PARTHENOS

Pooling Activities, Resources and Tools
for Heritage E-research Networking,
Optimization and Synergies

D6.2 Report on services and tools

AUTHORS: Alessia Bardi
George Bruseker
Matej Durco
Marc Kemps-Snijders
Matteo Lorenzini
Maria Theodoridou

DATE 28 April 2017

PARTHENOS is a Horizon 2020 project funded by the European Commission. The views and opinions expressed in this publication are the sole responsibility of the author and do not necessarily reflect the views of the European Commission.

HORIZON 2020 - INFRADEV-4-2014/2015:
Grant Agreement No. 654119

PARTHENOS

Pooling Activities, Resources and Tools for Heritage E-research Networking, Optimization and Synergies

NAME OF THE DELIVERABLE

Report on Services and Tools

Deliverable Number	D6.2
Dissemination Level	PUBLIC
Delivery date	28 April 2017
Status	Final

Authors	Alessia Bardi George Bruseker Matej Durco Marc Kemps-Snijders Matteo Lorenzini Maria Theodoridou
Contributors	Sheena Bassett Athanasios Karasimos Go Sugimoto

Project Acronym	PARTHENOS
Project Full title	Pooling Activities, Resources and Tools for Heritage E-research Networking, Optimization and Synergies
Grant Agreement nr.	654119

Deliverable/Document Information

Deliverable nr./title	D6.2 Report on Services and Tools
Document title	Report on Services and Tools
Author(s)	Alessia Bardi Matej Durco Marc Kemps-Snijders Matteo Lorenzini
Dissemination level/distribution	Public

Document History

Version/date	Changes/approval	Author/Approved by
V 0.1 2017-04-03	Initial version	Alessia Bardi
V 0.2 2017-04-05	Description of the methodology	Alessia Bardi
V 0.3 2017-04-12	Section for T6.2	Alessia Bardi
V 0.4 2017-04-19	Integrated description of X3ML engine and 3M Editor tools in section T6.2	George Bruseker Maria Theodoridou Alessia Bardi
V 0.5 2017-04-21	Integrated suggestions from Go Sugimoto. Integrated Sections T6.3, T6.5 and the NERLIX use case	Alessia Bardi Matej Durco Matteo Lorenzini Marc Kemps-Snijders
V 1.0 2017-04-22	Added figure of the high-level architecture of the Parthenos Content Cloud. Fixed crosslinks for T6.3, T6.5 and	Alessia Bardi

	NERLIX sections. Draft shared with Athanasios for review.	
V 1.1 2017-04-26	English corrections and formatting changes by Sheena Bassett. Updated description for X3ML tools. Added missing images (screenshots of X3ML toolkit and architecture image for resource discovery tools)	Alessia Bardi Maria Theodoridou George Bruseker
V 1.1 2017-04-27	Reviewed version from Athanasios	Athanasios Karasimos
V1.2 2017-04-28	Integrated Athanasios' revisions	Alessia Bardi
V1.3 2017-04-30	Corrected linebreaks in 5.1	Matej Durco

Table of content

Index of Figures

Index of Tables

1. Executive Summary

The objectives of WP6 are:

1. To set up a set of tools and services enabling cross-discipline interoperability;
2. To share tools and services across different communities.

This document reports on the status of actions carried out towards achieving those two objectives in the context of three tasks:

Task 6.2 - Tools and services enabling interoperability

This task addresses the provision of interoperability tools/services to the PARTHENOS research communities and contributes to the first objective of WP6. The provision status of tools and services for interoperability is summarised in Table 1 (page 12).

Task 6.3 - Sharing specialized tools

This task addresses the provision of tools/services for manipulating, presenting and publishing similar data within the disciplinary domains (or possibly outside) and contributes to the second objective of the WP. A selection of specialised services that will be considered for integration in the PARTHENOS infrastructure are provided in Table 3 (page 30) and the options to perform the integration are discussed.

Task 6.5 - Resource discovery tools

This task will deliver a number of tools for the discovery of resources available in the PARTHENOS ecosystem, thus contributing to the second objective of the WP with an API and GUI for the cross-communities and cross-discipline discovery of PARTHENOS resources. An overview of resource discovery tools is given in Section 5.3.

This deliverable is an interim report. The final version of the report will be delivered in M45 as D6.4 report/deliverable.

Outline of the report

The document is organised into five main sections. Section 2 describes the methodology adopted by T6.2 and T6.3 for the selection and the delivery of relevant tools and services to the PARTHENOS community. Section 3 describes the integration status of services and tools for interoperability into the PARTHENOS infrastructure (T6.2). Section 4 discusses the available options for integrating specialised tools into the PARTHENOS infrastructure and presents a selection of tools that that will be considered for integration (T6.3). Section 5 describes requirements and desiderata for setting up advanced resource discovery tools (T6.5). Section 6 concludes the deliverable with the description of the NERLIX use case, which will be used to test, and prototypically synthesize the various technical aspects of the PARTHENOS endeavour - resource aggregation & discovery, metadata mapping, integration of processing and visualisation services.

2. Methodology

Tasks 6.2 and 6.3 share a common methodology for the selection, assessment, design, implementation, evaluation and deployment of the selected tools and services. The flowchart in Figure 1 describes the sequence of steps that are performed by T6.2 and T6.3:

1. **Identification of a service or tool**: a service or tool is considered a candidate for integration in the PARTHENOS infrastructure if: (a) it is used by a significantly wide or strategically placed community, and (b) it addresses a requirement expressed by the research communities in the PARTHENOS consortium.
2. **Technical Assessment**: the selected service/tool is assessed as to whether it needs to be consolidated, extended, or integrated with additional tools in order to ensure: (a) it correctly addresses the requirement(s) for which it was selected in the first phase, and (b) the feasibility of its integration in the PARTHENOS infrastructure.
3. **User Evaluation**: after the technical assessment, the service/tool is made available to members of T2.3, who can verify the usability and usefulness of the service/tool and, if necessary, propose amendments. In the case of amendments, the service/tool is returned to the technical assessment phase, where the suggested changes are discussed for technical and resource feasibility and possibly implemented.
4. **Deployment in the PARTHENOS infrastructure:** after a positive evaluation, the tool is ready to be deployed in the PARTHENOS infrastructure. When deployed, the service/tool is available for use to all interested members of the PARTHENOS consortium. Users of the service/tool can report bugs or usability issues (that may have not been identified or encountered in the technical assessment and/or in the evaluation phase) and the service/tool is returned to the technical assessment phase where the proper fixes can be applied.

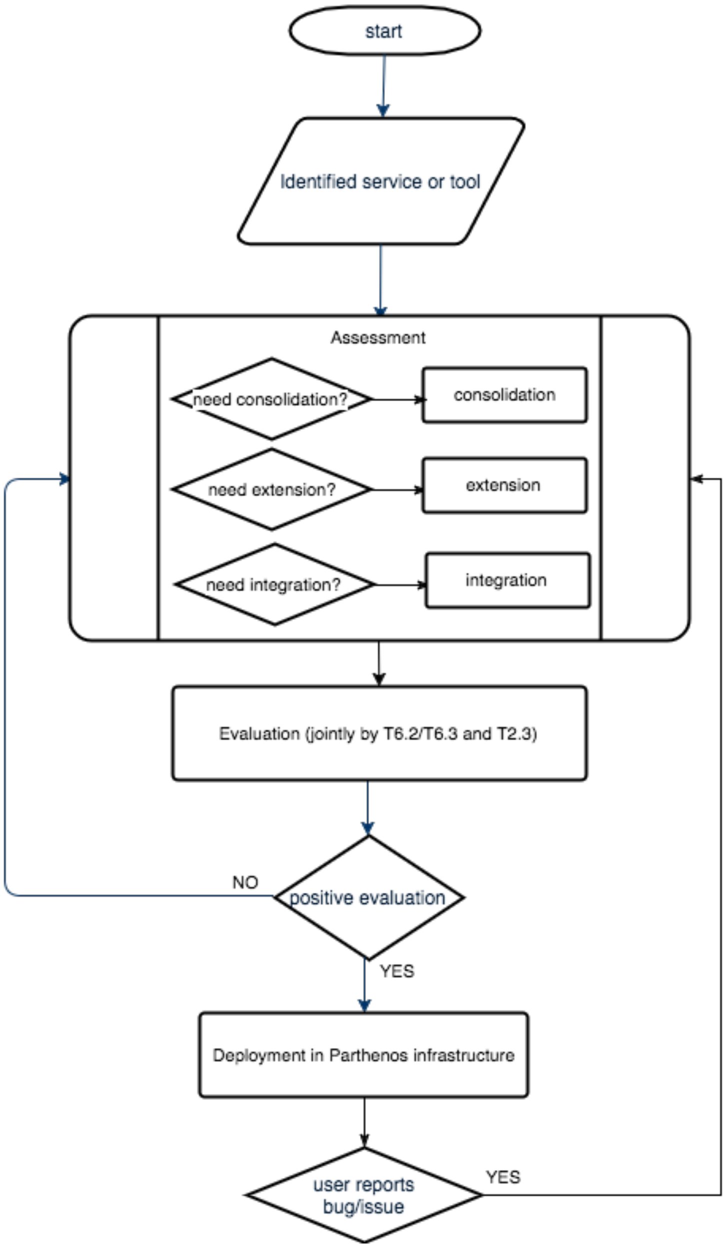


Figure 1: Methodology adopted by T6.2 and T6.3

3. Task 6.2: tools and services enabling interoperability

Task 6.2 addresses the provision of interoperability tools/services to the PARTHENOS research communities and follows the methodology described in Section 2 (see Figure 1). Table 1 summarizes the services and tools that have been already identified relevant for the PARTHENOS community and their integration stage into the PARTHENOS infrastructure.

Table 1 T6.2: tools and services enabling interoperability

Service/Tool	Description	Responsible partner	Addressed requirement(s)	Integration stage
X3ML toolkit	A set of open source components that assist the definition of mappings from XML to PARTHENOS Entities Model RDF for information integration.	FORTH	Metadata experts should be able to define the mappings to the PARTHENOS Entities model.	Deployed
D-NET Software Toolkit	Enabling framework for the realization and operation of aggregative metadata infrastructure	CNR-ISTI	Set-up of a PARTHENOS aggregator to enable cross-RI search and browse	Evaluation
X3ML engine	Transformation engine capable of applying mappings defined via the 3M Editor.	FORTH	Automatic application of mappings on the PARTHENOS side	Evaluation: integrated with D-Net

Metadata Cleaner	Service for the harmonization of values according to controlled vocabularies	CNR-ISTI	Adoption of common controlled vocabularies	Evaluation: integrated with D-Net
Metadata Inspector	GUI for the visualization of transformed metadata records and detection of “uncleaned” records	CNR-ISTI	Metadata quality control	Evaluation: integrated with D-Net

This set of tools are used in an integrated environment for the realisation of the PARTHENOS Content Cloud framework (see Figure 2 and deliverable D6.1 [1]).

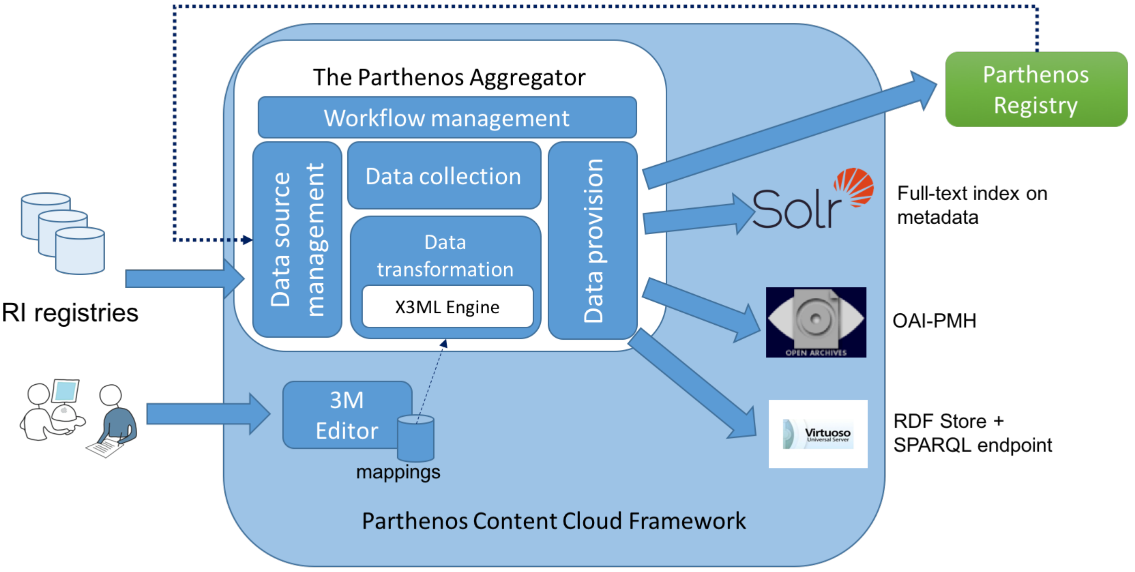
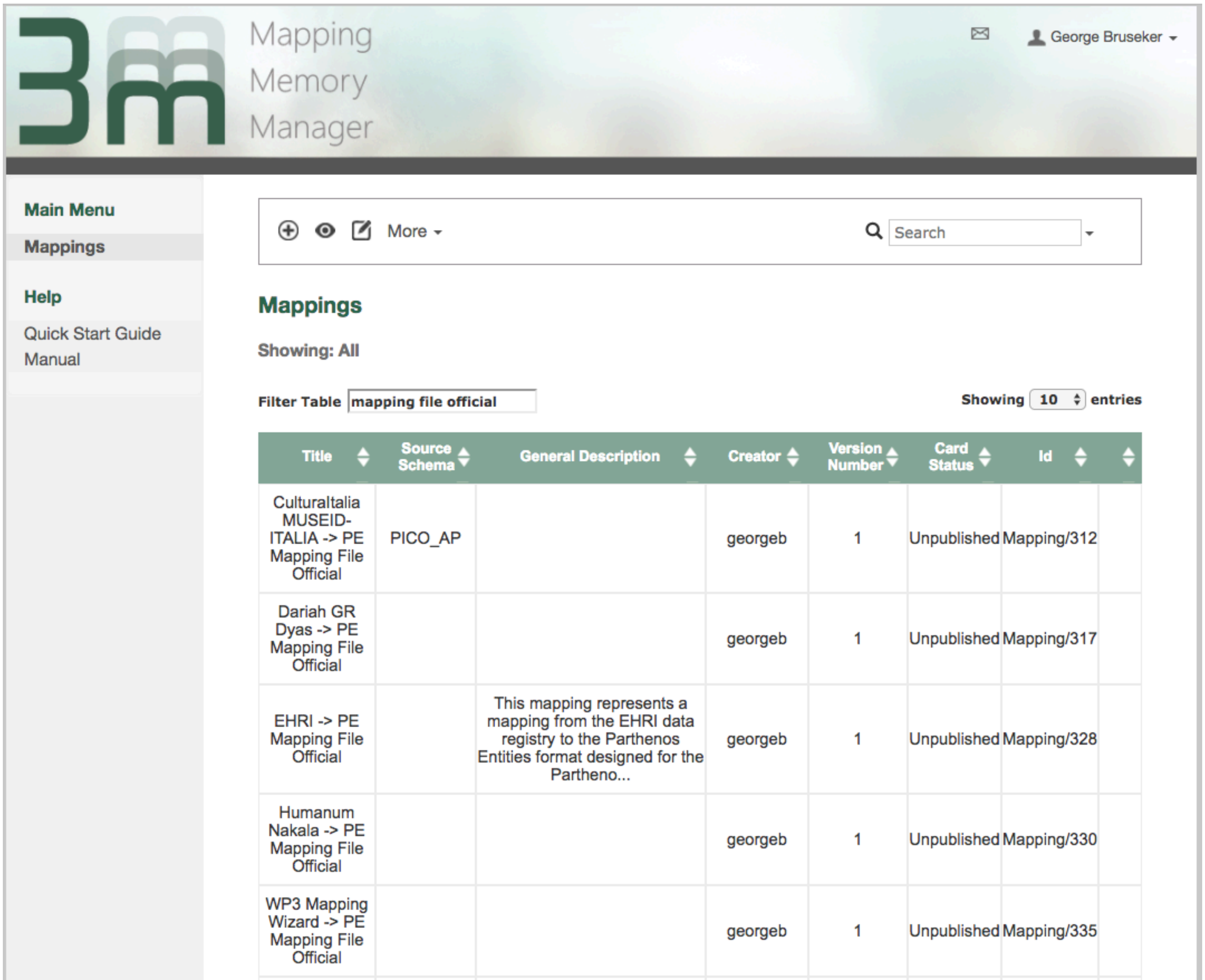


Figure 2: High Level Architecture of the Parthenos Content Cloud Framework

3.1. X3ML Toolkit

Description

The X3ML Toolkit [6] consists of a set of software components that assist the data provisioning process for information integration. It follows the SYNERGY Reference Model, a rich and comprehensive Reference Model for a better practice of data provisioning and aggregation processes, adapted from ongoing work of the CIDOC CRM SIG. The key components of the toolkit are: (a) 3M, the Mapping Memory Manager (see Figure 3), (b) the 3M Editor, and (c) the X3ML Engine.



Mappings

Showing: All

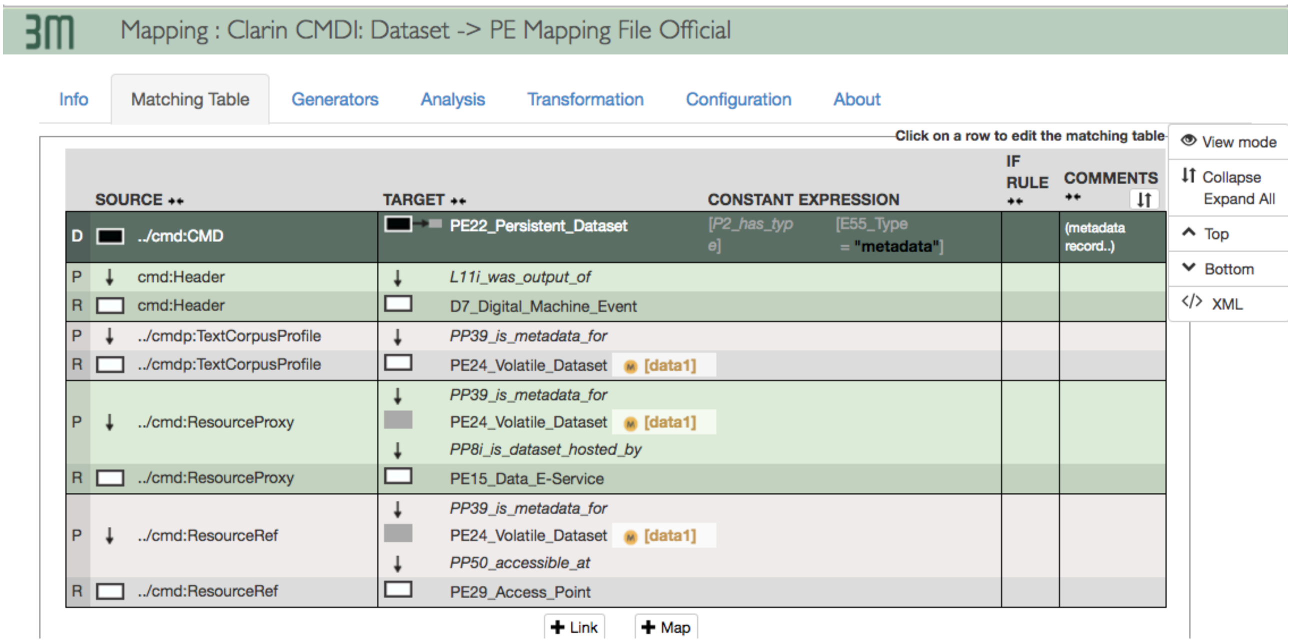
Filter Table mapping file official

Showing 10 entries

Title	Source Schema	General Description	Creator	Version Number	Card Status	Id	
Culturaltalia MUSEID-ITALIA -> PE Mapping File Official	PICO_AP		georgeb	1	Unpublished	Mapping/312	
Dariah GR Dyas -> PE Mapping File Official			georgeb	1	Unpublished	Mapping/317	
EHRI -> PE Mapping File Official		This mapping represents a mapping from the EHRI data registry to the Parthenos Entities format designed for the Partheno...	georgeb	1	Unpublished	Mapping/328	
Humanum Nakala -> PE Mapping File Official			georgeb	1	Unpublished	Mapping/330	
WP3 Mapping Wizard -> PE Mapping File Official			georgeb	1	Unpublished	Mapping/335	

Figure 3: 3M Mapping Memory Manager interface

The X3ML Toolkit is a web application suite containing several software sub-components that exploit several external services. Its main functionality is to assist users during the mapping definition process, using a human-friendly user interface and a set of sub-components that either suggest or validate the user input. Figure 4, Figure 5 and Figure 6 show the GUI of the 3M Editor for the definition and test of mappings.

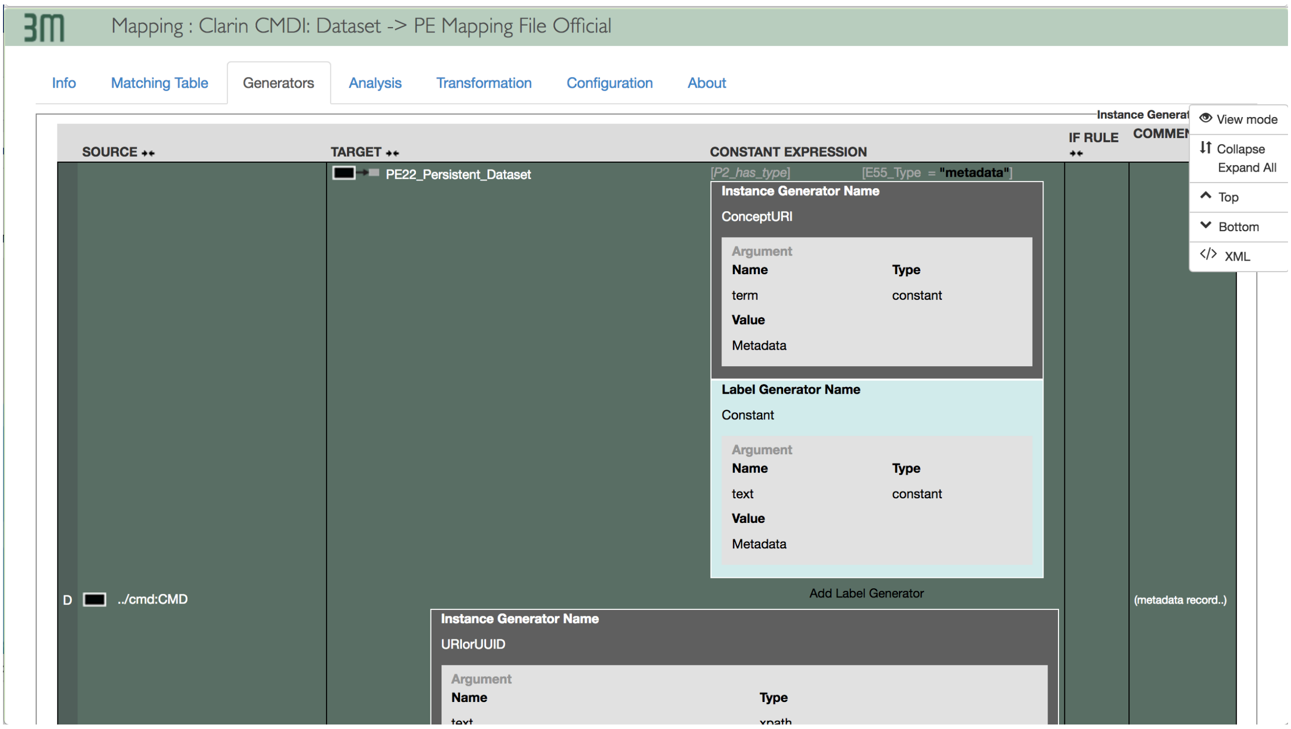


Mapping : Clarin CMDI: Dataset -> PE Mapping File Official

Click on a row to edit the matching table

	SOURCE	TARGET	CONSTANT EXPRESSION	IF RULE	COMMENTS
D	../cmd:CMD	PE22_Persistent_Dataset	[P2_has_type] [E55_Type = "metadata"]		(metadata record..)
P	cmd:Header	*L11i_was_output_of*			
R	cmd:Header	D7_Digital_Machine_Event			
P	../cmdp:TextCorpusProfile	*PP39_is_metadata_for*			
R	../cmdp:TextCorpusProfile	PE24_Volatile_Dataset [data1]			
P	../cmd:ResourceProxy	*PP39_is_metadata_for* PE24_Volatile_Dataset [data1] *PP8i_is_dataset_hosted_by*			
R	../cmd:ResourceProxy	PE15_Data_E-Service			
P	../cmd:ResourceRef	*PP39_is_metadata_for* PE24_Volatile_Dataset [data1] *PP50_accessible_at*			
R	../cmd:ResourceRef	PE29_Access_Point			

Figure 4: 3M Editor declarative mapping interface



Mapping : Clarin CMDI: Dataset -> PE Mapping File Official

Instance Generat

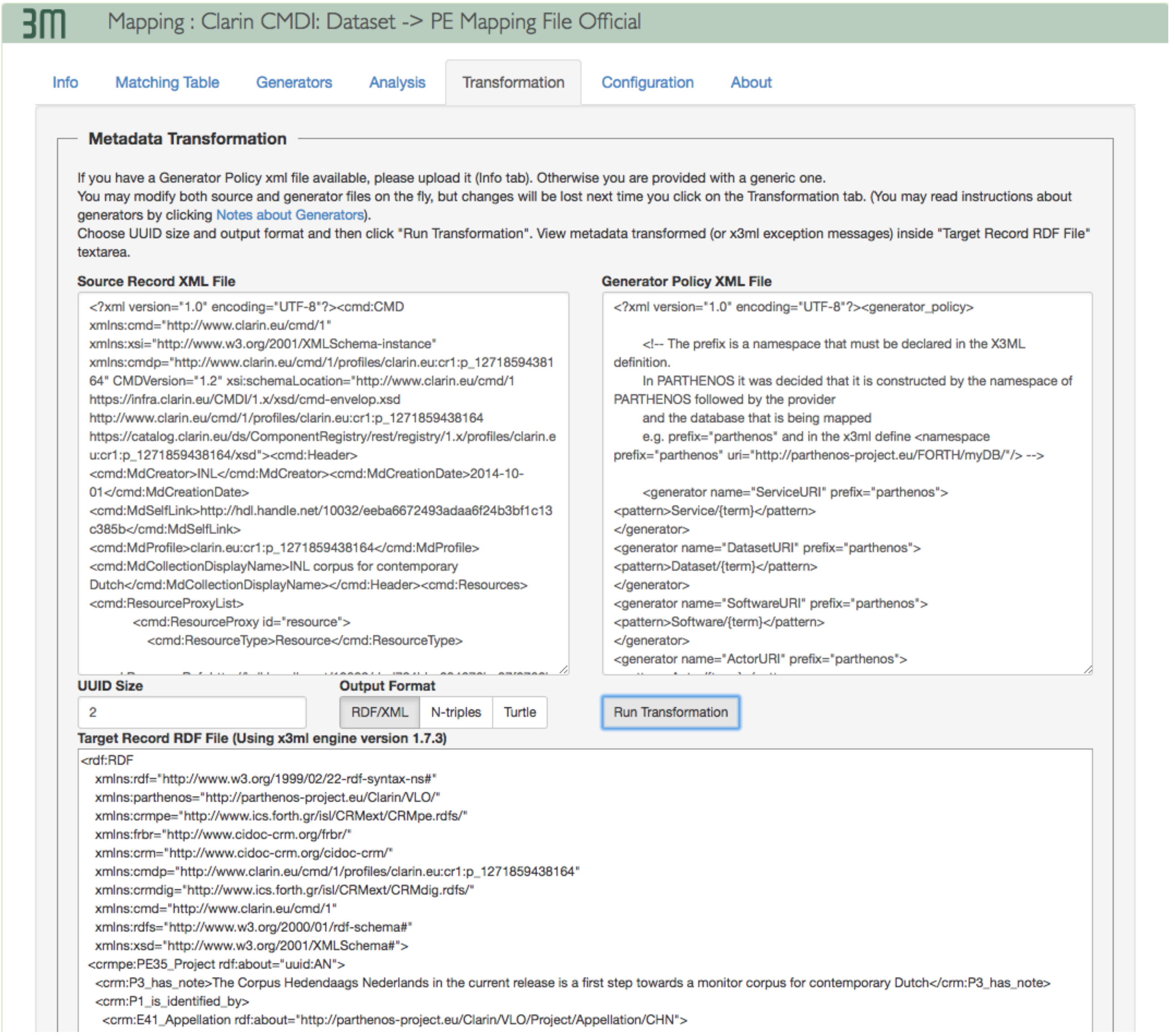
	SOURCE	TARGET	CONSTANT EXPRESSION	IF RULE	COMMEN
D	../cmd:CMD	PE22_Persistent_Dataset	[P2_has_type] [E55_Type = "metadata"]		(metadata record..)

Instance Generator Name
ConceptURI
Argument
Name: term — Type: constant
Value: Metadata

Label Generator Name
Constant
Argument
Name: text — Type: constant
Value: Metadata

Instance Generator Name
URIorUUID
Argument
Name: text — Type: xpath

Figure 5: 3M Editor instance generator interface



Metadata Transformation

If you have a Generator Policy xml file available, please upload it (Info tab). Otherwise you are provided with a generic one.
You may modify both source and generator files on the fly, but changes will be lost next time you click on the Transformation tab. (You may read instructions about generators by clicking Notes about Generators).
Choose UUID size and output format and then click "Run Transformation". View metadata transformed (or x3ml exception messages) inside "Target Record RDF File" textarea.

Source Record XML File

```
<?xml version="1.0" encoding="UTF-8"?><cmd:CMD
xmlns:cmd="http://www.clarin.eu/cmd/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:cmdp="http://www.clarin.eu/cmd/1/profiles/clarin.eu:cr1:p_12718594381
64" CMDVersion="1.2" xsi:schemaLocation="http://www.clarin.eu/cmd/1
https://infra.clarin.eu/CMDI/1.x/xsd/cmd-envelop.xsd
http://www.clarin.eu/cmd/1/profiles/clarin.eu:cr1:p_1271859438164
https://catalog.clarin.eu/ds/ComponentRegistry/rest/registry/1.x/profiles/clarin.e
u:cr1:p_1271859438164/xsd"><cmd:Header>
<cmd:MdCreator>INL</cmd:MdCreator><cmd:MdCreationDate>2014-10-
01</cmd:MdCreationDate>
<cmd:MdSelfLink>http://hdl.handle.net/10032/eeba6672493adaa6f24b3bf1c13
c385b</cmd:MdSelfLink>
<cmd:MdProfile>clarin.eu:cr1:p_1271859438164</cmd:MdProfile>
<cmd:MdCollectionDisplayName>INL corpus for contemporary
Dutch</cmd:MdCollectionDisplayName></cmd:Header><cmd:Resources>
<cmd:ResourceProxyList>
          <cmd:ResourceProxy id="resource">
             <cmd:ResourceType>Resource</cmd:ResourceType>
```

Generator Policy XML File

```
<?xml version="1.0" encoding="UTF-8"?><generator_policy>

      <!-- The prefix is a namespace that must be declared in the X3ML
definition.
      In PARTHENOS it was decided that it is constructed by the namespace of
PARTHENOS followed by the provider
      and the database that is being mapped
      e.g. prefix="parthenos" and in the x3ml define <namespace
prefix="parthenos" uri="http://parthenos-project.eu/FORTH/myDB/"/> -->

      <generator name="ServiceURI" prefix="parthenos">
<pattern>Service/{term}</pattern>
</generator>
<generator name="DatasetURI" prefix="parthenos">
<pattern>Dataset/{term}</pattern>
</generator>
<generator name="SoftwareURI" prefix="parthenos">
<pattern>Software/{term}</pattern>
</generator>
<generator name="ActorURI" prefix="parthenos">
```

UUID Size: 2

Output Format: RDF/XML | N-triples | Turtle

Target Record RDF File (Using x3ml engine version 1.7.3)

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:parthenos="http://parthenos-project.eu/Clarin/VLO/"
   xmlns:crmpe="http://www.ics.forth.gr/isl/CRMext/CRMpe.rdfs/"
   xmlns:frbr="http://www.cidoc-crm.org/frbr/"
   xmlns:crm="http://www.cidoc-crm.org/cidoc-crm/"
   xmlns:cmdp="http://www.clarin.eu/cmd/1/profiles/clarin.eu:cr1:p_1271859438164"
   xmlns:crmdig="http://www.ics.forth.gr/isl/CRMext/CRMdig.rdfs/"
   xmlns:cmd="http://www.clarin.eu/cmd/1"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
  <crmpe:PE35_Project rdf:about="uuid:AN">
   <crm:P3_has_note>The Corpus Hedendaags Nederlands in the current release is a first step towards a monitor corpus for contemporary Dutch</crm:P3_has_note>
   <crm:P1_is_identified_by>
    <crm:E41_Appellation rdf:about="http://parthenos-project.eu/Clarin/VLO/Project/Appellation/CHN">
```

Figure 6: 3M Editor transformation test facility

All the components of the X3ML Toolkit have been developed as open source components in the context of the projects CultureBrokers, ARIADNE[1], KRIPIS – POLITEIA[2], and VRE4EIC[3]. More specifically, the X3ML Toolkit components have been released under the European Union Public License whereas the X3ML engine has been released under the Apache 2 licence. FORTH (Foundation for Research & Technology – Hellas) is the main developer of the X3ML Toolkit and was supported by DelvingBV (www.delving.eu) in the initial implementation of the X3ML engine.

[1] ARIADNE: http://www.ariadne-infrastructure.eu

[2] KRIPIS – POLITEIA: http://politeia.ims.forth.gr

[3] VRE4EIC: https://www.vre4eic.eu

The X3ML Toolkit supports the data aggregation process by providing mechanisms for data transformation and URI generation. Mappings are specified using the X3ML mapping definition language, an XML-based, declarative, and human readable language that supports the cognitive process of a mapping in such a way that the generated maps can be collaboratively created and discussed by domain experts with little to no IT knowledge.

Unlike XSLT, that is comprehensible only by IT experts with training, the X3ML mapping definition language is designed from the ground up to be understood equally by IT experts as by non-technical domain experts whose data is the subject of transformation. This enables a domain expert to verify the semantics of a mapping by reading and validating the schema matching. This model carefully distinguishes between mapping activities carried out by the domain experts, who know and provide the data, from the activities of the IT technicians, who actually implement data translation and integration solutions. Usually, schema matching is used to describe the process of identifying that two different concepts are semantically related. This allows the definition of the appropriate mappings to be used as rules for the transformation process. However, a common problem is that the IT experts do not fully understand the semantics of the schema matching - it's not their data - and the domain experts do not understand how to use the technical solutions for creating mappings. For this reason, the X3ML Toolkit relies on two distinct components which separate task of schema matching from the URI generation processes. The schema matching can be fully performed by the domain expert and the URI generation by the IT expert, therefore solving the bottleneck that requires the IT expert to fully understand the mapping. Furthermore, this approach keeps the schema mappings between different systems harmonized since their definitions do not change, in contrast to the URIs that may change between different institutions and are independent of the semantics. Moreover, this approach completely separates the definition of the schema matching from the actual execution. This is important because different processes might have different life cycles; in particular, the schema matching definition has a different life cycle compared to the URI generation process. The former is subject to changes more rarely compared to the latter.

The development of the X3ML toolkit attempts to support the need to involve and record the knowledge of the data producers and ensure the best possible semantic and contextual mapping of cultural data.

The X3ML toolkit is a “community web application”. While it supports security for editing files (a user can edit a mapping only if the creator of the mapping authorized him/her), it allows the whole community to view a mapping file (with the 3M Mapping Memory Manager, Figure 3). This means that one can look at other users mapping definitions to learn techniques and see approaches to different types of information. It is also possible to share editing rights with others. Although only one person can edit at a time, this provides some capability to share the mapping process with someone else online – to suggest and consult, or to make corrections.

Identification

The PARTHENOS consortium expressed the requirement of setting up an aggregative infrastructure to collect metadata from the research infrastructures participating in the consortium. In order to build a homogenous information space, where metadata records collected from different providers are harmonised according to the PARTHENOS Entities model (CIDOC PE), providers must define metadata mappings from their registries to CIDOC PE. The X3ML toolkit supports the definition of mapping with a user-friendly GUI and has been used with success in several projects (e.g. ARIADNE, ResearchSpace, VRE4EIC).

Technical assessment

The X3ML toolkit was consolidated in order to be installable on Linux machines (the initial version worked only on Windows).

User evaluation

The X3ML toolkit has been made available to all PARTHENOS members.

The RIs participating in the PARTHENOS consortium were invited to a training workshop in Rome, October 2016, where FORTH members presented the PARTHENOS Entities model, the mapping techniques, the usage of the X3ML toolkit and provided the participants with guidelines for setting up and making a complete mapping. A set of mapping exercises have been implemented and made available to all the partners.

Following the training, seven partners (ARIADNE, CLARIN, CNR, CulturaItalia, DARIAH-GR, EHRI, Huma-Num) started using the toolkit in order to define the mapping of their source schema to the PARTHENOS Entities Model. FORTH continuously supports the partners during their mapping efforts and collects their feedback in order to resolve bugs and enhance the usability of the graphical user interface. Already a new version of the GUI is ready and will soon be deployed in PARTHENOS.

Deployment in the PARTHENOS infrastructure

The X3ML toolkit is deployed in the PARTHENOS infrastructure and integrated in the PARTHENOS Virtual Research Environment. In April 2017, the 3M Editor has:

- 43 Total Mappings
    - 9 Sample Mappings
    - 34 Active Mappings
- 25 Mappings for Registry
    - 20 with Semantic Mapping
    - 5 with official generators policy

3.2. D-NET Software Toolkit

Description

The D-NET Software Toolkit[4] (D-NET for brevity) is a service-oriented framework specifically designed to support developers at constructing custom aggregative infrastructures in a cost-effective way [2]. D- NET offers data management services capable of providing access to different kinds of external data sources, storing and processing information objects compliant to any data models, converting them into common formats, and exposing information objects to third-party applications through several standard access APIs. D-NET services are obtained by encapsulating advanced and state-of-the-art open source products for data storage, indexing, and processing – such as PostgreSQL, MongoDB, Apache Solr, and Apache HBase – in order to serve broad application needs. Most importantly, D-NET offers infrastructure enabling services that facilitate the construction of domain-specific aggregative infrastructures by selecting and configuring the needed services and easily combining them to form autonomic data processing workflows. The combination of out-of-the-box data management services and

[4] The D-NET Software Toolkit: http://www.d-net.research-infrastructures.eu/

tools for assembling them into workflows makes the toolkit an appealing starting platform for developers dealing with the realization of aggregative infrastructures.[5]

CNR-ISTI (Italy) is the main developer of D-NET, supported by teams from University of Athens (Greece), Athena Research Centre (Greece), and University of Bielefeld (Germany). D-NET is open source (Apache licence), developed in Java, based on Service-Oriented Architecture paradigms. Its first software release was designed and developed within the DRIVER and DRIVER-II EC projects (2006-2008). The motive driving its development was that of constructing the European repository infrastructure for Open Access repositories. The infrastructure had to harvest (tens of) millions of Dublin Core metadata records from hundreds of OAI-PMH repository data sources, harmonizing the structure and values of such records to form a uniform information space.

Identification

The PARTHENOS consortium expressed the requirement for setting up an aggregative infrastructure to collect metadata from the research infrastructures in the consortium, transform them according to the PARTHENOS data model to form a homogenous information space to expose to third-party consumers through a number of APIs.

D-NET offers out-of-the data management services and tools for assembly into workflows to facilitate the construction of domain-specific aggregative infrastructures like the one requested by the PARTHENOS consortium.

D-NET features several built-in plugins for the collection of metadata records (in XML, CSV, TSV and similar) via different exchange protocols (e.g. OAI-PMH, SFTP, FTP(S), HTTP(S), local file system). Additional plug-ins can be easily implemented and integrated if needed.

From the data processing point of view, D-NET features a Transformation Service capable of transforming metadata records from one format to another by applying mappings of different kinds (XSLTs, Groovy scripts, Java code, and D-NET transformation rules).

For the publishing of the generated uniform information space, D-NET can be configured to export metadata records via OAI-PMH and to index metadata records into a Solr index.

For the full list of features of D-NET please refer to [2].

[5] For a more detailed description of D-NET, its features, and its role in the PARTHENOS infrastructure please refer to D6.1, Section 2.5 [1].

The D-NET Software toolkit has been used with success in several EC projects for the provision of content to the European Cultural Heritage aggregator Europeana (e.g. EFG, EAGLE, HOPE), initiatives for Open Access (DRIVER, OpenAIRE) and national and international aggregators like CeON (Poland), La Referencia (South America), Recolecta (Spain), SNRD (Sistema Nacional de Repositorios Digitales, Argentina).

Technical assessment

During the technical assessment phase D-NET was consolidated and upgraded to Java 8. As described above, D-NET is a framework for the construction of aggregative data infrastructure. As such, part of the assessment phase focused on the selection of available data management services and the design of the data processing workflow to be set up for the PARTHENOS aggregator (which is a D-NET instance). The data processing workflow devised for PARTHENOS is shown in Figure 7. The workflow comprises two automatic sub-workflows: the *aggregation workflow* automatically collects input metadata records, transforms them according to a defined X3ML mapping, and makes the resulting RDF records available to metadata experts for inspection. If the metadata experts approve the results, then the *publishing workflow* is executed in order to officially publish the records. The publishing comprises: (1) the indexing of metadata on a Solr index - the Solr schema is defined in collaboration with T6.5 (see also Section 5.5); (2) the availability of an OAI-PMH export of the metadata (in CIDOC-CRM and oai_dc); (3) the update of the RDF store; and (4) the update of the PARTHENOS registry.

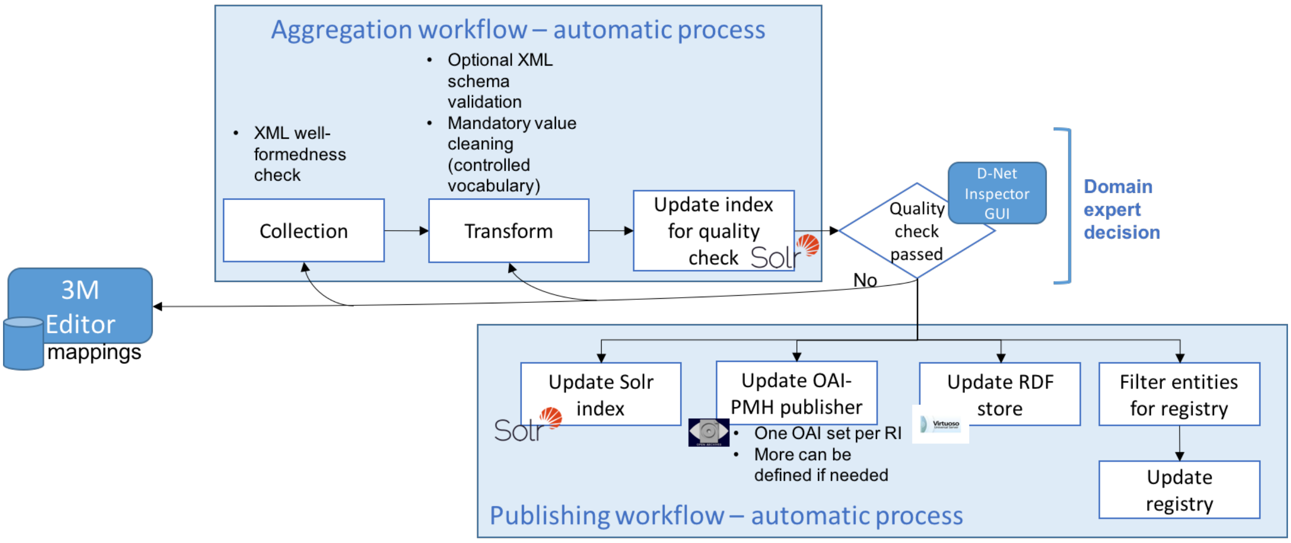

Aggregation workflow – automatic process
XML well-formedness check
Optional XML schema validation
Mandatory value cleaning (controlled vocabulary)
Collection
Transform
Update index for quality check
Solr
Quality check passed
D-Net Inspector GUI
Domain expert decision
No
3M Editor
mappings
Update Solr index
Solr
Update OAI-PMH publisher
One OAI set per RI
More can be defined if needed
Update RDF store
Virtuoso
Filter entities for registry
Update registry
Publishing workflow – automatic process

Figure 7: The workflow devised for the PARTHENOS aggregative infrastructure

An analysis of the APIs of the research infrastructures has been carried out to ensure that D-NET could collect metadata records from all the available sources. The analysis revealed that D-NET natively covers the majority of API protocols used by research infrastructures. A summary of the analysis is available in Table 2.

Table 2: Analysis of the APIs of PARTHENOS research infrastructure with respect to the collection plugins natively available in D-NET

Research Infrastructure	Registry	Endpoint	Type	D-Net collector plugin	Action
ARIADNE	Ariadne Registry	SFTP server	SFTP with public key authentication	SFTP plugin	Plugin extended to support public key authentication.
CLARIN	Virtual Language Observatory	https://vlo.clarin.eu/resultsets/	XML dumps via HTTPS	HTTP plugin	
CENDARI	CENDARI Registry	Not yet available	OAI-PMH	OAI-PMH plugin	
Cultura Italia	CulturaItalia Catalogue	http://www.culturaitalia.it/oaiProviderCI/OAIHandler	OAI-PMH	OAI-PMH plugin	
DARIAH GR/ΔΥΑΣ	ΔΥΑΣ Organizations and Collections Registry	http://registries.dyas-net.gr/en/developer	OAI-PMH	OAI-PMH plugin	
	LRE MAP	http://www.resourcebook.eu/lremap/owl/instances/	XML dumps via HTTP	HTTP plugin	
	METASHARE	Not yet available	XML dumps. Exchange protocol to be agreed.	One of: HTTP, SFTP, FTPS, local file system	
HUMA-NUM	Isidore	http://www.rechercheisidore.fr/sparql	SPARQL endpoint	Not available. SPARQL plugin under development.	
	Nakala	https://www.nakala.fr/	SPARQL endpoint		

Research Infrastructure	Registry	Endpoint	Type	D-Net collector plugin	Action
EHRI	EHRI ICA Archives	https://portal.ehri-project.eu/units/<ID>/export	HTTP API	Dedicated collector plugin under development.	

In order to be able to process the X3ML mappings produced by metadata experts of research infrastructures via the 3M Editor tool, D-NET has been extended by integrating the X3ML Engine, a transformation engine developed by FORTH capable of processing X3ML mappings (see Section 3.3).

New software modules are under implementation and testing for the integration with the PARTHENOS registry.

User evaluation

The instance of D-NET for PARTHENOS has been deployed on a beta server where developers and infrastructure administrators from CNR-ISTI and T6.2 are performing technical and usability tests on the data processing workflow and the new plugins devised in the assessment phase.

It is worth noting that the D-NET instance is not supposed to be used directly by end-users, but only by the administrators of the PARTHENOS infrastructure. Metadata experts of the research infrastructures (i.e. those who defined the mappings with the 3M Editor) will use the D-NET Metadata Inspector to verify the metadata quality before the records are officially published. We, therefore, plan a major involvement of T2.3 members in the evaluation of the Metadata Inspector (see Section 3.5).

Deployment in the PARTHENOS infrastructure

To be done.

3.3. X3ML Engine

Description

The X3ML Engine is a transformation engine that realizes the transformation of the source records to the target format. The engine takes as input the source data (currently in the form of an XML document), the description of the mappings in the X3ML mapping definition file and the URI generation policy file and is responsible for transforming the source records into a valid RDF document which corresponds to the input XML file, with respect to the given mappings and policy.

The X3ML Engine implementation was initiated by the CultureBrokers project, co-funded by the Swedish Arts Council and the British Museum. Currently, FORTH is maintaining the implementation and has received funding from the projects ARIADNE, KRIPIS POLITEIA, and VRE4EIC.

The source code is available on Github[6] and is licensed under the Apache License 2.0. A ready to use artefact is provided via the FORTH ISL Maven Repository[7].

Identification

The adoption of the X3ML Engine is required for the execution of mappings generated via the 3M Editor.

Technical assessment

The X3ML Engine is a Java library and the technical assessment focused on the feasibility of its integration into the PARTHENOS aggregator based on the D-NET software toolkit. In order for the integration to be possible, FORTH and CNR-ISTI collaborated to update the library with new Java methods and consolidate the code by upgrading some old library dependencies.

User evaluation

The X3ML Engine has been integrated into D-NET and it is not a tool that is directly accessible by end-users. The evaluation phase is instead carried out by members of T6.2 from CNR-ISTI.

From the functional point of view, no issues have been found.

From the performance point of view, CNR-ISTI performed some preliminary tests about the transformation speed. The first transformation workflows for evaluation purposes have been run on April 14th 2017 over a random set of 19,000 records collected from

[6] X3ML Engine source code: https://github.com/isl/x3ml

[7] FORTH ISL Maven Repository: http://www.ics.forth.gr/isl/maven/

ARIADNE, using a non-complete 3M mapping. The transformation of each record took an average of 125 milliseconds (max: 1219 ms, min: 35 ms). The execution of the workflow on the full set of records from ARIADNE (about 1,6 million of metadata records) took 3 days and 11 hours to complete, with an average transformation time per record of 184 ms (including X3ML transformation, value cleansing via Metadata Cleaner, storage of transformed records into the D-NET metadata store). Further tests will have to be performed with finalised mappings and the results will be shared with FORTH to discuss whether the application of the mapping can be optimised to reduce the execution time.

Deployment in the PARTHENOS infrastructure

The X3ML Engine is integrated into D-NET, hence its deployment in the infrastructure is implicit with the deployment of D-NET.

3.4. Metadata Cleaner

Description

The Metadata Cleaner is a D-NET service that harmonises values in metadata records based on a set of thesauri. A D-NET thesaurus consists of a controlled vocabulary that is a list of authoritative terms together with associations between terms and their synonyms. Data curators – typically based on instructions from data providers and domain experts – are provided with user interfaces to create/remove vocabularies and edit them to add/remove new terms and their synonyms. Given a metadata format, the metadata cleaner service can be configured to associate the metadata fields to specific vocabularies. The service, provided records conforming to the metadata format, processes the records to clean field values according to the given associations between fields and vocabularies. Specifically, field values are replaced by a vocabulary term only if the value falls in the synonym list for the term. If no match is found, the field is marked as ‘invalid’. The ‘invalid’ marker is exploited by the Metadata Inspector (see Section 3.5) to highlight non-cleaned records and suggest the update of D-NET vocabularies or the update of the values in the input record.

Identification

The inclusion of the Metadata Cleaner was not initially planned, because the value cleaning can also be performed by defining specific rules in the X3ML mappings. However, the PARTHENOS Consortium agreed that a mechanism to ensure that all controlled fields (i.e. metadata fields whose values must comply to a controlled vocabulary) contain valid values was needed. At this goal, CNR-ISTI proposed to include in the D-NET instance of PARTHENOS the Metadata Cleaner so that, in the transformation phase of the aggregation workflow (see Figure 7), each record is transformed by the X3ML Engine and, afterwards, the controlled fields are further cleaned by the Metadata Cleaner. If a controlled field cannot be harmonised according to the proper vocabulary, the record is marked in order to enable inspection via the Metadata Inspector.

Technical assessment

The Metadata Cleaner is natively integrated in D-NET and no issues had to be addressed at the technical level.

User evaluation

During this phase, the configuration of the Metadata Cleaner will be evaluated by T2.3 members (with the support of T6.2). Specifically, the Metadata Cleaner should be configured to clean all fields that are defined as controlled fields in the PARTHENOS data model. In order to proceed and finalize the configuration of the Metadata Cleaner, T6.2 needs input from WP4 (Standardization) for the choice of the standard vocabularies to adopt for the controlled fields of the PARTHENOS data model. If a standard vocabulary cannot be found or adopted, then custom controlled vocabularies will be created.

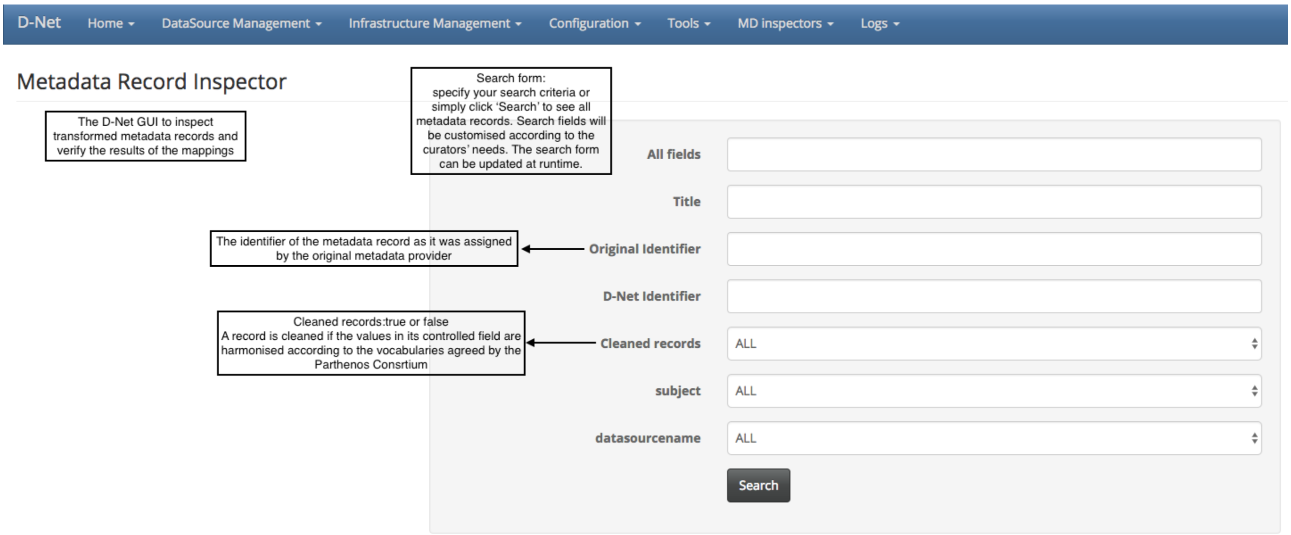
Deployment in the PARTHENOS infrastructure

The Metadata Cleaner is integrated into D-NET, hence its deployment in the infrastructure is implicit with the deployment of D-NET.

3.5. Metadata Inspector

Description

The Metadata Inspector is a Web GUI integrated into D-NET that provides data curators with an overview of the information space, where they can search and browse records and verify the correctness of the transformation phase (e.g. no mapping mistakes or semantic inconsistencies, no records marked as ‘invalid’ by the Metadata Cleaner). Upon positive verification of the records in the information space, data curators can inform the PARTHENOS infrastructure administrators that the records can be published (see ‘Publishing workflow’ in Figure 7). Figure 8 and Figure 9 show two screenshots of the Metadata Inspector.



Metadata Record Inspector

The D-Net GUI to inspect transformed metadata records and verify the results of the mappings

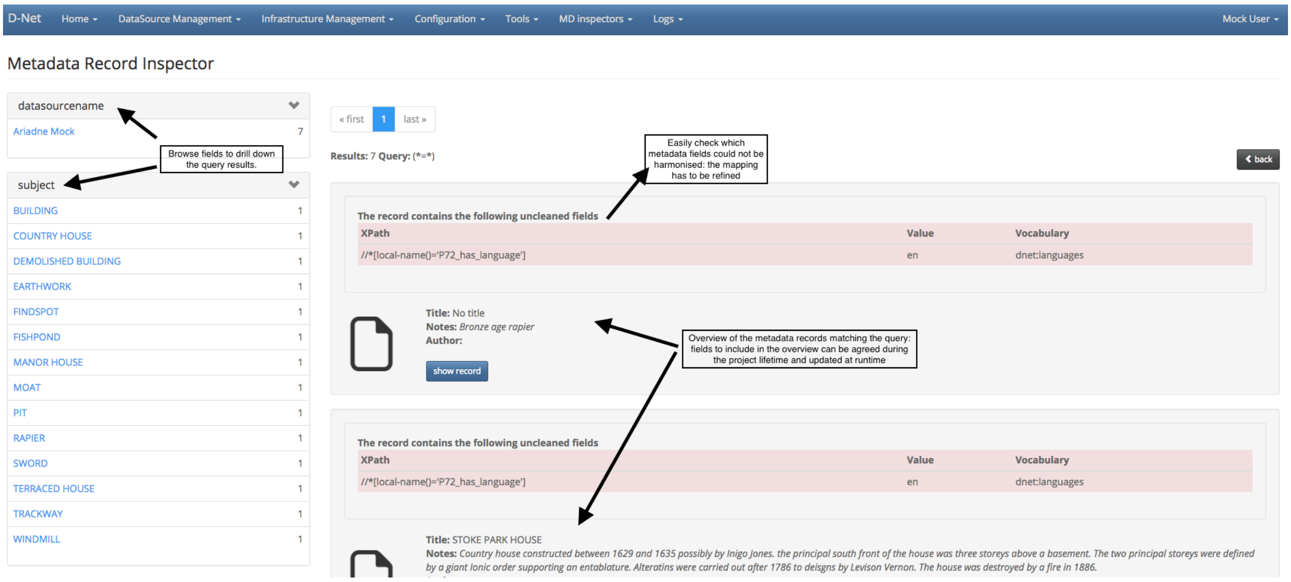
Search form: specify your search criteria or simply click ‘Search’ to see all metadata records. Search fields will be customised according to the curators’ needs. The search form can be updated at runtime.

The identifier of the metadata record as it was assigned by the original metadata provider

Cleaned records:true or false
A record is cleaned if the values in its controlled field are harmonised according to the vocabularies agreed by the Parthenos Consrtium

All fields	
Title	
Original Identifier	
D-Net Identifier	
Cleaned records	ALL
subject	ALL
datasourcename	ALL

Figure 8: The main search form of the Metadata Inspector



Metadata Record Inspector

Browse fields to drill down the query results.

datasourcename	
Ariadne Mock	7

subject	
BUILDING	1
COUNTRY HOUSE	1
DEMOLISHED BUILDING	1
EARTHWORK	1
FINDSPOT	1
FISHPOND	1
MANOR HOUSE	1
MOAT	1
PIT	1
RAPIER	1
SWORD	1
TERRACED HOUSE	1
TRACKWAY	1
WINDMILL	1

Results: 7 **Query:** (*=*)

Easily check which metadata fields could not be harmonised: the mapping has to be refined

The record contains the following uncleaned fields

XPath	Value	Vocabulary
//*[local-name()='P72_has_language']	en	dnet:languages

Title: No title
Notes: *Bronze age rapier*
Author:

Overview of the metadata records matching the query: fields to include in the overview can be agreed during the project lifetime and updated at runtime

The record contains the following uncleaned fields

XPath	Value	Vocabulary
//*[local-name()='P72_has_language']	en	dnet:languages

Title: STOKE PARK HOUSE
Notes: *Country house constructed between 1629 and 1635 possibly by Inigo Jones. the principal south front of the house was three storeys above a basement. The two principal storeys were defined by a giant Ionic order supporting an entablature. Alteratins were carried out after 1786 to deisgns by Levison Vernon. The house was destroyed by a fire in 1886.*

Figure 9: The Metadata Inspector shows metadata records with “uncleaned” fields

Identification

The PARTHENOS consortium expressed the requirement for having a tool to check the results of the transformation and the vocabulary harmonisation.

Technical assessment

The Metadata Inspector is a native feature of D-NET and no issues had to be addressed at the technical level.

User evaluation

Metadata experts of the research infrastructures (i.e. those who defined the mappings with the 3M Editor) will use the Metadata Inspector to verify the metadata quality before the records are officially published. We, therefore, plan a major involvement of T2.3 members in this phase, from whom we expect feedback, especially regarding the structure of the GUI and the metadata fields that must be searchable and browsable to support the record inspection at its best.

Deployment in the PARTHENOS infrastructure

The Metadata Inspector is integrated into D-NET, hence its deployment in the infrastructure is implicit with the deployment of D-NET.

4. Task 6.3: Sharing specialized tools

This task addresses the provision of tools/services for **manipulating**, **presenting** and **publishing** similar data within the disciplinary domains (or possibly outside). The task will follow the same approach as the *task* 6.2 for the selection, the assessment, the design, implementation, evaluation and deployment of the tools/services. For example*,* these include advanced visualization; annotation of texts/images/video/3D; workflow description; and more. The task will harmonize such tools and make them available after adapting or making the changes necessary for generalization. The results are documented in a section of the interim report on tools, D6.2, and likewise in the final report, D6.4, incorporating the amendments made after testing in D2.3.

Based on an exemplary use case, we have developed a blueprint for service providers (SP) describing the different options on how to integrate their services into the D4Science infrastructure.

4.1. Overview of tools and services

There is a hoist of specialized services provided in the context of the RIs, a selection of which is provided in Table 3, where services have been classified as either editing, processing, visualization, data hosting or discovery services/tools. Of these, the first three categories are of primary interest. Data hosting services represented in this table may be considered special cases of (data) resources and provide access to underlying concept definitions and (partial) schema information. Discovery services refer to RI specific discovery services. While these are of interest to the specific RI community, the data content of these discovery services should also be available through the PARTHENOS Resource Registry and/or Content Cloud representing a superset/union of all the individual datasets. Hence, the primary focus of the integration task will be on editing, processing and visualisation tools. WebAnno, for example, is a general-purpose web-based collaborative annotation tool for a wide range of linguistic annotations. Processing services, such as Weblicht, provide text processing facilities and GeoBrowser is used for visualizations of spatial and temporal data sets.

The services listed in the table represent just a small illustrative fraction of all available services. Hitherto, preliminary investigations revealed that CLARIN provides a few hundred services/tools, DARIAH around a few dozen and other RI's a few each.

Table 3: A selection of services offered by research infrastructures

Name	Provider	RI	Task
GeoBrowser	SUB, Göttingen / DARIAH-DE	DARIAH	Visualisation
WebLicht	SfS, Tübingen / CLARIN-D	CLARIN	Processing
Transkribus	University of Innsbruck	CLARIN/DARIAH	Edit
Datasheet editor	SUB, Göttingen / DARIAH-DE	DARIAH	Edit
Nederlab	Meertens Institute	CLARIN	Visualisation
GraphViewer	ACDH-OEAW	CLARIN/DARIAH	Visualisation
Digivoy	Uni Würzburg / DARIAH-DE	DARIAH	Visualisation
VLO	CLARIN ERIC	CLARIN	Discovery
Component Registry	CLARIN ERIC	CLARIN	Data Hosting
Concept Registry	CLARIN ERIC	CLARIN	Data Hosting
Dariah Collection Registry	SUB, Göttingen / DARIAH-DE	DARIAH	Discovery
Dariah-DE generic search	SUB, Göttingen / DARIAH-DE	DARIAH	Discovery
WebAnno	UKP, TU Darmstadt / CLARIN-D	CLARIN	Edit
TokenEditor	ACDH-OEAW	CLARIN/DARIAH	Edit
ARBIL	MPI, Nijmengen	CLARIN	Edit
COMEDI	Uni Computing, CLARINO	CLARIN	Edit

CENDARI Archival Directory	DARIAH	CENDARI	Processing
B2SHARE	EUDAT	EUDAT	Data Hosting

4.2. Service Integration - User stories

Based on the classification described above, we propose a more general approach towards service integration based on the following user stories:

1. As a Service Provider (SP), I would like to integrate my **editing** service/application into the PARTHENOS platform, allowing users to edit available resources directly inside a custom VRE;
2. As a SP, I would like to integrate my **processing** service into the PARTHENOS platform, allowing users to send resources available in the platform for processing to the service (and storing/making available results back into the platform (workspace));
3. As a SP, I would like to integrate my **visualisation** application into the PARTHENOS platform, allowing users to visualise eligible resources directly inside a custom VRE.

These user stories should provide an easy to use entry point for service/tool providers wishing to integrate their tools in the PARTHENOS infrastructure. Each of these user stories is to be accompanied by concrete implementation examples ('recipes') that assist service/tool providers in their integration process. Rather than relying on the gCube documentation, which is rather technical in its descriptions, these recipes should provide concrete examples of how to integrate existing software and services into the PARTHENOS infrastructure.

4.3. Modes of integration

Services/tools are launched from a hosting environment. In gCube terms, two types of hosting node typologies are distinguished, gCube-based and SmartGears based. With gCube-based nodes services are launched and operated on nodes of the D4Science infrastructure. An advantage of this approach is that it provides the possibility to automatically scale up. The SmartGears option requires installation.

gCube's enabling layer distinguishes between two types of hosting node typology: Software-as-Resource (SaR) and Container-as-Resource (CaR). In the SaR approach, a piece of software is considered a gCube resource if it can be managed in the gCube infrastructure. The CaR approach focuses on software that can be used over the network. They are typically deployed within containers, e.g. Tomcat, and may be registered to become gCube Hosting Nodes (gHN).

This flexible setup allows service providers to either keep offering their services on their own servers, adapted to become gHN, or to transfer the software to the facilities of the D4Science infrastructure. With this second option, the D4Science infrastructure is responsible for ensuring the availability of server-side capacities (storage, computing power, network bandwidth). Starting initially with the gHN approach for integration and testing and then moving to the central infrastructure for production is a very feasible and sensible scenario.

For most services delivered through the RIs, the CaR option appears to provide the most viable option as this allows for deployment and maintenance of the services from within the participating RIs. Participation of a container requires installation of SmartGears on the host, registration of the host with the infrastructure and acquisition of authorization tokens.

In order for hosted services to operate in the PARTHENOS infrastructure, the easiest manner appears to be Web Processing Service (WPS) compliant. In almost all cases, this requires a proxy to be deployed alongside the service or a wrapper encapsulating the service, with the benefit of not having to change the service (or the underlying software).

Tools requiring user interface interactions may be modified to gCube **portlets**. These portlets are the main constituents of a gCube-based portal offering selected gCube facilities via a GUI. In all the integration scenarios, the comprehensive documentation accompanying the gCube framework represents invaluable input.

In summary, the following options are available for service integration:

1. Local gCube hosting node - register service, allows monitoring, authentication/authorisation.

    a. Service as servlet (Tomcat or another servlet container)
    b. Not a servlet (requires a wrapper)

2. Access cloud storage
    a. Read
    b. Write
3. Run service hosted on the nodes of D4Science infrastructure (with the possibility to scale out)

From user's perspective, the integrated services become part of a custom on-demand VRE, as a main organizing unit for putting together users, resources and services around a specific research question or task. Still under discussion is whether PARTHENOS will work towards a small number of bigger VREs with a broad function spectrum or multiple smaller, more specialized, ones. The NERLIX VRE (see Section 6) combines the strategy of service integration ('How do I integrate my own service') with the end user perspective as expressed through the D2.1 Use cases and requirements [7].

5. Task 6.5: Resource discovery tools

Based on the Joint Resource Registry, this task will deliver a number of tools for resource discovery, such as faceted searches and localized searches concerning space and time (e.g.: which resources are available with information about Malta? Which resources cover the 7th century AD?), or advanced querying services for the discovery of similar or related resources (e.g. which datasets exist that have a similar content to this one? Which collections contains objects related to the contents of this collection? Which tools exists that have a functionality similar to this one?), offered both via a GUI and an API, including Linked Data publication of the registry. Advanced browsing services will also be considered, and implemented following a user-centred design approach. Such tools will be accessed through the project portal, through collaboration between this task and Task 8.1. The task outcomes are reported in D6.2 (draft) and D6.4 (final).

The task T6.5 is to deliver tooling for resource discovery (RD-tools), operating as exploitation stack on the large amount of data to be aggregated by the PARTHENOS infrastructure. These tools will form an interlinked array, offering different, complementary views on the data, covering the data space comprehensively, and that will allow the users to navigate/explore, but also to search along a number of dimensions.

5.1. The PARTHENOS Data Space

Crucial for the discussion on resource discovery is a clear understanding of the overall data space to be covered by the discovery tools to be made available to the users.

PARTHENOS positions itself “within the broad sector of Linguistic Studies, Humanities, Cultural Heritage, History, Archaeology and related fields”. Assuming a general goal of maximizing recall, PARTHENOS will try to obtain as much (meta)data from these fields as possible, mainly by means of close collaboration with participating research infrastructures to (re-)aggregate the metadata made available by the individual RIs.[8]

[8] Metadata offered by the RIs is normally already aggregated from the original content providers

CIDOC-PE, the target schema of PARTHENOS for mapping the different metadata formats onto one common semantic framework, is by design a minimal schema, trying to capture and represent main entities, and neglect all domain-specific details. While this is a sensible setup in order to establish a generic stable “data backbone” reference for identities and relations between them, from the discovery point of view, domain-specific information is equally relevant to the users and will need to be taken into consideration.

We would like to point out the following general dimensions/concerns with respect to the data:

- **Format heterogeneity**

  Given the situation of individual RIs, PARTHENOS will need to deal with huge heterogeneity of data and metadata formats. By abstracting to a the concept of “digital object”, the PARTHENOS infrastructure can certainly handle basic (CRUD) operations on any kind of data, but in order to offer a real added value to the users, format-specific functionalities for viewing, editing, processing need to be offered. This issue also represents a specific challenge for resource discovery, prompting a need for format harmonisation beyond the minimal metadata and for a robust handling of sparse search space, with potentially many properties applying only to a limited number of resources. In this respect, it is important to keep in mind that most providers and aggregators offer metadata in multiple formats, with most detailed information on the side of the providers.
- **Metadata vs. data**

  While metadata is openly available, data is often not, which is in contrast to user’s general interest for the content itself rather than the metadata. While offering links for downloading the content can be considered a baseline, the ambition has to be to offer the access to data in a more transparent/seamless way, increasing the added value and efficiency gain. See discussion on metadata & content search in Section 5.4.1.
- **Granularity**

  Are only high-level collections described and represented in the system, or also individual collection items? While more detailed information allows for more precise search, at the same time processing, indexing and querying millions of fine-grained metadata records may pose a major load on the infrastructure, besides representing an extraordinary challenge for the user interface and for intelligent presentations of result sets, to prevent overwhelming users.

Last but not least, fine-granular metadata may not be available at all for many datasets.

- **Meta/data quality**

  The quality of meta/data has major influence on the discoverability of data (e.g. missing, misspelled or misplaced values hamper recall).

  Summarising, we face at least two "entropy problems/sources":

  - Information not being available in the source data
  - Information loss through translation to common target format

  For detailed discussion, see section 0.

- **Datasets, Software & Services**

  While the focus in resource discovery is primarily on datasets, PARTHENOS semantic framework equally represents also Software and Services and sizeable amounts of these kind of entities are available across the community. These should be considered in the resource discovery as well.

5.2. Requirements

This section presents a preliminary list of requirements on resource discovery tools with the central goal of offering a comprehensive entry-point to the huge aggregated dataset, PARTHENOS is about to generate. Comprehensive in terms of scope (datasets, data items being represented), depth (information available about individual data items), and functionality (available modes of interaction).

As a guiding principle in this context, see the InfoVis Mantra by Shneiderman (1996):

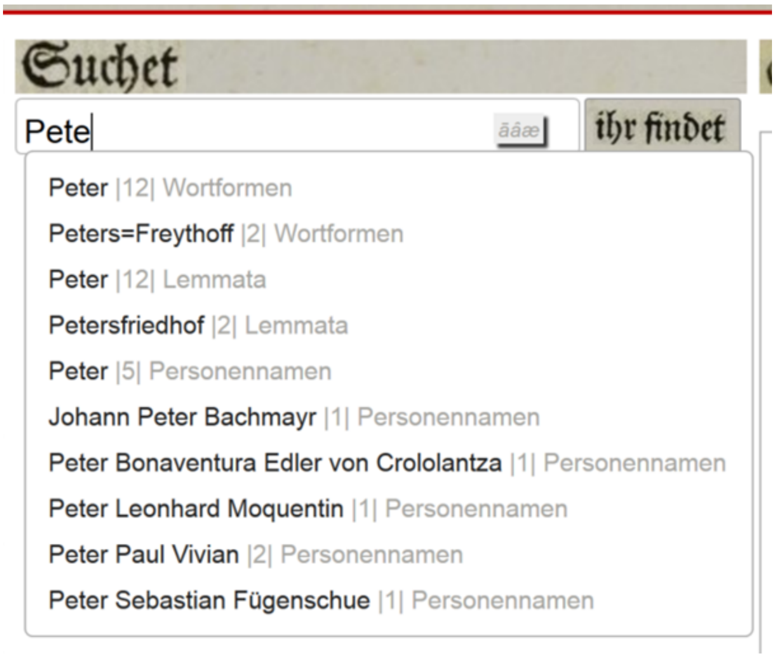
Overview first, zoom and filter, then details-on-demand [5]

1. **Combined faceted / index-based full-text search** (allowing for browsing, drilling down, as well as specific string-based queries).

   Nowadays, faceted search needs to be considered a minimum baseline, exposing the main facets/aspects of the data space as quantified lists of available values per facet, allowing the user to easily browse through, and drill down the explored dataset.

The faceted search is a powerful method to explore the dataset, which on the user interface side can be realized in a number of ways. Next to the traditional list of facets, it can be also plotted in a number of ways (like timeline, histogram, etc.) or the index values reused for supporting input in a simple search field through typed and quantified autocomplete.

2. **Full-text, Content-first search** (autocomplete/suggestions, ideally typed and quantified) well known from Google (or your browser).
   Examples: Wikidata (or Wikipedia), WolframAlpha.



Pete

Peter |12| Wortformen
Peters=Freythoff |2| Wortformen
Peter |12| Lemmata
Petersfriedhof |2| Lemmata
Peter |5| Personennamen
Johann Peter Bachmayr |1| Personennamen
Peter Bonaventura Edler von Crololantza |1| Personennamen
Peter Leonhard Moquentin |1| Personennamen
Peter Paul Vivian |2| Personennamen
Peter Sebastian Fügenschue |1| Personennamen

Figure 10: Example of typed and quantified suggestions in a content-first search

3. **Complex queries / Advanced search** (allow to query specific indexes, apply boolean operators, fuzzy search, regular expressions, joins).
   The harmonized metadata being available in RDF, SPARQL will be the natural default for full-fledged complex querying capabilities, but other query languages may be of use, such as Lucene/SOLR query language, or CQL.
   For further discussion see subsection 5.4.2.
4. **Metadata & content search**
   Ideally making accessible not just the metadata, but also (selected aspects of) the data, especially mentions of named entities, but also keywords, or full-text search, or even linguistic search over multiple annotation layers/tiers for appropriate resources.

This comes with major additional complexity and burden/load on the infrastructure (details in separate subsection 5.4.1).

5. **Semantic search** (esp. Named entities)

   Not just search for strings, but for entities, with a knowledge base in the background. Either the (meta)data is already semantically enriched, or one can employ query expansion on runtime. This opens a whole new chapter on curation and enrichment of the data being harvested and offered for querying
   (details in separate subsection 0).

6. **Handle arbitrarily deep hierarchies**

   Data being structured in oftentimes nested datasets/collection, we have to consider the hierarchy aspect, especially with respect to collection-level information/metadata when querying the item level information. For instance, the information about the project underlying the creation of given piece of data can be encoded in the metadata for the whole collection as opposed to being duplicated within every item's metadata
   In large-scale aggregators like PARTHENOS, Content Cloud is going to be one natural top-level of hierarchy which is constituted by the individual sources (content providers).[9]

7. **Handle arbitrary relations**

   By translating the individual records into entities with relationships among them, the resulting data set can be considered as a graph, with potentially a number of typed relations between the different paths. It is important to ensure that this additional structural level becomes available to the end user, by offering appropriate interaction/visualisation means, to navigate and query the underlying graph.

8. **Pointers to original location** [of data & metadata] (ideally all locations/identifiers)

   Ultimately, the users will want to not only discover, but also access and use the resources of interest. Therefore, links to the actual payload, or at least a clear indication of the mode of access must accompany any metadata representation of any given resource.

9. **Inform about available indexes**

   Examples of information needed to properly use the indexes are:

[9] Take also Europeana for comparison.

1. Explanatory remarks/definitions of individual indexes
2. Description of the source of individual index (i.e. which fields in original metadata contribute to given index)
3. Access to the distinct values covered by the index. This aspect is well covered thanks to the faceted search functionality.

10. **Invoke processing services (**on resources and result sets)

    This is a somewhat peculiar feature, not directly part of the resource discovery. It is motivated by the general idea of making available to the users an interactive VRE and assuming a default workflow of discovering resources not just for the sake of it, but to actually “do something with them”.

    From the technical point of view, this requires a kind of “basket”-functionality, where the user can select a set of resources (either explicitly, or implicitly by means of a query) and invoke operations on this arbitrary set.

    See also the NERLIX use case in section 6 for a planned implementation of this aspect.

11. **Harmonized access/API to heterogeneous sources**

    Naturally given by the aggregating setup. Actually, at the core of all the mapping and aggregation procedures central to PARTHENOS undertaking.

12. **Personalized Workspace**

    At least as important as extensive search capabilities as stated above is the possibility for a custom user-specific profile/mode, which is ideally building up through use. A ubiquitous example being the history-memory of the browsers.

13. **Harvesting/Download**

    Provide access to raw (original) metadata, both at the individual item level as well as the possibility to get all or query-defined subsets of metadata.

14. **Feedback**

    Allow for contextualized feedback anytime and make sure that it is received and acted upon.

5.3. Overview of Tools

Before we turn to the PARTHENOS-specific solution, we would like to have a brief look at the landscape of existing frameworks and solutions for resource discovery and especially visualisation.

Some of tools and services for discovery and visualisation provided by members of the PARTHENOS RI-communities are also already mentioned in section 4.1.

The discovery platforms of the individual RIs can serve as a natural point of inspiration, given that the PARTHENOS RD tools should ideally represent a superset/union in terms of scope/coverage and functionality.

The baseline for resource discovery, nowadays, is faceted and/or full-text search, with results sets presented as paged lists. While this is a necessary and useful functionality the user is used to, we would like to go well beyond this basic setup, with the focus towards Visual Analytics, i.e. strong adoption of information visualisation techniques. These can play a role in three distinct parts of the workflow:

- Support search - e.g. plotting available data points on a map, allowing to filter the dataset visually.
- Visualize result - e.g. all kinds of quantitative statistics over the metadata of the result set.
- Explore a specific resource - e.g. a resource being a graph representing the social network of Viennese society in 19th century, this is better approached via a dedicated tool for graph analysis and exploration

In the following listing, we, very selectively, mention a few existing visualisation frameworks and tools (not restricted to providers within consortium) grouped according to different types of visualisation. Further exploration is needed to see if these can be reused/integrated directly, or would rather only serve as source of inspiration.

5.3.1. Geospatial

- Pelagios: Pelagios is a digital classics network which connects heterogeneous data resources from the domain of Ancient World research: classical Greek and Roman literature, collections of coins, inscriptions, databases of museum items and archaeological artifacts, repositories of contemporary research articles, etc. To date, Pelagios lists more than 750,000 records, and is growing constantly.

- GeoBrowser: GeoBrowser is a tool for the visualization and analysis of temporal and special relations in the Humanities domain. GeoBrowser facilitates a comparative visualization of multiple requests and supports the display of data and its visualization in correlation between geographical spatial settings and corresponding points in time and time lapses. Researchers can thus analyse spatial-temporal relations of data and source collections and simultaneously establish correlations between these. Users can enter the data to be analysed with the aid of the Datasheet Editor developed by DARIAH-DE, refer to KML-, KMZ- or CSV-files available on the Internet or upload local files.

5.3.2. Tree/Graph/Network

- Gephi: Gephi is an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs. It can visualize directed, undirected, weighted, un-weighted, labelled, un-labelled, etc. supporting many network layout algorithms as well as accompanying graph manipulation functionality
- GraphViewer is a web application for visualizing and interactive exploration of graph data. It is implemented on top of JavaScript library **D3** the code available on **GitHub**. Being developed by one of the consortium partners, it would be a good candidate for integration, as potentially needed deployments could be realized easily.

5.3.3. Quantitative/Statistical

- R: is a free and cross platform software environment for statistical computing and graphics. R offers a comprehensive set of built-in functions and libraries for visualizations and present data such as gplot, Lattice and leafletR.
  - *Gplot* allows you to create graphs that represent both univariate and multivariate numerical and categorical data in a straightforward manner. Grouping can be represented by colour, symbol, size, and transparency.

    - *Lattice* attempts to improve on base R graphics by providing better defaults and the ability to easily display multivariate relationships. In particular, the package supports the creation of trellis graphs - graphs that display a variable or the relationship between variables, conditioned on one or more other variables.
    - *LeafletR*: represent a new package for the creation of a leafleft map in R. It is supported by RStudio and appears to provide the simplest, fastest way to represent interactive maps in R.
- *Matplotlib*: is a library for Python and its mathematics extension NumPY. Matplotlib is based on Matlab graphics command and is designed with the aim to create simple plots with just a few commands and export the representations to a number of formats, such as SVG, PS or PNG files.

Increasingly, complex frameworks/packages are available that cover a broad range of data types and visualisation techniques, just to name a few:

- Palladio: Palladio is a web based visualization tool for complex humanities data developed by the Humanities + Design lab at Stanford University. Palladio is designed for the easy visualization of humanities data as a map, graph, table, or gallery. It allows for the identification of patterns, clusters, and trends within data that may be difficult for an individual researcher interacting with the data to see. Palladio serves as a means of enhancing (not replacing) traditional qualitative humanities research methods. Data can be mapped, graphed to show network relationships, viewed and faceted as an interactive gallery, and more.
- D3.js: Data Driven document is a web-based data visualization library that utilizes JavaScript, HTML, SVG and CSS to create a variety of data visualizations. D3.js is a powerful library with a plethora of visualisation techniques implemented, e.g. TreeMap, Dendrogram, Sunburst, Chord Diagram, Flow Diagram, Force-Directed Graph Layout, and many more.
- Bokeh: is a Python interactive visualization and presentation of data libraries in modern web browsers. Bokeh allow the creation of interactive plots, dashboards, and data applications.

5.4. Architecture/Dependencies

In line with the overall architecture, discovery tools rely on the availability of search indices over the resources from individual RIs aggregated into the infrastructure (see Figure 11). This may be either covered by the minimal metadata represented in the Joint Resource Registry (JRR), or by more fine-granular and detailed information available in the Content Cloud. We expect the information available via the search indices of the Content Cloud to be a superset of the information in JRR and, therefore, assume the Content Cloud to be our primary source of information / endpoint.

This also implies that only information will be available for searching that has been extracted from the metadata and made available via the indices. In other words, any requirements on additional information (e.g. space, time coverage, named entities) available for querying/searching, needs to be percolated down to the Content Cloud indexing.

Furthermore, we assume that the information in the Content Cloud is available via well-defined interfaces. Specifically, via a synchronized combination of high-performance full-text search (Lucene/Solr), and a SPARQL endpoint on top of a triple store, nowadays quite a usual setup.

We also acknowledge that the mapping from original metadata to CIDOC-PE is by design lossy, trying to capture the main entities and the relationships among them, whilst neglecting a number of domain-specific attributes/features. The original metadata being available in the Content Cloud, it will be feasible to (conditionally) extend mappings and enrich the information in the JRR to introduce new, more specific indices. Domain or functional specific knowledge lays in specific sets of metadata in knowledge bases and can be accessed/discovered by hybrid search (combining query languages and datasets / knowledge bases).

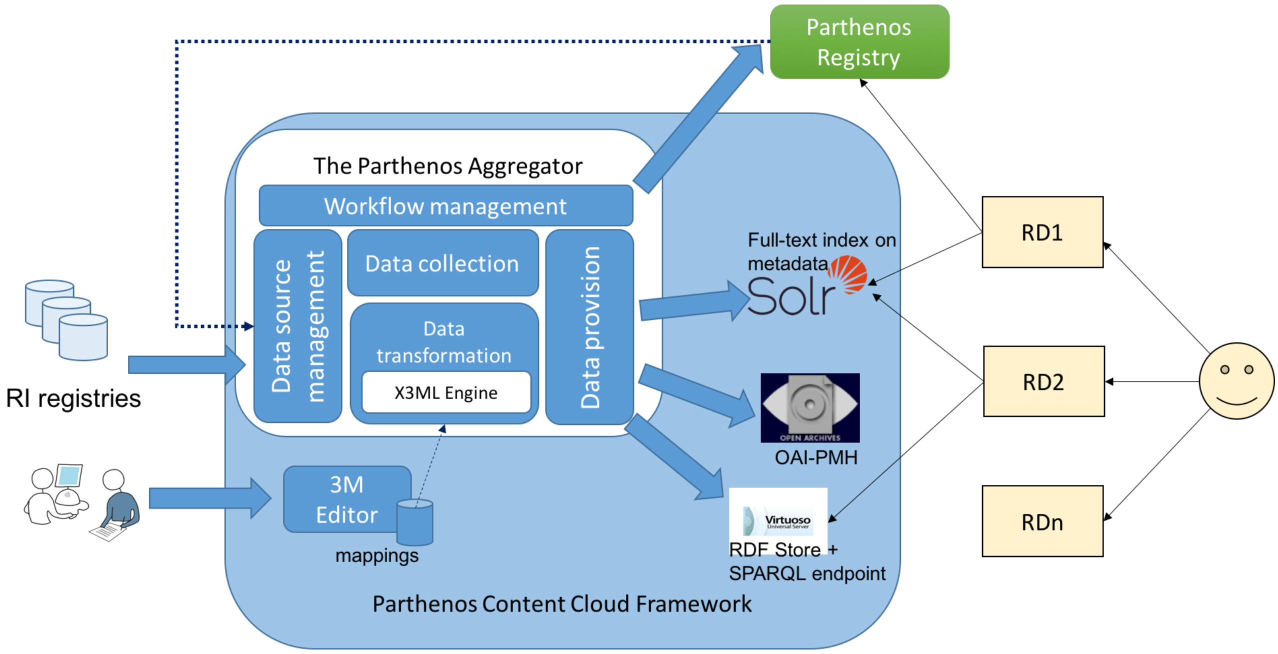


Figure 11: Resource discovery tools: using the PARTHENOS Content Cloud and the PARTHENOS registry

5.4.1. Metadata & content search

Until now, all the work on mapping done in WP5 is concentrating on providing a unified view of the PARTHENOS data space on the level of metadata, i.e. the descriptive information about the resources. This effort has already the potential to be very useful for the user. However, it would be interesting to also move to the level of the content of the resources themselves, i.e. to allow a combined metadata and content search. When in place, this would allow queries like mention of entity X (content query) in resources of a specific type, time of creation or project context (metadata query).

One major contribution of Parthenos' CIDOC-CRM based common semantic framework is the explicit distinction between the metadata record and the content resource, essentially making the metadata records "first-class citizens" that can be referred and related to. Such an explicit distinction is a necessary (but not sufficient) precondition for a combined metadata/content search.

Search in content comes with a number of complications, or additional challenges for the infrastructure: First, content is oftentimes not as readily available as the metadata. Subject to IPR restrictions, it may be impossible to pass the content for processing into a central index. Then, there is the issue of an even more heterogeneous format landscape at the content level and last, but not least, the potentially prohibitive size of a hypothetical resulting index.

Problem of centralized index - due to update frequency, size, licensing and other issues, not all (content) data is available for transmission to an external facility.

One approach to tackle these issues is a distributed setup, where content is not transferred to a central index, but stays on the side of the original content provider, who exposes an interface to query given content. A dedicated component, an aggregator, then allows the users to perform content queries on distributed resources offered by individual providers, by distributing user's query to the individual endpoints on the fly (query time), collecting back partial results and presenting these to the user as one merged results set. A precondition for such a complex setup is an agreement between the content providers and the aggregating infrastructure on a common protocol for such distributed querying mechanism. The viability of this approach is demonstrated by a number of real world scenarios, e.g. meta-search engines for booking flights or hotels.

One potential drawback of this approach is that it is only as good as its weakest (slowest, least expressive) component. This can be remedied to a certain extent by appropriate provisions on the side of the user interface (e.g. displaying partial results as they are being returned by individual endpoints; giving feedback to the user, about the status of the query processing).

CLARIN RI can contribute rich experience in this area, based on the so-called Federated Content Search (FCS) effort/initiative, in which individual partners agreed to offer a unified access point to search in textual resources (text corpora). In general, the gathered experience confirms that even though technically the task of providing an (additional) endpoint adhering to a specific agreed upon protocol is perfectly feasible, in practice it is a tedious challenge to:

a) find a common denominator with respect to the protocol definition (e.g. which query language to support, which complex query expressions beyond trivial keyword search can be realistically supported by at least the majority of the providers);
b) keep the distributed system operational (individual endpoints/partners may become (temporarily) unavailable).

Thus, to be of real value to the users, such a system has to be very tolerant and be able to gracefully cope with all kinds of errors inherent to such a distributed setup.

Also, exactly due to the overwhelming heterogeneity of the available data, FCS is currently restricted to serve/query textual data. This restriction maybe a sensible one in the context of the CLARIN project, but it would be, however, too reductionist/restrictive in the context of a project with such a broad domain and resource type/format coverage, like PARTHENOS.

5.4.2. Complex search

Even though experience of providers overwhelmingly shows that “expert search” is almost completely ignored by the users (1% using sometimes at best), with a Lucene-based setup it basically comes for free, so it should be considered and offered as well (even though it will be probably seldom used, these may be the most interesting findings.)

SPARQL endpoint can be considered as the complementary discovery/exploration means to the indexed full-text search. It offers most direct access and thus complete freedom in terms of navigating through the underlying knowledge graph, allowing to ask questions, not possible via the predefined search indices. As such, it comes with a somewhat steeper learning curve, but being a standard means to explore semantic web data, it needs to be considered as baseline, which increasingly many users will expect and use readily.

Next to a barebones SPARQL-endpoint requiring to formulate SPARQL queries, more user-friendly means of exploration become pervasive/used. Foremost is the interactive user interface harnessing the graph nature of RDF data that enables navigation through the knowledge graph translating to corresponding SPARQL queries in the background. The foremost example is LodLive.

A separate point would be the discussion on graphical user interfaces supporting construction of complex queries, of which many have been developed in the past and could be considered for integration/reuse.

5.4.3. Semantic search

Semantic search is used to denote different things, we mean here primarily the option to search for entities instead of string.

This requires a lot of work on curation/enrichment of the meta/data as, more often than not, entities in the source data are referred to with simple (string) values in the source format (esp. Actor, Service), as opposed to some globally unambiguous identifiers.

Another aspect is the use/integration of thesauri/vocabularies/reference resources into the discovery tooling, e.g. in the faceted search in order to improve recall and precision of the searches or also to help the users to perform and optimize the query thanks to the division into fundamental, high level categories (facets).

“Semantic search” can also mean trying to improve search by understanding the contextual meaning of the terms and trying to provide the most accurate answer from a given knowledge base.

However, thesauri are created in different languages, with different scopes and point of views and at different levels of abstraction. The mapping of two or more domain thesauri to an upper ontology like CIDOC-CRM is an attempt to solve the possible semantic discrepancy.

5.4.4. Curation/Quality of the (meta)data

It is a trivial but very true statement that the search/discovery capabilities are only as good as the underlying data. This is practically a universal issue in all harvesting/indexing/searching/discovery scenarios, contexts, projects. For example, CLARIN RI, in acknowledgment of the crucial importance of the issue of (meta)data quality, established a dedicated standing taskforce for Metadata Curation [3][4] that continuously monitors and tries to improve the quality of the large body of metadata harvested on regular basis by the infrastructure (~800,000 metadata records from ~60 providers).

In general, one can identify the following types of quality issues:

- **Missing values**

  Worst case - information not provided at all; implies bad facet coverage and consequently bad recall.
- **Variability of values**

  Especially in metadata formats where entities are represented/referred to by their names, instead of well-known stable identifiers, trivial issues of spelling and naming variation become very virulent.

  For real-world entities (particulars), this can be remedied (to a certain extent), by striving to convert these strings to globally unambiguous entity identifiers, ideally drawn from some of the global reference resources (DBpedia, VIAF, etc.) the (Information Extraction / Entity Linking).

The problem is even more prevalent for concepts in the broadest sense (universals), where disagreement on a definition is rather the rule, than an exception.

The traditional approach is elaboration of taxonomies or other kinds of more or less structured controlled vocabularies, and even though far from perfect, PARTHENOS has to try to make use of any (semantic) reference resources available wherever possible.

- **Underspecified semantics of fields**

  Oftentimes, problems are introduced already on the schema level, when the scope/range of individual fields is not specified well enough.

  A typical example would be all kinds of "date" fields, where e.g. it is not clear if the specified data refers to the edition of the original resource (e.g. book from 1658) or it's digitized version/edition (2017). Sometimes, even the time coverage (time period the dataset is about, e.g. 4th century BC) is mixed up with dates related to the creation of the resource. In the worst case, data does not provide any formal way to distinguish between 4th century BC, 1658, and 2017, all being just "dates".

- **Inconsistent use of fields**

  Even if individual fields are well-defined, on the instance level they are often filled with incorrect information, introducing problems and confusion on the exploitation side.

  One example would the "simple" field of "MdCreator" in the CMDI-framework, which should indicate the Actor responsible for creating the metadata, but is often filled with the name of the script used to create the metadata (e.g. the name of an xslt script, like "olac2cmdi.xsl")

5.5. Indices for Datasets, Service & Software

As stated before, central to the whole resource discovery functionality is the availability of well-curated comprehensive typed indices to be queried (either directly in a powerful query language, or mediated through interactive facilities like faceted search or graph navigation).

In the following, we make a first attempt to define an elementary set of indices expected to be made available by the Content Cloud to enable basic resource discovery. These should be covered by the minimal metadata. This initial set will quite probably be gradually refined and extended based on feedback received from testing this basic setup.
We assume that the primary entities of interest are Datasets, Services and Software (in terms of CIDOC-PE). The fourth central entity of CIDOC-PE, *Actor*, is considered a secondary (though crucial) entity. In other words, the Actor entities will not appear in result set, but rather as a dimension available for querying, as well as a related entity of the Dataset, Service and Software entities (in their specific roles: creator, rights holder, provider).

Additionally, we propose the corresponding entity-relationship-path in the CIDOC-PE representation, based on the preliminary experience with the mapping, however this is to be considered very experimental, and will have to be thoroughly validated in the initial aggregation phase.

Table 4: Initial list of index fields for basic resource discovery

index	CIDOC-PE source	Comment
title	<> → crm:P1_is_identified_by → crm:E41_Appellation	
description	<> → crm:P3_has_note → rdfs:Literal	
type_class	<> → rdf:type → rdfs:Class	
type_E55type	<> → crm:P2_has_type → crm:E55_Type	
context_collection	<> → crmpe:PP23i_is_dataset_part_of → crmpe:PE24_Volatile_Dataset → crm:P1_is_identified_by → crm:E41_Appellation	
context_provider	<> → pp2_provided_by→ crm:E39_Actor	
context_project	→ pp43i is project activity supported by PE35_Project	

context_RI_consortium	-> pp1 currently provided by -> PE25 RI Consortium	
actor	<> -*-> crm:E39_Actor -> → crm:E51_ContactPoint	* ideally, accompanied by information about the specific role of the actor with respect to the resource (rights owner, provider, …)
time	→ p4_has_time_span → E52_Time_span → P82_at_some_time_within → rdf-schema#Literal	Also needs to be typed: creation date, time coverage, etc.?
related_accesspoints	→ PP28_has_designated_access_point → PE29_Access_Point (+ PE29_Access_Point → P2_has_type → E55_Type)	Link to original resource, if available the kind of access point should be conveyed as well
related_metadata	→ crmpe:pp39i_has_metadata(?) → crmpe:PE22_Persistent_Dataset[metadata]	
text_metadata	* [Potentially any field with textual content]	Flat full-text index on all metadata
Text_resource	[Resource payload]	Subject to availability of the content for indexing, both in terms of format (text?) and permissions (can the data be accessed anonymously, to allow their indexing)

6. Use Case 1 - NERLiX

In order to test, and prototypically synthesize the various technical aspects of the PARTHENOS endeavour - resource aggregation & discovery, metadata mapping, integration of processing and visualisation services integration - a first use case has been developed and is being implemented.

NERLiX stands for Named Entity Recognition and Entity Linking and Extraction and tries to address a number of use cases formulated in PARTHENOS deliverable D2.1 Use cases and requirements [7], most notably:

- 2.1.1.5: Named Entity Recognition (NER) service to extract relevant entities
- 2.1.2.2: Tokenization and lemmatization of the texts. Performs NER on the texts

As opposed to the use cases from D2.1 that were reflecting only the user perspective, this use case is meant more as a demonstrator, meaning that the user requirements are to be translated into a technical solution, which will be implemented and made actually available for use by the users.

At the same time, it should serve as a testing ground for service providers to compose standard recipes for service/tools integration and gradually modify/improve on these as more services/tools are integrated. The standard use cases described in the service integration user stories apply.

Thus, the use case focuses on two main stakeholders in the overall process that collaborate through the VRE, *the service provider and the end user*.

As is given by the hosting infrastructure d4science, the use case will be contained in / implemented as a custom Virtual Research Environment hosted on the D4Science platform.

The use case can be functionally broken down into a combination of specific service/tool interaction and infrastructure component interaction that can be summarized as follows:

- As a **user,** I would like to search available resources
- As a **user,** I would like to select a resource
- As a **user,** I would like to view a selected resource

- As a **user,** I would like to edit a selected resource
- As a **user,** I would like to process a selected resource through a selected processing service
- As a **user,** I would like to access the result of an edited or processed resource from my workspace.

Specifically, the NERLiX VRE shall allow primarily **NLP and semantic enrichment of textual data**. The data can be in different languages, the enrichment/annotation can cover different dimensions (tokenisation, lemmatisation, Part of Speech tags, syntactic parsing, persons, places, etc.).

In the initial setup, we would integrate only a small sample of resources and one or two services. The fully implemented setup should give access to all language resources and all NLP services provided by CLARIN (and other RI partners).

The full setup should contain all **textual resources** made available through participating RIs.

As an initial set, we consider the Historical German texts from Deutsches Text Archiv (DTA) (~2.500 texts) or, alternatively, texts from Digitale Bibliothek. It is expected that these resources are made available through the PARTHENOS infrastructure, i.e. the original metadata of these resources has been mapped onto the PARTHENOS Entities model and is available through the Resource Registry. Similarly, all proposed services for the NERLiX VRE will be expected to be available through the PARTHENOS infrastructure, both at the level of metadata mapping and integration as well as technical integration.

The proposed workflow is aimed to be prototypical to many interaction scenarios and covers a full path from discovery to visualisation, focusing on the **processing** and **visualizing** aspects and comprises the following actions:

1. Select one or more texts from DTA (by metadata query, or hand-picked)
2. Select a service to process the data with (or just hit the button [enrich] in the most minimal setup)
3. **Invoke** selected service with parameters (especially also the desired storage location for the result)

4. (Optionally) **store** resulting (stand-off) annotation [JSON-LD or TCF/XML] in myWorkspace Content Cloud
5. Visualize the result (TCF/XML) with AnnotationViewer
6. **Convert** the information into KML
7. **Visualize** the data in a geo-spatial viewer like GeoBrowser; or at least provide a link from the VRE to GeoBrowser: GeoBrowser accepts any external data for display through passing the URL to the data as parameter).

6.1. Mapping

The mapping model for the use case is based on ten main entities both from CIDOC-CRM and PARTHENOS entity model PE1.11 and aims to integrate the resources from CLARIN into the VRE.

The mapping structure is based on the main mapping between CMDI to PARTHENOS Entities (PE) previously defined in the context of T5.2 Mapping.

Table 5 gives a simplified overview of the mapping of parts of the CMDI metadata records to CIDOC-PE main entities. This covers specifically the teiHeader-style CMDI records describing the resources of DTA, which is the primary dataset of interest in NERLiX use case.

Table 5 Overview of mapping from CMDI to CIDOC-PE

CMDI	CIDOC-PE
cmd:CMD	PE22 Persistent Dataset
cmd:Header	D7 Digital Machine Event
cmd:ResourceProxy	PE29 Access Point
cmdp:teiHeader	PE24 Volatile Dataset
cmdp:author	E21 Person
cmdp:editor	E21 Person

cmdp:respStmt	E39 Actor
cmdp:publicationStmt	F30 Publication Event
cmdp:publisher	E40 Legal Body

6.2. Implementation status

- A default “empty” VRE has been set up within D4Science and made available in a default configuration.
- The foreseen specific functionality is not yet in place, but the system design has been specified and the implementation has been started, with a first prototype of a WPS-compliant wrapper for the processing service Stanbol.
- Custom mappings for the metadata of the primary dataset are available.

Figure 12 shows a UML sequence diagram detailing the interaction flow between the individual components of the use case.

The two new components to be developed and integrated are:

A) Faceted search **portlet** for the resources in the content cloud;

B) A **wrapper** proxying communication from the VRE to the processing service (Stanbol), for sending the selected document by the user to the Stanbol wrapper.

The workflow is as follows:

1. User searches via faceted search (integrated as portlet inside the VRE) for resources.
2. User selects resources for further processing (basket-functionality)
3. User invokes the enrichment process on the selected resources These are sent to the wrapper service (3.1), which handles the asynchronous processing (3.2) and iterative calling to the actual enrichment service (3.3) When the enrichment service is finished with processing a resource it sends the enrichment information (in JSON-LD format) back to the wrapper (3.3.1), which stores it to user’s workspace (3.3.1.1)

4. During the asynchronous processing, the user can check for the status of the processing via job-ticket.
5. When the job is ready, the user is notified via VRE-notification mechanisms, about the completion and the location of the resulting data.

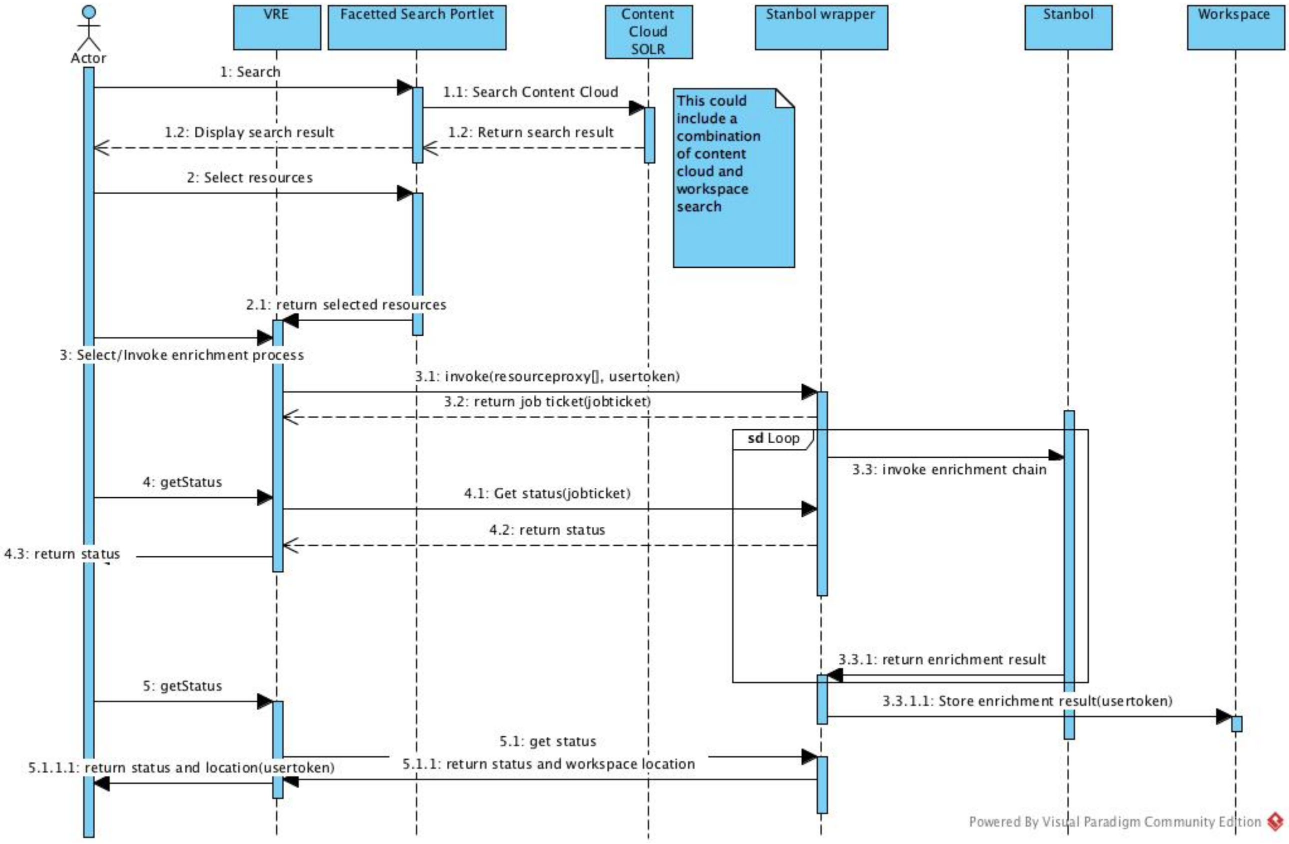

Actor
VRE
Facetted Search Portlet
Content Cloud SOLR
Stanbol wrapper
Stanbol
Workspace
1: Search
1.1: Search Content Cloud
This could include a combination of content cloud and workspace search
1.2: Display search result
1.2: Return search result
2: Select resources
2.1: return selected resources
3: Select/Invoke enrichment process
3.1: invoke(resourceproxy[], usertoken)
3.2: return job ticket(jobticket)
sd Loop
3.3: invoke enrichment chain
4: getStatus
4.1: Get status(jobticket)
4.2: return status
4.3: return status
3.3.1: return enrichment result
5: getStatus
3.3.1.1: Store enrichment result(usertoken)
5.1: get status
5.1.1.1: return status and location(usertoken)
5.1.1: return status and workspace location
Powered By Visual Paradigm Community Edition

Figure 12: UML sequence diagram of NERLiX use case

7. References

[1] Pagano, P. Candela, L., Assante, M., Frosini, L., Manghi, P., Bardi, A. & Sinibaldi, F. (2016). *"PARTHENOS Cloud Infrastructure"*. Deliverable D6.1.

[2] Manghi, P., Artini, M, Atzori, C., Bardi, A., Mannocci, A., La Bruzzo, S., Candela, L., Castelli, D. & Pagano, P. (2014). *"The D-NET software toolkit: A framework for the realization, maintenance, and operation of aggregative infrastructures"*, Program, Vol. 48 Issue: 4, pp.322-354, doi:10.1108/PROG-08-2013-0045

[3] King, M., Ostojic, D., Ďurčo, M., & Sugimoto, G. (2016). Variability of the Facet Values in the VLO – a Case for Metadata Curation. In *Selected Papers from the CLARIN Annual Conference 2015*, pp. 25–44. Linköping University Electronic Press. Retrieved from http://www.ep.liu.se/ecp/123/003/ecp15123003.pdf

[4] Ostojic, D., Sugimoto, G., & Ďurčo, M. (2016). Curation module in action-preliminary findings on VLO metadata quality. Retrieved from https://www.clarin.eu/sites/default/files/ostojic-etal-CLARIN2016_paper_22.pdf

[5] Shneiderman, B. (1996). The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pp. 336-343, Washington: IEEE Computer Society Press. Retrieved from: http://citeseer.ist.psu.edu/409647.html

[6] Marketakis, Y., Minadakis, N., Kondylakis, H. et al. (2016). X3ML mapping framework for information integration in cultural heritage and beyond. Int J Digit Libr (pp 1-19). doi:10.1007/s00799-016-0179-1

[7] Drude, S., Di Giorgio, S., Ronzino, P., Links, P., Degl'Innocenti, E., Oltersdorf, J. & Stiller, J. (2016). "Report on User Requirements". Deliverable D2.1.