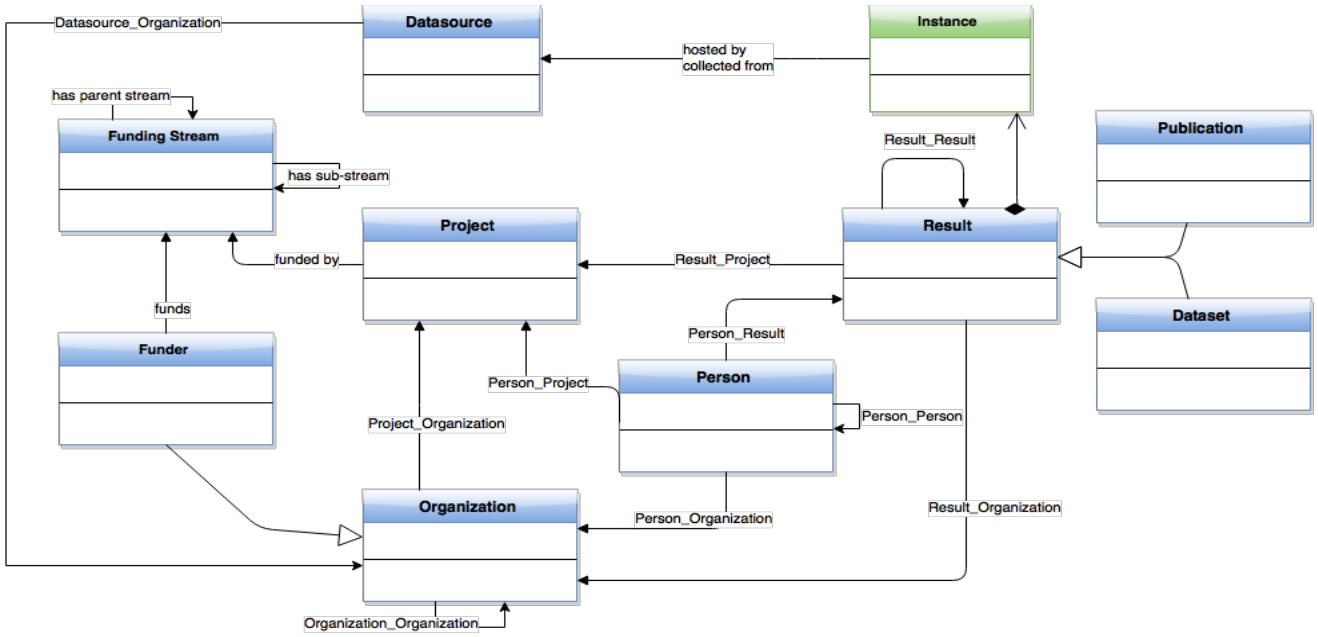

Figure 1 - The OpenAIRE data model

Results are intended as the outcome of research activities and may be related to *Projects*. OpenAIRE supports two kinds of research outcome: *Datasets* (e.g., experimental data) and *Publications* (*Patents* and *Software* entity types will be introduced soon). As a result of merging equivalent objects collected from separate data sources, a Result object may have several physical manifestations, called *instances*; instances indicate URL(s) of the payload file, access rights (i.e. open, embargo, restricted, closed), and a relationship to the data source that hosts such file (i.e. provenance).

Persons are individuals that have one (or more) role(s) in the research domain, such as authors of a Result or coordinator of a Project.

Organizations include companies, research centers or institutions involved as project partners or as responsible of operating data sources.

Funders (e.g. European Commission, Wellcome Trust, FCT Portugal, Australian Research Council) are *Organizations* responsible of a list of Funding Streams (e.g. FP7 and H2020 for the EC), which are strands of investments.

Funding Streams identify the strands of funding managed by a *Funder* and can be nested to form a tree of sub-funding streams (e.g. FP7 -- SP1 -- HEALTH).

Projects are research projects funded by a *Funding Stream* of a *Funder*. Investigations and studies conducted in the context of a *Project* may lead to one or more *Results*.

Finally, OpenAIRE objects are created out of metadata records (e.g. XMLs, CSV, txt, xls, JSON, HTML) collected from various *Data sources*.

In order to give visibility to the original data sources and to track down the origin of information, OpenAIRE keeps provenance information about each piece of aggregated information. Specifically, since de-duplication merges objects collected from different sources and inference enriches such objects, provenance information is kept at the granularity of the object itself, its properties, and its relationships. Object level provenance tells the origin of the object that is the data sources from which its different manifestations were collected. Property and relationship level provenance tells the origin of a specific property or relationship when inference algorithms derive these, e.g. algorithm name and version. Examples are:

- *Document classification properties*: e.g. subjects from a set of standard classification schemes, such as the Dewey Decimal Classification and Medical Subject Headings;
- *Research initiative properties*: e.g., information about the research initiatives, such as the European Grid Infrastructure, related to the research results presented in the publication;
- *Citation properties*: e.g., the list of references cited by the publication, extracted from the bibliography or reference section of the full-text;
- *Relationships* to projects, datasets, and similar publications.

The Information Space is obtained via the combined effort of three infrastructure sub-systems, depicted in *Figure 2*:

- *Harmonization* (aggregation sub-system): The OpenAIRE infrastructure collects metadata records from data sources and derives from them objects and relationships that form the information space graph (typologies and numbers of data sources currently included in OpenAIRE are available from https://www.openaire.eu/search/data-providers). For example a bibliographic metadata record describing a scientific article will yield one publication object and a set of person objects (one per author) with relationships between them. Objects of given entities are transformed from their native data models (e.g. physically represented as XML records, HTML responses, CSV files) onto the OpenAIRE data model in order to build an homogenous information space;
- *Merge* (de-duplication subsystem): objects of the same entity type are de-duplicated in order to remove ambiguities that may compromise statistics and impact (e.g. the same publication may be collected from different repositories as supposedly different objects)
- *Enrichment* (information inference sub-system): publication full-texts are collected and processed by text mining services capable of inferring new property values or new relationships between objects.

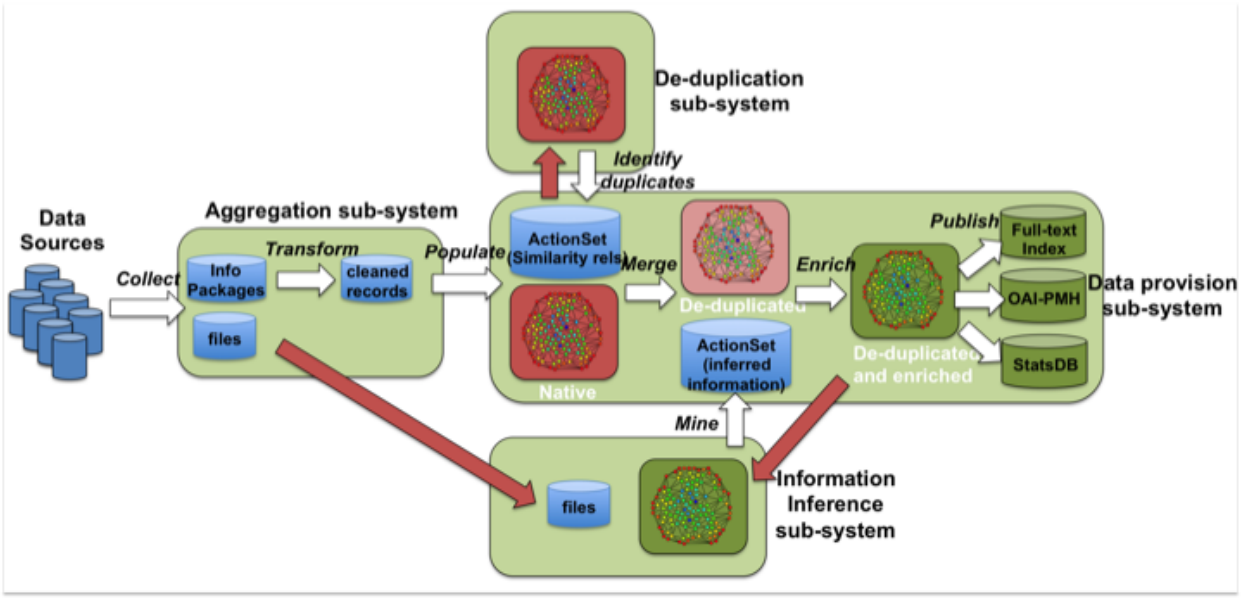

Figure 2 - OpenAIRE services high-level architecture

In order to give visibility to the contributing data sources, OpenAIRE keeps provenance information about each piece of aggregated information. Specifically, since de-duplication merges objects collected from different sources and inference enriches such objects, provenance information is kept at the granularity of the object itself, its properties, and its relationships. Object level provenance tells the origin of the object that is the data sources from which its different manifestations were collected. Property and relationship level provenance tells the origin of a specific property or relationship when inference algorithms derive these, e.g. algorithm name and version. The OpenAIRE Information Space is then made available for programmatic access via several APIs (Search HTTP APIs, OAI-PMH, and soon Linked Open Data) and for search, browse and statistics consultation via the OpenAIRE portal (www.openaire.eu).

4 | FUNCTIONAL REQUIREMENTS

The OLBS operates on top of the OpenAIRE information graph and supports repository managers with a Web Dashboard from which they can subscribe to (potential) "enrichment" and (potential) "addition" events occurring to the graph and of interest to their repository. *Figure 3* shows how the OLBS integrates with the existing OpenAIRE infrastructure. Data sources are aggregated, de-duplicated and enriched by mining techniques so as to populate the OpenAIRE Information space graph. Whenever a new information space is generated, the OLBS explores the graph to detect if any of the active subscriptions finds a match and in such case notifications are generated, delivered, and archived.

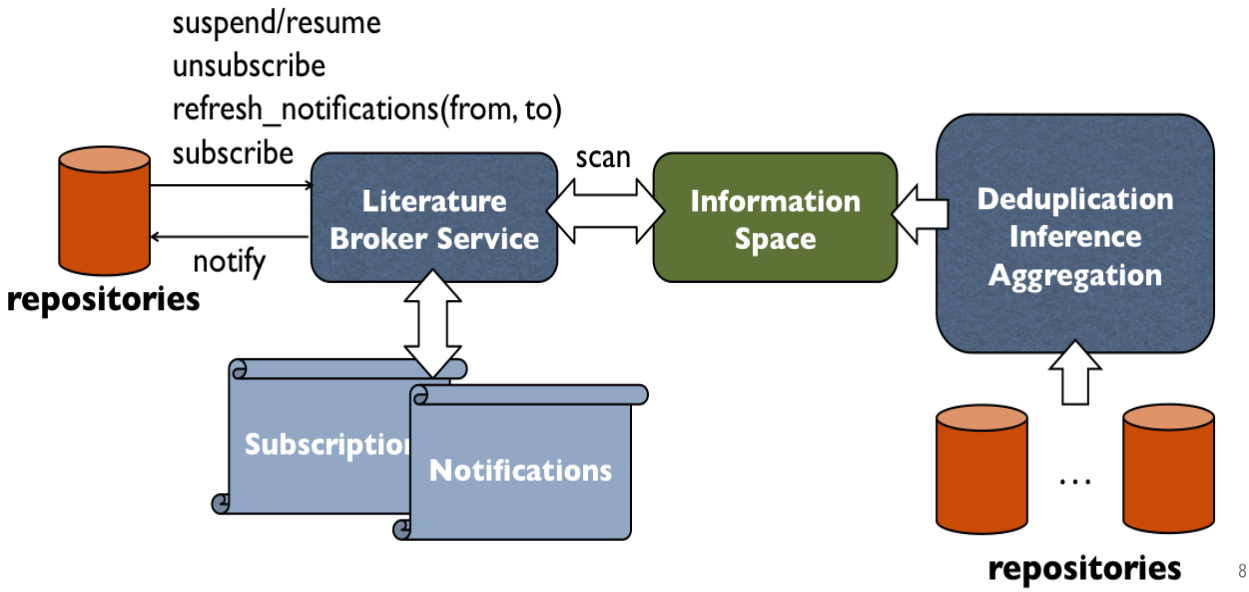

Figure 3 - The OLBS in the OpenAIRE infrastructure

4.1 Subscriptions

Repository managers will be able to subscribe to two main classes of subscriptions: "enrichment" and "addition".

4.1.1 Enrichment

The first class refers to notifications about publications that (i) were collected from the repository by OpenAIRE and (ii) have been enriched with properties or relationships to other objects by OpenAIRE inference algorithms (e.g. relationships to projects and datasets, citation lists, document classification properties) or by the side effect of being merged with richer publication objects (e.g. DOI of a publication, Open Access version of the publication). The identification of these events is straightforward as it is based on provenance of collection (i.e. selects publication objects collected from the given repository) and of enrichment (i.e. further selects objects of the given repository involved into a merge or enriched by inference algorithms). Repository managers will be able to fine-tune their subscriptions based on the bibliographic fields they would like to be notified about; e.g. "return the fields DOIs and Funding Project relative to my records".

4.1.2 Addition

The second class refers to notifications relative to publications that are "relevant to" the repository at hand, but are not present in the repository. The identification of these events requires the navigation of the Information Space graph, in an attempt to identify relationships between the subscribing institutional repositories (which are a specific OpenAIRE data source type) and publications that have not been collected from the given repository but are "relevant to" it. Three relationships have been identified, according to which a publication is relevant to a repository if one of the following chains of relationships exist in the OpenAIRE information graph:

Affiliation repository: publication-author-organization-repository The most intuitive criterion of publication *relevant to* a repository is that based on the relationships *authorship,* i.e. the publication has a given author, *author affiliation*, i.e. the author of the publication is affiliated to an organization, and *organizationRepositoryOfReference*, i.e. the institutional repository of reference of all authors of an organization. While OpenAIRE can collect from data sources relationships between publication-author (e.g. publication metadata) and data source-organizations (e.g. OpenDOAR returns the list of European publication repositories), affiliation relationships between publication and authors are generally not available in collected publication metadata. In fact, publication records provided by data sources (according to the OpenAIRE guidelines, but also according to accepted use of Dublin Core) do not provide author affiliation information or when they do, they follow patterns that vary from case to case and are hard to match automatically.

The inference service of OpenAIRE features a module for affiliation inference, which mines the publication full-texts to identify and extract pairs <author-organization> with a level of trust *T* between [0..1]. If the algorithm is able to determine which author is associated to which organization, then a relationship *affiliation* between the author and the organization is added to the graph, otherwise the *authoringOrganization* relationship is created between the publication and the organization. For the purpose of the Service there is no difference between the two (see Figure 4) as what matters is the identification of a relationship between the publication and repositories.

In summary, the publication has an author whose organization (affiliation) has a given institutional repository of reference; the affiliation relationship *publication-author-organization* is extracted by inference and has a level of trust *T:[0..1],* which represents the level of confidence of the inference algorithm for that specific statement; the relationship *organization-repository* is instead provided by an authoritative OpenAIRE data source, namely OpenDOAR[1], which maintains the directory of repositories and related responsible organizations.

[1] *The Directory of Open Access Repositories*, http://www.opendoar.org/

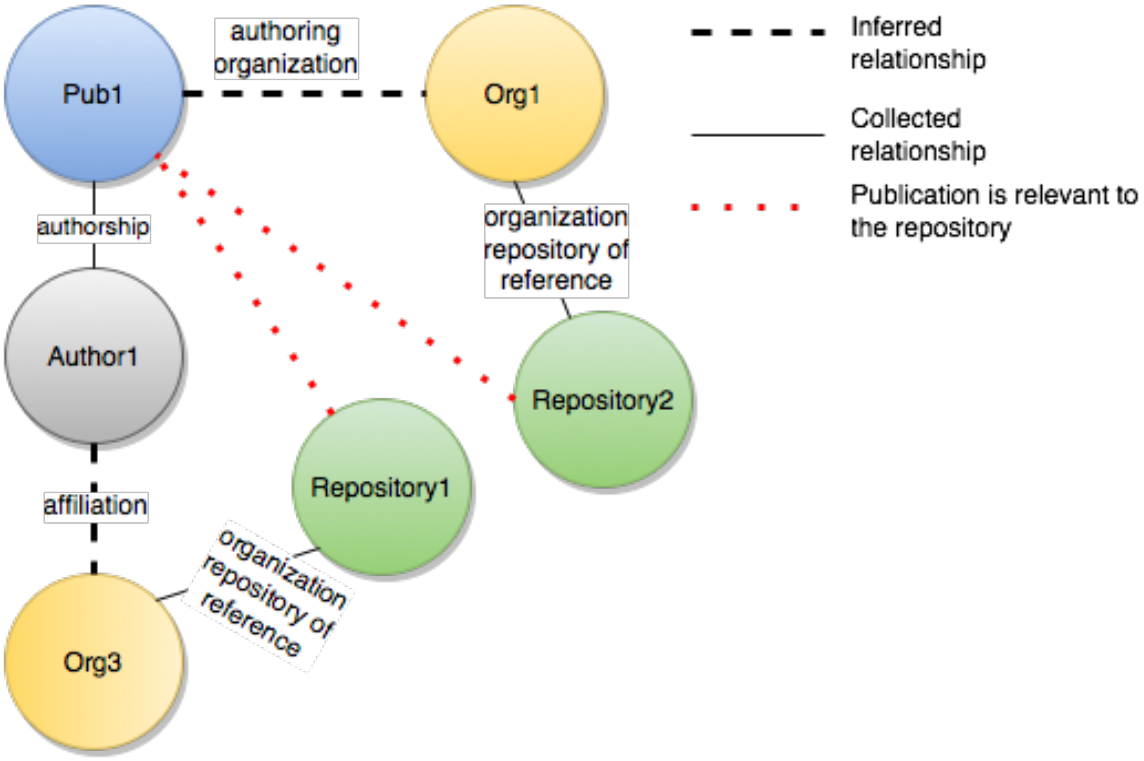

Figure 4 Detection of "relevant to" criterion via full-text mining

Reference repository: publication-author-repository The second criterion to detect which publications may be "relevant to" a repository is based on the relationships *authorship*, i.e. the publication has an author, and *authorRepositoryOfReference*, the author deposits her publication in the given repository. As aforementioned, *authorship* is generally provided by the collected metadata, while *authorRepositoryOfReference* needs to be inferred by OpenAIRE services. To this aim the services exploit the results of the de-duplication algorithms over authors and publications. Harvested metadata records contain authors' names in *dc:creator* fields, as simple strings. In the OpenAIRE information space, such "raw" author objects are initially created with a stateless identifier that makes them unique in the graph. Author identifiers are obtained from the OpenAIRE identifier of the repository, the OpenAIRE identifier of the publication that contains them (obtained from publication identifiers such as DOIs or OAI-PMH identifiers), and the author name string (see Figure 5 for an example). As such, before de-duplication, each occurrence of an author name in a publication from a given data source is considered as a unique author, which carries a pointer to the data source (e.g. the repository) and to the publication that brought it into the system. The result of de-duplication over author objects is a set of "anchor" authors obtained as the merge of several "raw" authors. Starting from "anchor" authors, and exploiting the pointers to institutional repositories of the raw authors they merge, OpenAIRE inference services calculate the notion of "author submission frequency" by counting the number of publications of the author across different repository data sources. In the majority of cases, the repository with the highest submission frequency turns out to be the repository of reference of the

author, namely the one onto which he is supposed to report its publications (in future work, this process will be further refined to identify the "migration" of an author to another institution, therefore to a different repository of reference; this condition may conflict with the "highest number of submissions" criteria, but may be identified using submission dates). Accordingly, with a given degree of approximation, when OpenAIRE collects a new publication from a given repository it is possible to state if some of its authors have a different repository of reference. An exemplification is shown in Figure 5. The same author string ("A. Turing") collected from four different data sources (three institutional repositories and one data source of a different typology) results in four different person objects. When the de-duplication is run, the four persons are merged into one new "anchor" object ("anchor::A. Turing", in the example). Table 1 shows the occurrences of submission of "anchor::A. Turing" considering the provenance of the four "raw" authors it merges. The table shows that the author deposited mostly in repository "Repo1", which can then be considered the repository of reference for the author. Consequently, "Repo1" may be interested in being notified about the publications of the author deposited in "Repo2", "Repo3" and "DS".

In order to avoid useless notifications, publication de-duplication permits to understand whether or not the repository of reference already has the publications or should be notified.

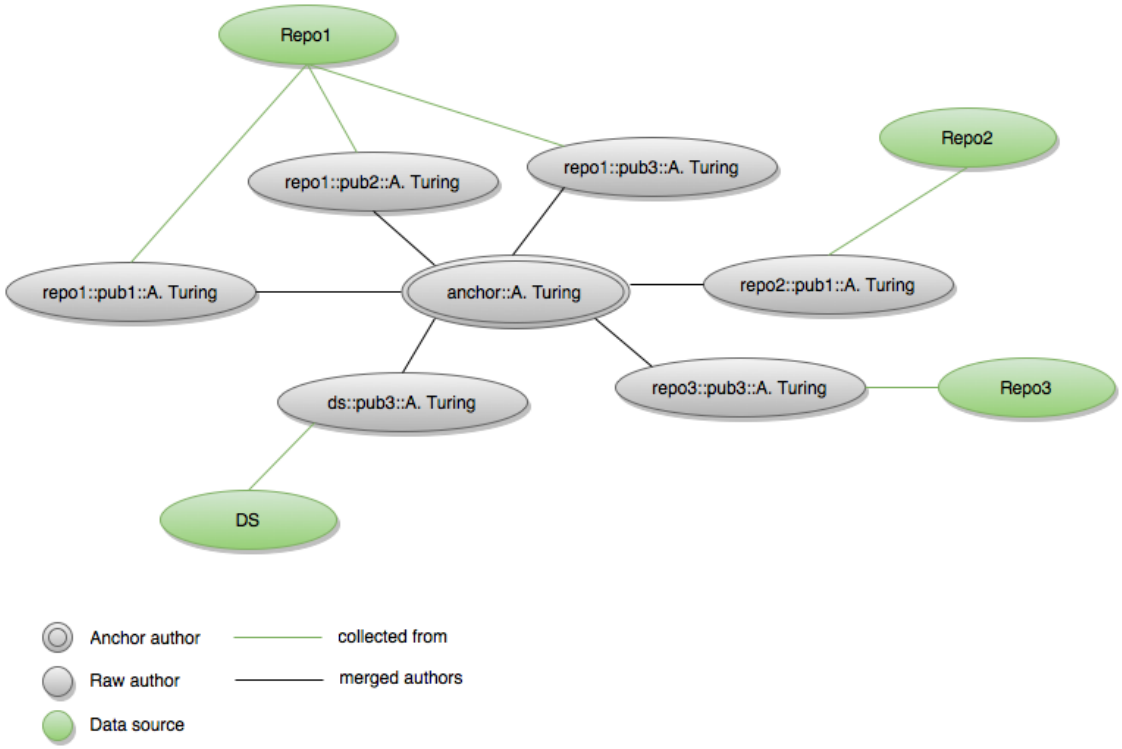

FIGURE 5 AFFILIATION DETECTION: USING DE-DUPLICATION TO COMPUTE THE CLOSENESS OF AN AUTHOR TO A REPOSITORY

TABLE 1 SUBMISSION FREQUENCY FOR THE GRAPH IN FIGURE 5

	Repo1	Repo2	Repo3	DS

anchor::A. Turing	3	1	1	1

The accuracy of the de-duplication algorithms is very important for the correct implementation of this strategy. In addition, repository managers should be able to fine-tune the parameters for the selection of "author submission frequency" (e.g. minimum number of submissions per data source or in total) in order to limit the number of false positive notifications.

A preliminary analysis of the OpenAIRE information space graph for the detection of "frequent submitters" has been carried out considering authors with at least 10 publications and with at least 4 publications in the repository of reference. The analysis is summarized in Table 2 and in the graph in Figure 6. From a total of 157,549 anchor authors from 426 institutional repositories, about the 19% submit their articles into one single repository (i.e. the 100% of the publications of each author has been collected from the same data source). Interestingly, about the 60% submitted publications in different repositories, but their repository of reference hosts from 50% to 99% of their publications. Most likely, repository managers will be interested in this subset of authors, as they are those that mostly deposit papers in one repository, but some of their papers can also be found in other locations. Finally, about the 20% submitted in their repository of reference only up to 50% of their publications.

TABLE 2 AUTHORS, REPOSITORIES OF REFERENCE AND PERCENTAGE OF DEPOSITION

Author publications appearing in repository of reference (%)	Number of authors in this category
10-19%	70
20-29%	4469
30-39%	12,316
40-49%	15,802
50-59%	20,659
60-69%	16,832
70-79%	15,805
80-89%	18,823
90-99%	22,572
100%	30,201

In summary, the relationship *author-repository* is inferred by mining the graph and identifying the correlations between authors, their co-authored publications and the repositories from which these publications were collected; each author can be associated to a repository with a weight of "repository reference-ness", obtained as the percentage of author publications occurring in the repository ("deposition rate"), and the repository with the highest percentage corresponds to the

repository of reference for the author; the information space contains a relationship between authors and their inferred repository of reference, with a degree of trust *T:[0..1]* that depends on the "dominance" of the repository over other repositories (variables are total of author publications and total of related repositories).

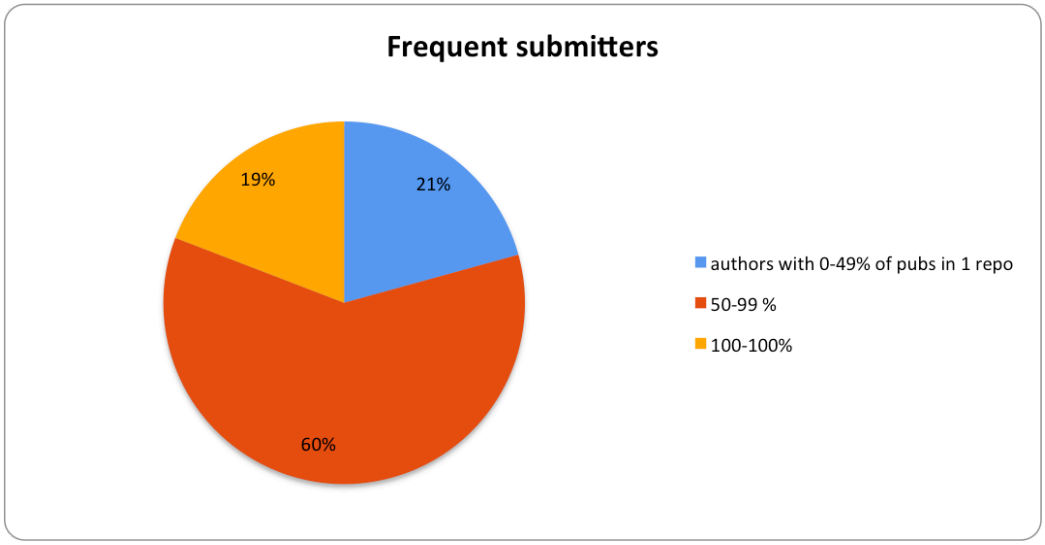

FIGURE 7 - PRELIMINARY ANALYSIS OF THE OPENAIRE GRAPH FOR "FREQUENT SUBMITTERS"

FUNDER REPOSITORY: PUBLICATION-PROJECT-ORGANIZATION-REPOSITORY THE THIRD CRITERION AVAILABLE FOR SUBSCRIPTIONS ON "RELEVANT TO" EXPLOITS THE RELATIONSHIPS *BENEFICIARYOF*, I.E. ORGANIZATIONS INVOLVED IN RESEARCH PROJECTS, AND *ORGANIZATIONREPOSITORYOFREFERENCE*, I.E. THE INSTITUTIONAL REPOSITORY OF REFERENCE FOR ALL AUTHORS OF AN ORGANIZATION. OPENAIRE COLLECTS THESE RELATIONSHIPS FROM PUBLICATION METADATA (E.G. REPOSITORIES, JOURNALS), PROJECT METADATA (FUNDERS), AND REPOSITORY METADATA (OPENDOAR).

Figure 8 exemplifies the concept: CNR-ISTI is an Italian research institute, whose institutional repository is PUblication MAnagement (PUMA). Researchers from CNR-ISTI should deposit their publications in PUMA. CNR-ISTI is involved in the EC funded project OpenAIRE2020 hence some

of the publications linked to the OpenAIRE2020 project may be "relevant to" PUMA because they may be co-authored by researchers working at CNR-ISTI.

The approach has high chances to yield false positive notifications, as some projects involve a considerable number of organizations (e.g. the OpenAIRE2020 EC-H2020 project involves 49 organizations). Repository managers will be able to fine-tune their subscription in order to include, for example, only projects from a given list or projects with a limited number of participants. the publication has been funded by a project whose participants (organizations beneficiaries of the grant) have a given institutional repository of reference; the relationship publication-project is either collected from the data sources or inferred by mining and has therefore a level of trust T:[0..1]; the relationship organization-repository is instead provided, as in the first case, by the OpenDOAR data source; with a degree of approximation, the repositories reached by such relationship may be interested in the given publication.

In summary, the publication has been funded by a project whose participants (organizations beneficiaries of the grant) have a given institutional repository of reference; the relationship *publication-project* is either collected from the data sources or inferred by mining and has therefore a level of trust *T:[0..1]*; the relationship *organization-repository* is instead provided, as in the first case, by the OpenDOAR data source; with a degree of approximation, the repositories reached by such relationship may be interested in the given publication.

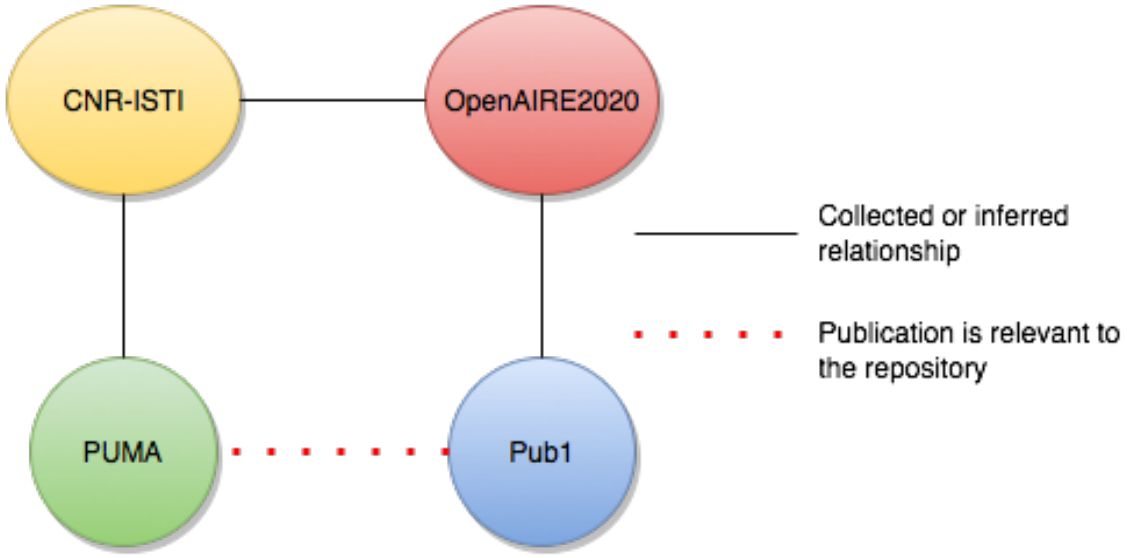

FIGURE 8

DETECTION OF PUBLICATIONS' AFFILIATION: EXPLOITING LINKS TO PROJECTS

Given a publication, if one of such relationships exists in the graph (collected or inferred) the OLBS may notify the subscribing repositories of the publication. Since the relationships are generally not authoritative (not collected from data sources), but inferred by OpenAIRE services, subscribing repository managers can also fine-tune the minimal threshold of trust T_m for their subscriptions.

4.2 Notifications

If a repository manager activates several subscriptions or modifies over time the parameters of existing subscriptions, the same record may meet the criteria of different subscriptions and be notified several times to the repository. Repositories should therefore not be notified more than once of each relevant publication, unless explicitly requested. As a consequence, the OLBS should keep the history of all past notifications to allow repository managers to consult them and avoid their re-sending to the repositories. Two different notification strategies are under evaluation in order to meet diverse requirements of subscribers:

- **Mail postcards** Subscribers may opt to be notified by email at given interval of times (e.g. daily, weekly, monthly) and with given granularity (individual records, digests, URL to a web user interface).
- **Programmatic access** *Pull mode*: APIs will be provided to retrieve notifications by status (e.g. read/unread), subscription typology, and filters (e.g. criteria on the metadata fields). *Push mode:* the SWORD protocol for automatic ingestion of records into repositories will be evaluated.

4.3 Web Dashboard

Repository managers are supported with the OLBS Web Dashboard, from where they can activate and configure subscriptions of the available types, how publications of these categories should be notified, and also explore the history of notifications they have so far received (e.g. searching and browsing notifications by type, publication title, authors publication date, notification date). Repository managers can fine-tune their subscriptions by real-time interacting with the graph and test the quality of the notifications they would collect before they actually submit the subscription. A summary of the available functionalities is:

- Configure subscriptions, one for each type of addition or enrichment subscription;
- Manage notification mode relative to each subscription typology: configure scheduling, suspend/resume notifications, opt for one the available notification modalities; select the format of the email notification digests to receive among a set of supported formats (e.g. Dublin Core XML, Bibtex and citations in ASCII);
- Manage history of notificationsView, download or re-send old notifications.

5 | DATA MODEL AND ARCHITECTURE SPECIFICATION

This section presents the data model specification, the high level architecture of the OLBS, and the proposed implementation aiming at satisfying the functional requirements described in section 2.

5.1 Data model

The data model is illustrated in Figure 9 and includes the classes: *repository, subscription, potential notification,* and *notification*. The model relies on a "topic tree"[2] that encodes the typologies of subscriptions supported by the OLBS, which are of two main classes *addition* and *enrichment*:

- *addition.<subclass>*: the values of *<subclass>* encode the three approaches for the identification of publications "relevant to" a repository, namely: *byAffiliation*, *byReference*, and *byFunder*.
- *enrichment.<subclass>*: the values of *<subclass>* encode the six publication enrichment notification use-cases: *open_access_version*, *project_link*, *dataset_link*, *subject classifications*, *DOI*, *author_PID*.

Repository Institutional repository registered to the OLBS.

Subscription Subscriptions are associated to a *repository* and are characterized by the following configuration parameters:

- *trustThreshold*: the minimum value of trust in the interval [0..1] that a *potential notification* must satisfy to be included in the *notifications* of this subscription;
- *criteria:* CQL query over the set of publication fields included in *potential notification*; the criteria must hold for the *potential notification* to beome a *notification* for the subscription;
- *type*: a path in the subscription tree, which identifies the *potential notifications* of interest
- *notification_scheduling:* how often the notifications for this subscription must be generated and sent (daily, weekly, monthly);
- *notification_granularity*: what must be included in each notification (full Dublin Core of all publications, digest, or a URL to the web dashboard)
- *notification_mode:* possible options will depend on the OLBS functionalities, examples are email and SWORD protocol.

[2] *OASIS Standard WS-Topics specification* 1.3 (2006), http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf

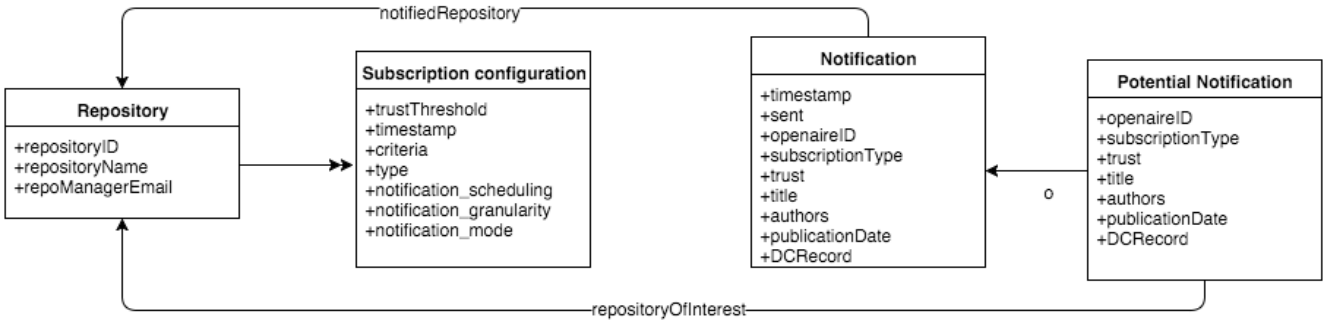

FIGURE 9 - DATA MODEL

Potential Notification Potential Notifications are generated by analyzing the OpenAIRE graph information space to identify the whole range of repository-publication pairs that may be of interest to repositories in OpenAIRE, i.e. independently from the existence of an active *subscription*. Each potential notification is associated to a *repository* and is characterized by the following properties:

- *openaireId*: the unique identifier of the publication in the information space;
- *title*, *authors*, *publicationDate* of the publication, to enable search and browse;
- *DCRecord*: the Dublin Core record of the publication;
- *subscriptionType*: path in the topic tree;
- *trust*: level of trust of the association between the publication and the *repository* w.r.t. the subscription type;

Notification When a *potential notification* matches a *subscription*, a corresponding notification is created and the *repository* is alerted, respecting *scheduling* and *mode* of the relative *notification configuration*. A notification must be persisted as the evidence that a publication has been notified to a repository. As such, it is a copy of the relative *potential notification*, which, due to the variability of the information space, may not necessarily persist as long as the notifications it has generated (publications in OpenAIRE may disappear). A notification includes all properties of the corresponding potential notification plus the *timeOfCreation* of the notification.

5.2 Architecture overview

The OpenAIRE information space is stored in an HBASE cluster (8 worker nodes, each of them with 8 CPUs and 24GB RAM) in order to support performance and scalability. Metadata records collected from data sources are harmonised and transformed into objects compliant to the OpenAIRE data model. The objects are then stored into HBASE, where each object is converted into one HBASE row. In December 2015, OpenAIRE collected more than 15,7M publication metadata records, corresponding to more than 30 millions OpenAIRE objects (and therefore HBASE rows). Once populated, the HBASE table is ready to be processed by inference and de-duplication algorithms for enrichment.

- The inference subsystem analyzes the OpenAIRE information space, supported by the available full-texts of publications to generate properties and relationships, including the *relationships publication-XXX-repository* needed by the three subscription methods mentioned

in Section 2.1. In December 2015, the inference subsystem enriched more than 1,2M publications (8% of the total publications) with relationships and properties.

- The de-duplication subsystem runs Mapreduce jobs on the HBASE table that implement heuristics for the detection of duplicates of publications, organisations and persons. Groups of duplicate objects are then merged into one disambiguated object. Depending on the configuration settings, the deduplication process for 15M publications takes 2-3 hours, identifying 4,1M of duplicates and merging them into 1,7M of disambiguated objects.

The enriched HBASE table is then processed for publishing via the OpenAIRE portal and standard APIs (http://guidelines.openaire.eu). The OLBS will have to further process the OpenAIRE information space stored on HBASE to identify addition and enrichment events to be notified to subscribers. Figure 10 shows the OLBS integration in the OpenAIRE infrastructure, which consists of a three-phase data workflow.

Phase 1: generation of potential notifications Whenever the OpenAIRE infrastructure generates a new version of the information space (i.e. hamornisation, deduplication, enrichment steps are performed), the OLBS will run MapReduce jobs on the HBASE table to identify the current *potential notifications*. These jobs produce tuples of the form:

openaireID	SubscriptionType	repositoryId	trust	title	authors	PublicationDate	DCRecord

For each publication-repository pair such that: *(i)* the publication was collected from the repository but is richer in properties or relationships in OpenAIRE (*trust* corresponds to the level of *trust* of the OpenAIRE enrichment), or *(ii)* there exist a chains of relationships *addition.affiliation, addition.reference,* or *addition.funder* between the two (*trust* corresponds to a combination of the inferred relationship *trust* level and other parameters, see section 2.1; the formulae will be refined over time).

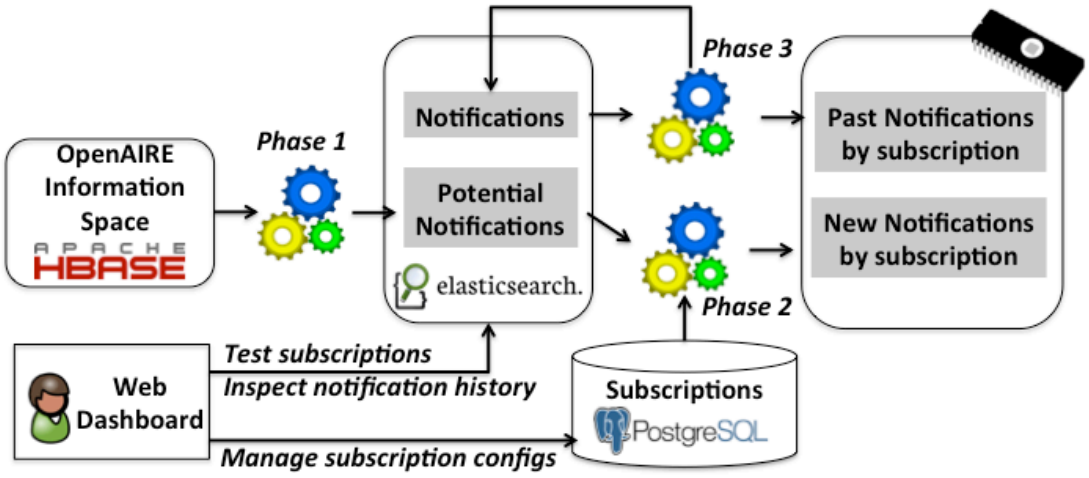

FIGURE 10 - OLB DATA FLOW

Scalability issues Potential notifications may reach very large numbers, which can be estimated as follows. For additions: since institutional repositories aim at collecting all publications of authors of reference, assuming the average of missing publications is 20% would reasonably be a worst-case scenario. Since the average repository in OpenAIRE counts 13,000 publications, its 20%

being 2,600, and since OpenAIRE counts around 400 institutional repositories, the estimate number in the average-case would be around 1M. For enrichments: number of institutional repository publications subject to de-duplication and therefore possibly affected by enrichment is 1,4M publications; inference is applied to publications with a PDF, today around 3M. In summary, the worst-case envisages 4,4M multiplied by the 6 potential "enrichment" notifications, for a total of around 27M. Hence, the order of magnitude of potential notifications is overall around 28M. In order to make them searchable (Web Dashboard requirement) the OLBS will adopt a scalable back-end, efficient on dropping and feeding entries, and also capable of supporting efficient queries. The current implementation plan finds in Elasticsearch[3] full-text index the best fitting candidate among the Open Source search engines because of its capability of scaling up horizontally and the expressivity of its data model and query language.

Phase 2: subscription matching

Based on the available subscriptions, especially on the *notification_schedule* property, the OLBS queries the index of potential notifications to identify those that match the subscription conditions: repository, subscription type, criteria, and minimal level of trust. Subscriptions, managed by repository managers via the Web Dashboard, are stored on a relational database (PostgreSql[4]). The results of the query are temporarily kept in memory, ready to be used in the following phase, when the OLBS identifies new notifications and avoids redundant notifications.

Phase 3: find new notifications

In this phase the OLBS identifies which among the potential notifications selected in phase 2 have not yet been sent to the repository of interest, as a consequence of previous notification schedules. Novel notifications are stored in the Elasticsearch index as separate *notification* entries of the following shape:

timestamp	openaireID	Subscription Type	repositoryId	trust	title	Authors	Publication Date	DCRecord

After phase 2, *notification* entries in the index represent notifications that have been previously sent to the repositories. The OLBS queries the index for *notifications* matching the same subscription type and intersects the result in memory with the potential notifications selected in phase 2. The action will keep only new notifications and feed them in the index, in order to avoid that the same publications are sent to repository under the same subscription type.

[3] Elasticsearch *https://www.elastic.co*
[4] PostgreSql *http://www.postgresql.org*