D12.2 – Mid-term report on data integration

Version 1.0 (*final*)

2021-01-20

Grant Agreement number:	**823914**
Project acronym:	**ARIADNEplus**
Project title:	**Advanced Research Infrastructure for Archaeological Dataset Networking in Europe - plus**
Funding Scheme:	**H2020-INFRAIA-2018-1**
Project co-ordinator name, Title and Organisation:	**Prof. Franco Niccolucci, PIN Scrl - Polo Universitario "Città di Prato"**
Tel:	**+39 0574 602578**
E-mail:	**franco.niccolucci@pin.unifi.it**
Project website address:	**www.ariadne-infrastructure.eu**

The research leading to these results has received funding from the European Community's Horizon 2020 Programme (H2020-INFRAIA-2018-1) under grant agreement n° 823914.

Author: **Alessia Bardi**

CNR-ISTI

Johan Fihn Marberg

SND

Maria Theodoridou

FORTH

Contributing partners: **Miriam Baglioni, CNR-ISTI**

Vittore Casarosa, CNR-ISTI

Pablo Millet, SND

Enrico Ottonello, CNR-ISTI

Quality control: **Stephanie Williams, PIN**

Document History

- 07.12.2020 – Draft Version 0.1
- 07.01.2021 - Task descriptions
- 12.01.2021 - Introduction and executive summary
- 14.01.2021 - Conclusions
- 16.01.2021 - Draft version 1.0 for internal review

Table of Contents

Document History **3**

Table of Contents **4**

1 Executive Summary **5**

2 Introduction and Objectives **6**

3 Activities **8**

3.1 T12.1 – JRA 1.1 Implementing the ARIADNEplus AC 8

3.1.1 Work performed 8

3.1.2 Deviation from work plan 12

3.1.3 Plans for the next period 12

3.2 T12.2 – JRA1.2 Implementing the ARIADNEplus Aggregative Data Infrastructure 13

3.2.1 Work performed 13

3.2.2 Deviation from work plan 16

3.2.3 Plans for the next period 16

3.3 T12.3 – JRA 1.3 Implementing the ARIADNEplus Portal 16

3.3.1 Work performed 16

3.3.2 Deviation from the work plan 22

3.3.3 Plans for the next period 22

3.4 T12.4 – JRA1.4 Item-level data integration 23

3.4.1 Work performed 24

3.4.2 Deviations from the work plan 25

3.4.3 Plans for the next period 25

4 Conclusions **25**

1 Executive Summary

This deliverable describes the activities carried out and the results achieved during the first two years of the ARIADNEplus project within four tasks of Work Package 12 (WP12).

The objectives are to develop, deliver and maintain the components of the ARIADNEplus infrastructure that support the integration and interoperability of the data provided by the members of the consortium. The catalogue data integrated by the ADI (the Aggregative Data Infrastructure developed in T12.2) are made available as RDF records compliant to the AO-Cat model to the ARIADNEplus portal (T12.3) and the pilots developed in WP16 via two services: (1) the ARIADNEplus AC (the data and knowledge cloud developed in T12.1) , which exposes a SPARQL API, and (2) an Elasticsearch server, which provides a full-text index of the content of the AC. The deeper integration of item level data (item-level integration) is investigated in task 12.4, in order to develop support for research questions that require information that is richer than what is available in AO-Cat.

The design, development and deployment activities have been guided by the requirements of all the members of the consortium, especially those involved in WP4 and WP5. For the development of the new features of the portal, a Portal Working Group has been formed including technical and non-technical members from SND, PIN, USW, CNR, ADS, and SRFG. By December 2020, WP12 delivered all the components and implemented the aggregation workflow devised in collaboration with WP5.

The ADI includes services and tools required to perform data collection, transformation, and harmonisation: the 3M Editor (definition of the mappings from local metadata format to AO-Cat) developed and maintained by FORTH; the Vocabulary Matching Tool (definition of mappings from local subject terms to terms of Getty AAT) developed and maintained by USW; and the ARIADNEplus aggregator developed and maintained by CNR (the data aggregator is based on the D-Net software toolkit: it collects the providers' XML records and integrates the X3ML toolkit for the execution of 3M mappings, and implements the aggregation workflows defined in collaboration with WP5).

The AC includes a knowledge graph implemented with GraphDB (free edition) and one Springboot application that acts as mediator for the interactions among the aggregator, GraphDB, and Elasticsearch.

Elasticsearch is used to provide a full-text index of the content available in the AC, to be used by the ARIADNEplus portal. The ARIADNEplus portal is developed using PHP, Vue.js, Javascript, Vuex, Tailwind, and Font Awesome. The portal provides standard free-text and faceted search options, but also advanced features based on the concepts of temporal, spatial, and topical coverage.

In order to enable data curators to check the quality of data before it is made available on the public portal, WP12 set up a staging environment where the collected data is aggregated, added to a staging AC and indexed on a staging portal. Upon confirmation of data experts, data is then pushed to the production environment. The production environment makes the ARIADNEplus portal available to the public, while the staging environment is only available to the consortium members to check data quality and test new functionality of the portal.

2 Introduction and Objectives

Introduction to deliverable and the objectives of the work.

This deliverable describes the activities carried out during the first two years of the ARIADNEplus project within Work Package 12 (WP12) by the different partners and describes the results achieved by this work package in the four tasks:

- T12.1 Implementing the ARIADNEplus data and knowledge cloud (AC) (JRA1.1)
- T12.2 Implementing the ARIADNEplus Aggregative Data Infrastructure (ADI) (JRA1.2)
- T12.3 Implementing the ARIADNEplus Portal (JRA1.3)
- T12.4 Item-level data integration (JRA1.4)

The objectives are to develop, deliver and maintain the components of the ARIADNEplus infrastructure that support the integration and interoperability of the data provided by the members of the consortium. The integrated data are made available to the ARIADNEplus portal and the pilots developed in WP16 via two services: (1) the ARIADNEplus AC (knowledge graph), which exposes a SPARQL API, and (2) an Elasticsearch service, which provides a full-text index of the content of the AC (see Figure 1).

Figure 1 Services and tools for data integration and interoperability in the ARIADNEplus infrastructure

The aggregative infrastructure (T12.2) includes services and tools required to perform data collection, transformation, and harmonisation according to domain-specific vocabularies and ontologies:

- 3M Editor: definition of the mappings from local metadata format to AO-Cat;
- Vocabulary Matching Tool: definition of mappings from local subject terms to terms of Getty AAT
- D-Net: the ARIADNEplus aggregator is based on the D-Net software toolkit. It collects the providers' XML records and integrates the X3ML toolkit for the execution of 3M mappings. It is configured to implement the aggregation workflows defined in collaboration with WP5.

The ARIADNEplus data and knowledge Cloud (AC) (T12.1) is a knowledge graph developed according to the Resource Description Framework (RDF), supporting the Semantic Web principles (Linked Data). It comes with a SPARQL access point, providing triples of the knowledge graph, and reasoning capabilities that can be exploited for the implementation of advanced knowledge discovery services.

The portal (T12.3) is the main entry point for humans willing to search, browse and access the aggregated resources.

Finally, the deeper integration of item level data (item-level integration) is investigated in task 12.4 in order to develop support for research questions that require information richer than what is available in AO-Cat.

Deliverable D12.1 briefly reported about activities carried out in the context of these tasks until the end of the first reporting period (Months 1-18). This report extends D12.1 with additional technical details and covers the period M1-M24, until December 2020.

Results obtained in the first 24 months

T12.1 ARIADNEplus data and knowledge Cloud (AC)

- Selection of the technology for the knowledge graph (GraphDB, free edition)
- Definition of the organization of aggregated data in "named graphs"
- Deployment of two instances of GraphDB, one staging instance for data curation, one public instance available to the public
- Development and deployment of a component in charge of the communication among the aggregative infrastructure, Elasticsearch and GraphDB

T12.2 Aggregative infrastructure

- D-NET framework toolkit instantiated for ARIADNEplus and configured to implement the aggregation workflow devised in collaboration with WP5.
- 3M Editor updated to allow the generation of RDF records compliant with AO-Cat
- Vocabulary Matching Tool for the generation of mappings from local subject terms to Getty AAT terms deployed on the D4Science infrastructure

T12.3 ARIADNEplus portal

- Elasticsearch server v7.4.0 available on the D4Science infrastructure and configured by migrating the schema of the old server to the new format
- Implementation of the new portal with a new technology stack
- Deployment of Elasticsearch v7.4.0 and portal in two environments: staging and production. The production environment features the publicly available portal, while the staging environment is only available to the consortium members to check data quality and test new functionality of the portal.
- Docker setup created for easy deployment of the portals
- A user requirements group has been formed to formulate the feature requirements and priorities regarding the development of the portal.

T12.4 Item-level integration

- Compilation of a first list of potential resources suitable for deeper integration. Started working on their mappings.

- Started a revision of experimental item level mappings that had been performed during the first ARIADNE.

3 Activities

3.1 T12.1 – JRA 1.1 Implementing the ARIADNEplus AC

The ARIADNEplus data and knowledge Cloud (AC) is a knowledge base where datasets of the Archaeological Research Communities are integrated, structured according to the CIDOC CRM-based ARIADNEplus Ontology, and interlinked with standard ontologies and gazettiers of the archeological domain, such as PeriodO for historical periods and Getty AAT for arts and archeological subject terms classification.

The objectives of task 12.1 are to design, develop, deliver and maintain the AC using standards and state-of-the-art web and Semantic Web languages and protocols (e.g. REST, RDF, XML, JSON, SPARQL) to maximize interoperability and foster re-usability.

APIs for feeding and querying the AC will also be implemented and made available for the ARIADNE portal (T12.3), discovery use cases (T12.5), and the pilots (WP15 and WP16). A SPARQL endpoint will also be made openly available.

3.1.1 Work performed

The first activity performed was the selection of the technology to adopt for the implementation of the AC. Three candidate technologies were identified for evaluation because they are widely used for the implementation of RDF knowledge bases: Virtuoso1, Metaphactory by metaphacts2, and GraphDB3. GraphDB free edition has been evaluated as the best compromise between functionalities, operational costs and user-friendliness for both developers and end-users. The performances of GraphDB are being monitored and, if needed, a plan for the migration to the commercial version of GraphDB will be proposed to the JRA supervisor.

Based on the requirements received from T5.4 “Ingesting data into the ADI”, the high-level architecture of the AC and the workflow for its population with content has been defined. The AC features two instances of GraphDB and two instances of the Publisher component. The “staging” instances (on the left of Figure 2) are needed to provide a controlled environment where data experts can evaluate the output of the aggregation and preview how the data will look like in a dedicated portal. The staging instances, including the staging portal, will only be accessible to the consortium members. Data is therefore made available first on the staging GraphDB and then, only upon

[1] https://virtuoso.openlinksw.com/
[2] https://metaphacts.com/
[3] https://graphdb.ontotext.com/

successful feedback from the data experts, it is fed to the public GraphDB (and eventually to the public ARIADNEplus portal).

The publisher component is in charge of the communication among the ARIADNEplus Aggregator, GraphDB and the Elasticsearch servers used by the ARIADNEplus portal. It is implemented as a Spring Bootstrap application exposing a REST API whose access is restricted to the ARIADNEplus Aggregative Data Infrastructure. The source code is available at CNR's Git repository accessible at https://code-repo.d4science.org/D-Net/AriadnePlus/src/branch/master/dnet-ariadneplus-graphdb-publisher .

The staging instances of GraphDB and the Publisher component have been deployed in M15 (March 2020) and have been fine tuned and adapted to the evolution of the AO-Cat ontology and the requirements of the portal and WP5.

The public instances have been deployed in M24 (December 2020).

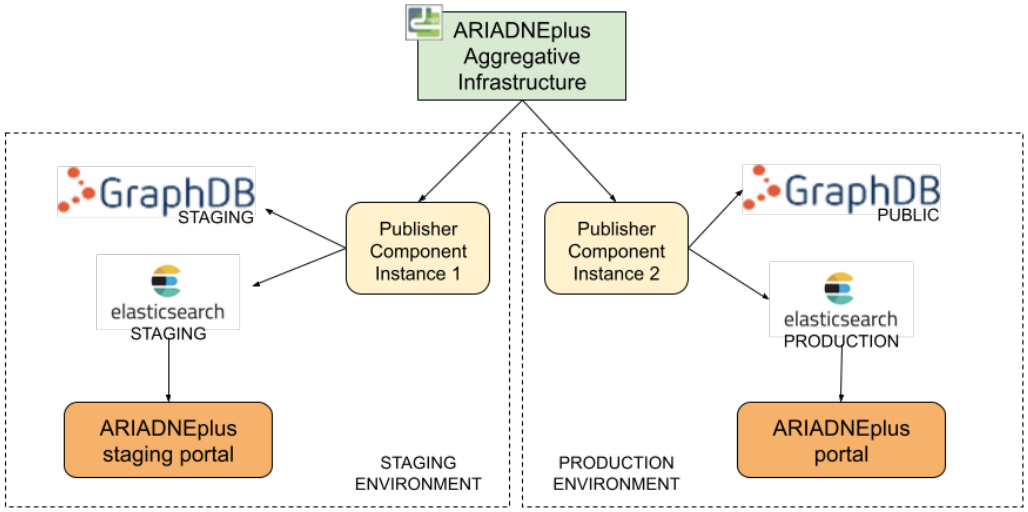


Figure 2 Staging and production environments

Data on GraphDB is organised in such a way that it is possible to perform incremental updates of its content. This has been done by using the concept of "named graph" in a specific way. For each set of data, the ARIADNEplus aggregator pushes the transformed RDF records to the publisher component, which in turns feeds one dedicated named graph of GraphDB. In other words, GraphDB features one named graph per dataset (where "dataset" is to be intended as a set of metadata records from a provider that can be transformed with the same 3M mapping). Having the dataset under one single named graph enables the ARIADNEplus aggregator to request the deletion of that specific dataset. The feature was clearly a fundamental requirement in order to support continuous aggregation and update of the AC. Every time a dataset is updated (because the input changed or the mapping changed), the ARIADNEplus aggregator requests the publisher component to delete the named graph that corresponds to the dataset at hand and then proceed with the feeding of the updated datasets. With the same logic, enhancements to the provided records are added to dedicated named graphs, so that it is easy to keep track of the provenance of each RDF statement. In particular, GraphDB features five named graphs for each provider, as shown in Figure 3:

- One named graph with the aggregated RDF triples of a given dataset. If the provider manages different datasets (e.g. two databases), then GraphDB will feature one named graph per dataset. The URI of the named graph is based on the identifier assigned to the dataset on the ARIADNEplus aggregator with the template *https://ariadne-infrastructure.eu/<DNET_API_ID>*, where *DNET_API_ID* identifies the interface from which the dataset is collected from. For example, the named graph for the Archaeology Database of HNM has URI *https://ariadne-infrastructure.eu/api_____::ariadne_plus::hnm::hnmad*. The URI shall be considered as a mere identifier local to GraphDB and, although it starts with 'https' it does not resolve to any content served by the HTTPS protocol. We may change this approach in the next period of the project.
- One named graph with the matches between local subjects and Getty AAT terms as generated using the Vocabulary Matching Tool. The URI of the named graph is based on the identifier assigned to the D-NET API from which the matches have been collected from. For convenience, all APIs of this type are named in the form *api_____::ariadne_plus::<providerAcronym>::aat*. For example, the named graph for the matches between HNM subject terms and Getty AAT has URI *https://ariadne-infrastructure.eu/api_____::ariadne_plus::hnm::aat*.
  The URI shall be considered as a mere identifier local to GraphDB and, although it starts with 'https' it does not resolve to any content served by the HTTPS protocol. We may change this approach in the next period of the project. The existence of this graph is optional, as the transformed RDF triples may already contain terms of Getty AAT.
- One named graph with PeriodO terms covered by the provider. The URI of the named graph follows the template *https://ariadne-infrastructure.eu/ariadneplus::<providerAcronym>::periodo*. The URI shall be considered as a mere identifier local to GraphDB and, although it starts with 'https' it does not resolve to any content served by the HTTPS protocol. We may change this approach in the next period of the project. The existence of this graph is optional, as the transformed RDF triples may already contain terms and dates of PeriodO or the provider might not have a dedicated PeriodO collection to import. In this latter case, the PeriodO information will be taken from the PeriodO collection created during the previous ARIADNE project[4], which has been imported in GraphDB under the named graph *https://ariadne-infrastructure.eu/ariadne/periodo*.
- One named graph containing the triples inferred by intersecting the aggregated data and Getty AAT based on the provided matching that are available in the *https://ariadne-infrastructure.eu/api_____::ariadne_plus::<providerAcronym>::aat* named graph. The named graph is assigned to a URI that follows the template *https://ariadne-infrastructure.eu/ariadneplus::<providerAcronym>::aatplus*. The URI shall be considered as a mere identifier local to GraphDB and, although it starts with 'https' it does not resolve to any content served by the HTTPS protocol. We may change this approach in the next period of the project. The existence of this graph is optional.

[4] PeriodO collection created in the previous ARIADNE project: http://n2t.net/ark:/99152/p0qhb66

- One named graph containing the triples inferred by intersecting the aggregated data and the PeriodO collection relevant for the provider (available either in the named graph *https://ariadne-infrastructure.eu/ariadne/periodo* or *https://ariadne-infrastructure.eu/ariadneplus::<providerAcronym>::periodo*). The named graph is assigned to a URI that follows the template *https://ariadne-infrastructure.eu/ariadneplus::<providerAcronym>::periodoplus*. The URI shall be considered as a mere identifier local to GraphDB and, although it starts with 'https' it does not resolve to any content served by the HTTPS protocol. We may change this approach in the next period of the project. The existence of this graph is optional.

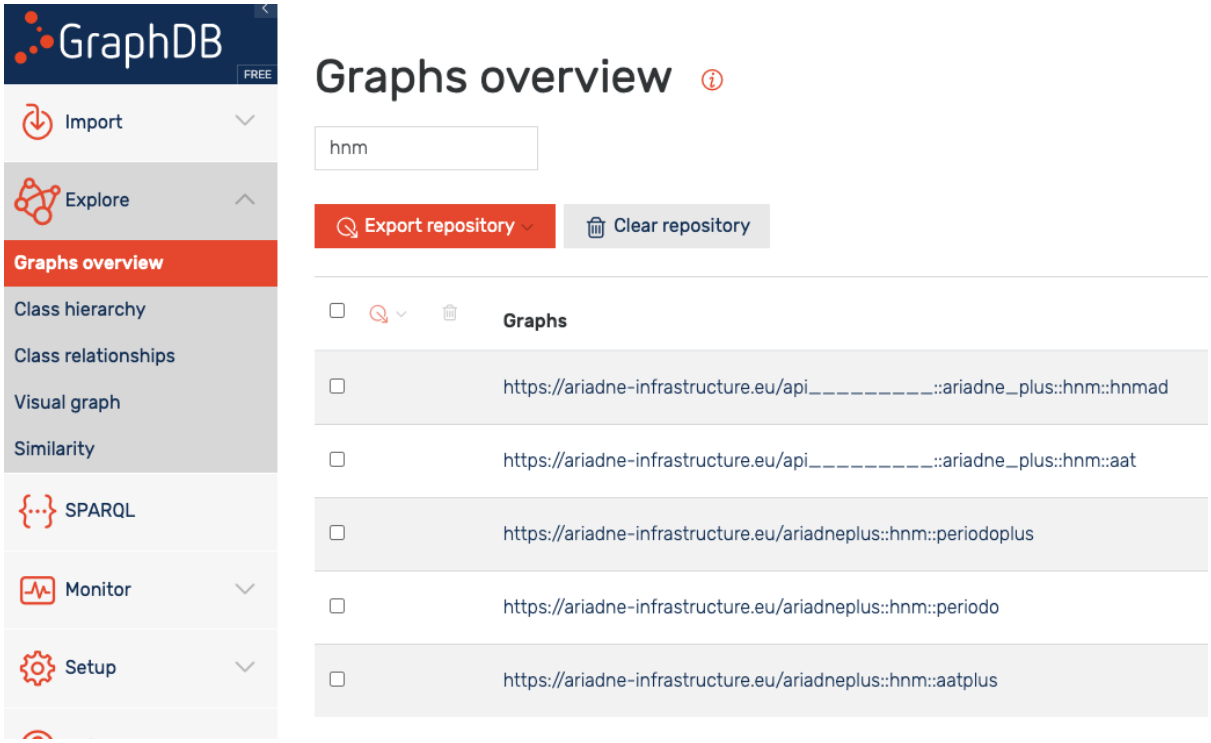
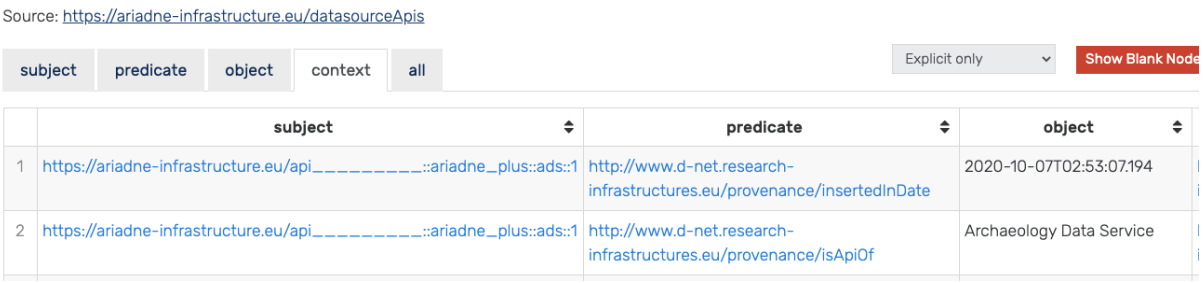


Figure 3 The named graphs associated to data provided by HNM

In addition, The ARIADNEplus aggregator also adds information to a special graph, named *https://ariadne-infrastructure.eu/datasourceApis*, to keep provenance information. It contains information about which endpoints and which datasets have been added to GraphDB and when. As an example, Figure 4 shows part of the content of the graph that tells us that the DNET API with id *api_____::ariadne_plus::ads::1* is an API of the Archaeology Data Service (ADS) and that the content from it was inserted into GraphDB in 2020-10-07. Please note that the subject of the triples (*https://ariadne-infrastructure.eu/api_____::ariadne_plus::ads::1*) has the same URI as the named graph that groups all RDF triples generated from the data collected from this API.


Source: https://ariadne-infrastructure.eu/datasourceApis

subject | predicate | object | context | all — Explicit only — Show Blank Node

	subject	predicate	object
1	https://ariadne-infrastructure.eu/api_____::ariadne_plus::ads::1	http://www.d-net.research-infrastructures.eu/provenance/insertedInDate	2020-10-07T02:53:07.194
2	https://ariadne-infrastructure.eu/api_____::ariadne_plus::ads::1	http://www.d-net.research-infrastructures.eu/provenance/isApiOf	Archaeology Data Service

Figure 4 Sample triples in the provenance graph

The benefits of such a partition of content on GraphDB target data curators, aggregation managers, and end-users in different ways:

- Machine discoverability of new content and new providers in the AC;
- Easy identification of what has been aggregated and what has been inferred;
- Easy update of each subset of triples. If there are mistakes in the inference rules, only the "plus" graphs can be deleted, the inference rules updated and the "plus" graphs regenerated;
- Continuous updates of PeriodO terms, Getty AAT matching, and input data do not affect each other and can be run in isolation.

The main drawback consists in not having all triples about a dataset in the same named graph: in order to execute SPARQL queries that involve aggregated and inferred data, multiple named graphs must be included in the query. This may be not very convenient, especially for end-users who might not be fully aware of how the data is organised. The problem can be addressed by providing public and clear documentation with examples. A real use-case example is represented by the publisher component itself, which must perform a query on multiple graphs in order to obtain the information needed by the ARIADNEplus portal and feed it to the Elasticsearch server. The query used for collection level record is available at https://code-repo.d4science.org/D-Net/AriadnePlus/src/branch/master/dnet-ariadneplus-graphdb-publisher/src/main/resources/eu/dnetlib/ariadneplus/sparql/read_collection_data_template.sparql, the query for individual resources is available at https://code-repo.d4science.org/D-Net/AriadnePlus/src/branch/master/dnet-ariadneplus-graphdb-publisher/src/main/resources/eu/dnetlib/ariadneplus/sparql/read_record_data_template.sparql. Both queries are SPARQL CONSTRUCT queries whose goal is to provide information in a way that is easy to package into JSON records, compliant with the format expected by Elasticsearch.

3.1.2 Deviation from work plan

No deviation from the work plan.

3.1.3 Plans for the next period

In the next period, the Publisher component will be updated to address possible changes to the AO-Cat ontology and new requirements from the portal.

Documentation about the organisation of content in GraphDB will be produced to support data curators and end-users at performing SPARQL queries.

The possibility of adopting existing ontologies for the triples in the provenance named graph, without losing simplicity, will be investigated.

3.2 T12.2 – JRA1.2 Implementing the ARIADNEplus Aggregative Data Infrastructure

The aggregative infrastructure includes services and tools required to perform data collection, transformation, and harmonisation according to domain-specific vocabularies and ontologies:

- 3M Editor: definition of the mappings from local metadata format to AO-Cat;
- Vocabulary Matching Tool: definition of mappings from local subject terms to terms of Getty AAT;
- D-Net: the ARIADNEplus aggregator is based on the D-Net software toolkit. It collects the providers' XML records and integrates the X3ML toolkit for the execution of 3M mappings. It is configured to implement the aggregation workflows defined in collaboration with WP5.

The objectives of task 12.2 are to design, develop, deliver and maintain the ARIADNEplus ADI.

3.2.1 Work performed

The aggregative infrastructure has been set up and made available, as depicted in Figure 1. The aggregative infrastructure enables the integration and harmonisation of the data provided by the members of the consortium. The catalogue data integrated by the ADI are made available to the ARIADNEplus portal and the pilots developed in WP16 via two services: (1) the ARIADNEplus AC (the data and knowledge cloud developed in T12.1) , which exposes a SPARQL API, and (2) an Elasticsearch service, which provides a full-text index of the content of the AC.

During the first two years of the project, the D-NET framework toolkit has been instantiated for ARIADNEplus and configured to implement the aggregation workflow and requirements devised in WP5. Tests have been run on data provided by ADS and a stable version of the aggregator has been released in M10 (October 2019).

Additional updates and enhancements have been applied to the aggregation workflow in the ARIADNEplus aggregator in order to address new requirements and better serve the use cases of the project and peculiarities of partners in the consortium. Currently, the ARIADNEplus aggregator features three types of workflows that can be associated to a so-called API, i.e. a description of an entry point from which the aggregator can collect metadata records that can be transformed according to the same 3M mapping: the workflow for importing PeriodO collections, the workflow for integrating matchings to Getty AAT, and the aggregation workflow.

Integrating Getty AAT matchings

Each providers' data experts use the Vocabulary Matching Tool to associate the subjects used in the local data to terms of the Getty AAT. The output is one (or more) JSON file(s) that need to be transformed into RDF and added to the AC. The ARIADNEplus aggregator features one type of workflow to address this integration aspect. For each provider with a Getty AAT matching, the aggregation managers create one API configured to point to the proper JSON files. A workflow is then assigned to perform collection, transformation (with the 3M mapping 648), feeding to the staging AC and, eventually, feeding to the public AC.

Importing PeriodO collections

The ARIADNEplus aggregator supports two use cases. The first use case is applied to providers that were involved in the previous ARIADNE project and contributed to the ARIADNE PeriodO collection available at http://n2t.net/ark:/99152/p0qhb66. A dedicated API has been created in order to import the collection in the AC. A workflow is then applied that gets the data and calls the Publisher component (see T12.1) to feed the triples into GraphDB. If a provider updates the content of the collection, the workflow can be run again so that the new triples will be included in the AC.

The second use case applies to providers that were not involved in the previous ARIADNE project or that have produced a more up-to-date PeriodO collection since 2015. In this case a dedicated API is created and the same workflow applied in the first use case is associated and executed.

In both cases, when the data is ready to go public on the AC, another workflow of the same type is associated and configured to feed the public AC instead of the staging AC.

The aggregation workflow

As of December 2020, the aggregation workflow features a number of steps for the collection, transformation, and publishing into the AC of metadata records. The aggregation manager configures the workflow based on the information given by the partner responsible for the data in the Google form "ARIADNEplus: info for metadata aggregation" (https://forms.gle/eo54mRJL7wSSPvALA). The workflow is composed of "sub-workflows", as depicted in figure 5. Each sub-workflow includes the necessary steps for a given activity.

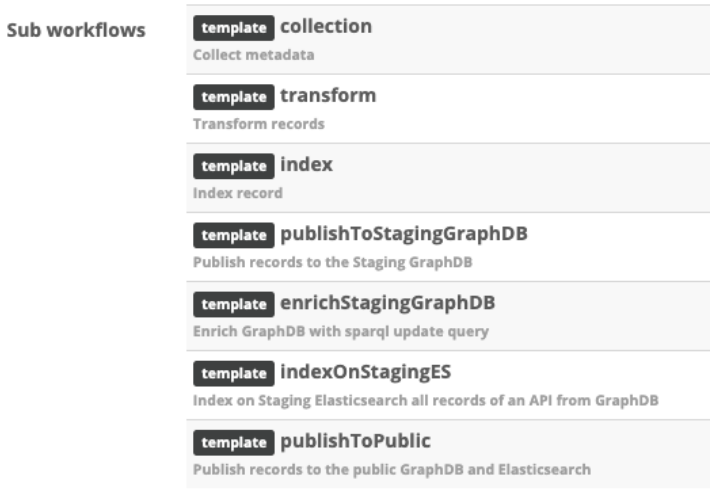


Figure 5 The aggregation workflow implemented in the ARIADNEplus aggregator

1. *collection*: collect native records (adds OAI header, if missing);
2. *transform*: apply 3M mapping to each record;
3. *index*: add records to Solr for the DNET Metadata Inspector;
4. *publishToStagingGraphDB*: add records to staging GraphDB. The automatic execution of the workflow stops here to allow the aggregator manager and the data curators to check the triples in GraphDB and set up the needed instructions for the next sub-workflow;
5. *enrichStagingGraphDB*: runs SPARQL INSERT queries to add triples about derived subjects and periods (and anything else that is needed to complete the records, such as propagation of fields from collection to individual records) on the staging GraphDB;
6. *indexOnStagingES*: reads from the staging GraphDB and sends the constructed records to the Elasticsearch that serves the staging portal. At this point the workflow pauses and the aggregation manager waits for the feedback from the data experts. If there are no mistakes in the data available via the staging AC and portal, then the data experts ask the aggregation manager to proceed with the publication of the data on the public portal, which is done by launching the last sub-workflow;
7. *publishToPublic*: executes the same processes as 4, 5, 6 but on the public instances of GraphDB and Elasticsearch without the need for additional manual intervention.

In addition to the configuration and deployment, the following development activities have been carried on the D-NET framework toolkit:

- Update to support Solr7 as back-end technology for the D-NET Metadata Inspector;
- Update to use version 1.9.3 of the X3ML-engine library;
- Development and testing of a new collector plug-in for ADS dumps;

- The Transformation step of the aggregation workflow adapted to use a remote 3M mapping (i.e. do not require local copies of mappings).

The 3M Editor has been updated to allow the generation of RDF records compliant to the AO-Cat ontology.

The Vocabulary Matching Tool for the generation of mappings from local subjects to Getty AAT has been deployed on the D4Science infrastructure.

A new tool called Activity Dash, developed in T14.1 to support the coordination of the ingest procedures, was released in M19 and deployed on the D4Science infrastructure.

3.2.2 Deviation from work plan

No deviation from the work plan.

3.2.3 Plans for the next period

In the next period no development activities are planned. The activities will focus on maintenance and operation of the aggregative infrastructure.

Some development might be needed to support the integration of data exposed via non standard exchange protocols that are not already supported by the aggregator.

3.3 T12.3 – JRA 1.3 Implementing the ARIADNEplus Portal

This task comprises the user-centred design, implementation and testing of a web application making the services exposed by the current ARIADNE Portal available through a new enriched Portal. First, the Portal developed by this task will rely on the ARIADNEplus Data Infrastructure instead of the ARIADNE Catalogue, the former being a large expansion of the latter with all the data collected and enriched by the ARIADNEplus Project. Second, the discovery service of the new Portal will significantly extend the Data Infrastructure search of the current ARIADNE Portal by adding the functionality developed by Task 12.5. This task will design and implement the GUI of the discovery service as part of the Portal. The GUI will also offer a browsing facility, allowing the visual exploration of the ARIADNEplus AC according to semantic categories defined in the ARIADNEplus Ontology. Finally, the new Portal will also expose the ARIADNEplus pilots and make the related data and knowledge accessible to the public. Therefore, special sections for the different pilots will be added to the application.

3.3.1 Work performed

Portal Working Group

Development of the portal follows a sprint-based methodology, where the functionality to be developed is discussed, evaluated, prioritized and approved by a set of experts from various partner

organizations. To support this effort, a Portal Working Group has been formed consisting of technical and non-technical members from SND, PIN, USW, CNR, ADS, and SRFG. The group meets regularly on a monthly or bi-monthly basis to discuss the upcoming development plan and to sign off on the developed functionality.

Staging and production environment

To automatize the deployment of new functionalities for the portal, all source code has been contained within a Docker[5] image and uploaded to Docker Hub[6]. Upon image updates, the image is then automatically downloaded from Docker Hub and deployed in a cluster of Docker engines, called a swarm[7]. This swarm is running on the D4Science[8] platform, which is used to run all VREs produced in the ARIADNEplus project. This automatization means deployment of new functionalities for the staging and the production environments can be done effortlessly, with minimal involvement of personnel resources.

The production environment of the portal, available to the public, is run as a Docker image deployed in the Docker swarm. For development and showcase purposes, a staging environment has also been deployed on the Docker swarm. This gives the developers the opportunity to show and test new functionalities with a smaller number of users before launching the new feature in production.

New technology stack

The old ARIADNE portal was developed using technology chosen with the intention of having a search portal and a backend feature where partners could login to curate their metadata. A backend framework was selected with this feature in mind that was very fit for that purpose. The backend curation functionality saw little use, and for the ARIADNEplus portal this feature has been removed. Therefore, the login functionality is no longer required. As the code base of the old framework was several major releases behind and in need of security updates, as well as additional code maintenance, a choice was made to migrate the existing functionality of the portal to a new code base, not based on a framework. This choice would also give enough flexibility to the developers in developing the new functionalities of the portal, where using a framework would add too much unneeded overhead.

Elasticsearch[9] is used for persistence of the metadata of the resources. Elasticsearch is a fit-for-purpose database with built in search engine where all metadata is stored as JSON[10] documents. Elasticsearch has been developed as a scalable database especially focused on searching large amounts of data in near real-time. An Elasticsearch mapping, which contains rules of the structure of the documents, has been developed to ensure that all documents have the same structure and are interpreted in the same way. The mapping document also provides various structures, which facilitates the various search options, described below, which the ARIADNE portal offers.

[5] https://www.docker.com/
[6] https://hub.docker.com/
[7] https://docs.docker.com/engine/swarm/
[8] https://www.d4science.org/
[9] https://www.elastic.co/products/elasticsearch
[10] https://www.json.org/json-en.html

The backend core of the ARIADNEplus portal is developed using PHP[11], which is a common, general purpose, programming language for web development. The core is responsible for communicating with and performing searches on the resources in the Elasticsearch software and delivering the results of these queries to the front end. For the communication with Elasticsearch, the official PHP Elasticsearch API[12] is used.

The frontend of the ARIADNEplus portal is developed using the Vue.js[13] Javascript[14] framework. Vue.js is a framework focusing on the view layer of an application and is used mainly to build user interfaces. In addition, Typescript[15] is used to provide type checking of the Javascript produced. Vuex[16] is used for state handling of the application to manage the various states a component of the frontend application has at any given time. Tailwind[17] is used as a CSS framework to build the look-and-feel of the application and Font Awesome[18] is used to provide scalable fonts to the application.

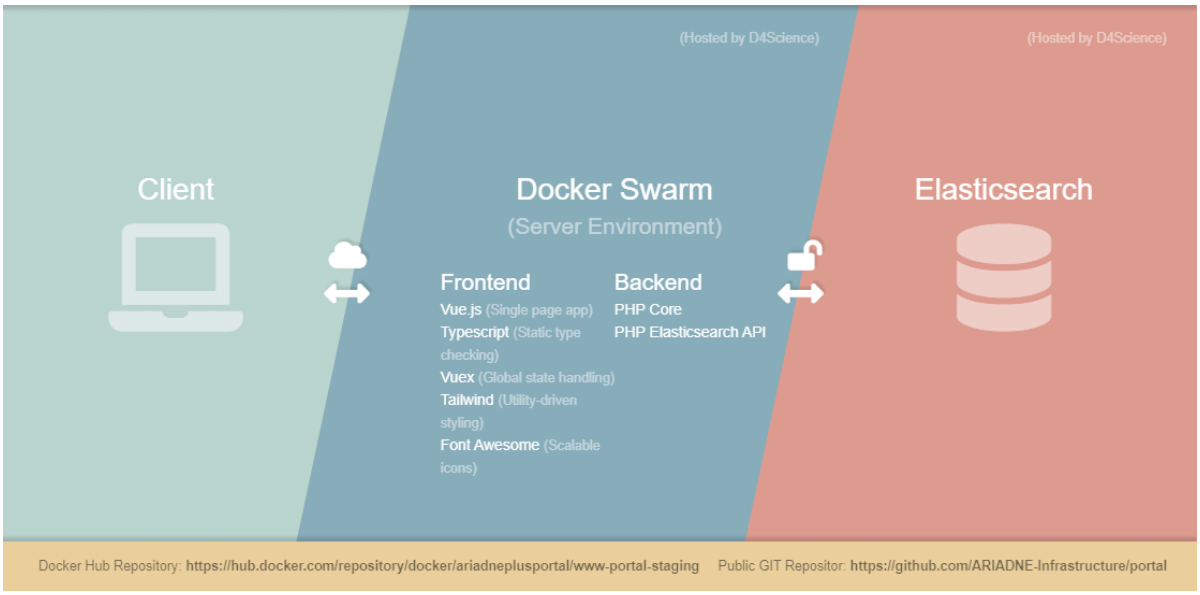


Figure 6 Technology stack of the portal

Portal functionality

The portal is developed with three keywords in mind, in addition to the standard free-text and faceted search options. These keywords, when – where – what, revolve around the concepts of temporal coverage, spatial coverage, and topical coverage. These keywords are focal points in the different findability options a user has when using the portal.

[11] https://www.php.net/
[12] https://www.elastic.co/guide/en/elasticsearch/client/php-api/current/index.html
[13] https://vuejs.org/
[14] https://www.javascript.com/
[15] https://www.typescriptlang.org/
[16] https://vuex.vuejs.org/
[17] https://tailwindcss.com/
[18] https://fontawesome.com/

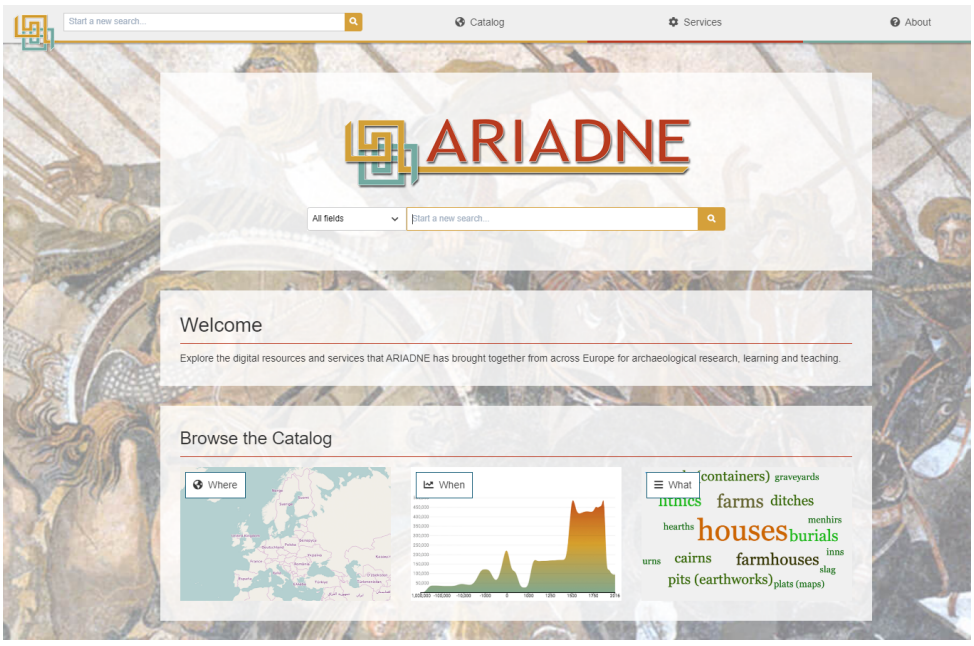


Figure 7 Front page of the portal

The “when” keyword represents the time period a resource is covering and is enabled by a Timeline search option where the user can choose a year span for browsing resources covering a specific period. In addition to the time span, resources can be filtered using various facets to further narrow down the search results.

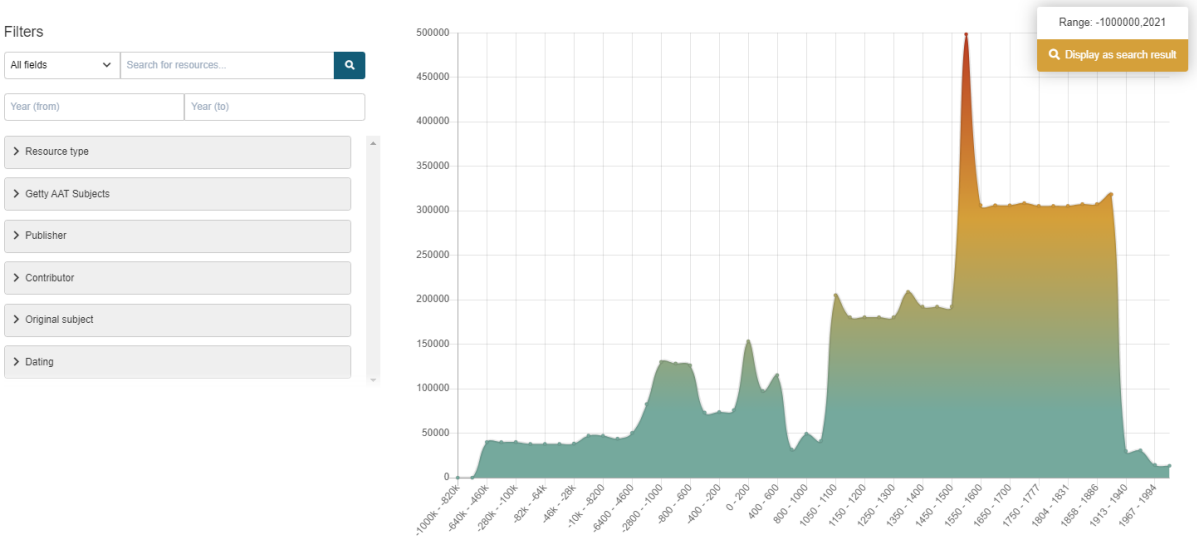


Figure 8 Time line search

The “where” keyword represents the physical location a resource is covering and is enabled by a Map based search option where the user can choose an area on a world-wide map to browse resources within that specific area. Several visualization options are available to help the user narrow down the

search results. On a high zoom level, clustered resources are shown using a "heat map"[19], which allows the user to see areas with a high density of resources. On lower zoom levels, marker clusters or individual markers are shown where the density of resources for each map tile is lower. Additional visualization options are available using various map layers where the user can choose a suitable option for their preferences. As with the timeline search, additional filtering is possible using facets.

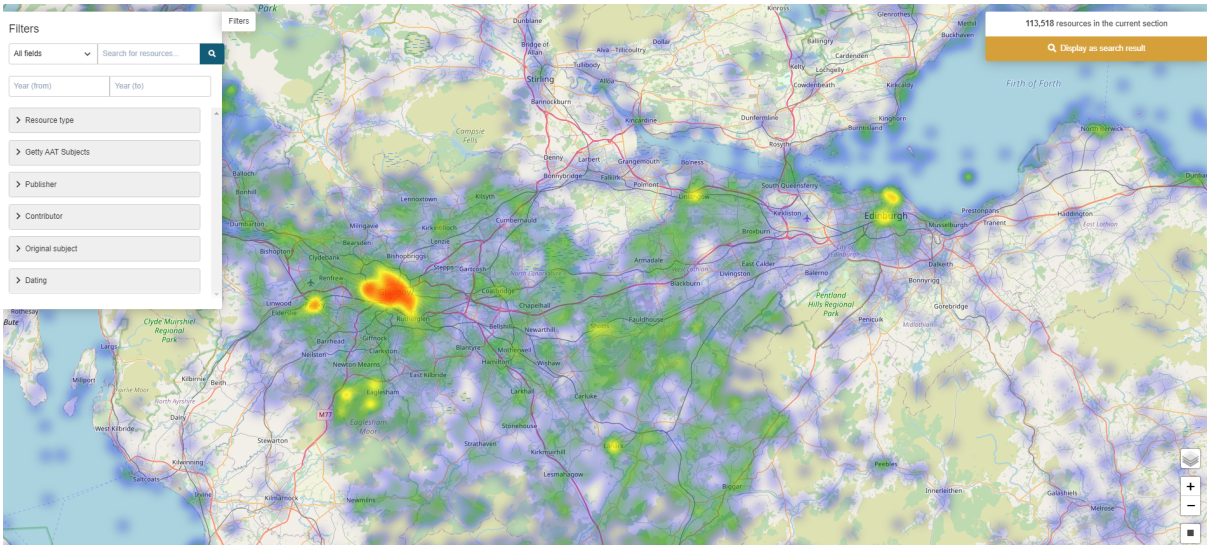


Figure 9 Map search with heatmap

The "what" keyword is represented by a word cloud using terms from the Getty Arts and Architecture Thesaurus (Getty AAT)[20]. The topical coverage of the resources is normalized in the ingest phase so that all resources are using the same vocabulary. The user can, using the word cloud, explore resources by clicking on terms of their interest. Size and colors of the terms in the word cloud are calculated based on the number of resources using each respective term. A word cloud is a non-sophisticated way of searching and will be explored as a complement various other search options using Getty AAT in the next phase of the development of the portal.

[19] A "heat map" represents graphically the number of resources available in one point, with colors ranging from green (few) to red (many).
[20] https://www.getty.edu/research/tools/vocabularies/aat/

Figure 10 Word cloud

Resource landing pages have been developed to include as much information as possible about the resources. This includes presenting all useful metadata available about the resource. Links to the original resource located at the provider are readily accessible, and a map showing the resource is provided if the metadata contains spatial information.

The user is also given the opportunity to find similar resources, either by using the map to find nearby resources, or by using the "Thematically similar" feature, where similarity is calculated on a set of different metadata. Much of the metadata provided is also clickable, allowing the user to browse further similar resources.

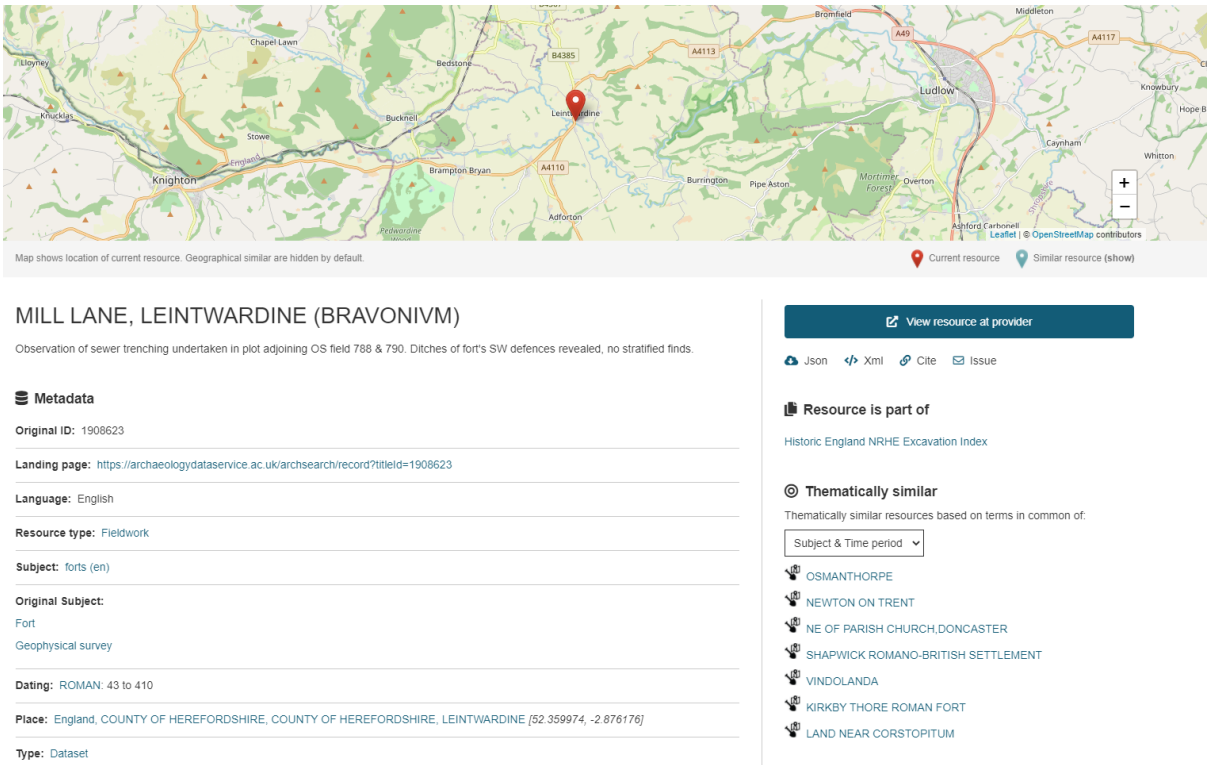


Figure 11 Resource landing page

Source code and licensing

All source code is released on the well-known source-code hosting platform Github[21] under the organization ARIADNE-infrastructure[22]. Source code is distributed using the European Union Public License v1.2[23].

3.3.2 Deviation from the work plan

The launch of the new portal, initially planned in M12 (December 2019) has been postponed due to the major corrections and updates required, as described above. A first beta version has been available to the consortium members since M15 (March 2020). The new version is planned to be launched on 1 February 2021.

3.3.3 Plans for the next period

Development of the ARIADNEplus portal will be ongoing for the remainder of the project. The Portal Working Group will set the priorities for the development during this time. Preliminary studies on possible new features have been conducted to prepare for the advancement of developments. Listed below is a non-exhaustive list of features which are suggested as candidates for development.

Multilingual capabilities

As the multilingual thesaurus Getty AAT is used for enrichment of the ingested metadata in order to harmonize the keywords all resources are tagged with, we have the opportunity to use this ontology in various ways to enhance the findability in the portal:

1. The previously mentioned “what” keyword search options should be enhanced, allowing to browse resources through a multilingual search function of terms found in Getty AAT.
2. The auto-complete functionality of the free-text search should be expanded to include resources found in other languages than original search term, e.g., searching for “sword” should also include search results where “spada” (Italian) or “espada” (Spanish) is mentioned.
3. Multilingual options in the search facets should be added, where possible.

Map search enhancements

As more detailed spatial information is ingested from partners, more advanced features related to the map-based search can be developed.

1. Visualization of polygons and other geometric structures related to the resources on the map - currently only a single point per resource is displayed.
2. Drawing of polygons on map to create a boundary to select resources - currently only a rectangular bounding box is possible.

[21] https://github.com/
[22] https://github.com/ARIADNE-Infrastructure
[23] https://joinup.ec.europa.eu/collection/eupl/eupl-text-eupl-12

3. Distinguishing between actual points of resources on the map and coordinates derived from third-party resources. When partners only provide the name of a geographic location for the resource, a third-party service called GeoNames[24] is used to enrich the metadata with the point for that location. This point, however, is less exact than an actual point.

Temporal search enhancements

Currently, the temporal search function is a timeline based on a start year and an end year, which the resource covers. Considering the fact that archaeological time periods vary enormously from region to region, additional options need to be developed to normalize these time periods and give the researcher the opportunity to find and compare resources belonging to the same time period but in different regions.

3.4 T12.4 – JRA1.4 Item-level data integration

The overall objective of ARIADNEplus is to integrate the datasets of the Archaeological Research Communities. WP4 focuses on implementing the ARIADNE ontology and specific extensions, the application profiles, to specific sub-domains of archaeology and archaeological science. The AO-Cat supports the registration and the integration of data resources into the ARIADNEplus Content Cloud, focusing mainly at collection level granularity. The application profiles, developed in WP4 and WP14 as part of the ARIADNE Ontology, support a finer description of the available resources and thus allow deeper integration.

The objective of Task 12.4 is to provide a deeper integration of data resources by identifying and federating multiple data resources to enable integrated searches. These deeply integrated datasets would then be made available in a specialized Virtual Research Environment (VRE) that would be specifically designed to support specific searches of interest to the relevant research community/group. There is no limit to the extent and scope of deeply integrating resources.

The work is unfolding in three directions:

1. Identifying research questions where relevant data from multiple sources are available.

2. Developing the application profiles, thus adding the expressive power needed to represent these data to the ARIADNEplus ontology.

3. Building the mappings from the original schemas to the so extended ARIADNEplus ontology, using the relevant application profiles, and transforming the data into Linked Open Data (LOD) to be accessed via a SPARQL endpoint, thus making them available to WP16 where their use will be demonstrated.

These activities, in the order given above, can also be seen as the successive steps of a cycle concerning a single research question. As explained below, the execution of one such cycle for each application domain or research question will be the core activity of this Task until the end of the Project.

[24] https://www.geonames.org/

3.4.1 Work performed

The work performed in this task is closely related to the work of WP4 and WP14. During the first year of the project, all the effort went towards dataset aggregation, as described in WP4 and the mappings to AO-Cat. The details of this work are presented in deliverable D4.2 - Initial report on ontology implementation. Following the first mappings, and after validating the suitability of AO-Cat to describe the resources provided, we started compiling a first list of potential resources suitable for deeper integration and started working on their mappings. The following table summarizes the mappings that are currently going on. We also imported the experimental item level mappings that had been performed during the first ARIADNE project, and are revising them accordingly.

Table 1 Ongoing mappings for item-level data integration

ARIADNEplus sub-domain	Data Resource	Data Provider	Mapping in 3M	Comments
4.4.14 Burials	In Touch with the Dead: Early Medieval Grave Reopenings in the Low Countries	OEAW	684	Details for this mapping are presented in D14.1 Section 6.
4.4.14 Burials	Franzhausen-Kokoron (UFK)	OEAW	681, 682	Mapping performed during ARIADNE 1 and now is being revised
4.4.2 Bio-archaeology & Ancient DNA	aDNA lab projects	FORTH-IMBB	687-693	Mappings are available for each step of the scientific analysis workflow of the Allentoft protocol. Details for these mappings are presented in D4.2 Section 6 and D14.1 Section 4.
4.4.0.1 Site/monument	AKB	NIAM-BAS	694	Mapping performed during ARIADNE 1 and now is being revised
4.4.0.2 Fieldwork	Paliambela	PP	668	

3.4.2 Deviations from the work plan

There is no deviation from the work plan. The work in this task started, as expected, during the second year of the project due to its strong dependency to the work performed in WP4 and WP14. Its major activity will be during the second half of the project. The lack of meetings and workshops posed some difficulties and the communication between the involved parties has not been as tight as we would have liked.

3.4.3 Plans for the next period

The development of the ARIADNEplus Catalogue will continue full speed during the second half of the project (years 3-4), until all the partners' collection-level data has been integrated into the ARIADNE Cloud through Aggregation. This implies that new mappings from the local formats of the Catalog data to the AO-Cat will need to be developed. This development is part of this Task.

In parallel, item level integration will also continue at full speed during the second half (years 3-4) of the ARIADNEplus project. As already mentioned, the lack of meetings and workshops posed some difficulties and the communication between the parties involved in the task has not been as tight as we would have liked. In the next period, we intend to strengthen this aspect and have already planned virtual meetings between WP4, WP12, WP14 and WP16 to finalize the use cases and the resources that will be used for the final integration.

The activity on item-level integration will consist in executing the 3-step cycle described above for each relevant application domain or research question. In particular,

- Step number 1: this task will study research questions identified by subtasks in WP4, where relevant data from multiple sources are available, and use them as drivers for the next steps.
- Step number 2: this task will collaborate in the development of application profiles, suggesting ways of capturing the relevant domain knowledge, based on the significant experience accumulated in many years of usage of the CIDOC CRM ontology in the area of Cultural Heritage and of Archaeology. It will also include work on the proper alignment of the application profiles.
- Step number 3: this task will continue developing the mappings from the local formats to the application profiles. This is the core activity of the Task and it is expected that it will extend until the end of the project.

4 Conclusions

Data integration and interoperability call for tools and services capable of addressing different scenarios and requirements. In the first two years of the ARIADNEplus project, WP12 delivered the infrastructure to create, update, enrich and access a knowledge graph containing information collected from the members of the consortium interlinked with standard ontologies of the cultural heritage domain like PeriodO and Getty AAT. The infrastructure has been designed and developed according to the requirements of the consortium, giving high importance to data curation and

automation of the aggregation, enrichment, and publishing processes. In fact, the infrastructure features two environments: a staging environment where the collected data is aggregated, added to a staging AC and indexed on a staging portal where data experts can manually verify the quality of the data. Upon confirmation from data experts, data is then pushed to the production environment, featuring the public ARIADNEplus portal. The staging environment is only available to the consortium members to check data quality and test new functionality of the portal. The tools offered to data experts for mapping the data into the AO-Cat model (3M Editor) and to link it to Getty AAT subject terms (Vocabulary Matching Tool) are fundamental to ensure the quality of aggregated data. Content integration and harmonisation is facilitated by the ARIADNEplus aggregator, built on top of the D-NET software toolkit that enables the realisation of highly configurable and autonomous aggregation systems. The knowledge graph is stored on a GraphDB server and the data is organised in such a way that the graph can be fed automatically by the workflows running on the aggregator.

The knowledge graph is accessible via two endpoints: a SPARQL endpoint for semantic queries and an Elasticsearch service, whose API is used by the new ARIADNEplus portal developed in T12.3.

Task 12.4 started to address the topic of item-level integration, that is a deeper integration of item level data to support research questions that require information richer than what is available in AO-Cat. Major results from this task will be available in the second part of the project.