On Constructing Repository Infrastructures
The D-NET Software Toolkit

Paolo Manghi,[1] Marko Mikulicic[1], Katerina Iatropoulou,[2]
Antonis Lebesis,[2] Natalia Manola,[2]

[1] Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo"
Consiglio Nazionale delle Ricerche, Pisa, Italy
{paolo.manghi, marko.mikulicic}@isti.cnr.it

[2] Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens, Greece
{kiatrop,antleb,natalia}@di.uoa.gr

1. Repository infrastructures

Due to the wide diffusion of digital repositories, organizations responsible for large research communities, such as national or project consortia, research institutions, foundations, are increasingly tempted into setting up so-called *repository infrastructure systems* (e.g., OAIster,[1] BASE,[2] DAREnet-NARCIS[3]). Such systems offer web portals, services and APIs for cross-operating over the metadata records of *publications* (lately also of *experimental data* and *compound objects*) aggregated from a set of repositories. Generally, they consist of two connected tiers: an *aggregation system* for populating an *information space* of metadata records by harvesting and transforming (e.g., cleaning, enriching) records from a set of OAI-PMH compatible data sources, typically repositories; and a *web portal*, providing end-users with advanced functionality over such information space (search, browsing, annotations, recommendations, collections, user profiling, etc). Typically, information spaces also offer access to third-party applications through standard APIs (e.g., OAI-PMH, SRW, OAI-ORE).

Repository infrastructure systems address similar architectural and functional issues across several disciplines and application domains. On the one hand they all deal, with more or less contingent complexity, with the generic problem of harvesting metadata records of a given format, transform them into records of a target format and deliver web portals to operate over these records. On the other hand, they have to cope with arbitrary numbers of repositories, hence administering them, from automatic scheduling of harvesting and transformation actions, definition of relative transformation mappings, to the inherent scalability problems of coping with ever growing incoming records.

Existing solutions tend to privilege customization of software, neglecting general-purpose approaches. Typically, for example, aggregation systems are designed to generate metadata records of a format X from records of format Y, and not be parametric with respect to such formats. Similarly, the participation of a repository to an infrastructure is driven by firm policies and administrators often do not have the freedom of specifying their own workflow, by combining as they prefer logical steps such as harvesting, storing, transforming, indexing and validating.

In summary, repository infrastructure systems typically provide advanced and effective solutions tailored to the one scenario of interest, while can hardly be applicable to different scenarios, where similar but distinct requirements apply. As a consequence, an organization willing to set up a repository infrastructure system with peculiar requirements has to face the "expensive" problem of designing and developing a new software from scratch. In this paper, we present a general-purpose and cost-efficient solution for the construction of customized repository infrastructures, based on the *D-NET Software Toolkit* (`www.d-net.research-infrastructures.eu`), developed in the context of the DRIVER and DRIVER-II projects (`http://www.driver-community.eu`). D-NET offers a service-oriented framework, whose services can be combined by developers to easily construct *customized* aggregation systems and *personalized* web portals. D-NET services can be

[1] http://www.oaister.org
[2] http://www.base-search.net
[3] http://www.narcis.info

customized, extended and combined to match domain specific scenarios, while distribution, sharing and orchestration of services enables the construction of scalable and robust repository infrastructures. As we shall describe in the following, D-NET is currently the enabling software of a number of European projects and national initiatives.

2. D-NET Software Toolkit

The *D-NET Software Toolkit* software is open source, developed in Java and available for download at www.d-net.research-infrastructures.eu. D-NET implements a Service Oriented Architecture (SOA) – based on the Web Service framework – capable of operating repository infrastructures. In D-NET a repository infrastructure is a run-time environment, enabled by an overlay network of core services (called *enabling services*), where *multiple organizations* can dynamically share D-NET services to *collaboratively* construct customized repository aggregation systems and personalized web portals.

In particular, a D-NET repository infrastructure has two main logical layers: the system core, called *enabling layer*, whose function is to support the operation of the *application layer*, which consists of the services in the *data management* and *end user functionality* areas (see Figure 1) used by the developers to form the applications. In the following, we introduce the enabling layer services and then describe how developers can construct their custom aggregation systems and web portals.

Enabling Services The enabling area comprises three main services:

Information Service (IS). The service addresses service *registration* and *discovery*. As in peer-to-peer systems, developers can dynamically add or remove services from the infrastructure. Hence, services in the need to interact with others, must be able adapt to the latest system's "service map". To this aim, services *register* to the IS their *profile* (information about their location, the functionality they expose and their current status) and *discover* the services they need by searching profiles in the IS.

Manager Service (MS). The service addresses service *orchestration*. The MS can be configured by developers to autonomously execute *workflows*, i.e., distributed transactions, involving a number of services, as a consequence of "events" (e.g., new repository available). The MS reacts to an event, locates through the IS the services needed for the relative workflow, instructs them and monitors their behavior. Examples of workflows are: every week, harvest records from repository A, transform them according to a given mapping, and send them to three distributed index service replicas; monitor index replicas, if one of the index replicas is not reachable (network failure or service de-registration), find another index service available and create another replica.

Authentication and Authorization Service. The service addresses AA communication. i.e., services may concede access only to Authorized services, which in turn must have properly Authenticated to the system at registration time; the service describes AA policies through the eXtensible Access Control Markup Language (XACML) standard.

Customized Aggregation Systems and Personalized Web Portals in D-NET. D-NET services are organized in two areas from which organization developers can fish to assemble their service-oriented applications: the data management and the end user functionality area. The former contains services for aggregation system management, i.e., *construction of information spaces*: harvesting of repositories, record transformation and validation, feature extraction, storage and indexing, compound object management, collection construction, OAI-PMH publishing, and others. The latter contains services for web portal construction, i.e., *web access to information spaces*: user and community profiles management, recommendations and alerts, search and browse over the metadata records. In particular, D-NET services are designed by following principles of general-purpose software, with the goal of leveraging reuse, personalization and composition principles:

Modularity: services provide minimal units of functionality so that they can be arbitrarily composed to meet custom data management workflows. A system may harvest metadata records, store them and then index them, while another one may instead need to harvest and then index them straightaway. Similarly, some web portals might include browse and recommendation functionalities, while other would not.

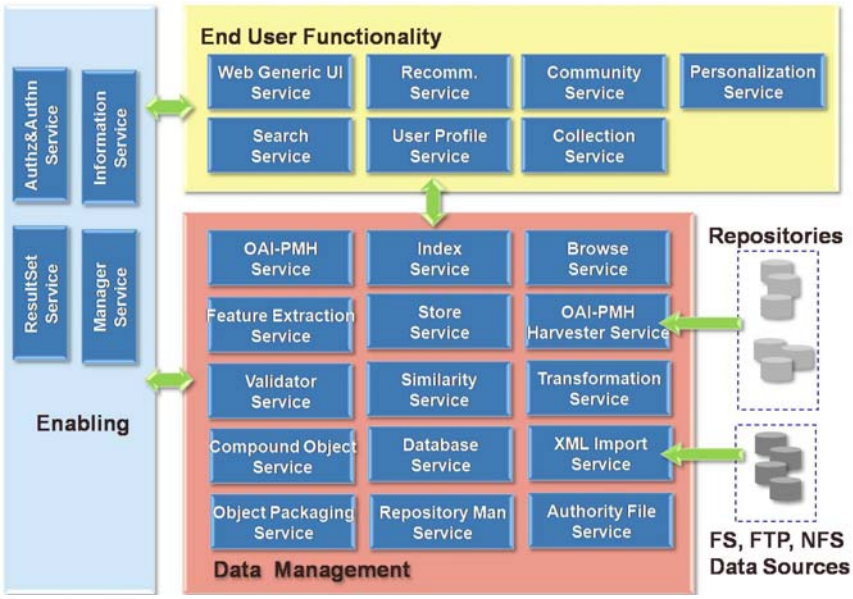

Figure 1 – D-NET service architecture.

Customizability: services support polymorphic functionalities, thereby operating over metadata records whose format matches a generic structural template. For example, the metadata record index service is designed to index records of any given format and the transformator service can, given the proper mapping (defined by administrators through user interfaces), transform any format onto any format. Similarly, search and browse functionalities of web user interfaces should adapt to any format required by the application.

Extensibility: D-NET is open to the addition of new services, in order to introduce new functionality, whenever this is required and without compromising the usability of other services.

In summary, developers using D-NET can benefit from the following key properties:

- *customizability*, *extensibility*, *modularity* of service software: enabling personalization of the applications by configuring D-NET services to match domain application needs, adding new services to compensate absence of functionality, building tailored service workflows;
- *distribution*, *sharing* and *orchestration* of its service applications: enabling the operation of scalable, robust and autonomous applications.

Accordingly, developers build their aggregation systems and web portals by customizing the selecting the D-NET services they need, configuring them to match local requirements (e.g., metadata formats to be harvested and generated), combining them in a LEGO-fashion to match the workflows of interest, possibly building new D-NET services to add missing functionality, and finally distributing and orchestrating services on a network to tune and automate robustness and availability control.

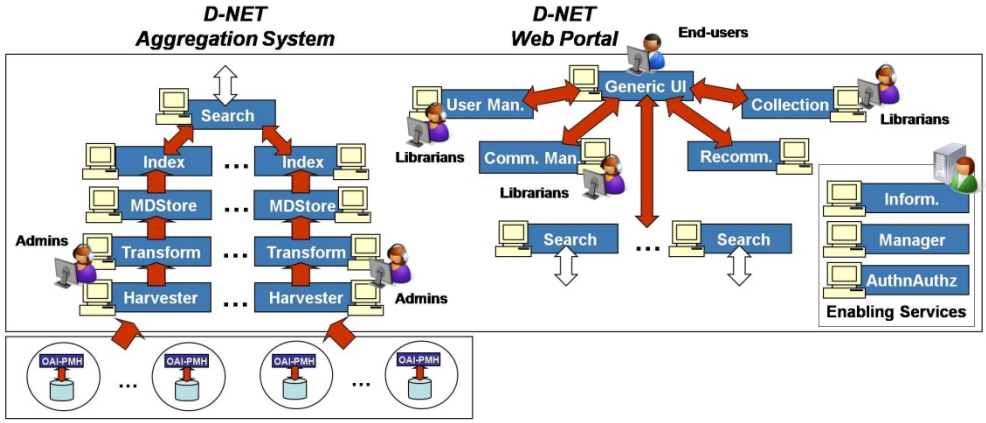

Figure 2 – Examples of aggregation systems and web portal settings.

For example Figure 2 shows how an aggregation systems may run services in multiple instances so as to build information spaces whose records are replicated or segmented (by vertical or horizontal partitioning) at different sites. Services can be combined through customized automated workflows. For instance, OAI-PMH harvester services could fetch and deliver records to transformation services and then to MDstore services, or instead directly to MDstore services if no transformation is needed. Similarly, portals consist of a combination of services, which appropriately combined can form a variety of web portals. Web portal services are modularly independent from aggregation systems and can automatically adapt to the metadata format they support (e.g., provide advanced search on the metadata fields, offer search on a subset of the fields) or be configured to access only a subset of it. Given a D-NET information space, hence D-NET aggregation system, the time needed to set up a D-NET portal ranges from an hour (e.g., Google-like portal) to a day, depending on the complexity.

3. D-NET's uptake

D-NET is currently used to operate the DRIVER Infrastructure[4] of Open Access European repositories, the European Film Gateway infrastructure[5] of European Film Archives, the Spanish-Recolecta infrastructure[6] of Spanish repositories and the Slovenian infrastructure[7] of Slovenian repositories. In addition, it is the software that will be used to operate the OpenAIRE EC project system[8] and the HOPE infrastructure[9] of the Heritage of People in Europe EC Project. Finally, a number of organizations, e.g., Greek, Bulgarian, Indian and Chinese national consortia, have shown their interest and are today establishing networks for supporting their national installations.

The DRIVER Infrastructure. The DRIVER Infrastructure maintains the European Information Space of Open Access publications. The system populates MDstore services by harvesting Dublin Core (DC) metadata from more than 250 Open Access repositories for a total of beyond 2,200,000 metadata records. Harvested records are transformed into DRIVER Metadata Format (DMF) records. DMF model captures (*i*) bibliographical information (*ii*) full-text extracted from PDFs reachable from URLs in the DC records, and (*iii*) provenance information, i.e., details of the repository of origin. Manager service workflows are configured to monitor and maintain 3 distributed replicas of the DC and DMF metadata records relative to the workflow of one repository in MDstore and index services. The DRIVER Information Space numbers two web portals: the main DRIVER portal, offering all functionalities and operating over the whole information space, and the Belgium national portal, offering an advanced search over the subset of records harvested from Belgium repositories.

The EFG Infrastructure. The European Film Gateway infrastructure aggregates records relative to compound objects from 14 film OAI-PMH archives in Europe, for a total of 300,000+ compound objects. Object records are harvested and converted into a common EFG compound object record format, describing film objects, with related person and audio/video objects. EFG records are in turn transformed into ESE metadata records, in order to be exported through an OAI-PMH Publisher Service to the harvester application of the Europeana project.[10] The EFG infrastructure has one portal, built by the external company *[Init]*, interfacing with the information space through the search service SRW standard APIs.

Acknowledgments. This work would have not be possible without the inestimable day and night work of the D-NET development team: Michele Artini, Alessia Bardi (ISTI-CNR, Italy), Marek Horst, Jaroslaw Wypychowski (ICM, Poland), Elena Nicolaki, Alexandros Mouzakidis, Thanos Papapetrou, Vassilis Stoumpos, (University of Athens, Greece), Marek Imialek, Jochen Schirrwagen, and Friedrich Summann (Bielefeld University, Germany); and the invaluable experience of Dr. Donatella Castelli, Dr. Wolfram Horstmann, Prof. Yannis Ioannidis and Dr. Wojtek Sylwestrzak. Research partially supported by the INFRA-2007-1.2.1 Research Infrastructures Program of the European Commission as part of the DRIVER-II project (Grant Agreement no. 212147).

[4] http://search.driver.research-infrastructures.eu,
[5] http://www.europeanfilmgateway.eu
[6] http://search.recolecta.driver.research-infrastructures.eu
[7] http://search.slovenia.driver.research-infrastructures.eu
[8] http://www.openaire.eu
[9] Project under EC negotiation, starting in April 2010
[10] http://www.europeana.eu