An Open Cultural Digital Content Infrastructure

Ioanna-Ourania Stathopoulou
National Documentation Centre / National Hellenic Research Foundation, Greece
iostath@ekt.gr

Haris Georgiadis
National Documentation Centre / National Hellenic Research Foundation, Greece
hgeorgiadis@ekt.gr

Vangelis Banos
National Documentation Centre / National Hellenic Research Foundation, Greece
vbanos@gmail.com

Panagiotis Stathopoulos
National Documentation Centre / National Hellenic Research Foundation, Greece
pstath@ekt.gr

Nikos Houssos
National Documentation Centre / National Hellenic Research Foundation, Greece
nhoussos@ekt.gr

Evi Sachini
National Documentation Centre / National Hellenic Research Foundation, Greece
esachin@ekt.gr

ABSTRACT
We present an Open Cultural Digital Content Infrastructure, a platform providing a coherent suite of loosely-coupled services that aim to promote quality in repositories and facilitate metadata and digital content reuse. The key functions of the infrastructure are the aggregation of metadata and digital files and the automatic validation of metadata records and digital material for compliance with quality specifications. The system that has recently moved to production, is currently being employed to ensure the quality standards of the output of more than 70 projects that support Greek cultural heritage organisations and are funded by the European Union structural funds. These projects are expected to produce more than 1.5 million digitized and born-digital items accompanied with detailed metadata. The validation is based on a set of quality and interoperability specifications that have been developed for the purpose. The infrastructure has been developed using an open source technology stack and tools and in particular reuses a number of components of the publicly available Europeana aggregator and portal software platform.

Categories and Subject Descriptors
H.3.7 [**Digital Libraries**]: Collection, Dissemination, Standards, System issues.

General Terms
Management, Design, Standardization.

Keywords
Metadata aggregation, Metadata quality, Metadata validation, Digital content aggregration, Digital content aggregration validation, Cultural Heritage Infrastructures, OAI-PMH, Interoperability guidelines, Metadata harvesting.

1. INTRODUCTION
The issue of appropriate digitization, documentation and preservation of cultural heritage has been widely recognized for its significance, resulting in a great number of large-scale efforts worldwide. A key issue in achieving the appropriate return from the corresponding investments is ensuring, to the greatest extent possible, the quality of the output at the level of both the metadata records and the digital files and their appropriate safe-keeping, dissemination (including provision via open APIs according to established standards) and preservation.

In the past, issues that are hampering the reuse and added value of the documented and digitized cultural heritage items have been observed such as inadequate and non-standards compliant documentation (e.g. use of custom data models instead of established international schemata), poor quality in digitization and relevant processing (e.g. low image resolution, omission of Optical Character Recognition in scanned texts), non-availabity of standard system interfaces for opening up metadata (lack of OAI-PMH support), failure to secure appropriate safe-keeping of digital files and interruptions (sometimes permanent) in the operation of web applications hosting the material.

To avoid these phenomena in current digital cultural heritage funded projects in Greece, a scheme has been created for the development of an infrastructure to aggregate centrally metadata records and digital files produced in the frame of these projects and automatically validate their conformance with interoperability and quality specifications. A set of such specifications has been developed at the initial stage of the funding programme [1].

The infrastructure that has been built to support this effort is presented in this contribution. It contains a Metadata Aggregator that aggregates metadata at national level and supplies a series of value-added services based on them, such as Search Engine and unified provision of content as Linked Data, and a Content Validator that validates the registered repositories against interoperability requirements and the provided content, including both metadata and digital files, against a large and extendible pool of specifications. The Validator is to be used both by the Aggregator personnel to ensure the quality of the metadata and digital files that are to be ingested and published by the Aggregator, and by the content managers and project contractors of the repositories in order to validate their content and to take the necessary steps towards conforming to the specifications. The Validation comprises two components, a front-end and a back-

end, detailed in Section 2.2. A shared, autonomous harvester component supports the operations of both the aggregator and the validator. A registry of repository providers is also maintained as a separate component. Communication among the components is performed via REST APIs. The architecture of the infrastructure is depicted in Figure 1.

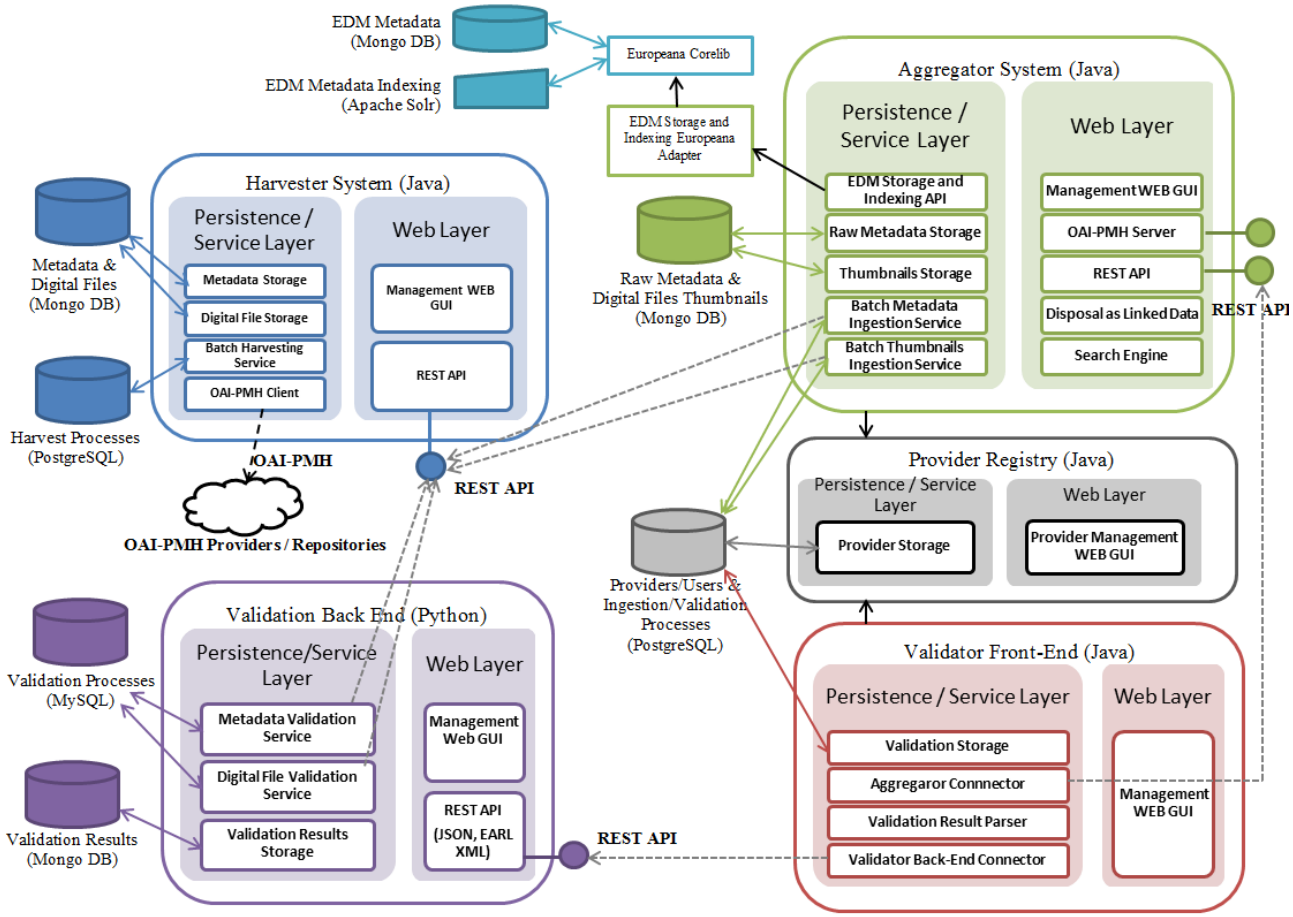

Figure 1. The Architecture of the infrastructure and service components

2. Infrastructure architecture, design and implementation

2.1 Metadata and digital file harvester

An autonomous harvester system is built which harvests and stores both metadata and digital files. In order to improve the efficiency and the throughput of the harvester the execution is implemented in a pipelined workflow scheduling [11] which adopts the pull data workflow model needing no materialization of intermediate results. The OAI-PMH XML responses are parsed on-the-fly using the Streaming API for XML and as soon as the parsing is finished, they are being forward to dependent tasks in the workflow which are being executed simultaneously. REST API has been implemented that allows external software components (such as the Validator or the Aggregator) to trigger and manage harvests.

2.2 Automatic validation of metadata records and digital files

The Validator is created with the aim to implement a validation model which operates in many levels: repository interoperability validation, metadata validation and digital file validation.

An important design choice considered the representation of the validation rules and the introduction of validation logic into the system. The approach of hard-coding the validation logic directly into the application code was rejected from the start, since it would result in a platform which would be hard to maintain and modify in the future, especially in view of the expected continuous evolution of validation requirements and rules. Instead, a dynamic platform was developed to support the definition of arbitrary validation models outside the application code with the use of a novel Validation Domain-Specific Language (VDSL).

2.2.1 Validator architecture

The Validator consists of two autonomous systems, the *back-end* and the *front-end*. The former provides the facilities to specify complex validation rules at any level (repository, metadata, digital files), execute validations and record detailed results, while the latter provides a user interface for personnel responsible to carry out validation procedures, aggregate reporting and connection with the repository providers and administrative procedures of validations. Some key system architecture points of the entire system are summarized in the following:

- **Interfaces:** Both systems have a web GUI for controlling every aspect of the validation processes. Furthermore, the Validator back-end is featuring a REST API that allows external software components to trigger and manage validation processes.
- **Input:** The input (metadata records or digital files) comes from the Harvester. A validation process may trigger a new harvest process in the harvester or utilize metadata/digital files that are have been already harvested by the harvester.
- **Output:** The validation results are stored permanently and can be served anytime both in analytical (per record/digital file) or aggregative form, via the REST API and the web GUI.

2.2.2 Validation Domain-Specific Language (VDSL)

We design and implement the VDSL as part of the Validator component of the OpenCDCI. Using VDSL we are able to define arbitrary repository, metadata and digital file validation rules and express validation logic in a precise and expressive way. The VDSL consists of building blocks such as:

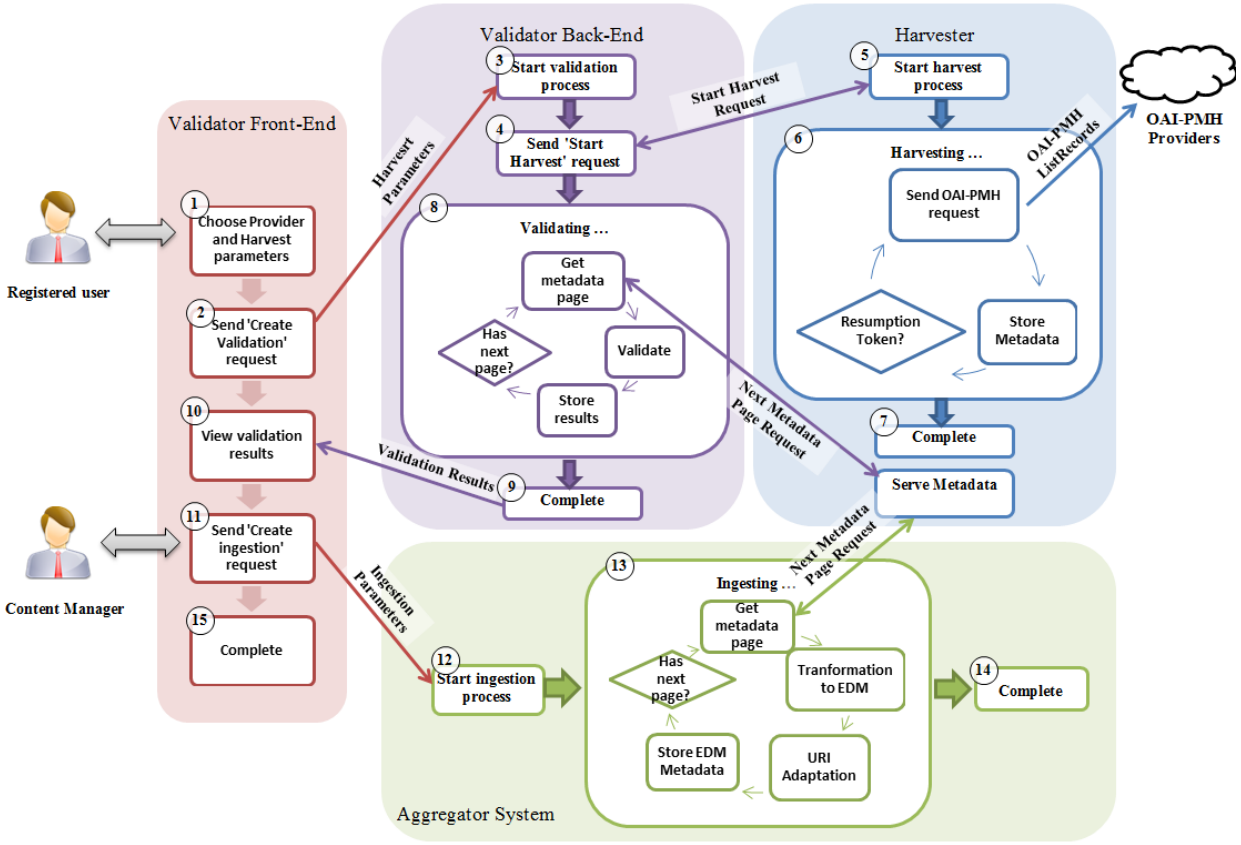

Figure 2. Validation and Ingestion workflows

- Metadata element validation rule constructs,
- Metadata document validation rule constructs,
- Digital file validation rule constructs,
- Boolean operators (AND, OR, etc)
- Control flow operators (IF, ELSE, etc)

Using these constructs, we can define validation rule sets using a JSON format, such as the one presented in Figure 3.

Using the VDSL yields many advantages:

- We facilitate the definition of validation models closer to the conceptual model, making it easy for non programmers to define and update validation models according to business rules.

```
"dc:identifier": {
    "url.exists": {"spec_id": "repository09", "if": "url.syntax"},
    "url.handle": {"spec_id": "repository09", "at_least_one": 1}
},
"dc:language": {"language.iso639": {"spec_id": "repository13"}},
"dc:date": {"date.iso8601": {"spec_id": "repository14"}},
"dc:creator": {
    "author.check": {"spec_id": "repository12"},
    "controlled_vocabulary.check": {"spec_id": "repository18"}
},
"dc:subject": {"controlled_vocabulary.check": {"spec_id": "repository17"}},
"dc:coverage": {"controlled_vocabulary.check": {"spec_id": "repository19"}},
"dc:publisher": {"controlled_vocabulary.check": {"spec_id": "repository20"}},
"dc:type": {"controlled_vocabulary.check": {"spec_id": "repository11"}}
```

Figure 3 Excerpt from validation ruleset

- We provide extreme flexibility in defining and updating validation models, matching evolving requirements without issues.
- We have a flexibleValidator system architecture which enables us to develop and test new rule constructs as plugins, without affecting the existing system and validation logic.

2.2.3 Related work

Several repository validator systems are in production operation since a number of years, such as the OpenAIRE validator based on the OpenAIRE guidelines for repositories [6][7], the ARIADNE validator of learning object repositories [4][5], the OAI-PMH validator [3] and validator systems produced in the context of Europeana [2] such as the VAMP semantic validation Service for MPEG-7 profile descriptions [8]. Some key features and differentiations of our solution are summarized in the following:

- We validate not only metadata records but also digital files and in addition we check the correct mapping of digital files with the corresponding metadata records.
- Certain existing systems limit themselves to only syntactic validation using technologies such as XML Schema and the schematron assertion language [9], while we exercise also semantic validation (as does VAMP [8] and to some extent also OpenAIRE [7]).
- Semantic validation is provided with a great degree of flexibility. For instance, we check whether metadata values of key attributes (e.g. locations, subjects, languages, time periods) belong to formally defined (e.g. with SKOS) controlled vocabularies – the vocabularies can be dynamically configured in the validation rules and do not need to be known a priori to the system.
- Combinations of validation rules are expressed using a domain specific language which is designed to be usable by non-programmers.
- Validation administrative procedures (e.g. connection with repository owners, checking for compliance using rules of a specific funding programme mandate, reporting) are decoupled with the actual validation logic and implemented at a separate component (validator front-end).

2.3 Aggregator

The Aggregator in our solution is the system that aggregates metadata from registered content providers, creates and stores thumbnails for digital files and provides with a series of value-added services, such as a metadata search engine and the disposal of the ingested metadata as Linked Data and for harvesting via OAI-PMH. The metadata records and the digital files from which thumbnails are created derive from the Harvester.

The aggregator ingests metadata in the Europeana Data Model (EDM) [10], which is the new proposal of Europeana for structuring cultural metadata. The Aggregator supports an extensible pool of transformations from well-known metadata formats to EDM.

The Aggregator uses the Europeana EDM Storage component, named Europeana Corlib, as an EDM Storage and Indexing backend. The integration with the Europeana Corelib is done through a generic EDM Storage and Indexing API which is agnostic to the storage backend that is actually used and can cover any storage system able to persist and index EDM structures. The Europeana Corelib stores EDM metadata records in Mongo DB and indexes them using the Apache Solr.

3. Conclusions

We designed and implemented a solid and extensible ingestion workflow that starts from the retrieval of metadata/digital files from the Harvester (using its REST API), provided that are already validated by the Validator, the enforcement of the appropriate transformations, including metadata format transformations and URI conversions, and ends up in the persistence of the EDM metadata records in the EDM Storage and Indexing System and of the thumbnails derived from the digital files in a NoSQL database. The workflow, which is illustrated in Figure 2 is implemented in a pipe-lined fashion, consisting of a series of modular operators and adopts the pull data-flow pattern, needing no materialization of intermediate results.

4. Acknowledgements

The presented work has been partly supported by the project "Platform for provision of services for deposit, management and dissemination of Open Public Data and Digital Content" (Ref: 327378). co-funded by Greece and the European Union through the Operational Programme "Digital Convergence" (NSFR).

REFERENCES

[1] Stathopoulos P. et al. (2013). *Specifications and features for the interoperability of open digital content* [Online]. Available: http://helios-eie.ekt.gr/EIE/handle/10442/8887

[2] *Europeana Validation tools* [Online]. Available: http://pro.europeana.eu/web/guest/thoughtlab/improving-metadata-quality

[3] *OAI PMH validato*r [Online]. Available: http://validator.oaipmh.com/

[4] *Ariadne validator* [Online]. Available: http://ariadne.cs.kuleuven.be/validationService/validateMetadata.jsp

[5] Klerkx J, Vandeputte B, Parra G, Santos JL, Van Assche F & Duval E, "How to share and reuse learning resources: the ARIADNE experience," Sustaining TEL: from innovation to learning and practice, vol. 6383, pp.183-196, 2010.

[6] *OpenAire Validator* [Online]. Available: http://www.openaire.eu:8380/dnet-validator-openaire/

[7] Schirrwagen, J., Manghi, P., Manola, N., Bolikowski, L., Rettberg, N., & Schmidt, B, "Data Curation in the OpenAIRE Scholarly Communication Infrastructure," in Information Standards Quarterly, vol. 25, no. 3, pp. 13-19, 2013.

[8] Troncy, R., Bailer, W., Höffernig, M., & Hausenblas, M., "VAMP: a service for validating MPEG-7 descriptions wrt to formal profile definitions," Multimedia Tools and Applications, vol. 46, no. 2-3, pp. 307-329, 2010.

[9] Jelliffe, Rick. "The schematron assertion language 1.5." Academia Sinica Computing Center, 2000.

[10] *Europeana Data Model Primer* [Online]. Available: http://pro.europeana.eu/edm-documentation

[11] Anne Benoit, Ümit V. Çatalyürek, Yves Robert, and Erik Saule, "A survey of pipelined workflow scheduling: Models and algorithms," ACM Computing Surveys (CSUR), vol.45, August 2013.