Grant Agreement ECP-2007-DILI-527003

ARROW

D6.4 Rights information infrastructure - release 2

Deliverable number	*D6.4*
Dissemination level	*PU*
Delivery date	*14-03-2011*
Status	*Final*
Author(s)	*Gabriella Scipione, Cinzia Caroli, Elda Rrapi, Giuseppe Trotta, (CINECA), Sally Chambers, Nuno Freire, Willem Vermeer (TEL-KB).*

e*Content*plus

This project is funded under the *e*Content*plus* programme[1],
a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

[1] OJ L 79, 24.3.2005, p. 1.

Table of content

List of figures

Executive Summary

The objective of the present document is to provide a comprehensive description of the results achieved during the project for the realisation of the Arrow system. The ARROW System is a comprehensive service to support any diligent search model adopted by libraries, by facilitating the identification of rightholders (authors/publishers) and the identification of the rights status of works with particular concern to orphan and out-of-print works. ARROW System is made up of the following macro components that will be described in the following sections:

- Arrow Web Portal Services
- The Rights Information Infrastructure (RII)
- The ARROW Work Registry (AWR)
- The Registry of Orphan Works (ROW).

There where two main releases of the ARROW System: the first release (Beta) in May 2010 and the second Release (Open Beta) in February 2011. The main changes between the two (see § "Release changes") regard the continuous enhancements of the first two components (RII/Web Portal) as well as the creation of the last two ones (AWR/ROW). Between the first and second release of the ARROW System, each group of components passed trough several intermediate releases summarised below:

RII / Arrow Web Portal Releases

Alpha: delivered in March 2010 and including the core services, only German workflow integrated.

- Germany with VLB as BiP and Vg Wort as RROs

Beta: delivered in May 2010 and enhancing and improving both core services and Arrow workflows. Two other European countries have been integrated:

- United Kingdom with Nielsen BookData as BiP and CLA/PLS/ALCS as UK-RROs
- Spain with Dilve as BiP and CEDRO as RRO

The output of the Alpha and Beta releases was delivered in D6.1 Rights Information Infrastructure – 1st release.

Closed beta: delivered in November 2010 including the French workflow:

- France with Electre as BiP and CFC as RROs

This was released to partners and external individuals for technical validation and usability.

Open beta: delivered in February 2011 including the creation of the AWR/ROW (see below) and further improvements at Arrow Web Portal level. This release serves mainly the purpose of demonstrating the system to interested parties and stakeholders of different countries.

The output of the last releases is delivered in the present document.

AWR / ROW Releases

Based on the requirements specified in the deliverable D6.2 Registry of Orphan Works Management System, the AWR/ROW, brand new system components (in respect to the first release), have been set up and passed through the following intermediate releases:

Alpha: delivered in October 2010 and including set up and feeding of the AWR/ROW.

Beta: delivered in February 2011 and including some additional functionality for the management of the AWR/ROW. Further enhancements will performed according to the progress in the definition of the legal framework and the requirements emerging at national level ("System evolution").

Although a global overview of the current Arrow system will be provided in the present document, a clear list of the enhancements and improvements performed in the last releases of RII/Web Portal (open and closed Beta) and AWR/ROW (Alpha and Beta) will be presented in the paragraph "Release changes".

This document starts by providing a brief overview of the ARROW system (§ 1) and afterwards describes thoroughly each single macro component: Arrow Web Portal Service (§ 2), RII (§ 3), AWR/ROW (§ 4). The last chapter (§ 5) reports how the Arrow system has been deployed.

Release changes

This paragraph lists all the changes and new services that have been introduced in the Arrow System from the first release (May 2010) to the second one (February 2011).

RII / Web Portal enhancements

Messages update: a new version of the schemas was released (v 0.26)

- the description of works and contributors was extended (i.e. integration of VIAF data)
- RROs' end point messages was extended

Workflow and Mappings (MARC21XML - ONIX 2.1) has been upgraded accordingly to the above changes.

Work Publishing status algorithm has been further refined.

RII presentation layer (Web Portal) has been reviewed:

- The Validation oriented interface was reviewed in order to display enriched information coming from the workflow
- Validation oriented interface was reviewed in order to improve overall look and feel.
- A new User oriented interface was set up, February 2011

Secondary clusters were handled in order to manage the related works.

VIAF integration was deployed by TEL in September 2010 and the RII was upgraded accordingly.

Clustering algorithm was refined and released

Public Domain Algorithm: Cineca implemented the algorithm that enables to establish if a work is in public domain or copyrighted, according to the "70 years" rule .

Countries updates: four countries were integrated in RII system. Here follows a table summarizing the current RII country data sources and data providers

	Libraries	BIPs	RROs
DE ALFA RELEASE: February 2010	DNB (through TEL)	VLB	VG Wort
FR BETA RELEASE: by September 2010	BNF (through TEL)	Electre	CFC
UK BETA RELEASE: by May 2010	BL (through TEL)	Nielsen Book Data	CLA PLS ALCS
ES BETA RELEASE: by May 2010	BnE (through TEL)	DILVE	CEDRO

DE workflow was updated and is complete.

- VLB: queries have been extended and enhanced by Cineca to handle related works.

FR workflow integration was completed.

- BNF Unimarc records were converted into MARC 21 in July by TEL and therefore ingested in the TEL central index ready for matching and clustering procedures,
- Electre (BiP):
  - Arrow-Electre connector was set up
  - Electre internal workflow was implemented by Electre for the BIP response.
- CFC:
  - Arrow-CFC connector was set up
  - CFC internal workflow was implemented by CFC for the RRO response.

ES workflow was updated and is complete

- BiP(Dilve)-Arrow Connector through CEDRO is complete
- CEDRO internal workflow for the BIP and RRO response was enhanced, tested and fine tuned.

UK workflow was updated and is complete

- BiP(NBD)-Arrow Connector through UK-RRO is complete
- CLA internal workflow for the BIP and RRO response was enhanced, tested and fine tuned.

TEL - RII components (external to the Arrow system)

Implementation of a UNIMARC to MARC21 conversion

- This conversion is applied at The European Library when ingesting the catalogue from a national library. It was deployed for Bibliothèque nationale de France

Manifestation clustering improvements

- Work metadata extraction was improved
    - More strict extraction of subtitles and parallel titles where the ISBD punctuation was not strictly followed by cataloguers.
- Matching of titles was improved
    - Higher emphasis on the first words of the title
    - Different similarity requirements depending on the length of the title
        - More words may differ in larger titles than on smaller ones
        - Special handling for very short titles such as “Le mur” and “Teatro”
- The overall result is more precise clusters
    - Less unrelated secondary clusters
    - More secondary clusters for manifestations with short titles

Contributor data enrichment with VIAF

- Additional information about the work contributors is retrieved from VIAF
    - Name forms, dates of birth and death, and nationalities
- This information is added to the work metadata sent to ARROW on the clustering results
    - Available for Germany, France and Spain

AWR / ROW

The ARROW Work Registry and the Registry of Orphan works have been designed, set up and integrated with the RII workflow. Both registries are continuously fed and updated by the RII workflow. Work Rights Status change over time is also traced.

The following AWR/ ROW services have been implemented:

Search and/or browse services:

- "search and browse" works (Simple Search)
- "search and browse" manifestations (Simple Search)
- browse of work history

Claiming Request service:

- claiming of rights ownership (claiming request)
- browse of own claiming requests and their status – claimer's view

Services for the ROW Management:

- browse of claiming requests
- management of claiming requests: approval or refusal of claiming requests
- updates of the work rights status and history
- notification of the ROW Manager validation result to the claimer

Methodology adopted

For the management of the technical implementation of the project Cineca and The European Library decided to adopt the Agile Project Management[2].

Agile Project Management (APM) can be an effective project management model, if the team works in complex and dynamic environments, even if small, with high volatility of requirements. Small working group were defined:

- one focused on the Arrow system implementation including Cineca and The European Library,
- others focused on the integration of the different services of the data providers and on the implementation of the Arrow connectors including Cineca and the relative data provider (the BiP organisation and the RRO one)

When necessary the team included not only those involved in programming, but also all the bearers of knowledge, all persons that are able to define how the product should be done, particularly AIE or the domain coordinators.

To decrease the risk of failure in software development, Cineca and The European Library proceed with the temporal subdivision of the project in iterations. Each iteration is a small independent project lasting few weeks (a release every 3 weeks) that must have inside all that is needed to release a small advancement in software functionality: planning (planning), requirements analysis, analysis, implementation, testing and documentation. Each single project released as a result of a single iteration does not contain all the features needed to be considered complete. The various iterations, issued one after the other, containing small and continuous improvements allowed to approaches closer and closer to projects needs. The progressive release of iterations allowed to review priorities during construction of project.

The tools adopted by the technical working group are various. Cineca for example has been using Request Tracker[3], an enterprise-grade ticketing system which enables a group of people to manage tasks, issues, and requests submitted by a community of users.

[2] Managing Agile Projects, Sanjiv Augustine, Robert C. Martin Series
[3] Request Tracker website: http://www.bestpractical.com/rt/

1. The ARROW System

The ARROW System is a comprehensive service to support any diligent search model adopted by libraries, by facilitating the identification of rightholders (authors/publishers) and the identification of the rights status of works with particular concern to orphan and out-of-print works. ARROW System is made up of the following macro components that will be described in the following sections:

- Arrow Web Portal Services
- The Rights Information Infrastructure (RII)
- The ARROW Work Registry (AWR)
- The Registry of Orphan Works (ROW).

The figure below shows a schematic representation of the Arrow system. The results and the information collected during the RII workflow form the basis for the AWR and therefore for the ROW which is a subset of the above mentioned AWR as described in the following paragraphs.

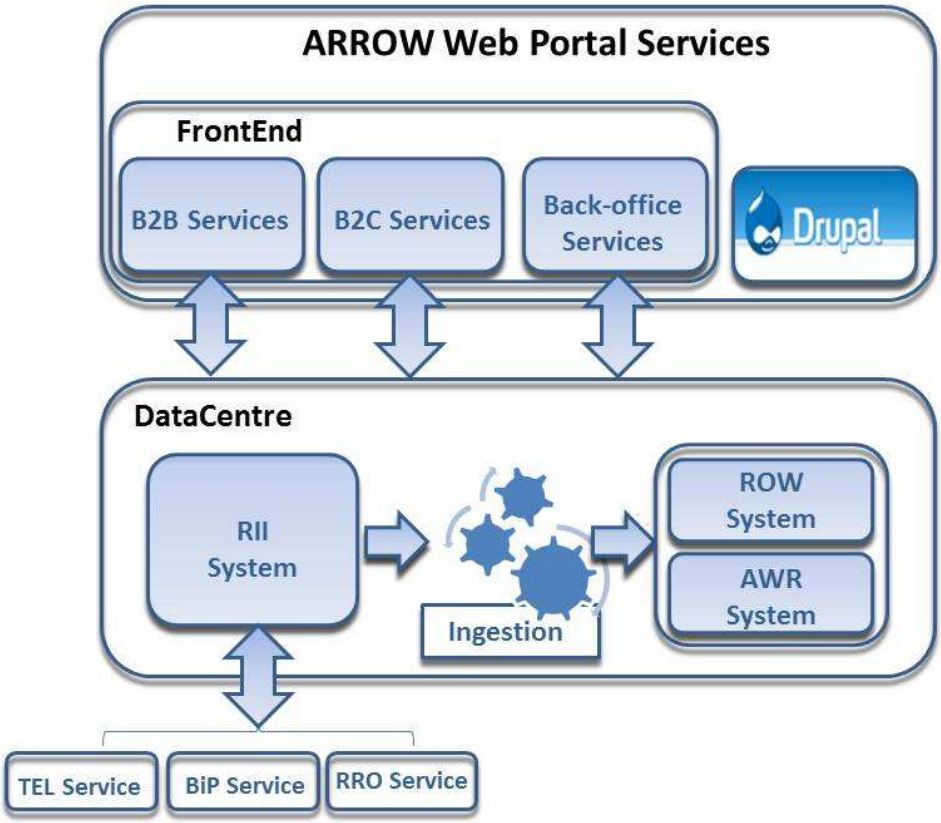


Figure 1: ARROW System

2. Arrow Web Portal Services

The Arrow Web Portal Services comprises the following two main components: the FrontEnd and Drupal Content Management System (CMS).

The FrontEnd is responsible for collecting the user input in various ways, validating and processing it and finally submitting the request to the DataCentre. In other words it represents an interface between the user and the DataCentre. The interaction can be through the Arrow web portal (B2C services) or directly querying the Arrow web service (B2B services).

The CMS is a software tool designed to facilitate the management of the web site content. Arrow uses Drupal as CMS, a free and open source CMS distributed under the GNU General Public License. The Back Office is conceived as a set of services designed to help the management of the entire system, such as the administration of the users and their roles.

The Arrow DataCentre constitutes the back end and performs all the business logic of the entire system, including both the RII and the Arrow Work Registy flow. The business logic of the DataCentre is based on a well defined workflow that requires the exchange of information with other external data providers, exposing data via different interfaces and protocols.

2.1. FrontEnd Software components

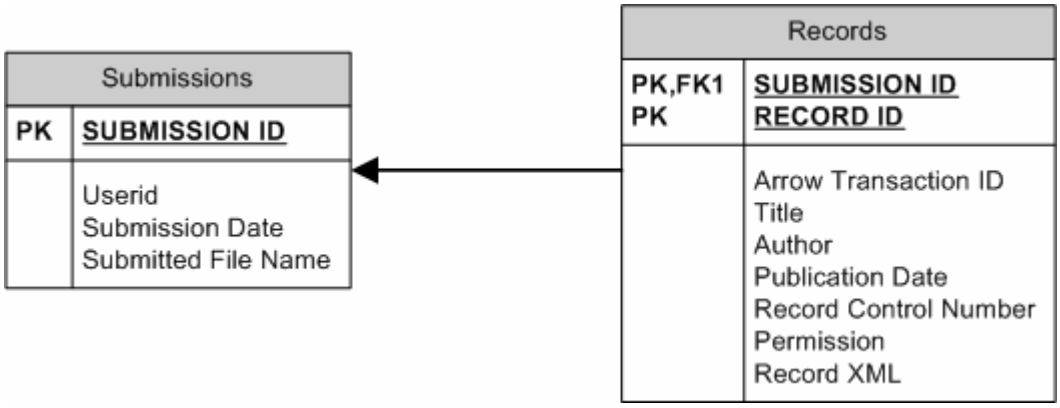
The main components of the Arrow Web Portal Services are hosted in Cineca and are listed in the table below:

Component Name	Description	Hosted at
FrontEnd ArrowPublicService Client	Allows FrontEnd to interact with the DataCentre in order to upload and validate marc requests, get different diligent search results.	Cineca
FrontEnd ArrowReviewService Client	Allows FronEnd to interact with DataCentre in order to obtain partial matches and to forward to DataCentre the user selected match	Cineca
FrontEnd Presentation Layer component	Allows users to submit requests, validate matching results and display submissions information	Cineca
FrontEnd Data Access Object	Allows FrontEnd to interact with data stored in local repositories	Cineca

FrontEnd Security	Enables to verify authentication and authorisation of the current users against Drupal	Cineca
FrontEnd Service	This is the FrontEnd core component. It implements all the business logic.	Cineca
Drupal CMS	Basic features: user account registration and maintenance, menu management, RSS-feeds, page layout customization, system administration.	Cineca

2.2. ARROW FrontEnd Database

The FrontEnd database stores all the relevant information regarding user (i.e. library) submissions. A simplified representation is displayed in the following figure:



3. The Rights Information Infrastructure

The Rights Information Infrastructure (RII) is at the backbone of the ARROW system[4] and the engine that enables ARROW to query and retrieve information from a multiplicity of data providers, in multiple formats, to make the formats interoperable, to process this information and take decisions on the successive elaboration and finally to exchange information according to a planned workflow, described in the figure below.

Building on the RII, the ARROW System receives a request for permission to digitise and use a manifestation of a work (for instance a book) from a library and after querying the data providers included in the workflow (TEL/VIAF, Books in Print, RRO) and elaborating the gathered results, provides information on the work rights status.

To obtain right status information ARROW RII performs three subsequent processes: the TEL process, the BIP process and the RRO process.

- the TEL process in which ARROW exchanges and elaborates the information coming from TEL through the messages M2 and M4,
- the BIP process in which the data coming from TEL are further elaborated and enriched with the information gathered by the BIP through the message M6,
- the RRO process that sends to the RRO the library request enriched by all the data at work and manifestation level collected and processed by the previous data sources through the message M7Q and gather the RRO response in the message M7R.

[4] More details and terms on ARROW RII are provided in D6.1 *Rights Information Infrastructure*

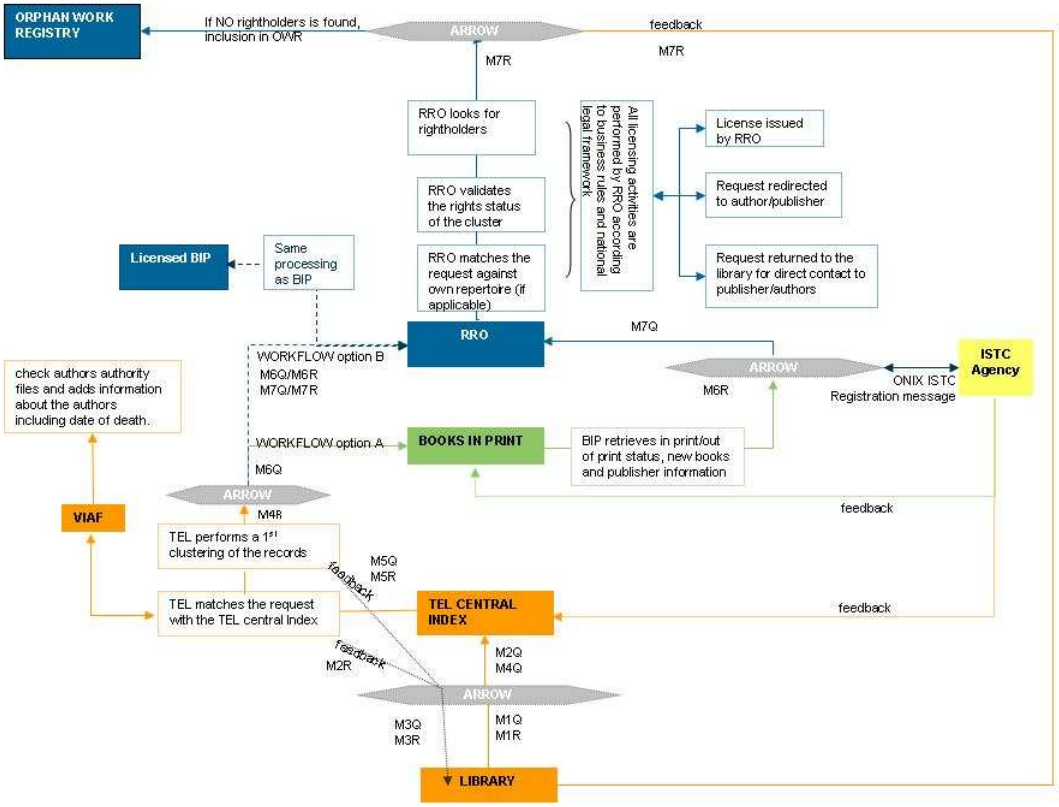


Figure 2: ARROW Workflow diagram

The initial library request is performed at manifestation level[5], whereas the response at the end of the workflow is provided at work level. This means that the initial request passes through stages of identification and matching, work and manifestation clustering and the identification of related works and manifestations; each process adds a piece of relevant information towards the identification of the rights status of the work.

At the end of the ARROW workflow, the following pieces of information have been retrieved in the message exchange and stored in the RII repository.:

- Work information
- Manifestation information

[5] To be more precise the initial library request refers to a "resource", where the term resource identifies an instance of a manifestation, for example a particular copy of a printed edition of a book. For more information about terms used in ARROW, see *D4.3.2 ANNEX II ARROW Glossary of terms* available for downloading in the Resources area of the ARROW website (www.arrow-net.eu)

- Relation between each manifestation and the work they belong to
- Relation between works
- Authors and other contributors information
- Relation between each identified author and the work they have contributed to
- Relation between each piece of information (work, manifestation, author) and the reference source that provided that information (TEL, VIAF, BIPs, RROs)
- A set of so called ARROW Assertions on each work: Copyright Status, Publishing Status and Orphan Status

ARROW Work Registry (AWR) stores and maintains all these pieces of information for every request processed by ARROW.

The Registry of Orphan Works (ROW) is based on a subset of the AWR, respecting specific criteria, that will be made publicly available to specific categories of users for specific purposes.

3.1. The Arrow messages

During the project it has been defined and implemented a set of messages in ONIX format that EDItEUR has developed to support the ARROW project based on the business requirements and feedback of the ARROW technical working group.

The messages are designed to be used by various players in the ARROW workflow and to cover the basic stages in the process, from an initial library request through to the grant or denial of a license. In the current pilot release, Version 0.2, fourteen messages (seven request and response pairs) have so far been defined and deployed. Starting with the initial request from a library, the resulting "transaction" passes through stages of resource identification, work and manifestation clustering and the identification of related resources, before submission to an RRO for a licensing decision. For the time being, the ARROW workflow runs on a national basis based on the country of publication of the target resource. The pilot process workflow concludes with the RRO sending its considered response to ARROW: this may involve the grant or denial of a license and/or the provision of additional information to help the library bring the request to complete resolution.

The table below summarizes the messages in Version 0.2, the name and business purpose of each, and the sender and addressee for each exchange. Note that all the documentation for version 0.2 of

the ARROW Message Descriptions can be found in project deliverable D4.3.2_Specification_metadata_messaging_formats_2010702; the terminology used there is further explained in a companion document, the ARROW Glossary: D4.3.2_ANNEX_II_20100531_ARROWGlossary of terms.

No.	Sender	Addressee	Message name* and purpose
M1Q	Library	ARROW	**InitialResourceAndUsageRequest** Asks ARROW to identify a "target resource" held by the library, assist in locating rightsholder(s) for the corresponding work, and convey details of the usages for which license or other permissions are sought
M1R	ARROW	Library	**InitialResourceAndUsageResponse** Acknowledges the library's request and assigns a persistent transaction identifier to the request to support further operations or inquiries
M2Q	ARROW	TEL	**ResourceIdentificationRequest** Asks a "reference source" (initially TEL) to identify or confirm the identity of a published "target resource" held by a library
M2R	TEL	ARROW	**ResourceIdentificationResponse** Communicates the results of attempts to match details of a library's "target resource" with the reference source's own records
M3Q	ARROW	Library	**ManifestationMatchingReviewRequest** Asks the library to review the results of attempts by a reference source (initially TEL) to match details of the library's "target resource" with the reference source's own records
M3R	Library	ARROW	**ManifestationMatchingReviewResponse** Communicates the results of the library's review of matches submitted by a reference source (initially TEL) between the library's "target resource" and the reference source's own records
M4Q	ARROW	TEL	**ClusterCreationRequest** Conveys details of a particular manifestation and asks a reference source (initially TEL) to return details of one or more clusters of other manifestations/works from its own records that appear to be related to that manifestation
M4R	TEL	ARROW	**ClusterCreationResponse** Communicates the results of attempts to identify clusters of manifestations/works from the reference source's own records, based upon the original manifestation originally submitted
M5Q	ARROW	Library	**ClusterReviewRequest** Asks the library to review the results of attempts by a reference source (initially TEL) to identify clusters of manifestations/works from the reference source's own records, based upon the manifestation originally submitted
M5R	Library	ARROW	**ClusterReviewResponse** Communicates the results of the library's review of clusters submitted by a reference source (initially TEL) based upon manifestations related to the library's "target resource" and the reference source's own records
M6Q	ARROW	BIP	**RelatedBooksInPrintRequest** Conveys details of a particular manifestation and asks the BIP to return details of any manifestations from its own records that appear to be related to that manifestation, together with information on the publishing status and availability of each one
M6R	BIP	ARROW	**RelatedBooksInPrintResponse** Communicates details of any manifestations (or "related ISBNs") identified by the BIP's matching between a particular manifestation and its own records, together with information on the publishing status and availability of each one
M7Q	ARROW	RRO	**FormalLicenseRequest** Identifies a "target resource" for which a library requests a formal license or other permissions. It specifies the usage permissions sought and presents supporting information gathered through the ARROW process, including details of any identified related works or manifestations and the apparent publishing status or availability of each one
M7R	RRO	ARROW	**LicenseProposalOrRefusal** Conveys the RRO's decision or other responses concerning the original request from a library. Responses may include the proposal or refusal of a license, advice that a license is unnecessary, and a range of other advice

3.2. Requirements

There are several actors in the Arrow system: the library, the ARROW system itself that can be seen as composed by two main parts: the Arrow Front End (FE) and the Arrow DataCentre (DC), the data providers that in the actual alpha release are the following: The European Library system (TEL) , the Book in Print (BiP) and the Reprographic Rights Organisations systems (RRO).

In the following paragraphs we will present the Use Cases between the following pairs of actors:

- Librarian – Arrow Front End
- Arrow Front End - Arrow DataCentre
- Arrow DataCentre – External Data Providers.

This can be accomplished by using the use case diagram in the Unified Modelling Language (UML)6 that presents a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

3.2.1. Library - The Arrow Front End

The Librarian as the actor of the Arrow Front End: Use case diagrams for the librarians

The aim of the present paragraph is to show what functions are foreseen at this stage in the Arrow front-end system for an actor like a library. The following figure represents the Use Case diagram that displays the list of functionalities that Front End must provide to the Librarians.

[6] The Unified Modeling Language - UML - is the most-used specification in the field of software engineering.. http://www.uml.org/

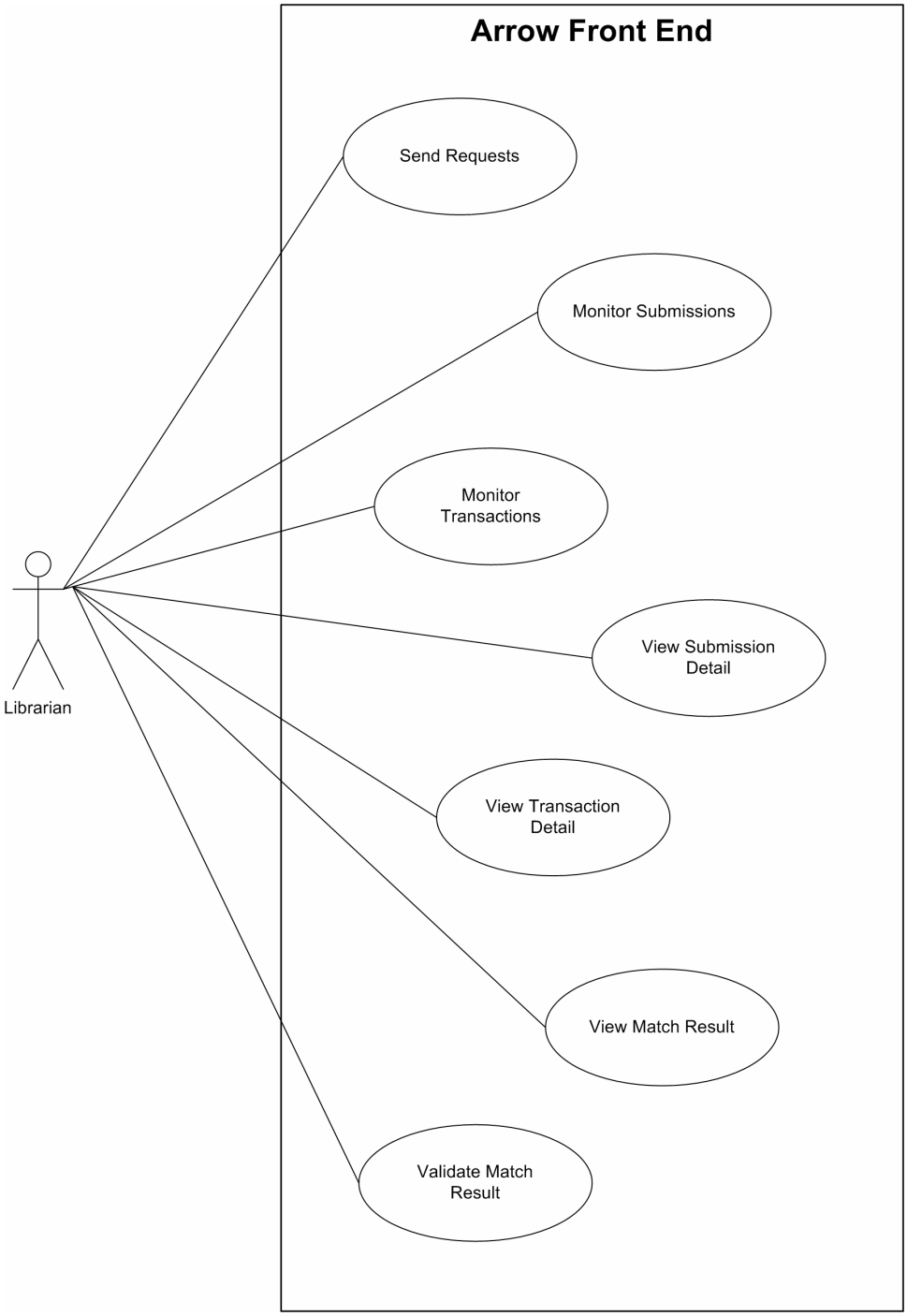


Figure 3: Library Use case diagram

Follows a detailed explanation of the above identified Use Cases. In the Use Cases (1-7) the system refers to Arrow Front End and the two terms may be used interchangingly.

In some of the following Use Cases it is also important to distinguish between the submission ID and the transaction ID. The submission ID identifies the Librarian's request and is used to track and reference it univocally. Since the request may contain one or more book records it is always split, and each book record starts a distinct diligent search. The Arrow Transaction ID identifies univocally the diligent search performed on a book record.

Use Case: Send Requests
ID: UC1
Actors: Librarian
Preconditions: The Librarian has been authenticated by the system
Events sequence: The use case starts when the Librarian selects the functionality "Diligent Search" from the main site. The Librarian can choose to upload a MARC21 XML File or fill a Query Form. 1. In case the Librarian selects "File Upload" (default option), the system displays to the librarian the possibility to upload the file 1.1. the Librarian uploads the MARC21XML File (containing one or more book records) 1.2. the system validates the request file. The validation is performed against the MARC21 XML schemas s well as against a core set of mandatory data (book title, contributor etc...) 1.2.1. In case the file is not valid, the system provides feedback to the Librarian on the validation error. 1.2.2. In case the file is valid, the system splits the original request, in order to retrieve each record. 2. In case the Librarian selects "Query Form" the system displays to the Librarian the form that must be filled by the Librarian 2.1. the Librarian fills in the form fields with the appropriate book record data 2.2. the system checks that all the mandatory fields are filled

2.3. the system generates a valid MARC21 XML record from the form

3. Upon successful validation, the Librarian proceeds by specifying the Permission Set (what rights are requested) and checking the Licensee information.

4. The system provides to the Librarian a summary page of the submission and provides the Librarian the possibility to review its requests before submitting it definitively.

5. The system stores in its database all the necessary information and assigns a submission ID to the Librarian's request

6. The system builds a valid M1Q request for each book record (obtained either via File Upload or Query form) and enters its own Use Case (Upload Request (UC8) – described below) to the Arrow DataCentre.

   When all the M1Q requests have been successfully uploaded in the Arrow DataCentre, the Arrow Front End provides to the library the feedback (submission confirmation) that its request has been successfully loaded into the Arrow System for the necessary elaboration.
   The submission ID related to the Librarian's request is provided as well. Such Id is necessary for the Monitoring service.

7. The system displays to the Librarian the "View Submission Detail" page (see UC4)

Post conditions:
Frontend stores in its database all information about the submitted request and its record(s), among which the Arrow transaction ID returned by the DataCentre.

Use Case: Monitor Submissions
ID: UC2
Actors: Librarian
Preconditions: The Librarian has been authenticated by the system
Events sequence: The use case starts when the librarian selects the functionality "Monitoring area" (submitted requests).in the main site and selects the "Submissions" tab. 1. The system displays: 1.1. the list of all the submissions (default 10 submission per page) performed by the Librarian. For each submission the system displays the Submission ID of the request, Date of the submission, the name of the file originally submitted by the

Librarian as well as a “View Detail”) button.

1.2. a mask for filtering submissions by date

1.3. a search mask

1.4. a paging mask for choosing the number of submission to be displayed per page

2. In case the Librarian enters a start and a stop date, the system filters the submissions and displays the only ones performed in the specified period

3. In case the Librarian enters some term on the search mask, the system filters the submissions and displays the only ones that contain the specified term in one of the fields (Submission ID of the request, Date of the submission, the name of the file originally submitted by the Librarian)

4. In case the Librarian selects a different value in the submission paging mask (i.e. 25), the system formats the submission list appropriately

5. In case the Librarian selects the “View Detail” button the system displays the “View Submission Detail” page (see UC4)

Post conditions:

Use Case: Monitor Transactions
ID: UC3
Actors: Librarian
Preconditions: The Librarian has been authenticated by the system.
Events sequence:

The use case starts when the Librarian selects the functionality “Monitoring area” (submitted requests).in the main site and selects the “Transactions” tab.

The system displays the list of all the transactions performed by the Librarian. For each transaction the system shows:

1. The summary of the book (manifestation) main metadata (i.e. title, author(s), year of publication, country of publications)

2. The status of the transaction or some action to be taken by the Librarian;

2.1. The status of the transaction may be: Just Submitted, Diligent Search in progress or

Completed

2.2. The action to be taken by the Library consists in validating the match result obtained from the Tel

3. The transaction result

3.1. In case the transaction is completed this field contains the Licence Request result (Refused of Granted) and a button Detail that enables the Librarian to see the Transaction Detail (see UC5)

3.2. In case the transaction is not yet completed the corresponding field is empty

Post conditions:

Use Case: View Submission Detail
ID: UC4
Actors: Librarian
Preconditions: The Librarian has been authenticated by the system.
Events sequence:

There are two entry points to this Use Case; it may start as soon as the Librarian sends a request (see last step of UC1) or when the Librarian selects “View detail” for one of the submissions displayed in Submissions Monitoring Area (see UC2).

The system displays:

1. The submission's summary (submission ID, Sent date, original file name)
2. The list of all the transactions contained in the selected submission. For each transaction the system shows:

2.1. The summary of the book (manifestation) main metadata (i.e. title, author(s), year of publication, country of publications)

2.2. The status of the transaction or some action to be taken by the Librarian;

2.2.1. The status of the transaction may be: Just Submitted, Diligent Search in progress or Completed

2.2.2. The action to be taken by the Library consists in validating the match result obtained from the Tel

2.3. The transaction result

2.3.1. In case the transaction is completed this field contains the Licence Request result (Refused of Granted) and a button Detail that enables the Librarian to see the Transaction Detail (see UC5)

2.3.2. In case the transaction is not yet completed the corresponding field is empty

The system automatically refreshes the status/action of each transaction in case it is not completed.

See (UC 10– Get Status List)

Post conditions:

Use Case: View Transaction Detail
ID: UC5
Actors: Librarian
Preconditions: The Librarian has been authenticated by the system and the transaction the Librarian wants to view has been completed successfully.
Events sequence:

There are two entry points to this Use Case; it may start as soon as the Librarian selects "Detail" button for a transaction from the Submission Detail Page or selects "Detail" button for a transaction from the Monitor Transactions page

The system displays a page containing:

1. The summary of the main metadata of the book that the Librarian is asking rights information about (target manifestation)
2. A four tab structure that is used to break the content result into multiple sections: These sections include:
   - ARROW Results
   - Target Expression
   - Related Manifestations
   - Related Works
3. In case the Librarian selects "ARROW Results" tab the system retrieves and displays:

   3.1. The set of ARROW Assertions (Copyright Status, Publishing Status, Orphan status related to the work at which the target manifestation belongs

3.2. The permission requested by the Librarian

3.3. The response to the permission requested by the Librarian and eventual suggested actions

4. In case the Librarian selects the "Target Expression" tab, the system retrieves and displays the metadata of the work (Primary Cluster) at which the target manifestation belongs

5. In case the Librarian select the "Related Manifestations" tab, the system retrieves and displays a list of manifestations, that share the same work with the target manifestation (other manifestations belonging to the same work). For each manifestation the system shows the main set of metadata.

6. In case the Librarian selects the "Related Works" tab, the system retrieves and displays a list of all the "Target Expression" related works (all the Secondary Clusters obtained). For each work the system shows its main metadata as well as the list of its manifestations. For each manifestation shows the main set of metadata

Post conditions:

Use Case: View Match Result
ID: UC6
Actors: Librarian
Preconditions: The librarian has been authenticated by the system. The Matching Process for the current request has been completed and the Status of this submission is "Waiting for Library Validation"
Events sequence: There are two entry points to this Use Case; it may start as soon as the Librarian selects "Please validate" action from the Submission Detail page (UC4) or selects "Please validate" action from the "Monitor Transactions" page (UC3). 1. The system builds a Message request containing the Arrow Transaction ID 2. The system Sends the message to the Arrow DataCentre and obtains a M3Q response message (see UC 11) 3. The system display the formatted result to the Librarian
Post conditions:

Use Case: Validate Match Result
ID: UC7
Actors: Librarian
Preconditions: The librarian has been authenticated by the system. The match of the Librarians book (manifestation) metadata with the records in the TEL Central Index may provide an exact, a partial or a no match result. In case of an exact match the library has to confirm that the found manifestation is the one of his interest. In case a partial match is provided, the library has to select just one manifestation or reject all of them.
Events sequence: 1. The librarian selects among the different manifestations returned in the UC6 (View Match Result) the one he is asking for information. 2. Based on such selection, the system builds a M3R message that contains the Arrow Transaction ID in order to identify the corresponding DataCentre transaction. 3. The system sends the M3R message to Arrow DataCentre and receives an acknowledgement (see UC12) 4. The system displays to the Librarian the Submission Detail page (pertinent to the current transactions). The Status/Action value for the current transaction is no longer “Please Validate”, but “Diligent Search in progress”
Post conditions:

3.2.2. The Arrow Front End – Arrow DataCentre

The Arrow Front End as the actor of the Arrow DataCentre: Use cases for the Front End

In this paragraph are shown the use cases of the Frontend as actor of the DataCentre.

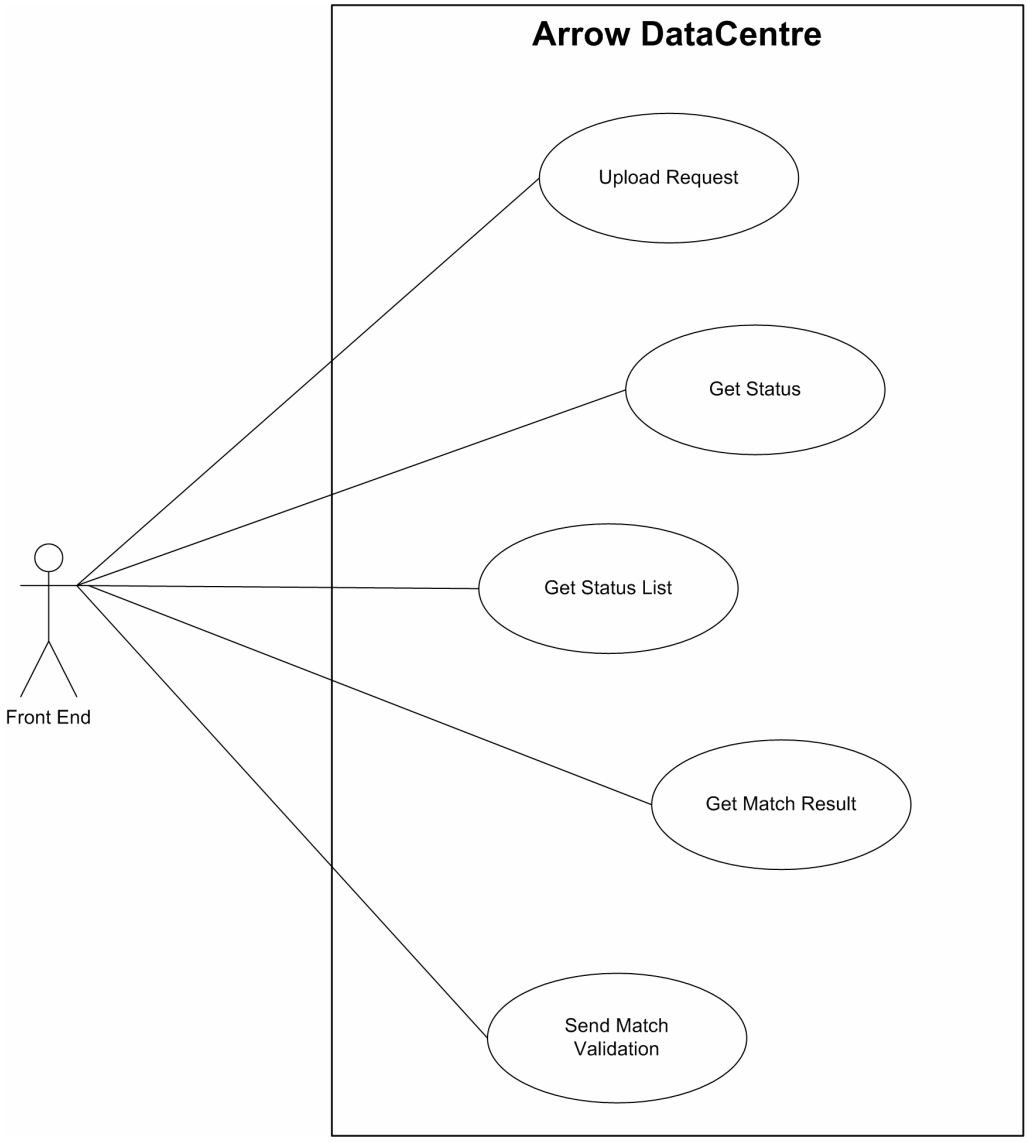


Figure 4: Front End Use case diagram

Use Case: Upload Request
ID: UC8
Actors: Front End
Preconditions: Frontend sends an M1Q request to DataCentre
Events sequence: 1. DataCentre validates the request 2. DataCentre generates the Arrow Transaction ID associated to the request 3. DataCentre puts the request in the correct queue (processing line) for the asynchronous elaboration 4. DataCentre builds M1R message (containing the Arrow Transaction ID) 5. DataCentre sends M1R message to Frontend.
Post conditions:

Use Case: Get Status Request
ID: UC9
Actors: Frontend
Preconditions: The Front End has asked DataCentre the elaboration status of a transaction
Events sequence: 1. Front End builds a Get Status request containing the Arrow Transaction ID to identify the corresponding DataCentre transaction 2. Front End then sends the request to the DataCentre web service 3. DataCentre answers to the Frontend with the Status of that transaction: the Status is composed by a code and a message.
Post conditions:

Use Case: Get Status List Request
ID: UC10
Actors: Front End
Preconditions: The Librarian has asked to view the “Monitor Transactions”(UC3) or the ”Submission Detail”(UC4) page
Events sequence: 1. Frontend builds a get multiple Status request containing more than one Arrow Transaction ID to identify the corresponding DataCentre transactions 2. Front End sends the request to the DataCentre web service 3. DataCentre answers to the Frontend with the Status of all those transactions: the Status is composed by a code and a message.
Post conditions:

Use Case: Get Match Result
ID: UC11
Actors: Front End
Preconditions: The Librarian has asked to view the data obtained by Tel matching (see UC6)
Events sequence: 1. DataCentre validates the request checking that the Arrow Transaction ID does exist and that the corresponding submission Status equals “Waiting for Library Validation” 2. DataCentre builds the M3Q message, retrieving all the manifestations obtained during the Matching Process that are contained in the M2R. 3. DataCentre sends the M3Q message to Frontend
Post conditions:

Use Case: Send Match Validation
ID: UC12
Actors: Front End
Preconditions: Front End sends a M3R message.
Events sequence: 1. DataCentre validates the M3R message 2. DataCentre replaces all the manifestations obtained in the M2R response with the one chosen by the library. This manifestation is eligible to follow on the Arrow workflow. **3.** DataCentre returns an acknowledgement to the Front End.
Post conditions: The unique manifestation that must enter the arrow clustering step is being established.

3.2.3. The Arrow DataCentre – External Data Providers

In this paragraph are shown the use cases of the Arrow DataCentre as actor of the External Data Providers.

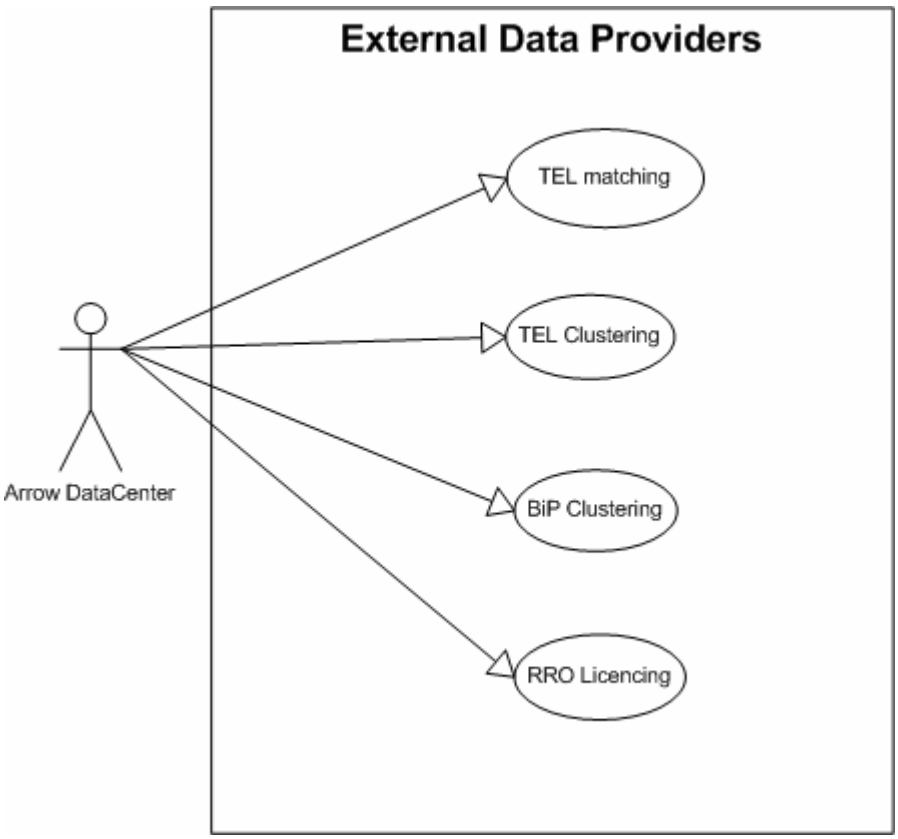


Figure 5: Arrow DataCentre Use case diagram

Use Case: Tel Matching
ID: UC13
Actors: Arrow DataCentre
Preconditions: Frontend has uploaded a valid M1Q request containing a manifestation record in MARC21Xml format. Such request is enqueued to Arrow Request queue
Events sequence: 1. The system updates the Arrow Transaction Status to “Waiting for Tel Matching ” 2. The system builds a correct M2Q message in order to query the Tel match service for the

specified manifestation 3. The system sends the match request to Tel Match service and receives a M2R message. 4. The system stores the M2R 5. If the response contains at least a found matching manifestation (either exact or partial) 5.1. The system updates the Arrow Transaction Status to “Waiting for Library Matching Validation” 6. If there is no matching manifestation from Tel 6.1. The system updates the Arrow Transaction Status to “No Match Result”
Post conditions: Tel match response has been stored and the DataCentre has established the type of Match: “Exact Match” or ”Partial Match” or ”No Match”

Use Case: Tel Clustering
ID: UC14
Actors: Arrow DataCentre
Preconditions: The unique manifestation has been established after the Matching Validation (UC5 and UC10)
Events sequence: 1. The system updates the Arrow Transaction Status to “Waiting for Tel Clustering” 2. The system extracts the from the identified manifestation all the necessary information for building the M4Q message 3. The system builds the M4Q message 4. The system queries the Tel Clustering service (by sending the M4Q message) and receives the response (M4R) 5. The system determines the Copyright Status (ARROW Assertion) for each identified work in the clusters and adds it in the ARROW Assertion of the work. **6.** The system stores the M4R containing clusters information
Post conditions: The TEL clustering response is being stored, and the workflow is ready to enter the BiP Clustering Process (UC15).

Use Case: BiP Clustering
ID: UC15

Actors: Arrow DataCentre
Preconditions: The Tel Clustering response has been correctly obtained, elaborated and stored.
Events sequence: **1.** The system updates the Arrow Transaction Status to “Waiting for BiP Response” **2.** If the BiP that has to be queried is M6 compliant, the system builds a M6Q message starting from the M4R **3.** If the BiP that has to be queried provides its own query API, the system extracts from the M4R the necessary information for performing the query **4.** The system queries the appropriate BiP **5.** If the BiP response is not in M6R, the system builds the M6R message from the gathered results 6. The system determines the work Publishing Status (ARROW Assertion) for each identified work in the clusters and adds it in the ARROW Assertion of the work. 7. The system stores the M6R message
Post conditions: The M6R message containing information on all the clusters (works, related manifestations and ARROW Assertions) is being stored and the workflow is ready to enter the RRO Licensing Process (UC16).

Use Case: RRO Licensing
ID: UC16
Actors: Arrow DataCentre
Preconditions: The BiP clustering response has been correctly obtained, elaborated and stored.
Events sequence: 1. The system updates the Arrow Transaction Status to “Waiting for RRO Response” 2. The system builds the M7Q request using: 2.1. the M6R message built in the BiP Clustering Use Case (UC13) 2.2. the Permission Set (what kind of permission is being requested) and the Licensee information contained in the M1Q 3. The system queries the appropriate RRO (by sending the M7Q) 4. The system obtains the results in M7R format in one of the following ways:

4.1. the queried RRO automatically send the response to the appropriate DataCentre web service (Web Service:: Provider) 4.2. the system itself performs polling to the RRO data provider service in order to retrieve the result. 5. The system stores the result 6. The system updates the Arrow Transaction Status to “Completed Successfully”
Post conditions: The rights status of the work (subject of the library request) as well as the license information has been established.

3.3. RII Software components

In this section we will describe the main RII components included in the Arrow DataCentre and the way they interact with each other. Each of this components provides a different task, such as asynchronous message management; data storage; workflow management, services for querying external data providers as well as services provided in order to be queried by different actors.

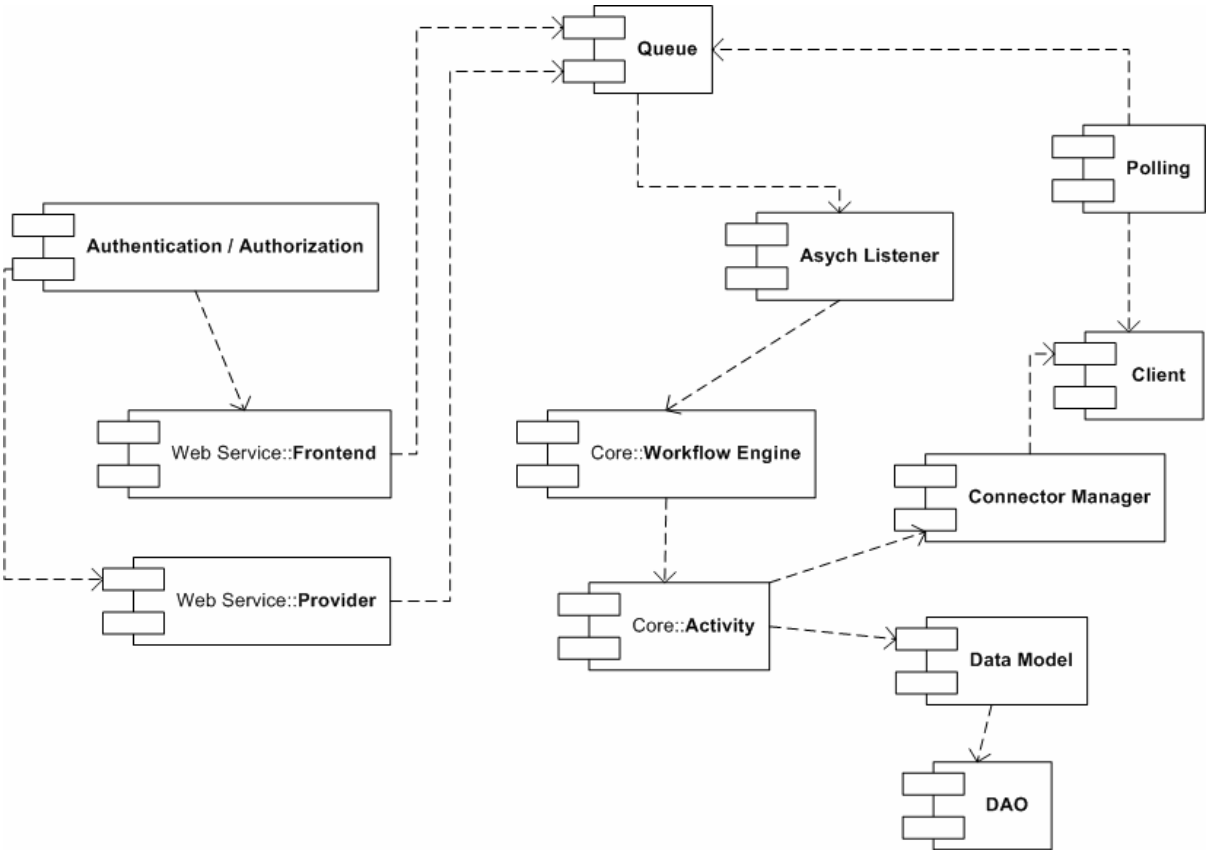


Figure 6: RII Software Components

Below, a short description and functional role of each component and the way they interoperate with each other are listed.

Component Name	Description	Hosted at
RII Authentication / Authorization	This component is a common part of many web applications. It's able to accept and filter only requests coming from authenticated users and with the proper access role.	Cineca
RII Web Service::Frontend	This component exposes several services to the Frontend system enabling it to upload requests, to query request's current status, and to validate requests as needed.	Cineca
RII Web Service::Provider	Like the previous component, this one exposes services for external data providers in order to allow asynchronous message exchange.	Cineca
RII Queue	This component provides asynchronous behaviour so that messages are temporarily cached on a specific queue (e.g. request queue, rro queue, etc.) for computing afterwards. It provides offline asynchronous processing of tasks and activities that can be run independently of the end user experience.	Cineca
RII Asynch Listener	This component is strongly related to the previous one. Its main task is listening on a specific queue and waiting until a new message arrives. When it happens, this component pops the messages from the queue and invokes the appropriate event handler.	Cineca
RII DataCentre::Workflow Engine	This component is surely one of the most important. Its role is to manage submissions from the beginning to the end. It starts activities performing a specific task, handles the request transitions through both synchronous and asynchronous states of the workflow by moving it in a wait state, stores all necessary information and resumes it.	Cineca
RII DataCentre: Activity	This component performs all the business logic of the ARROW system. It contains logic to accomplish all workflow tasks: mainly to query remote data providers and store data locally. As well it contains all decision logic to allow the workflow engine to move from one state to another.	Cineca
RII Data Access Object (DAO):	the RII workflow uses this component to save the incoming data. It has been designed in order to disjoin the application logic from the persistence one. This guarantees that a possible substitution of the database behind (RDBMS, xml file, flat file, …) does not require to modify the application logic.	Cineca
RII Data Model	This component provides definition and format of data that ARROW DataCentre system manages. Data structure is now defined based on all messages that the system needs to handle. Moreover, it contains logic for storing data in a persistent way as well as for accessing them.	Cineca

RII Connector Manager	This component is responsible for carrying out the necessary queries to retrieve data from external services. It uses client components to accomplish this task making proper communication possible with each provider.	Cineca
RII Client	This component is used by Connector Manager to send requests to the data provider. Different services need different communication protocol, so it is able to perform send requests in the right way.	Cineca
RII Polling	This component is responsible for polling remote services that are not able to reply to the ARROW system. Since some computations require unpredictable time, this component needs to query remote service for data retrieval periodically. As for the connector manager, it needs to use the proper client to accomplish its task.	Cineca

3.4. Data model

The ARROW DataCentre, at the current stage, has the main task of managing the whole workflow sending and retrieving data to/from the data providers. All of these data are what the system needs to store, in addition to other data that result from some specific business activity (e.g. publishing status). The resulted data model can be better explained through the following figure.

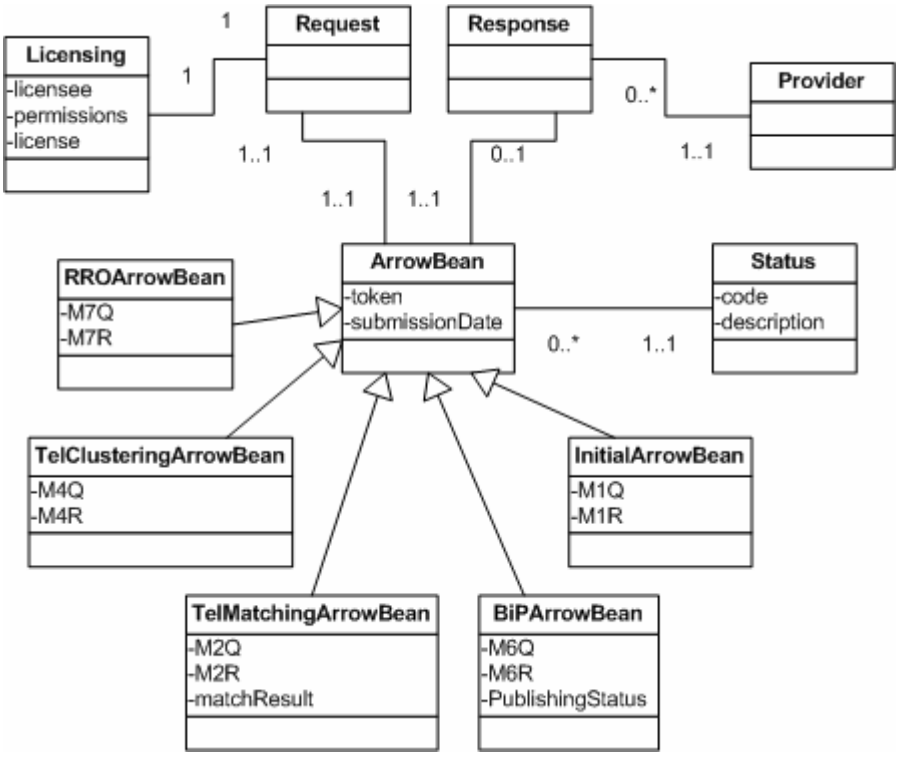


Figure 7: General data model

All submitted requests are bound to a unique arrow transaction identifier (i.e. token) used to retrieve all the needed information. The main data are wrapped into the ArrowBean which is

extended by each specific process bean in order to manage specific request/response messages. By now, only licensing data are kept separately.

3.5. Processes supporting the use cases

In this section we are going to describe the processes that take place in the Arrow system in order to perform all the necessary elaborations requested for the accomplishment of the functionalities requested in the use cases described earlier in this document.

We are going to describe these processes (the communication sequence between actors and system components) using the UML sequence diagrams.

Being that many of the use cases described earlier, are not independent from each other, each sequence diagram that we are going to present, may include more then one Use Case.

The Arrow processes are in many cases asynchronous, that means that some of them are activated upon some particular action (Library validation, external provider asynchronous answer). The order in which the Sequence diagrams will be presented, take in consideration even this asynchronous behaviour.

Sequence Diagram: Initial upload request

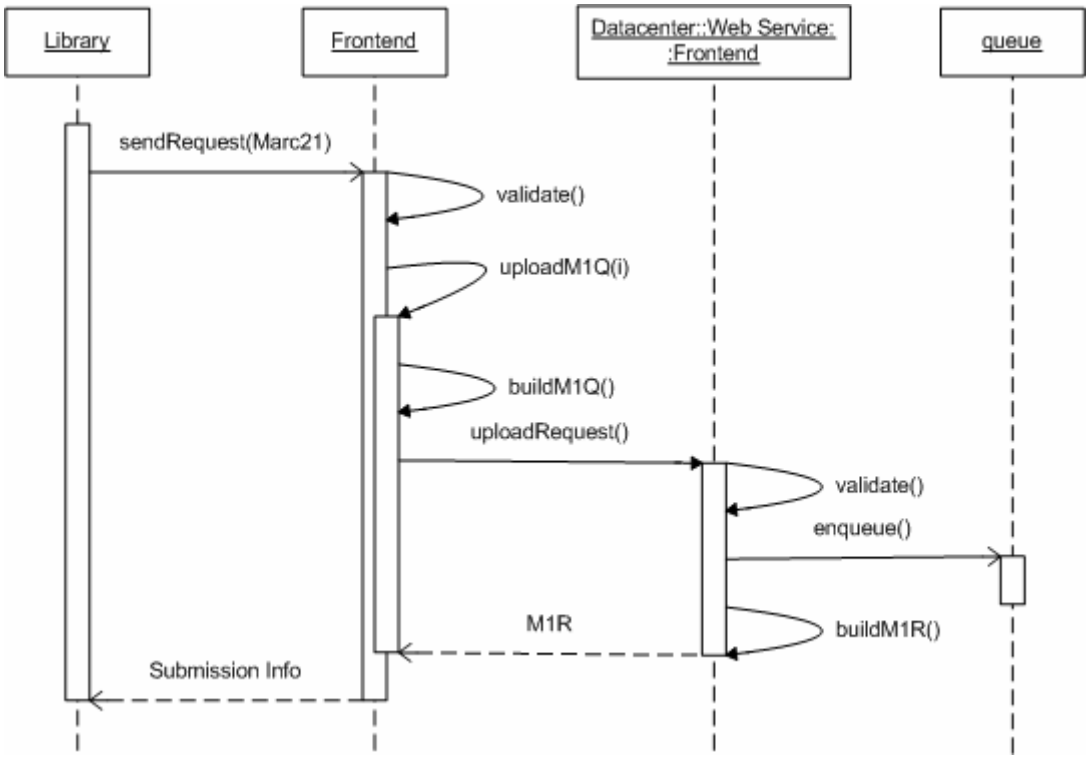


Figure 8: The execution of initial upload request

This diagram shows how the Librarian interacts with both the Arrow Frontend and the Arrow DataCentre in order to upload a request in the Arrow System. It comprises the following use cases:

- UC1 - Send Requests (Librarian-Frontend)
- UC8 - Upload Request (Frontend - DataCentre)

The librarian can send requests either by uploading a MARC21 XML file or filling the Query Form.

In case the Librarian uploads a request file containing a list of MARC21 Xml records, the Frontend checks first of all that the submitted file is valid against the MARC 21 XML schema and then for each of the records checks all the mandatory fields that must be present in order to be able to perform the arrow necessary elaboration. Upon validation completion Frontend splits the file according to the number of the MARC records it includes, and for each of these records builds a M1Q request. M1Q is build according to the message specification provided by EDItEUR for the communication between Library-Frontend (see deliverable 4.3). After the M1Q construction, Frontend invokes the web service (DataCentre: Webservice Frontend) that the Arrow DataCentre exposes in order to enable the uploading of such requests into the Arrow System.

This web service validates the incoming M1Q message, enqueues it in the Arrow queue (the elaboration of the enqueued message will be explained in “Tel matching” sequence diagram) and soon after builds a M1R response that is returned to FrontEnd (according to the message specification provided by EDItEUR).

Upon correct submission of all the records contained in the submission file, Frontend answers back to the Librarian communicating that all the records has been correctly submitted and providing a Submission ID that is necessary to identify the librarian submission.

In case the Librarian sends the request via query form, Frontend checks that all the mandatory fields and builds a valid MARC21 XML file. The rest of the elaboration is the same as in the upload file case.

Sequence Diagram: Tel matching

This sequence diagram supports UC13

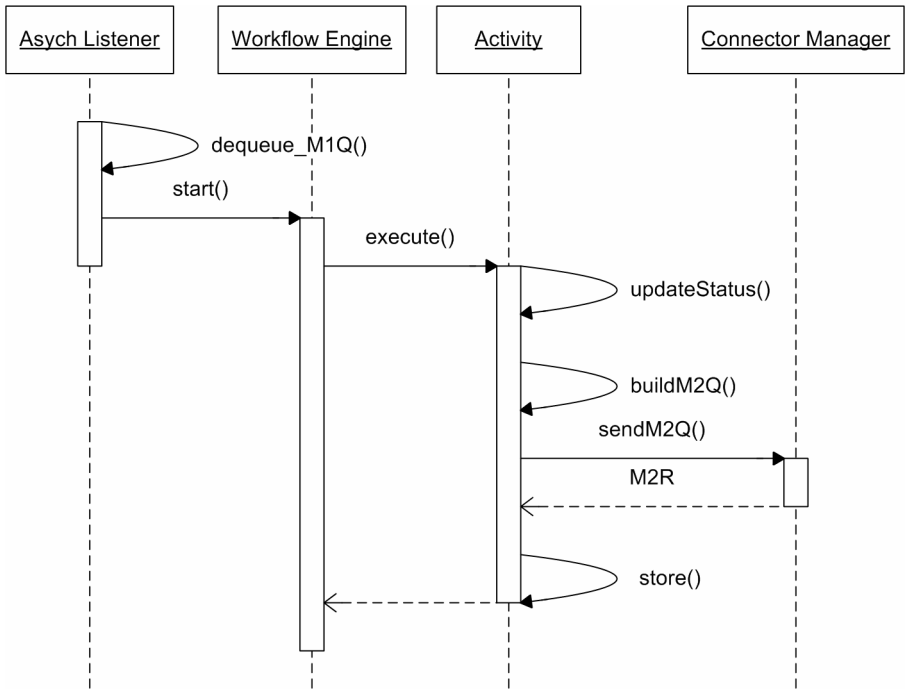


Figure 9: Arrow DataCentre - Execution of Tel matching process

This diagram shows how the Arrow DataCentre interacts with its own components as well as with Tel data provider in order to perform a matching request. The Asynch Listener is a component always listening in the queue where the previous process (initial upload) deposited the request.

As soon as the Asynch Listener pops the request from the queue, the control of the elaboration is taken by the Workflow engine component. According to the type of the request, such engine activates the Tel Matching Activity Manager. The Activity Manager updates the elaboration status of the request to “Waiting for Tel Matching” (The elaboration status of a request is always traced). Further on, Tel Activity builds a Tel Match request (M2Q) according to the EDItEUR specifications regarding Arrow-Tel match communication messages. Upon message build, the elaboration control is taken by the Connector Manager which is responsible for identifying the external data provider to be invoked, loading the corresponding client, sending the correct request message to the matching service provided by Tel in this case, and successively obtain the response. In this case the Tel match response is compliant with the M2R message.

As soon as the ConnectorManager completes its task, the control is returned to the Match Activity Manager which proceeds by:

- Identifying the match type, in case the match is “exact” or “partial” it updates the request status to “Waiting for Library Validation”. In case no match is found in Tel, the request status is updated to “No Match found”
- Storing the result in the arrow repository.

At this stage, the request under consideration is suspended, pending on the validation of the response by the Library. The validation process is described in the following sequence diagram.

Sequence Diagram: Library Validation

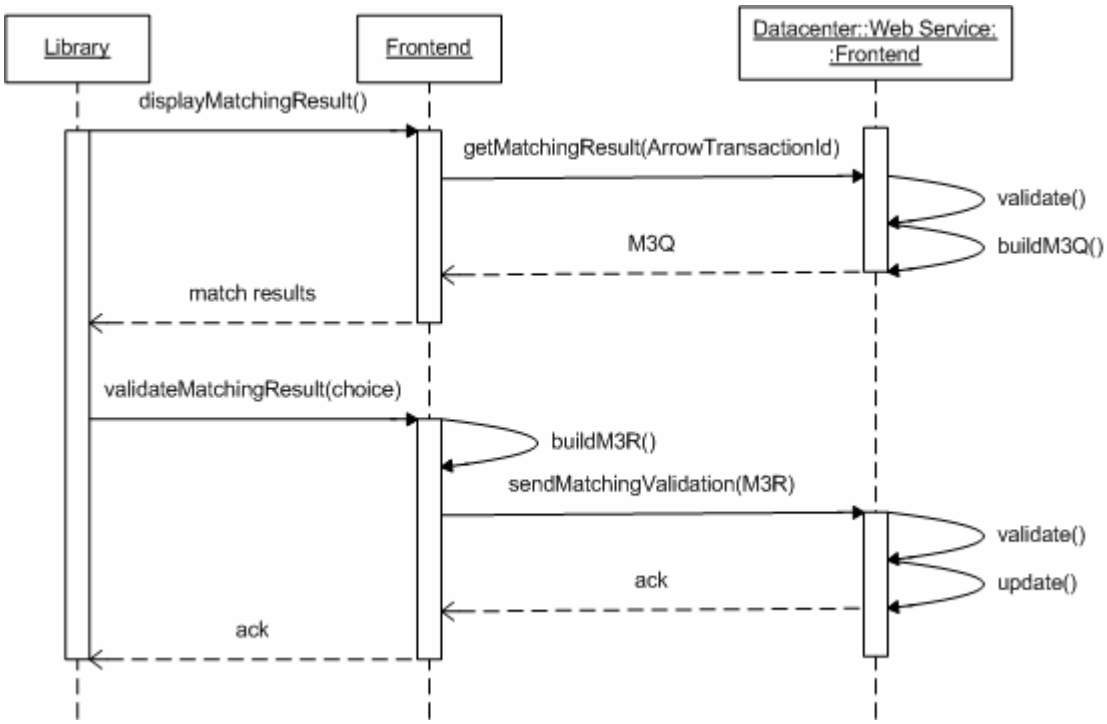


Figure 10: Arrow DataCentre -The execution of library validation process

Each time a Tel exact or partial match response is obtained, the Tel result must be validated by the Library before proceeding with the remaining steps of the Arrow workflow. In case of an exact match the Library has to confirm that the found record in the Tel repository is the one of its interest. In case of a partial match the Library has to select just one record from the whole set of the returned one by the Tel Matching process.

This diagram shows how the Librarian interacts with both the Arrow Frontend and the Arrow DataCentre in order to validate a Tel match response. It comprises the following use cases:

- UC6 - View Match Result (Librarian-Frontend)
- UC11 –Get Match Result (Frontend-DataCentre)
- UC7 - Validate Match Result (Librarian-Frontend)
- UC12 –Send Match Validation (Frontend-DataCentre)

Such uses cases, are represented in the same sequence diagram since are conceptually related to the library validation process and the display of the Tel match results is preliminary to the validation (single manifestation selection)

The validation process starts when the Library wants to view the matches that Tel has provided. As soon as the Frontend receives a displayMatchingResult request, it extracts the Arrow Transaction Id of the request and invokes the web services that Arrow DataCentre exposes in order to provide to Frontend the results of the Tel Matching. The role of the provided web services is to validate the incoming Frontend request, retrieve from the arrow repository the Tel match response, build a M3Q message and send the response to Frontend. Upon response receipt, Frontend formats it and displays the result to the Library (End UC6, UC11).

The Library views the results and selects one of them (Starts UC5). Frontend builds a M3R message based on this selection and sends it to the Arrow DataCentre Webservice for Frontend (UC10). The web service validates such request, replaces all the manifestations returned by the Tel matching with the one selected by the library. A success acknowledgement is sent to Frontend, and Frontend propagates such acknowledgement to the Library.

The end of the validation process awakens the process that was waiting for the library validation (the one suspended at the end of Tel matching process) and now the library validated request is ready to enter the Clustering process that follows below.

Sequence Diagram: Tel Clustering

This sequence diagram supports UC14

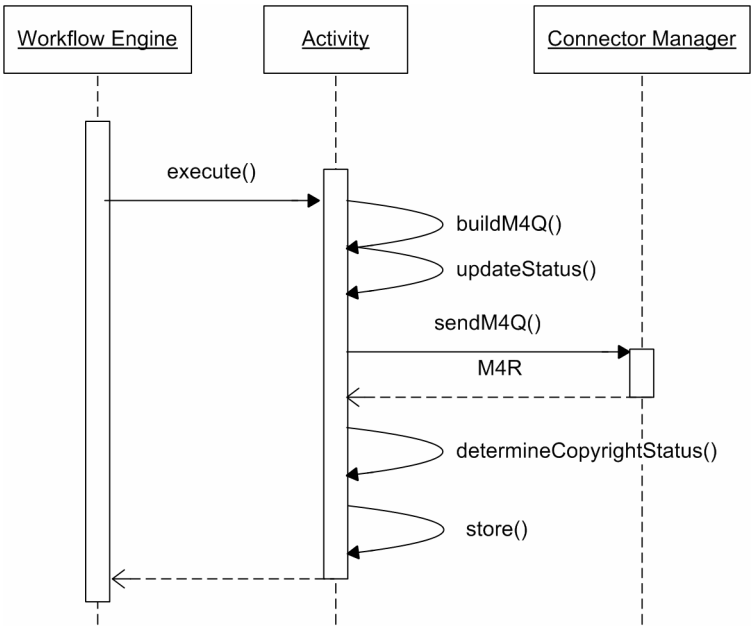


Figure 11: Arrow DataCentre -The execution of TEL clustering

This process begins as soon as the Library ends its validation process. Once the Arrow DataCentre owns the manifestation record that it should further elaborate, the Workflow engine activates the Tel Clustering Activity Manager, whose role is that of: updating the request status to "Waiting for Tel Clustering", extract the Tel manifestation Identifier from the incoming manifestation, build a M4Q message based on it and leave the control to the Connector Manager which is responsible for identifying the external data provider to be invoked, loading the corresponding client , sending the correct request message to the clustering service provided by Tel in this case, and successively obtain the response. In this case the Tel Clustering response is compliant with the M4R message.

As soon as the ConnectorManager completes its task, the control is returned to the Tel Cluster Activity Manager which proceeds by deducing the Copyright Status (see § 3.9.3) for each of the Clusters (Primary and Secondaries) contained in the M4R and finally stores the result in the Arrow repository.

Soon after the control is returned to the Workflow Manager, which is now responsible for handling the BiP Clustering activity that is explained in the following sequence diagram.

Sequence Diagram: BiP Clustering

Once the Tel Clustering data is obtained, the scope of the Arrow workflow is to enrich the Tel Clusters with data gathered by the BiP querying (find in the appropriate BiP all the manifestations that match with the ones found in Tel Clustering) and to establish the Publishing Status for each work contained in these clusters. The system is able to query either BiPs that provide their own query API or BiPs that are M6R compliant (information exchange format is performed via M6 messages). The communication sequence between the involved actors and system components may vary according to the BiP that has to be queried. For greater clarity, in this paragraph, two separate sequence diagrams will be provided in order to support the BiP Clustering process (UC15): the first one describes the communication sequence in case the DataCentre has to query VLB (own API) and the second one describes the communication sequence in case the DataCentre has to query a M6 compliant BiP (ELECTRE, CEDRO, CLA etc..)

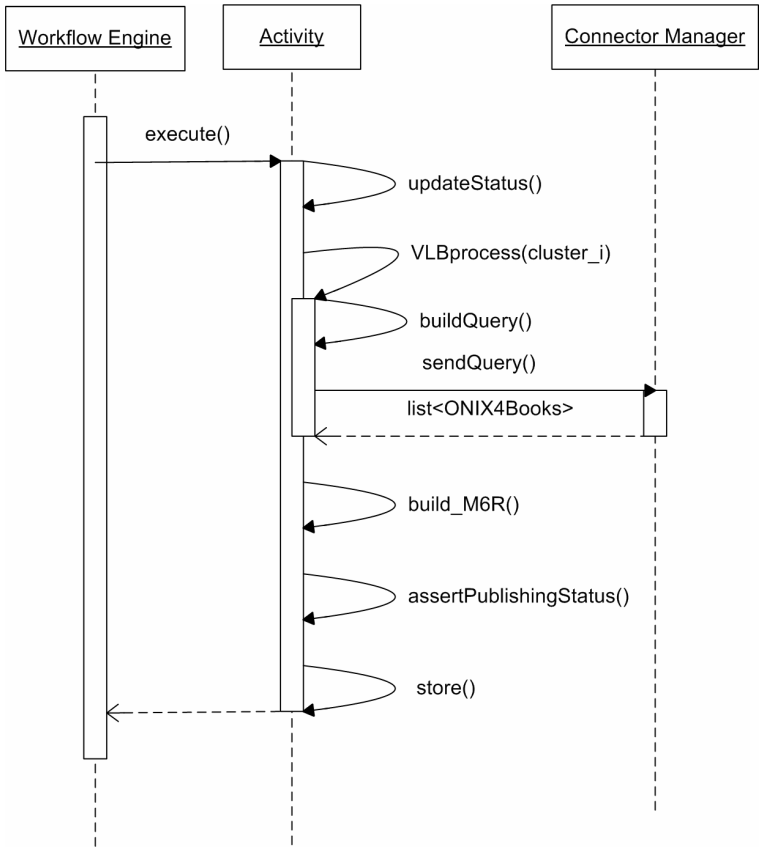


Figure 12: Arrow DataCentre -The execution of BiP clustering process (1/2)

The Workflow engine activates the BiP Clustering Activity Manager. The role of this Activity Manager is that of: updating the status of the submission to “Waiting for BiP Response”, identifying the appropriate BiP to be queried, and consequently extracting the necessary data for querying it. In this case, the BiP Clustering activity extracts all the necessary data from the Tel Clustering Response which contains MARC21 data and builds the queries for Vlb according to Vlb specific API. The Vlb service is invoked as many times as necessary in order to gather the necessary results for all the clusters. Once the querying is over the BiP Clustering Activity Manager proceeds with generating an M6R message based on the results of Tel-BiP Processes. While creating it, the Activity Manager also performs all the necessary transformations from ONIX4Books to ShordDescription (since the BiP manifestation metadata is provided in Onix4Books format. As soon as the M6R is generated, the Activity Manager proceeds by deducing the Publishing Status (see § 3.9.3) for each of the Clusters (Primary and Secondaries) contained in the M6R and finally by storing the result in the Arrow repository.

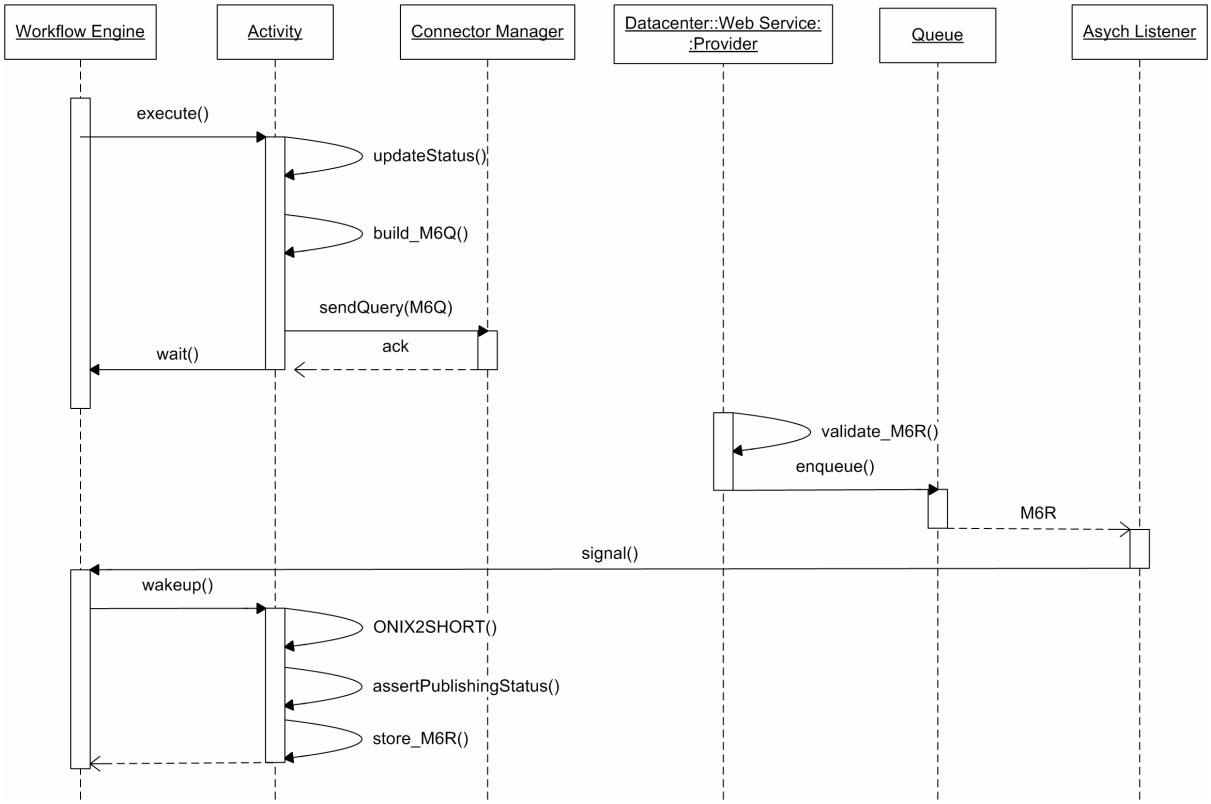


Figure 13: Arrow DataCentre -The execution of M6 compliant BiP clustering process (2/2)

Even in this case, the Workflow engine activates the BiP Activity Manager which is responsible for : updating the request status to “Waiting for BiP Response”, building a correct M6Q request based on the M4R message obtained from TEL. As soon as the M6Q request is built, the control is passed to the Connector Manager that is responsible for sending the request to the appropriate BiP provider (in this case ELECTRE, CEDRO, CLA). Since BiP service is asynchronous, as soon as the request is uploaded in their service the ConnectorManager returns the control to the BiP Activity Manager, and this one to the Workflow engine. At this stage, the request under consideration is suspended, pending on the asynchronous response from the BiP.

Once the BiP has elaborated the requests, it invokes the DataCentre Web Service::Provider in order to send the M6R response. The Web Service::Provider, validates the incoming message against M6 schema and enqueues it in an Arrow DataCentre queue. The Asynch Listener upon new message receipt awakens the suspended process described above. The control is taken by the Workflow engine, which activates the BiP Activity Manager. The Activity Manager proceeds with performing the necessary transformations from ONIX4Books to ShordDescription; by deducing the Publishing Status (see § 3.9.3) for each of the Clusters (Primary and Secondaries) contained in the M6R and finally by storing the result in the Arrow repository.

In both cases (Figure 12 and Figure 13), soon after the Activity Manager stores the M6R, the control is returned to the Workflow Manager, which is now responsible for handling the RRO Licensing activity that is explained in the following sequence diagram.

Sequence Diagram: RRO Licensing

The information exchange format with the RROs is M7, but since the communication modality may vary from RRO to RRO, for greater clarity, in this paragraph, two separate sequence diagrams will be provided in order to support the RRO Licensing process (UC15): the first one describes the communication sequence in case the DataCentre has to perform a polling at the RRO service (VGwort) in order to retrieve the results (M7R) and the second one describes the communication sequence in case the RRO service itself sends the response (M7R) to the DataCentre Web Service::Provider (ELECTRE, CEDRO, CLA etc..)

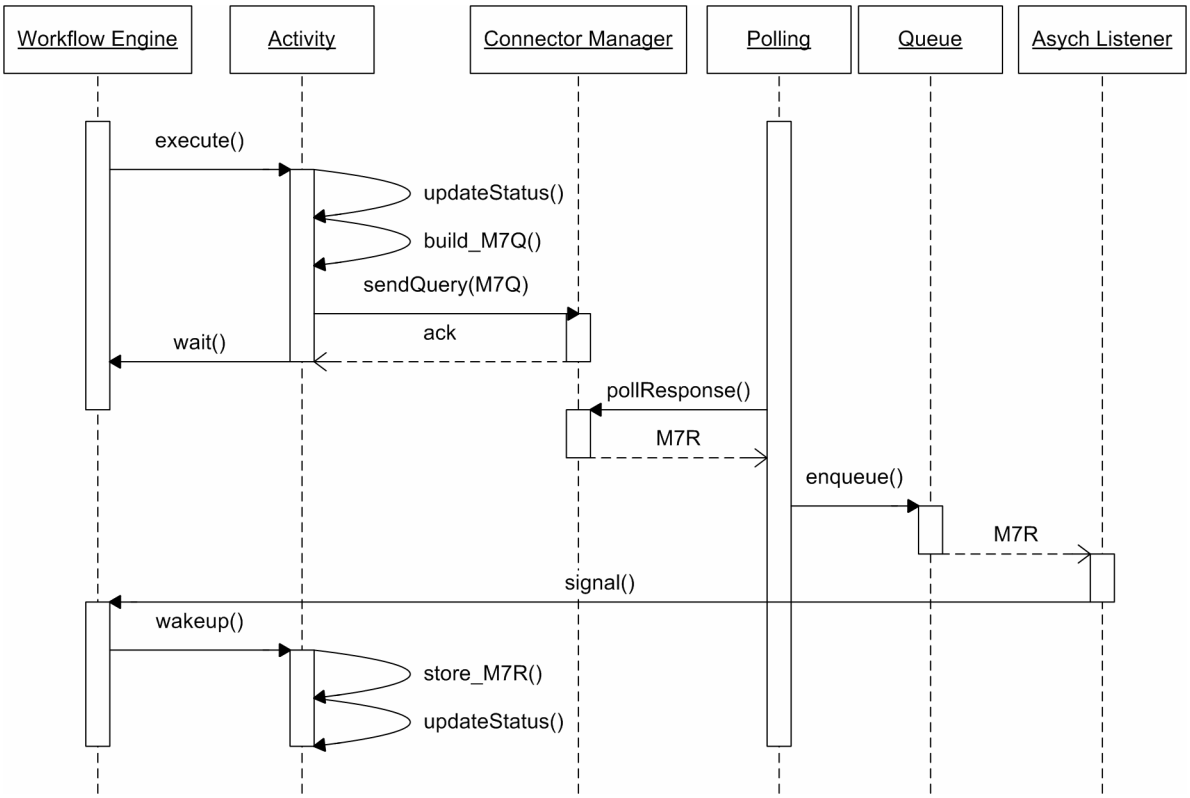


Figure 14: Arrow DataCentre -The execution of RRO licensing process (1/2)

The RRO Licensing process starts as soon as the previous activity has been completely accomplished. The Workflow engine activates the RRO Activity Manager which is responsible for: updating the request status to "Waiting for RRO Response", building a correct M7Q request based on the M6R constructed in the previous process and the licensee requestor retrieved by the M1Q request. As soon as the M7Q request is built, the control is passed to the Connector Manager that is responsible for sending the request to the relevant RRO provider. Since RRO service is asynchronous, as soon as the request is uploaded in their service the ConnectorManager returns the control to the RRO Activity Manager, and this one to the Workflow engine. At this stage, the request under consideration is suspended, pending on the asynchronous response from RRO.

Arrow is gathering the RRO responses by performing a continuous polling in their system. As soon as the Polling Manager founds an elaborated request in the RRO service, it retrieves the response (M7R message) and enqueues it in an Arrow DataCentre queue. The Asynch Listener upon new message receipt awakens the process described immediately above. The control is taken by the Workflow engine, which activates the RRO Activity Manager. The manager is responsible for storing this result

in the Arrow repository and updating the request status to “Completed Successfully”. Here the arrow workflow ends.

The following diagram displays the communication sequence in case RRO service itself sends the response (M7R) to the DataCentre Web Service::Provider (ELECTRE, CEDRO, CLA etc..).

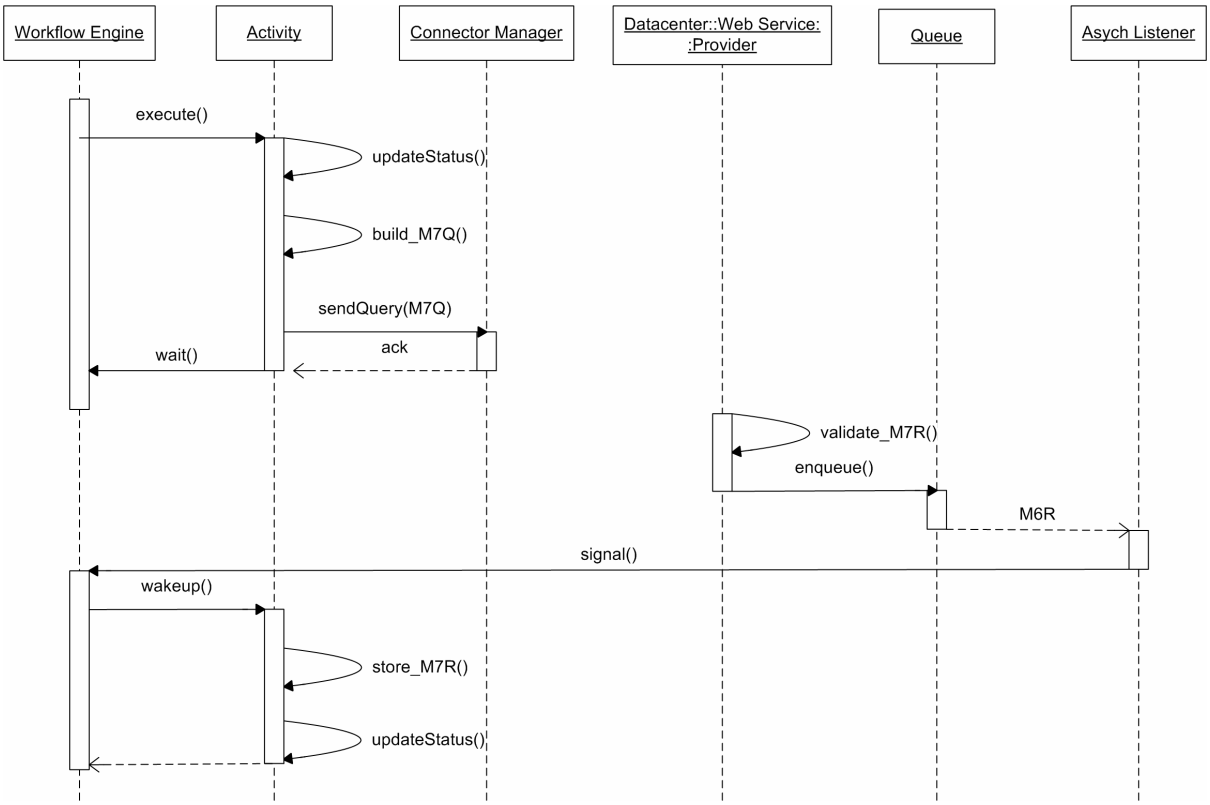


Figure 15: Arrow DataCentre -The execution of RRO licensing process (2/2)

3.6. Other Arrow RII components external to the Arrow system (TEL Service, BIP Service, RRO Service)

3.6.1. The role of The European Library

Bibliographic data from the catalogues of Europe’s national libraries is one of the key data sources in ARROW. Originally, it was envisaged that the ARROW system should query each of the national library catalogues separately. However, as such bibliographic data is already being aggregated

through The European Library[7], it was decided that by enabling the ARROW system to query a single access point to catalogues of the National Libraries of Europe, the complexity of the ARROW system can be significantly decreased. With this solution, interoperability would only be necessary between the ARROW system and The European Library instead to each of the European National Libraries catalogue systems.

In addition, as The European Library network extends to 46 of the 48 National Libraries in the Council of Europe, scaling the ARROW system to all the countries in the European Union would not be an issue. From a sustainability view point, The European Library is also a good choice, as it is built on a sustainable business model, paid for directly from subscription fees by the National Libraries themselves.

The European Library system for ARROW serves three main purposes:

- It allows the libraries to identify the bibliographic record describing the manifestation whose rights are to be cleared. The bibliographic record should be identified in the catalogue of The National Library of the country where it has been published.
- It identifies all other manifestations that potentially share, in part or totally, intellectual work with a manifestation, for further processing in the ARROW workflow.
- It provides detailed information about the contributors of works, as it exists in national libraries' authority files.

Therefore, the role of The European Library is centred on the processing of bibliographic data. To fulfil its purpose, the TEL system makes extensive use of data cleaning, information extraction, and duplicates detection techniques.

In spite of many great efforts that libraries undertake to standardize cataloguing practices and bibliographic data formats, we still find very heterogeneous data.

[7] Launched as an operational service in March 2005, The European Library provides a single point of access to the bibliographic and digital collections of the National Libraries of Europe. In spring 2010, 46 from the 48 national libraries in Europe have made their collections available in The European Library. The 7 national libraries, who are partners in the ARROW project, are all full-members of The European Library network. For more information about The European Library please refer to: http://www.theeuropeanlibrary.org/

Cataloguing rules still leave room for different interpretations, and the information that libraries record in their catalogues is often too complex to be encoded retaining its complete semantics for machine processing. Librarians often have to resort to general note fields to record specific information, or they have to work with the limitations of the information systems that are not always up to date with the standards, or do not fully enforce and validate cataloguing practices. Therefore the same information may be represented quite differently from library to library, and even within the same library.

Library data has been created for several decades. For this reason, the catalogues of the National Libraries contain legacy data which was created in accordance with older cataloguing rules or in older library management systems.

In addition to the data encoding practices in the library domain, these data sources are also subjected to typical data quality problems: typing errors, misspellings, synonyms, homonyms, abbreviations, etc.

Consequently, in ARROW comparison and matching of bibliographic records has to be carried out with special respect to the heterogeneity of this data. Without it, the task of right clearance could be performed on incomplete information. For this reason, the TEL system was designed to minimize the chances of missing any relevant manifestation for the process of rights clearance.

The European Library system also serves as a gateway between libraries authority files and ARROW, by its use of VIAF. VIAF is a joint project of several national libraries aiming to increase the usefulness of library authority files. In the context of ARROW, VIAF is relevant due to its data about the contributors of works. The authority files from all libraries are matched, linked and published on the Web, both in human and machine readable formats.

Bibliographic records contain references to persons and/or corporate bodies that contributed to the work described in the records. These references are short. Typically, they contain at most the name of the contributor, the years of birth and death (in the case of persons) and an identifier given to the contributor in the authority file of the library. If a contributor uses different forms of his name in different publications (for example, in some cases middle names may be omitted, or a pseudonym is used), libraries will adopt one form of the name to be the preferred form and will use it throughout all bibliographic records to refer to that contributor. The other variants of the name and additional details are registered in the authority file.

More comprehensive data about the contributors may be of value for other tasks of the ARROW workflow that take place after the manifestation clusters are formed in The European Library. For this reason, the manifestation records and the work descriptions provided by the clustering process are enriched with data from VIAF. For example, dates of birth and death may be relevant in the rights clearance process of a manifestation and the records from a national library may not have that information, but they may be obtained from VIAF.

This section describes The European Library system of the ARROW infrastructure.

3.6.2. Requirements

There are two actors in The European Library system (TEL system): the ARROW system which sends ARROW workflow requests to The European Library; and the TEL system's administrator who manages the data available in The European Library for processing the ARROW requests.

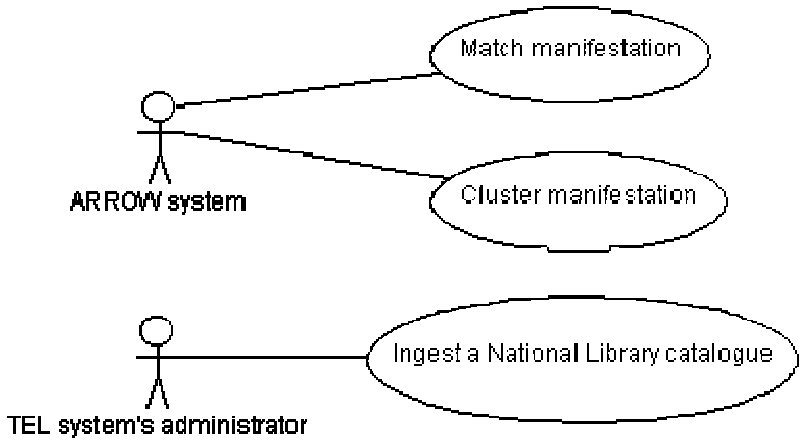


Figure 16: Use Case: Match Manifestation

Use case: Match manifestation

Actors: The ARROW system

Description: Identify the bibliographic record of a particular manifestation within the catalogue of the National Library of the country where the manifestation was published.

Functional requirements:

Input: A bibliographic record about the manifestation. The bibliographic record must have the minimum mandatory data defined in ARROW.

Description: The manifestation should be matched with those within the catalogue of the National Library of the country where it was published. The match must consider that the data in the bibliographic records, both the input record and those in the catalogues, can be incomplete and can have general data quality problems, such as typing errors, use of abbreviations, different cataloguing practices, etc. Identifiers like ISBNs may also be affected by data quality: typing errors, reuse of ISBNs in multiple manifestations, ISBN10 and ISBN13, etc. Therefore matching on identifiers should not be used solely for matching two manifestations. The system should assign a probability score for the matching manifestations.

Output: A list of bibliographic records that may describe the manifestation given in the input record. The list should be sorted by matching probability. If the input record comes from the National Library of the country of publication, then only that specific record should be returned.

Use case: Cluster manifestation

Actors: The ARROW system

Description: Identify all other manifestations that potentially share, in part or totally, intellectual work with a particular manifestation of a National Library catalogue.

Functional requirements:

Input: A bibliographic record from a National Library about a manifestation. The record must exist in The European Library.

Description: The manifestation should be compared to similar manifestations within the same catalogue. Comparisons of the manifestations must take in considerations that the data in the bibliographic records can be incomplete and can contain general problems with data quality like for example typing errors, use of abbreviations, different cataloguing practices, etc. Identifiers, such as ISBNs, may also be affected by data quality: typing errors, reuse of ISBNs in multiple manifestations, ISBN10 and ISBN13, etc. The similar manifestations must be grouped according to the "intellectual work" that they describe (clusters). Those that share the same work with the target manifestation should form the primary clusters. Other manifestations, with possibly related works, should be grouped in secondary clusters. The system should implement a criterion that will form the clusters based on the comparison of the works described in the manifestations. This criterion should form

the clusters according to the requirements of the following steps in the ARROW workflow. Information about work contributors should be complemented with data available in VIAF, such as variant forms of the names, dates of birth and death, and nationality.

Output: All manifestations that share the same "intellectual work" grouped in a primary cluster. Manifestations of related intellectual works, which may be relevant for rights clearance, should be grouped in secondary clusters. Each cluster is described by work level metadata, which includes contributor data enriched by VIAF.

Use case: Ingest a National Library catalogue

Actors: The European Library's systems administrator

Description: Ingest a National Library Catalogue into The European Library system, in order to serve as a bibliographic data source in ARROW.

Functional requirements:

Input: An export of a National Library's catalogue.

Description: Ingest into the system a new or updated export of the catalogue of a National Library. The catalogue should be processed by the system in order to make it available for the manifestation matching and clustering use cases.

Output: The catalogue available in The European Library system for serving ARROW requests.

3.6.3. Software Components

The system is comprised by 6 main software components, as shown in the following diagram (using the notation of the UML component diagrams[8]).

[8] http://en.wikipedia.org/wiki/Component_diagram

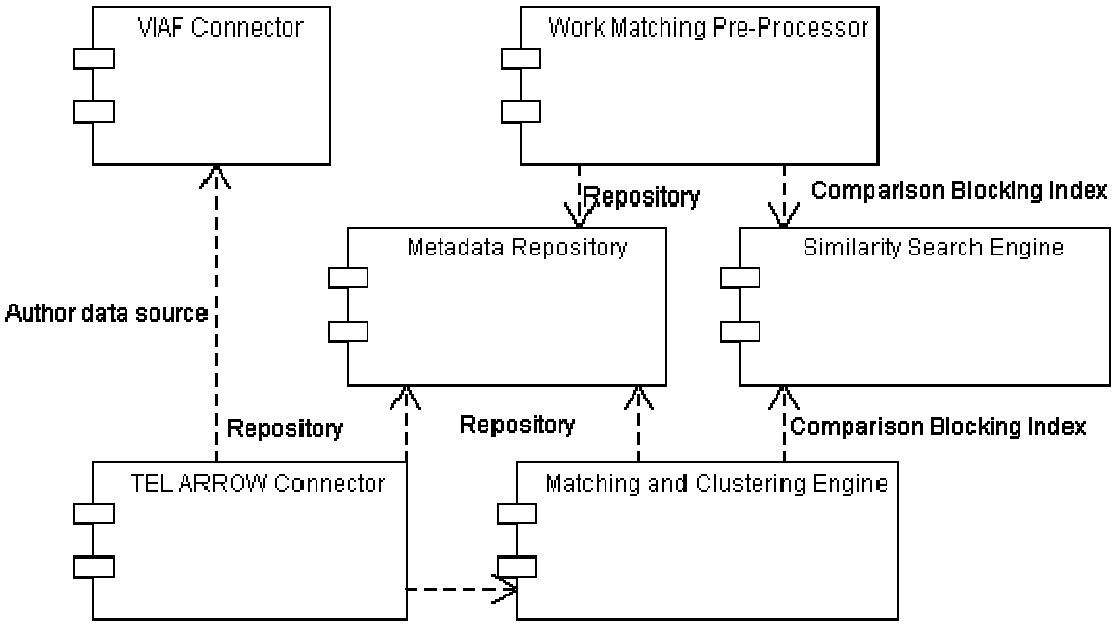


Figure 17: The main software components of the TEL system

The individual descriptions and functional roles of the six components can be found in the following tables:

Component Name	Description	Hosted at
Work Matching Pre-processor	This component is the main processor of the bibliographic data from the National Libraries' catalogues. It addresses all data heterogeneity and data quality issues, so that the rest of the system may function on consistent data formats. It applies data cleaning, information extraction and data transformation techniques. It extracts work metadata from bibliographic records, normalization of work metadata, Data preparation for similarity search.	TEL
Similarity Search Engine	This component provides the means to make similarity queries to work metadata by similarity. It supports the functions of the Matching and Clustering Engine. It allows the processing for matching and clustering requests to be restricted to a subset of the catalogue' data, that is, just to the subset of records that contain a minimal level of similarity at the title and author level with the target manifestation. Function: Indexing and retrieval of works by similarity	TEL
Metadata Repository	This component provides storage for the bibliographic records of the catalogues of national libraries and the corresponding work metadata. Function: Storage of data	TEL

Matching and Clustering Engine	This component is responsible for carrying out the requests from ARROW regarding the matching and clustering of manifestations. It applies duplicate detection techniques that do not require data to be encoded in the same way, or the existence of identifiers, to detect matches between records. Both kinds of requests are a form of duplicate detection, although with some differences. In manifestation matching, it is a case of duplicate detection on the manifestation specific part of the bibliographic records. In manifestation clustering, it is applied to the work metadata extracted from the bibliographic records. Separate criterions, which define how matches and clusters are carried out, were defined for both requests. It supports uncertainty in the matches of manifestations, by calculating a matching probability. For the clusters that it detects, a consolidated work metadata is built, comprising all the work metadata found in the manifestations that form the cluster. Functions: • Execution of the manifestation matching and clustering requests from ARROW • Creation of cluster work metadata	TEL
VIAF Connector	This component provides retrieval of authority records about contributors from VIAF, for enrichment of the work metadata.Function: Client for VIAF's REST web service	TEL
TEL ARROW Connector	The software component responsible for handling the communication between The European Library and Arrow. Function: Interface between TEL and ARROW	TEL

3.6.4. Data model

This section describes the major data units that are processed within the TEL system. The diagrams follow the notation of class diagrams[9] of UML.

The European Library system contains the National Libraries' catalogues with their data organized by manifestations. Typically, each bibliographic record describes one manifestation and any data concerning the intellectual work is not explicit, so it needs to be extracted. Therefore, any data about works within The European Library system is extracted from the bibliographic records that describe manifestations.

[9] http://en.wikipedia.org/wiki/Class_diagram

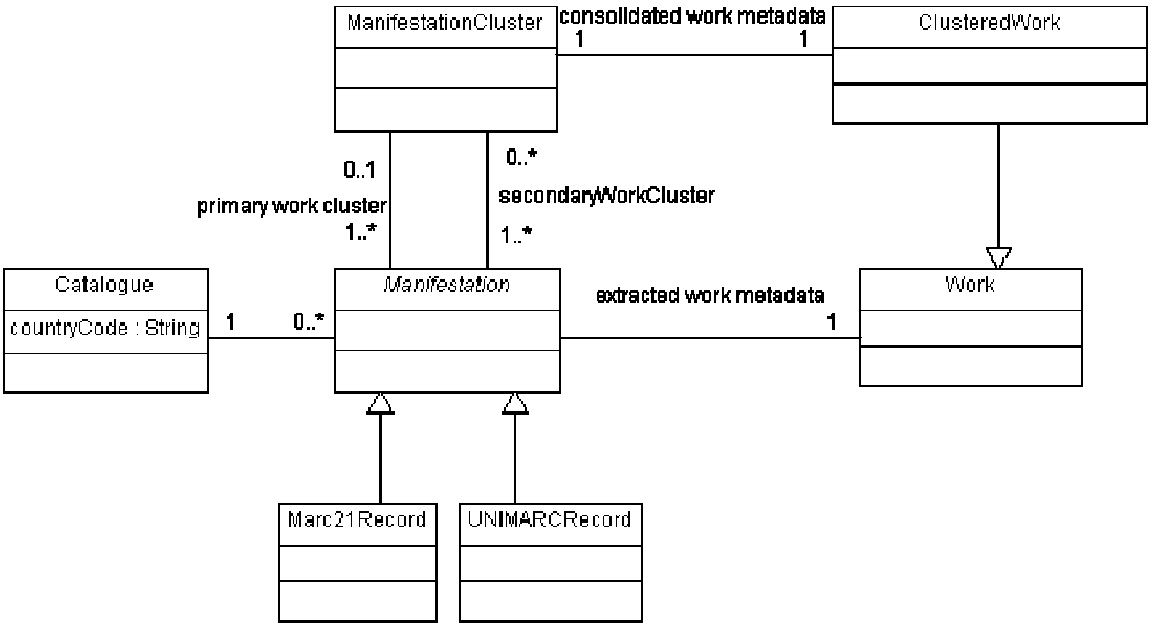
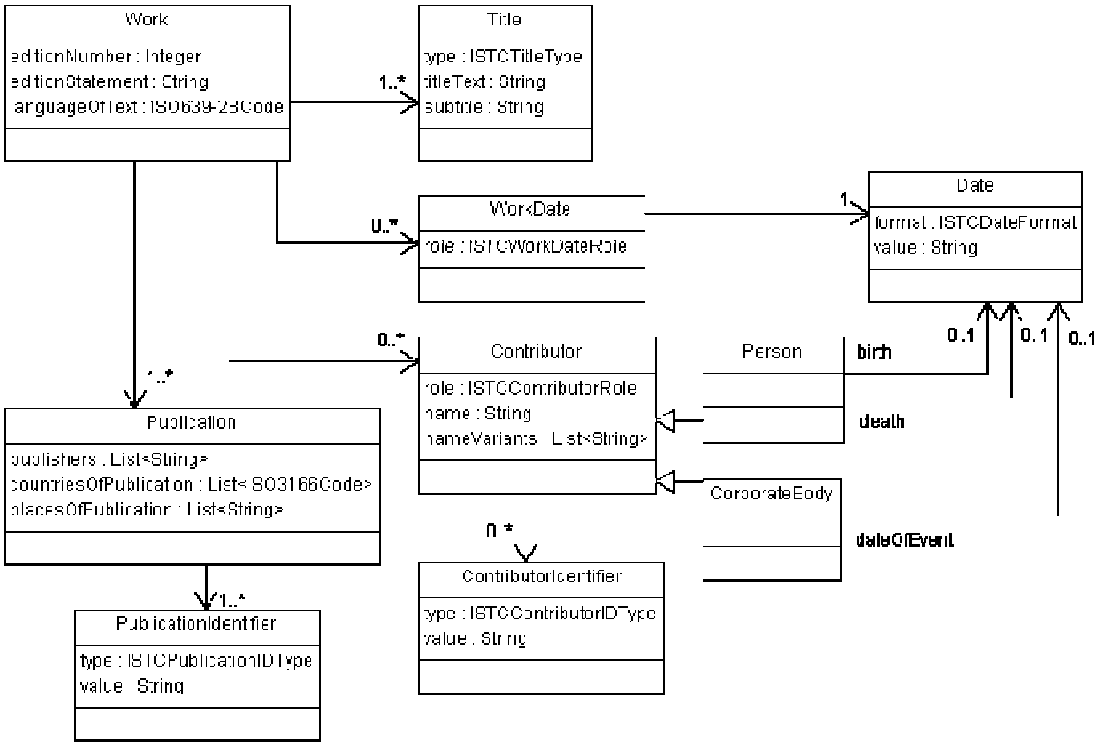


Figure 18: The main information units of the TEL system

Several data formats are used at the National Libraries, typically from the family of MARC formats. Although at this stage, only MARC21 is supported in ARROW, The European Library system was designed to be independent of the MARC format. In this way, support for other formats such as UNIMARC, or even national variations of MARC21, can be added to the system at a later stage without implications to most parts of the system.

Besides manifestation and works, the TEL system also processes clusters of manifestations. These clusters are created for the “cluster manifestation” use case, where a primary cluster will be created containing all manifestations that share the same work with the target manifestation. Secondary clusters will be created containing other manifestations with possibly related works.

Like individual manifestations, the clusters are also described by work metadata. This clustered work is a consolidation of all the data contained in the works of the manifestations that form the cluster.



"Work" data model

A key entity of information of the TEL system is the Work, whose data model is further elaborated on the figure above. The Work data is extracted from the manifestation ARC records by the Work pre-processor component and is used by the Clustering and Matching engine. This model was design to support the following requirements:

- Support the data requirements of the "match manifestation" use case
- Support the data requirements of the "cluster manifestation" use case
- Support the creation of ONIX short descriptions of manifestations
- Support the creation of ONIX short descriptions of clusters of manifestations

This data model was based on the ISTC metadata for works[10] and uses ONIX code lists extensively, but it contains adaptations for this particular application. The structure is a subset of the ISTC because the libraries' MARC records do not contain all the data of ISTC. On the other hand, data that

[10] For information on ISTC work metadata see the ONIX for ISTC specifications available at: http://www.editeur.org/files/ONIX%20for%20ISTC/ONIX-ISTC%20overview%20v1.0.pdf

is not in ISTC was added to potentially support the match and cluster manifestation requests (for example, countries of publication, contributor name variants, and life dates).

3.6.4.1. Processes supporting the use cases

This section elaborates on how processes that support the use cases are handled by the TEL system. The diagrams in this section follow the notation of sequence diagrams[11] of UML.

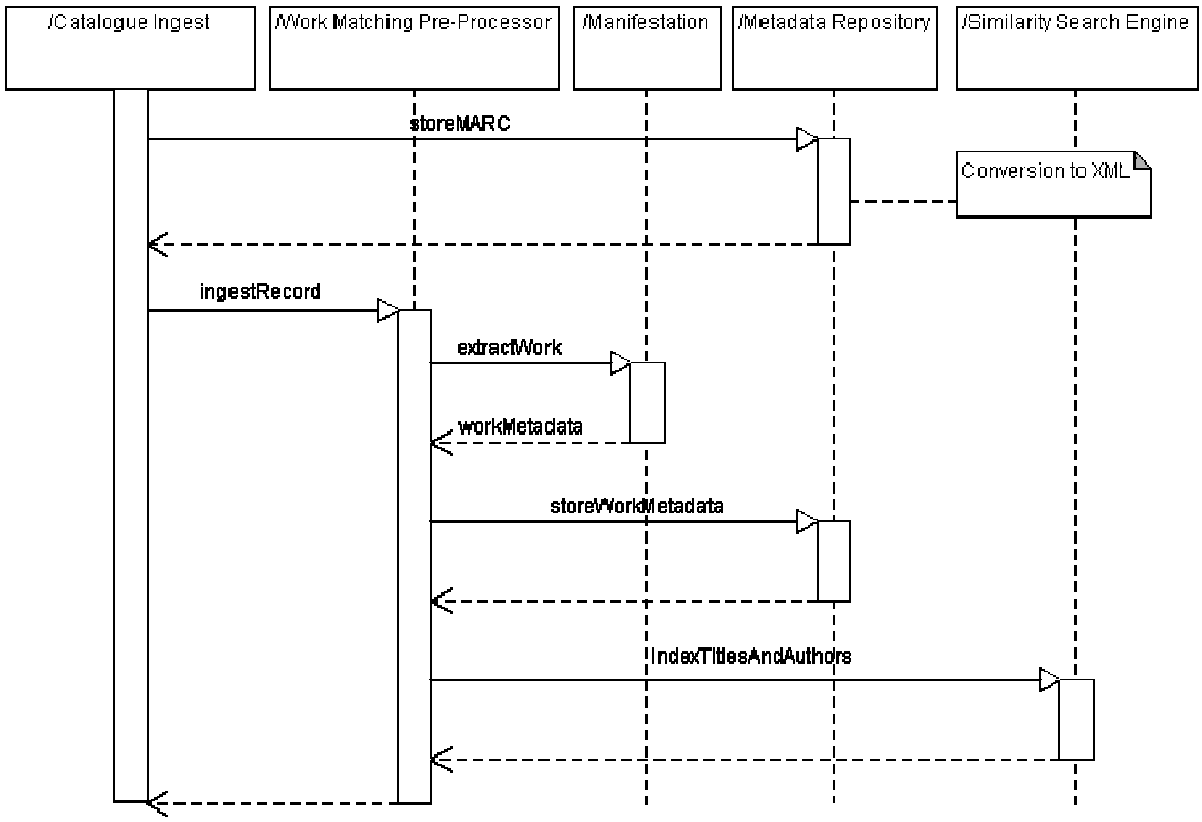


Figure 19: **The execution of the ingest of a National Library catalogue**

The ingestion process of a catalogue of a National Library starts after an export of the catalogue has been received by The European Library. Each exported record is processed in two components of the TEL system: the Metadata Repository and in the Work Matching Pre-Processor.

Ingestion into the Metadata Repository consists of converting the MARC records to XML, their storage, and indexing by identifier.

[11] http://en.wikipedia.org/wiki/Sequence_diagram

Ingestion into the Work Matching Pre-Processor consists of a three step process that aims to extract the data that describes the work of the manifestation from the MARC record, and prepare it for later processing within the system.

Work extraction consists of mapping, parsing and transformation between MARC21 fields and the systems' own data structure to represent a Work (described in the previous section). More details on work extraction are provided in ANNEX V (D6.1_ANNEX_V_WorkExtraction_TEL).

The extracted work is then stored in the Metadata Repository for retrieval. The Similarity Search Engine indexes the titles and contributors so that they can be efficiently searched by similarity during the execution of the manifestation matching and clustering process.

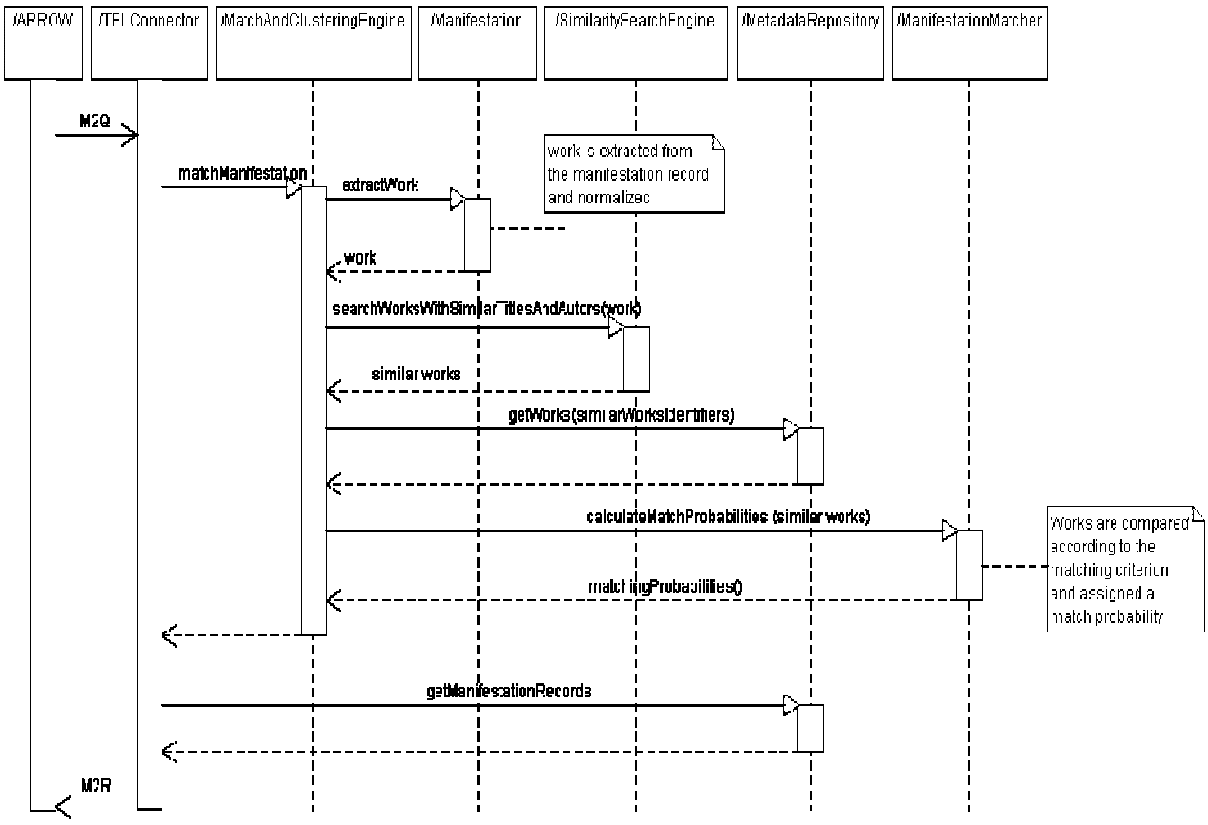


Figure 20: The execution of a match manifestation request

The implementation of the Match Manifestation use case consists of the exchange of two ARROW workflow messages between ARROW and TEL. Message M2Q, sent by ARROW to TEL, triggers the process. The MARC21 record describing the manifestation to be matched is processed in order to extract its work metadata (the same process that is applied on ingests). It is the work metadata that is used in the remainder of the matching process.

The titles and authors in the work metadata are then queried in the similarity search engine, and a first set of similar works is chosen. This first selection of similar works serves as a filter to reduce the number of works to be processed in following steps.

The metadata of the selected works are fetched from the Metadata Repository and then compared in detail to the target work. The comparison is based on both work and manifestation specific data. The main data used for comparison are:

- Title
- First author
- ISBN
- Other contributors
- Publisher
- Languages
- Edition
- Date
- Country of publication

Based on the results of the comparison some records are excluded (if they don't have a minimal level of similarity), others are considered to be an exact match, and for others a matching probability score is calculated. The works are ordered first by exact match and then by matching probability in descending order. Details on the comparison of data fields, on the criterion for exact match, and on the matching probability formula can be found in the Supporting documentation (D6.1_SD_Ib_100331_MatchingPrototype_v2)

After having the results, the TEL Connector, fetches the MARC records for matches from the Metadata Repository and sends the response message (M2R) to ARROW.

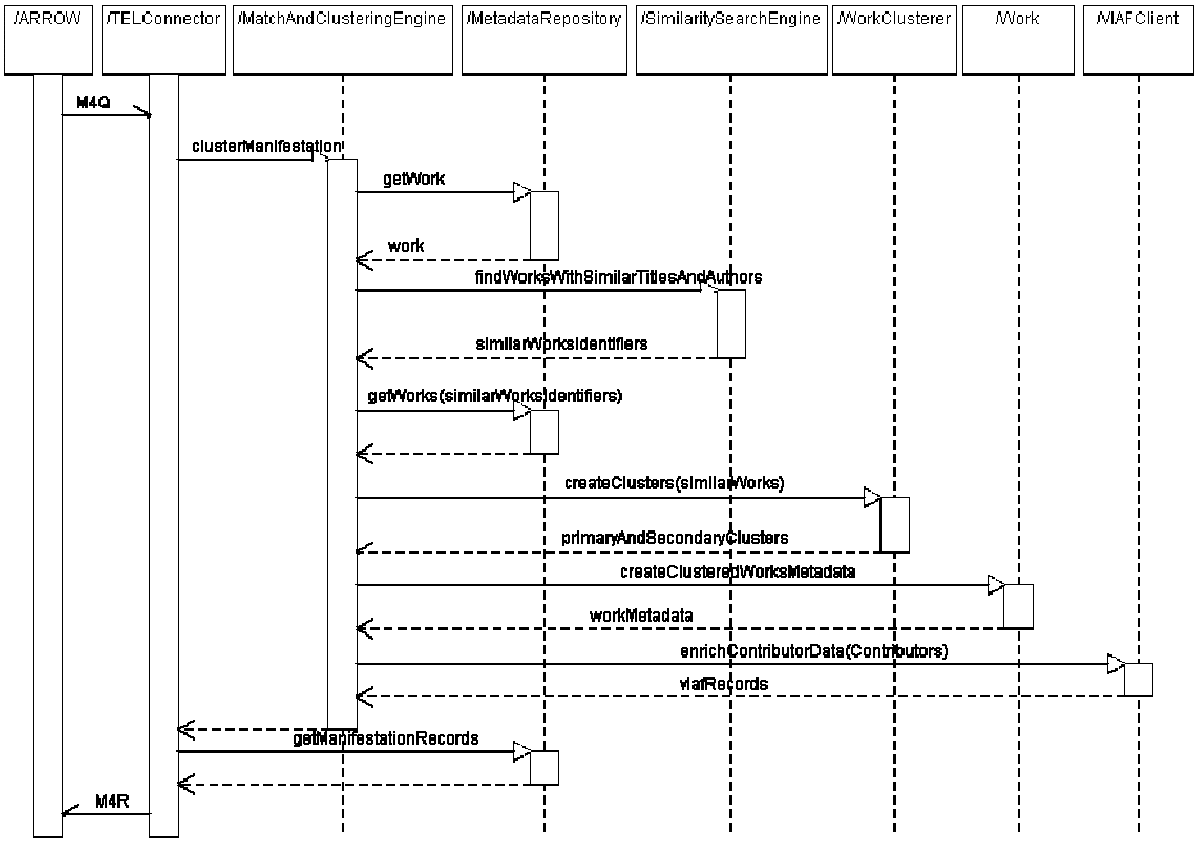


Figure 21: The execution of a cluster manifestation request

The implementation of the Cluster Manifestation use case consists of the exchange of two ARROW workflow messages between ARROW and TEL. Message M4Q, sent by ARROW to TEL, carries an identifier of a manifestation that exists in the TEL system. This manifestation is used by the TEL system to build the required clusters.

The system starts by looking up the identifier in the Metadata Repository and retrieve the work metadata for the target manifestation to be used in the remainder of the clustering process.

The titles and authors in the work metadata are then queried in the similarity search engine and a first set of similar works is chosen. This first selection of similar works should reduce the number of works to be process in following steps.

The metadata of each of the selected works is fetched from the Metadata Repository and then compared in detail to the target work.

The comparison is based on work metadata and manifestation specific data. The main data that is compared is:

- Title
- First author
- Other contributors
- Publisher
- Languages
- Country of publication

According to the results of the comparison on individual data fields some records are excluded (if they don't have a minimal level of similarity) and the others are organized in primary and secondary clusters.

The primary cluster contains manifestations which match the target manifestation on one title, all contributors, and all languages.

Titles can match even when they are not exactly equal, but are very similar. In other words, they may differ in one or two words (depending on the length of the title) and the differences between those words must be small. With this approach one can deal with typing errors or misspellings

Contributor names can also match by similarity in order to account for typing errors and spelling variations. The similarity calculation takes into account that the surname is the most important name to be matched and that the middle names may be omitted.

The same rules apply to the formation or the secondary clusters, that is, all manifestations within a secondary cluster match with another one on one title, all contributors, and all languages.

The relationship between the primary cluster and the secondary clusters is that they match with less similarity restrictions. Only the first author must match, and the similarity of the title is less restrictive than the one applied inside the clusters. It allows more words to be different, words may have larger differences, and words may be omitted.

After forming the clusters, the work metadata for each cluster is created by consolidating all the data contained in the works of the manifestations that form the respective cluster.

More detailed data about work contributors is added to the work metadata via the VIAF Client. It gets VIAF records of contributors which contain an authority identifier in the respective national

library authority file. To achieve this enrichment the national library must be a member of VIAF, and the authority identifiers have to be sent to TEL with the catalogue. The following data is obtained from VIAF:

- All name variants
- Years of birth and death
- Nationality

In the last step, the TEL Connector fetches the MARC records that are contained in the clusters from the Metadata Repository and sends the response message (M4R) to ARROW.

3.6.4.2. Implementation

This section provides implementation specific details about the deployment of the software components used by The European Library.

The following diagram displays the deployment of the software components of the TEL system, using the notation of deployment diagrams[12] of UML. The software components form two applications.

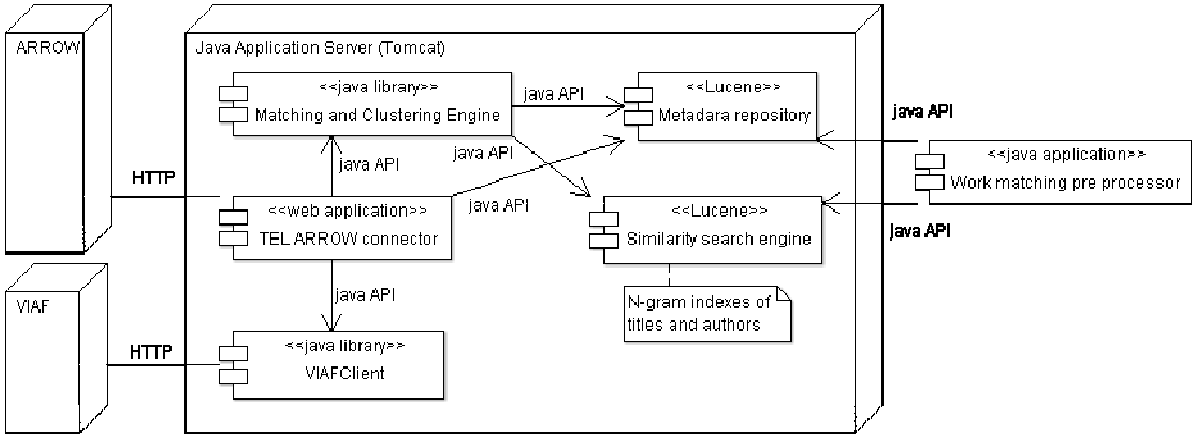


Figure 22: The deployment of the TEL system

The Work Matching Pre-processor is deployed as a standalone application that is operated by the system's administrator from The European Library during ingestion of the catalogues into the system. It is associated with the Similarity Search Engine and the Metadata Repository components that run in the main web application of the TEL system.

[12] http://en.wikipedia.org/wiki/Deployment_diagram

The TEL ARROW connector is deployed as a web application. It integrates the Matching and Clustering Engine and the VIAF Client as java libraries.

The TEL ARROW connector communicates between The European Library and the ARROW system by an exchange of XML messages[13]. The internal structure of these XML messages is based on the definition provided by EDItEUR and the web services adhere to the basic standards for Service Oriented Architectures such as WSDL and SOAP. The web services and the messages are implemented with JAXB 2.0 for object-XML mapping, and Spring Web Services for the WSDL based interaction.

The similarity search engine is based on the Lucene search engine. In order to execute similarity searches, character n-grams[14] are extracted from the titles and authors and indexed. By indexing the n-grams, works can be retrieved even if words are written differently due to typos, misspellings, abbreviations, spelling variations, etc.

The metadata repository uses Lucene as well. The main requirement for the metadata repository is to store the MARC records and the works, allowing retrieval by identifier. Three implementation options were trialled and Lucene proved to be the simplest to deploy and the most efficient. The other options that were tested were a direct implementation on the file system and an implementation using a relational database.

The VIAF Client uses the XML over HTTP interface of VIAF to send queries and retrieve the authority records. It supports queries by the identifier used in the national libraries' authority files.

3.6.5. BIP and RRO services

In the project the BiP and RRO organisations available in the different countries and involved in the pilot maintain information necessary to establish the right status of the work. To collect this information Arrow sends a message (M6Q for BIP, M7Q for RRO) to the web service of the BIP/RRO organisations. Upon Arrow message receipt BIPs and RROs systems do: (1) elaborate the request internally and (2) provide an answer sending back a response in a synchronous or asynchronous manner (respectively M6R and M7R).

[13] Specified in ARROW *Deliverable D.4.3 Specification of Rights Expression Metadata*.
[14] http://en.wikipedia.org/wiki/N-gram

The individual description and functional role of the components implemented by MVB and CLA partners acting as data providers can be found in the following table:

Component Name	Description	Hosted at
VLB web services enhancements	The pre-existing VLB web service was modified for Arrow purposes adding the following functionalities: - Processing and inclusion of "out of print" data in VLB catalog and WS and adding privileged access to WS for ARROW - Add-ons VLB web service for ARROW • Statistics on VLB data • Boolean search: increase number of parameters • ONIX reference names • New field Archiving date	MVB
Standard BIP query WS	For the ingesting of M6Q messages.	CLA, CEDRO, ELECTRE
Standard RRO query WS	For the ingesting of M7Q messages.	CLA, CEDRO, CFC, VG-WORT
Standard BIP response WS client	For sending M6R messages	CLA, CEDRO, ELECTRE
Standard RRO response WS client	For sending M7R messages	CLA, CEDRO, CFC, VG- WORT
BiP or RRO Internal ARROW workflow manager	This component provides the business rules and internal decision making process to enable appropriate M7R responses to be returned.	CLA, CEDRO, CFC, ELECTRE, VG- WORT

The build for the various CLA ARROW components uses some core library components developed by Arc Software Consultancy. Arc Software Consultancy has granted a one-off licence for their use by RROUK (CLA, ALCS, PLS) in its participation in ARROW.

3.7. Architecture overview

This section provides a more detailed overview of all the main software components and relationships between them. The figure below shows a simplified DataCentre architecture containing

some of the components described previously as well as the current workflow represented by the activity diagram highlighted in grey.

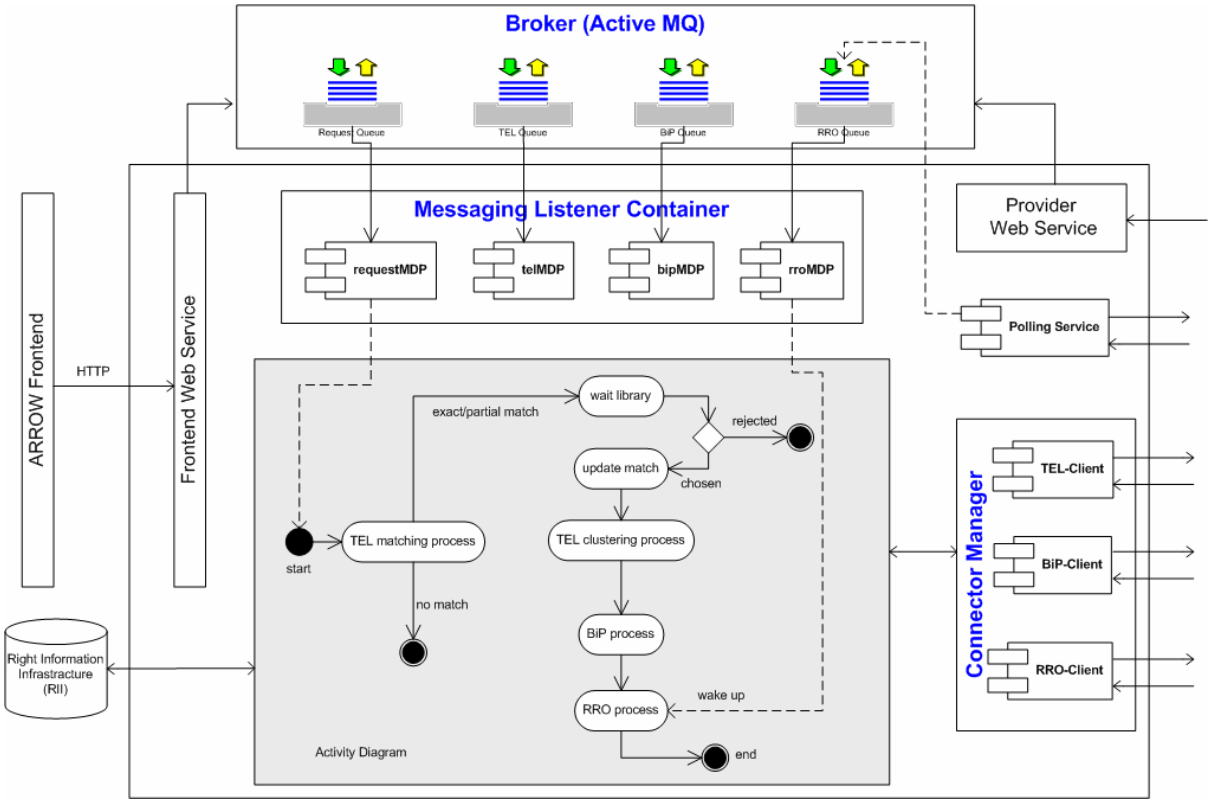


Figure 23: DataCentre Architecture

As the figure shows, all external communications with data providers pass through the Connector Manager component. In case of synchronous communication, the provider client components send the requests according to provider's protocol and implementation specification and the retrieved responses are immediately returned to the workflow engine.

In case of asynchronous communication, the provider client components send the requests to the relevant service and the relevant responses will be later retrieved in two ways:

1. ARROW service performs a polling on the providers service
2. External providers invoke the ARROW Provider web service

In both cases, the obtained responses are moved to the messaging broker.

This asynchronous mechanism has been implemented using an external service (ActiveMQ) which fully supports transient, persistent and transactional JMS messaging. The DataCentre uses JMS

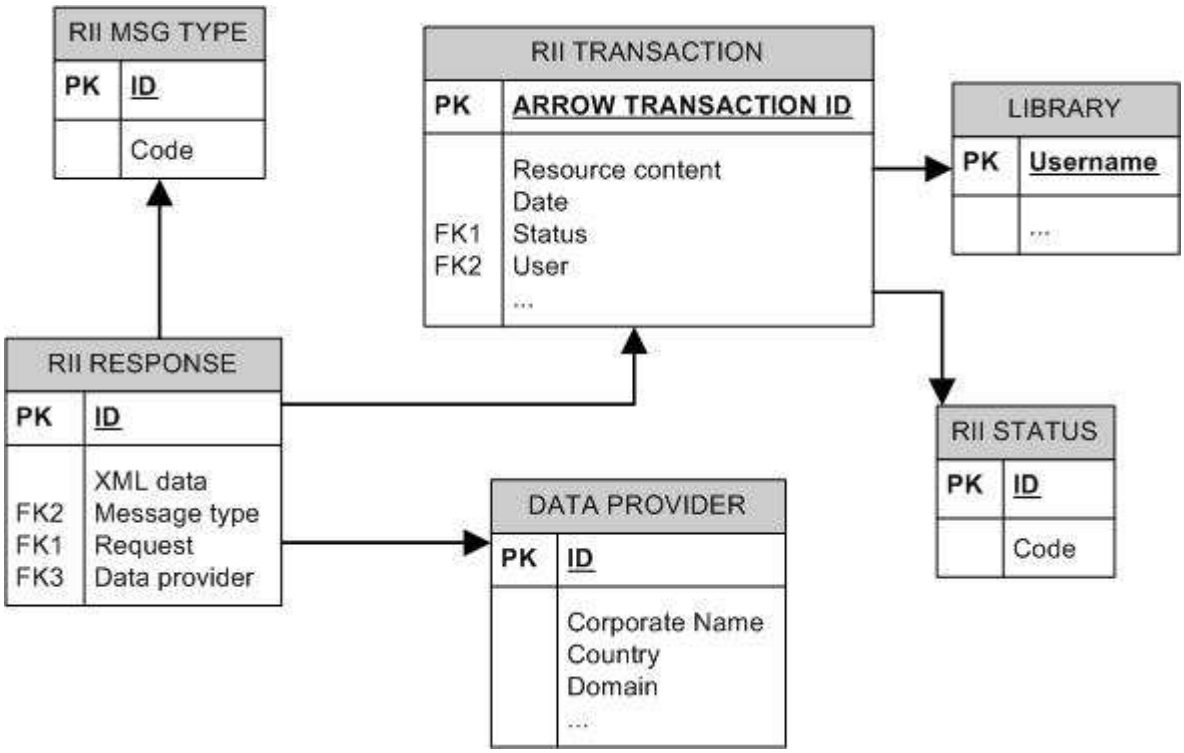
listeners to fetch responses from the proper queue and delegate the responses to the workflow engine.

Workflow engine component is implemented using jBPM framework. jBPM manages the process instances described by a process description document. This framework enables us to deal with workflow declaratively and in a more flexible manner.

3.8. ARROW RII Repository

It's worth to underline that the RII database stores all the relevant information coming from the different actors of the system, i.e. TEL/VIAF, BIP, RRO in the different Arrow messages (M1-M7).

A simplified representation of the RII database is presented in the following figure:



The following table displays the type of data retrieved by the Arrow system, the format in which they are stored in RII repository as well as the format in which they are forwarded to possible other data providers.

Data Provider	Data provider data	Type of stored Arrow information based on message number	Other Data provider receiving the same kind of information in different format
TEL	M2R containing Marc	M2R where	

	bibliographic records matching details of library's target resource	bibliographic records are transformed in Onix Short Description	
TEL	M4R containing identified clusters of manifestations works for the library's selected resources	M4R where bibliographic records are transformed in Onix Short Description	VG Wort (DE), Electre, CFC (FR), CEDRO (ES), CLA (UK) M6Q containing Onix Short Description
VLB (DE)	Onix for Books bibliographic records that complement or enrich TEL identified clusters. Each incoming record contains information on the publishing status and availability	M6R created by Arrow based on incoming VLB data. Onix for Books are transformed in Onix Short Description	VG Wort (DE) M7Q containing Onix Short Description
Electre (FR), DILVE-CEDRO (ES), NBD-CLA (UK)	M6R Onix for Books bibliographic records that complement or enrich TEL identified clusters. Each incoming record contains information on the publishing status and availability	M6R with bibliographic records in Onix Short Description	CFC (FR) M7Q containing Onix Short Description
CFC (FR), CEDRO (ES), CLA (UK)	M7R including license proposal or refusal, advice that a license is unnecessary as well as other advice	M7R	

It's worth to underline that the messages M4R and M6R are further enriched by the Arrow RII system with Work Assertion information: respectively CopyRight status and Publishing status.

3.9. RII Algorithms

3.9.1. Matching algorithm (TEL)

The Matching and Clustering Engine hosted at TEL includes the implementation of the matching algorithm used for ARROW messages M2R. This algorithm compares two manifestation records on work and manifestation specific data, and outputs a probability of the two manifestation records being about the same manifestation.

3.9.2. Clustering algorithm (TEL)

The Matching and Clustering Engine hosted at TEL includes the implementation of the clustering algorithm used for building the work clusters for ARROW messages M4R. It is applied in the comparison of work metadata from the target manifestation and other works with similar titles and first contributor. The comparison is based on work metadata and manifestation specific data

3.9.3. Copyright Status and Publishing Status

Based on the information gathered in the messages M4R and M6R, Arrow deduces some important information regarding the work status. Such information includes:

- Copyright Status: determines if the work is in Public Domain or under Copyright starting from Contributors Dates in Work Description.
- Publishing Status: determines if the work is In Print or Out of Print.

The Copyright Status of the work is deduced from the copyright status of each contributor through a Copyright Status algorithm that is implemented within the RII DataCentre Workflow engine hosted at CINECA.

The Publishing Status of the work is deduced from the status of each manifestation retrieved in the BIP process through a Copyright Status algorithm that is implemented within the RII DataCentre Workflow engine hosted at CINECA. Elements and action of algorithms are described in project deliverable D6.4 Rights information infrastructure - final release.

4. The ARROW Work Registry (AWR) and the Registry of Orphan Works (ROW)

One of the objective of the ARROW project is to set up a registry containing information about works that are considered orphan at the end of the processes and elaborations along the ARROW workflow (Registry of Orphan Works), and make them publicly searchable, thus enabling public information on works which have been declared to be orphan, among other things, to enhance the possibility for rightholders (individually or through collective representative organisation or agent) to claim their rights in order to decrease as much as possible the number of orphan works.

The starting point for the design of a Registry of Orphan Works within the ARROW system has been provided by the High Level Expert Group (HLEG) Final Report on Digital Preservation, Orphan Works and Out of Print Works, including also recommendations and key principles for rights clearance centres and databases for orphan works[15]. In addition, in order to define the ROW requirements, the different stakeholders communities represented in the ARROW consortium and in the countries piloting the system have been consulted, in particular that of Collective Management Organisations and Reproduction Rights Organisations, identified, by the HLEG, as natural candidates in the current situation to run Orphan Works databases and Rights Clearing Centres. Furthermore, it has been considered that, at the present time, the definition of the legal framework concerning Orphan Works is still in progress and operative solutions are under discussion in many European countries, as described in the deliverables pertaining the Work Package on Legal framework and business model[16].

Hence, the chosen approach takes into consideration the open-ended situation and is intended to be as neutral as possible to the evolution of the European and national legal frameworks concerning Orphan Works as well as to operative solutions and business models that could emerge at national level. The technical specifications for the implementation of the ARROW ROW, resulting from the

[15] The Final Report on Digital Preservation, Orphan Works and Out of Print Works is available at http://ec.europa.eu/information_society/activities/digital_libraries/doc/hleg/reports/copyright/copyright_subgroup_final_report_26508-clean171.pdf

[16] see D3.1 *Report on legal framework, Edition 1,* D3.2.1 *Guidelines for the Definition of Orphan Works;* D3.2.2.*Evaluation of compliance of ARROW workflow with the HLG guidelines on diligent search*; D3.3.2 *Correspondence of ARROW infrastructure with emerging clearing centres and the needs of their users*; D3.5 *Report on legal framework, Edition 2,* available for downloading in the Resources area of the ARROW website (www.arrow-net.eu)

requirements gathered so far, are presented in the current document, along with background information for ROW design and interaction among components in the ARROW system.

4.1. AWR/ROW Feeding

The ARROW System starting point is the workflow of the Rights Information Infrastructure (RII), where each search submitted consists in a transaction, that is a set of message exchanges gathering and processing information from different data sources (TEL, VIAF, BIPs, RROs), grouped under the same Arrow Transaction ID (the unique and persistent identifier assigned by ARROW to each RII transaction) and stored in the RII repository. As described later in this document, these transactions are fundamental for the ROW history. The Arrow Transaction ID also bounds the RII repository and the ARROW Work Registry.

The figure below provides a high level overview of the AWR and its relation with the RII.

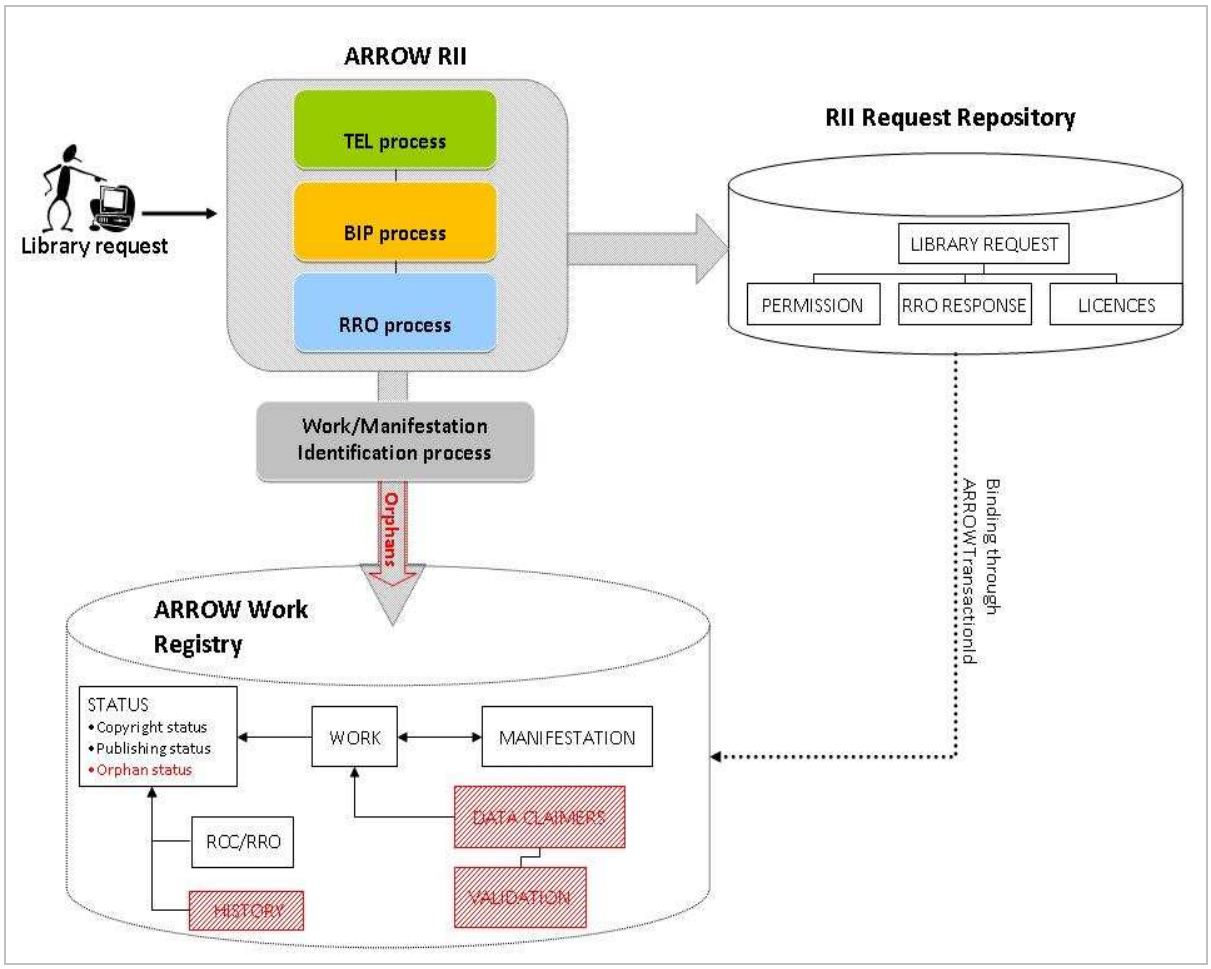


Figure 24: AWR/ROW Feeding

The ARROW RII has been simplified representing three subsequent processes: the TEL process, the BIP process and the RRO process.

In synthesis at the end of the ARROW workflow, and hence of these three processes, the RII comes out with two different outputs: the first one constitutes the bases for the RII repository, while the second one those for the AWR. When the ARROW system proceeds with the construction of the M7Q it also initialises the Work/Manifestation Identification process responsible to identify uniquely for each request coming from the library the underling work and manifestation metadata. Once this process is completed the ARROW system proceeds with the storage of these information on the ARROW Work Registry.

The ARROW Work Registry (AWR) stores all the relevant pieces of information collected by the RII workflow in a structured way that allows the retrieval and use of that information in the framework of ARROW services. The gathered information mainly regards to:

- Work and manifestation metadata
- Authors and other contributors metadata
- A set of so called ARROW Assertions on each work: Copyright status, Publishing Status and Orphan Status
- Reference source (TEL, VIAF, BIPs, RROs) of work metadata, manifestation metadata, authors and contributors metadata

From the above sections can be inferred that the implementation of the RII, and in particular the ARROW Work Registry (AWR), constitutes the fundament for the Registry of Orphan Works (ROW). The ROW core database in other words can be seen as a view of the AWR, result of the RII workflow. The design and set up of the ROW hence strongly depends on the AWR design. Being the ROW a subset of the AWR, the ROW starts ‘empty’ and gets populated by digitisation requests being processed through the ARROW system in an automatic way in the case the output of the RII process indicates that the work can be an orphan.

The works stored in the AWR that at any point in time have an Orphan status marked as “ProbablyOrphan” constitute the ARROW ROW. (For more details see Del 6.2 Registry of Orphan Work management system).

The ARROW Work Registry (AWR) and its data model were inspired by the ISTC metadata for works in order to guarantee the interoperability with the services provided by the ISTC international agency for ISTC registrations.

Here it is worth to recall the principles for the identification of relevant entities within the AWR, as they are the necessary background to illustrate the ROW structure and functions, being the ROW a subset of the AWR.

4.1.1. Identifiers

This section describes the principle of the identification of the relevant AWR entities. In order to univocally identify such entities the introduction of ARROW identifiers was necessary, for the following reasons:

- ARROW has been designed to retrieve data from multiple data sources, therefore the same entity can be provided to ARROW by more that one source. The identifiers used by each data source for the same entity may not be the same, especially when a standard identifier is not available
- For some of the AWR relevant entities, standard identifiers are not yet in place or their adoption is still underway.

The use of ARROW identifiers within the system allows to maintain the relation among the different entities with a single identification framework, keeping also the association with the identification schemes used by external data sources and at the same time being interoperable through the use of standard identifiers whenever available.

Arrow Transaction ID

A unique and persistent identifier assigned by ARROW to each RII transaction, i.e. to each request processed by ARROW; once assigned, the Arrow Transaction ID is mandatory in all messages exchanges triggered by a library request. Storage of the Arrow Transaction ID in the AWR allows to trace and group all the message exchanges performed within a specific transaction and link to the RII repository. Being the primary key for querying and tracking each instance of ARROW workflow, Arrow Transaction ID is also fundamental for the ROW history.

Arrow Resource ID

A unique and persistent identifier assigned by ARROW to the record originally submitted by the library, identifying the manifestation (resource) the library wishes to digitise and use. The Arrow Resource ID is correlated with one ore more identifiers assigned to that resource by the requesting Library (ex. library's control number) as well as with any standard identifier (ISBN) available for that resource.

Arrow Manifestation ID

A unique and persistent identifier assigned by ARROW to each manifestation retrieved in the workflow, in addition to any other identifier available for the manifestation.

Actually the ARROW workflow may obtain the same manifestation from different sources (TEL or BIP). Each provided manifestation may already have one or more identifiers such as a proprietary identifier of the data provider or an ISBN when available. The Arrow Manifestation ID will be associated with all these identifiers as well as with the set of manifestation metadata. If the same manifestation is provided from more sources the Arrow Manifestation ID keeps the association to both sets of metadata.

Arrow Work ID

A unique and persistent identifier assigned by ARROW to each work in addition to any other identifier available for the work. In the current ARROW workflow, a first work identification is performed by TEL to support the clustering process. TEL assigns each identified work with a unique TEL Work ID. Once ARROW receives work information from TEL, ARROW assigns its own Arrow Work ID, in addition to the TEL Work ID. This allows ARROW to persistently identify works regardless the reference source of the information, ensuring the system scalability. The Arrow Work ID is also used to establish during each elaboration of the RII workflow if the work is new or if it already exists in the AWR.

Reference Sources and Record Sources Identifiers

Each data provider interconnected within the ARROW RII is identified with a Reference Source Identifier, i.e. an identifier for the source of information or authority files against which a matching, clustering or searching operation is performed in ARROW workflow. Each record exchanged within the ARROW RII is associated to a Record Source Identifier, which identify the source or organisation that provided that record. The Reference Source Identifiers and Record Source Identifiers are

essential to the compliance of ARROW workflow to HLEG guidelines for diligent search, as it contributes to the documentation of the search process that ARROW performed.

4.1.2. Work/Manifestation Identification process

All works requested for digitisation should be checked against the AWR to confirm whether this is a new work or an existing one (already present in the AWR).

In case of new work, the ARROW workflow proceeds according to the following steps:

- The system generates a new Arrow Work ID and associates it to the TEL Work ID
- Work metadata are inserted in the AWR
- For each manifestation related to the work, ARROW verifies if such manifestation is already present in the repository or not and, if the manifestation is not present, a new Arrow Manifestation ID is generated and associated to it. Manifestation metadata are inserted in the registry.

In case of an existing work already the ARROW workflow proceeds according to the following steps:

- Existing work metadata are replaced
- Existing manifestation associated to the work
    - If the manifestation is provided by the same data provider its metadata will be completely replaced
    - If the same manifestation is provided by a different data provider it is added in the registry which also traces of the data provider in question.
- New manifestations associated to the work are added
- “Missing manifestations”: if the incoming work contains less manifestations that the existing manifestations in the AWR these “missing manifestations” will be removed and maintained only in the history.

As the picture of a work at any point in time should be considered the most accurate to date, the previous work and manifestation metadata are maintained in the history.

The set of ARROW Assertions on each work (Copyright status, Publishing Status and Orphan Status) will be added and updated accordingly.

The works stored in the AWR that at any point in time have an Orphan status marked as "ProbablyOrphan" constitute the ARROW ROW. In the Figure 22 this concept has been underlined in the workflow by using a "red arrow". It's worth to underline that the ROW is not only a subset of the AWR but it extends the entity relations defined for the AWR introducing new relations in the data model necessary to guarantee the consistency of the ROW with the HLEG guidelines. The logic view of these extensions has been represented in Figure 22 with red boxes such as: History, Claimers data, Validation. A more detailed description of the ROW data model has been provided in § 4.9.

4.2. Feeding of the ROW - Orphan Criteria

In compliance with the HLEG guidelines for the definition of orphan works[17], a work in ARROW is considered to be orphan if its rightholders can not be identified or, if identified, it is not possible to locate them.

Moreover it has to be considered that an Orphan work is a work protected by copyright but the current owner is unknown or untraceable by diligent search, therefore works in the public domain are in principle not relevant for the scope of the ROW.

Establishing if a work is orphan or not is not a simple and straightforward task but is the result of a diligent search for the rightholders, according to a procedure and methodology compliant with the principles contained in the HLEG guidelines on diligent search. Such diligent search is supported by the ARROW Rights Information Infrastructure (RII) that enables to search distributed sources of information to find out the rights status of a work and its rightholders[18].

As mentioned above, at the end of the RII workflow, each work is associated with a set of ARROW assertions on the work rights status: Copyright Status, Publishing Status and Orphan Status. The works that have an Orphan Status marked as "ProbablyOrphan" feed the ROW.

The following table is an outline of the possible combinations of the ARROW assertions on the work rights status that lead to the inclusion of a specific work into the ROW.

Copyright Status	Publishing Status	Orphan Status	Inclusion in the ROW
arrow:Copyright	ARROW:CurrentlyActive	ARROW:ProbablyOrphan	YES

[17] see D3.2.1 *Guidelines for the Definition of Orphan Works*, available for downloading in the Resource area of the ARROW website (www.arrow-net.eu)

[18] Compliance of the ARROW workflow to the HLG guidelines on diligent search has been documented in *D3.2.2 Evaluation of compliance of the ARROW workflow with the agreed HLG guidelines on diligent search* available for downloading in the Resources area of the ARROW website (www.arrow-net.eu)

		ARROW:NotOrphan ARROW:Unspecified	NO NO
arrow:InPublicDomain	ARROW:CurrentlyActive	Not relevant	NO
ARROW:Unspecified	ARROW:CurrentlyActive	ARROW:ProbablyOrphan ARROW:NotOrphan ARROW:Unspecified	YES NO NO
arrow:Copyright	ARROW:NotCurrentlyActive	ARROW:ProbablyOrphan ARROW:NotOrphan ARROW:Unspecified	YES NO NO
arrow:InPublicDomain	ARROW:NotCurrentlyActive	Not relevant	NO
ARROW:Unspecified	ARROW:NotCurrentlyActive	ARROW:ProbablyOrphan ARROW:NotOrphan ARROW:Unspecified	YES NO NO
arrow:Copyright	ARROW:Uncertain	ARROW:ProbablyOrphan ARROW:NotOrphan ARROW:Unspecified	YES NO NO
arrow:InPublicDomain	ARROW:Uncertain	Not relevant	NO
ARROW:Unspecified	ARROW:Uncertain	ARROW:ProbablyOrphan ARROW:NotOrphan ARROW:Unspecified	YES NO NO

The above outline of the possible combinations of the ARROW assertions reflects the current status of the discussion on Orphan Works, considering a work Orphan when none of the rightholders is known or traceable.

However, it is clear that the concept of “being orphan” might be more complex, when considering that for a work only some of the rightholders can be not known or traceable while others are perfectly known. A typical example is a book with photos or other illustrations where the writer is known and the illustrator not.

The complexity might increase yet again when considering that for a work the owner of a specific right might be unknown, while for other rights the owners are perfectly known. A typical example is a book for which it is unclear who is the digital rights owner.

Enough flexibility will be left to the ARROW system to be able to include in the ROW also the two above mentioned categories of works, however, in absence of a formal definition of “Orphan Works” under this perspective, either at European or at national level, no specific implementations are foreseen at this stage to address this issue.

Finally, it's worth noting that a work can be declared "ProbablyOrphan" in ARROW independently from the possibility of licensing that work, because the latter depends on the legal framework in a country and does not affect the "being orphan" of the work.

To sum up, the ROW will be fed according to the following criteria:

- Works with Orphan Status = arrow:ProbablyOrphan will be part of the ROW, as diligent search has been completed
- Works with Orphan Status = arrow:NotOrphan will not be part of the ROW
- Works with Orphan Status = arrow:Unspecified will not be part of the ROW, as their status has not been cleared yet.

4.3. Functional requirements - "Shall lists"

Functional requirements are the functions and actions that specific categories of users will be able to perform on the ROW, according to specific purposes, ie. to make information on Orphan Works publicly searchable and enhance the possibility for rightholders to declare their rights, thus reducing the number of Orphan Works.

Search and/or browse services:

- "search and browse" works (Simple Search)
- "search and browse" manifestations (Simple Search)
- browse of work history

Claiming Request service:

- claiming of rights ownership (claiming request)
- browse of own claiming requests and their status – claimer's view

Services for the ROW Management:

- browse of claiming requests
- management of claiming requests: approval or refusal of claiming requests

- updates of the work rights status and history
- notification of the ROW Manager validation result to the claimer

4.4. Actors & Roles

Based on the above functional requirements, the following actors, roles and use cases can be foreseen, also taking into account possible evolution of the European and national legal framework. ROW shall be designed to be scalable to emerging actors in ARROW scenario. The actors here identified are to be interpreted as "Roles" that can be played by different organisations in the ARROW environment, regardless who will actually play each of the roles. From the design and technical point of view, it is therefore necessary to define roles at the most granular level possible, to allow the system to be neutral to the solutions for managing Orphan Works that will be adopted in each national legal framework. Under this perspective, it might be possible that one single organisation plays two or more roles, thus embodying two or more actors. This approach is also neutral to the model for managing Orphan Works adopted (Centralised infrastructure or National ROW(s)).

Actors & Roles:

- End Users – The end users have the possibility to search and browse the ROW, as well as to request a claiming account
    - any Internet user
    - any interested organisation (including libraries, publishers and authors associations, collecting societies)
- Claimers – The claimers have the possibility to search and browse the ROW and claim for right ownership and to be notified of the approval or refusal of the claiming request and change of the Orphan status
    - Authors and other contributors and their heirs (single person)
    - Institutions such as libraries and foundations, acting as rightholders
    - Publishers
    - Literary agent (on behalf of one or more persons)

- Collective management organisations/RRO
- Authors Collective management organisations
- Publishers Collective management organisations[19]

In the last four cases, when someone is claiming on behalf of a rightholder, the Claimer is not the rightholder themselves.

- ROW Manager – an authorized entity responsible to manage a ROW on a country basis. The ROW Manager has the possibility to search and browse the ROW, to browse the work history, browse the claiming requests and is entitled to approve or reject the pending claiming requests of the claimers, and change the Orphan Status of the work, following an approved claiming request. The validation process of the claiming request is outside the scope of the present document, being it entirely between the ROW manager and the organisation in charge of validation.
  - In the current ARROW workflow Collective management organisations such as RROs are candidate to act as ROW Manager
  - Any other authorised entity that will be appointed in each country legal framework
- Diligent Search Agency – an authorized entity that according to each country legal framework is appointed to do the diligent search, or endorses as "diligent" a search done by a user, and to declare that a work is orphan. The Diligent Search Agency have the possibility to search and browse the ROW, browse the work history, declare that a work is orphan and change the Orphan Status of the work, following a diligent search
  - In the current ARROW workflow Collective management organisations/RROs practically act as Diligent Search Agency if they declare that a work is "ProbablyOrphan"
  - Any other authorised entity (libraries and other public bodies) that will be appointed in each country legal framework
- Rights Clearing Centres for Orphan Works - an authorized entity that according to each country legal framework is appointed to issue a license for an orphan work[20]. The Clearing

[19] List can be updated in future

Centres have the possibility to receive a notification from the ROW whenever the Orphan Status of a licensed work changes

- Any authorised entity (including RROs) that will be appointed in each country legal framework. Not existing yet.

It is foreseen that, according to each country legal framework, actors like ROW Manager and Rights Clearing Centres can operate at country level and their specific interaction with the ROW is limited to the works within their jurisdiction. At the moment in ARROW "works within a jurisdiction" are defined as works declared orphan by the RRO in the same country of the ROW Manager and Rights Clearing Centres. In case it is needed, a specific ROW Manager and Rights Clearing Centres can be associated to more than one jurisdiction.

[20] see also D.3.3.2 Correspondence of ARROW infrastructure with emerging clearing centres and the need of their users.

4.5. Functions & Actors matrix

	End Users	Claimers (rightholders)	Claimers (on behalf of rightholders)	ROW manager	Diligent Search Agency	Rights Clearing Centre
Declare that a work is Orphan					X	
Search works (simple search)	X	X	X	X	X	X
Search works (bulk search)			X			
Search manifestations (simple search)	X	X	X	X	X	X
Search manifestations (bulk search)			X			
Browse works	X	X	X	X	X	X
Browse manifestations	X	X	X	X	X	X
Claim ownership		X	X			
Browse claiming requests				X		
Assess claiming request (approve/refuse)				X		
Update work rights status				X	X	
Browse work history				X	X	
Receive notification of claiming request assessment result		X	X			
Receive notification of updated rights status		X	X	X	X	X
Issue licenses on Orphan works						X

4.6. ROW models supported in ARROW

As mentioned previously, the ARROW system has been designed to support different models for the management of Orphan Works, in the first instance the set up of a centralised infrastructure and the set up of interoperable National ROWs. As described in §4.2 Feeding the ROW, the information gathered along the workflow by the RII, is the basis for the initial feeding and updates of the centralised ROW and in case of the National ROWs.

In both cases it is fundamental to make the public information on Orphan Works publicly searchable and enhance the possibility for rightholders to declare their rights, thus reducing the number of Orphan Works. As described in §4.3 Functional Requirements, to fulfil the purpose the following functions/services are envisaged:

- Search functions
- Claiming request functions
- Functions for the Management of ROW (management of claiming request and history)

To maximise the European impact of the ROW, a central index for searching will be built to allow any user to search on the whole corpus orphan works in Europe from a single access point.

Once the central index for search will be in place, rightholders will be able to search for a work and declare their rights. The claiming request function will be implemented at centralised level on Orphan Works managed in the centralised ROW. In case a Orphan Works are managed by other systems (as National ROWs) and they have their own claiming service, appropriate redirection mechanisms will be implemented from the central layer for searching to the national service.

Function for the Management of the ROW – assessment of claiming requests (approve/refuse), update of the work rights status, access to work history – will be implemented at centralised level for claiming requests on Orphan Works managed in the centralised ROW. In case Orphan Works are managed by other systems (as National ROWs) and they have their own management functions, mechanisms to ensure the update of the work status in the central layer for search shall be provided.

The following figure represents how the different model supported in ARROW can coexist in a single framework built on the ARROW workflow and RII. In the following figure, RROs are assumed to play the role of the ROW Manager and maintain national ROWs.

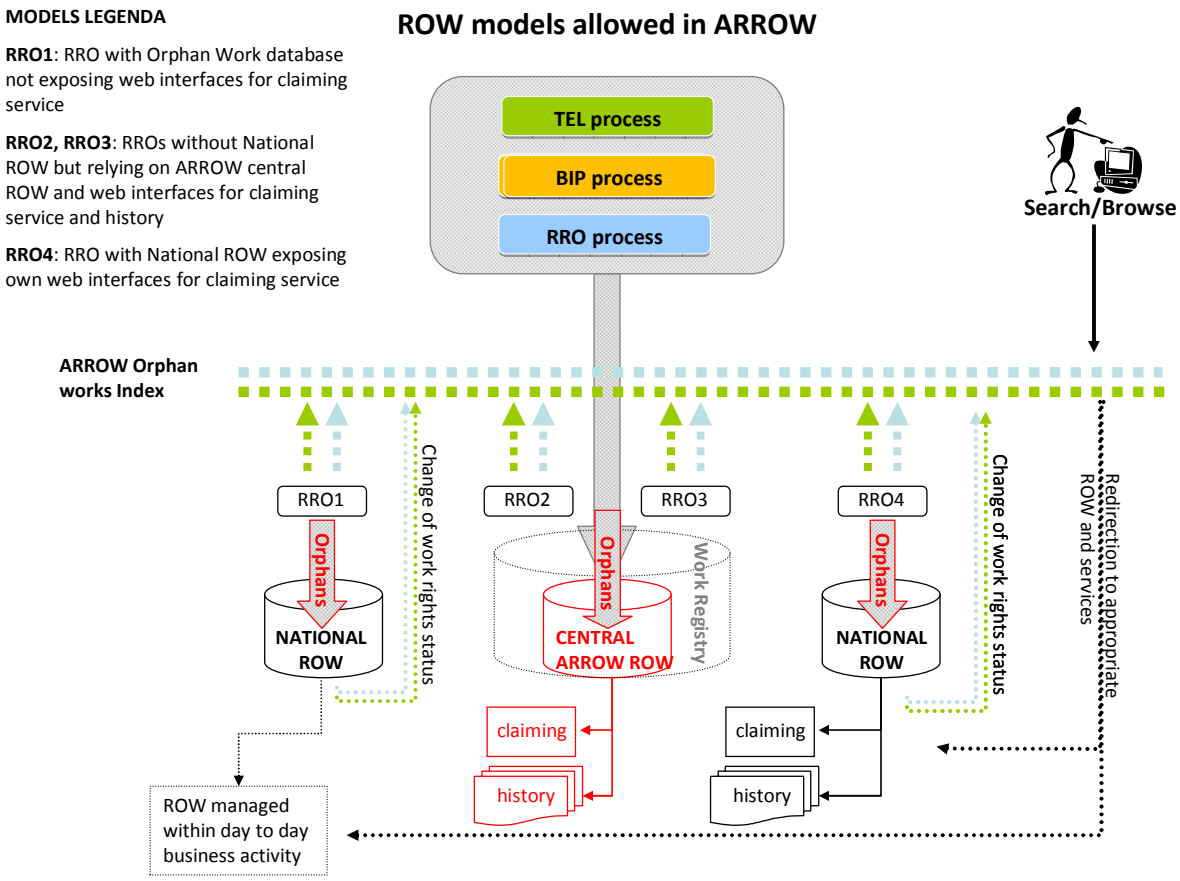


Figure 25: ROW Models in ARROW

4.7. System Requirements - Use Case Identification

Based on the Functional Requirements (§4.3) and the Actors & Roles (§4.4) the set of services that the ROW Management System shall provide to the different categories of actors has been modelled. The system has been designed in order to support all the kinds of actors (End User, Claimer, ROW Manager, Diligent Search Agency, Rights Clearing Centres for Orphan Works), but since at this stage of the project the last two categories of actors do not yet exist, in the following diagram only the Use Cases for End User, Claimer, ROW Manager will be displayed.

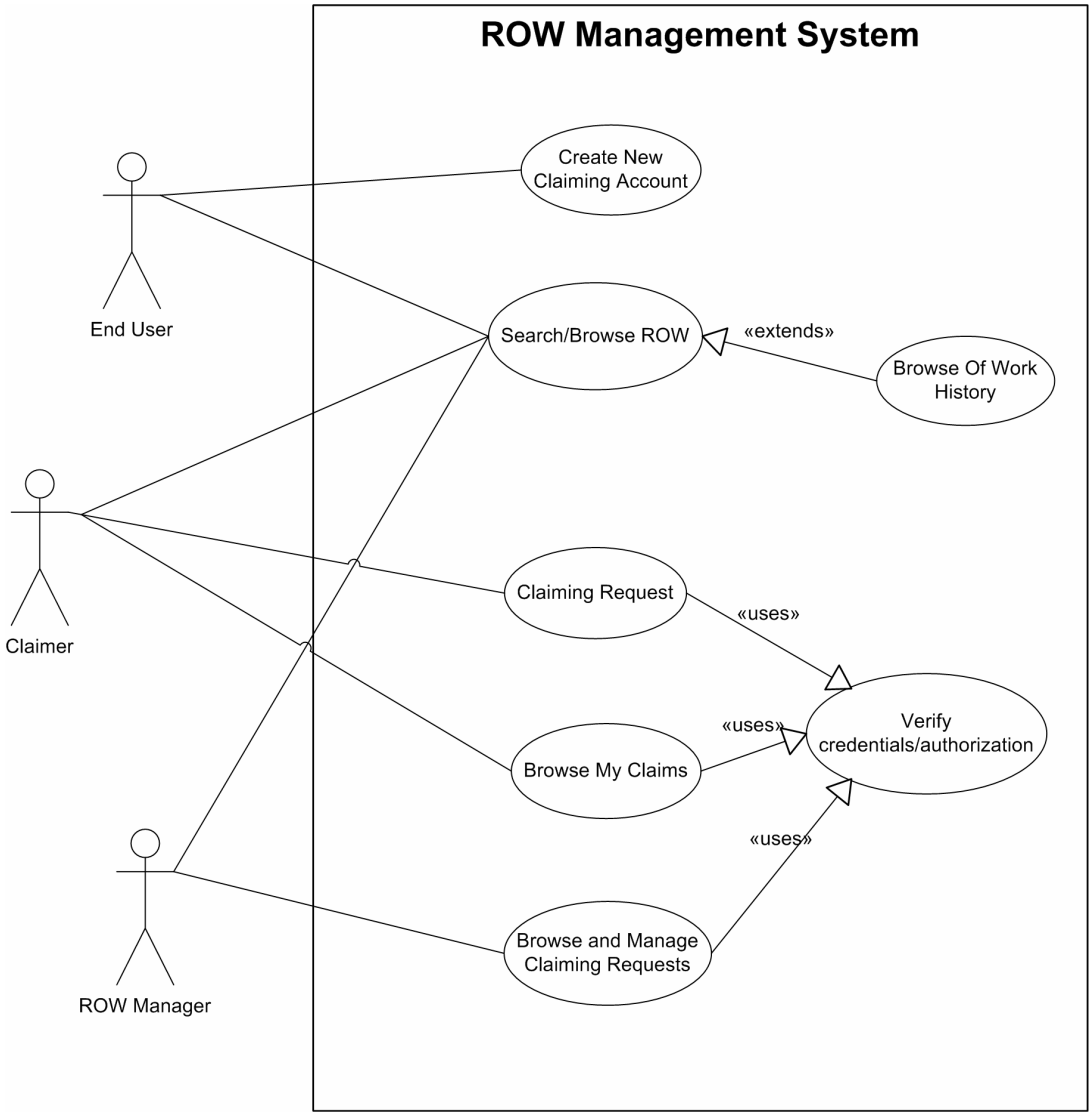


Figure 26: ROW Management System - Use Cases

Use Case: Search/Browse ROW
ID: UC17
Actors: End User, Claimer, ROW Manager
Preconditions:
Events sequence: The use case starts when the actor selects the functionality “Search/Browse ROW” from the main page. 1. If the actor selects “Search by Work” (default option), the system displays the work search

mask (default option) where the actor can search works by Title, Contributor(s) and Language of Text. The actor fills in the work data and then selects “Search” button.

1.1 The System processes the request and a list of works matching the required criteria is displayed. For each found work, its metadata as well as the list of corresponding manifestations are displayed. For each work it is possible to browse its history- (see how the work rights status; metadata and list of manifestations -from different reference sources- has changed over time). (In the Use Case Diagram the Browse Of Work History is modelled as an extension of the Search/Browse ROW one, since it is possible to browse the History only after retrieving the work search result.

2. If the actor selects “Search by Manifestation”, the search mask is enriched by further fields that comprise: ISBN (10, 13 or EAN), Publisher or Imprint, Country of Publication.

2.1 The System processes the request and a list of manifestations matching the required criteria is displayed. For each manifestation, the metadata provided by different reference sources as well as the related work metadata can be viewed

Post conditions:

Use Case: Create New Claiming Account
ID: UC18
Actors: End Users
Preconditions: The end user does not have a claiming account.
Events sequence: The use case starts when an End User wants to make a claiming request on an orphan work. 1. The System redirects the End User to the registration form. 2. The End User fills in the following information: • his/her personal info

- account type (author, publisher, agent)
- contact method (e-mail, telephone, fax)
- sign in information

3. The End User selects “Register”
4. The System performs the necessary checks to verify if the user already exists or if the mandatory fields are provided
5. The System creates a new system user authorized for performing claiming requests.

Post conditions:
A new claiming account has been created and its user is enabled to perform claiming request on orphan works

All the following Use Cases correspond to system services that are accessible only to authenticated users that own appropriate privileges (authorisation).That’s why all the following Use Cases use the Verify credentials/authorization Use Case.

Use Case: Claiming Request
ID: UC19
Actors: Claimer
Preconditions: The user is correctly authenticated and has the appropriate role for performing a claiming request.
Events sequence: The Use Case starts when the claimer selects the Claim link on a work after having performed a search in the ROW (see UC15) 1. The System displays the Claiming Request form that contains: • A brief summary of the work metadata that is subject to the current claiming • If the claimer is of type Author or Publisher, the system loads in the form the rightholder information • If the claimer is of type Agent, the Claimer enters the rightholder information data

2. The Claimer selects the Role of the rightholder with respect to the work 3. The Claimer submits the request 4. The system performs all the necessary checks (form required information provided) 5. The system updates and displays the list of own claiming(s) to the Claimer (see UC18)
Post conditions: A new claiming request is added in the system

Use Case: Browse My Claims
ID: UC20
Actors: Claimer
Preconditions: The user is correctly authenticated and has the appropriate role for performing a claiming.
Events sequence: The use case may have two starting points: if the Claimer selects “My claims” service or if the Claimer has just performed a claiming request (see UC17). 1. The system displays the list of all the claiming requests that the Claimer has performed. Each element of the list contains the work title, Rightholder Information, Request Status (pending, accepted, refused), claiming request date and claiming response date if available 2. The claimer can browse all the work metadata and filter the requests based on their status
Post conditions:

Use Case: Browse and Manage Claiming Requests
ID: UC21
Actors: ROW Manager
Preconditions: The user is correctly authenticated and has the appropriate role for browsing and evaluating claiming requests.
Events sequence:

The use case starts when the ROW Manager selects the ROW Manager service. 1. The System displays the list of all kinds of the claiming requests performed by all the claimers. Each request contains a summary of the work metadata, Claimer info, Rightholder info, Request Status (pending, accepted, refused), request date. 2. The ROW Manager can filter the Claiming Request by status. 3. In case the request is still pending the ROW Manager may choose to accept or refuse the claim. 3.1 In case the ROW Manager selects “Accept Claiming” the system updates the work Orphan status. 3.2 In case the ROW Manager accepts or refuses the claiming requests, the system notifies the result the claimer via e-mail
Post conditions: In case the ROW Manager accepts a claiming request, the corresponding work Orphan status is changed.

4.8. Software components

The figure below represents a list of the main AWR/ROW components included in the Arrow DataCentre and the way they interact with each other.

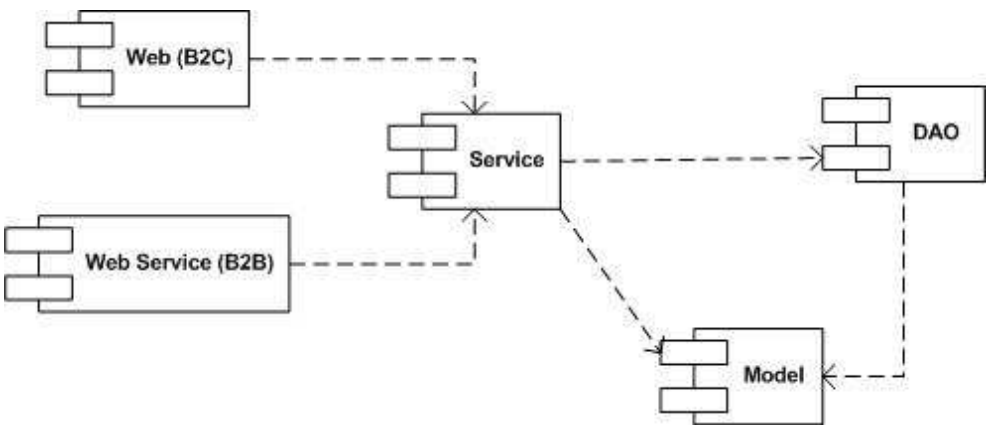
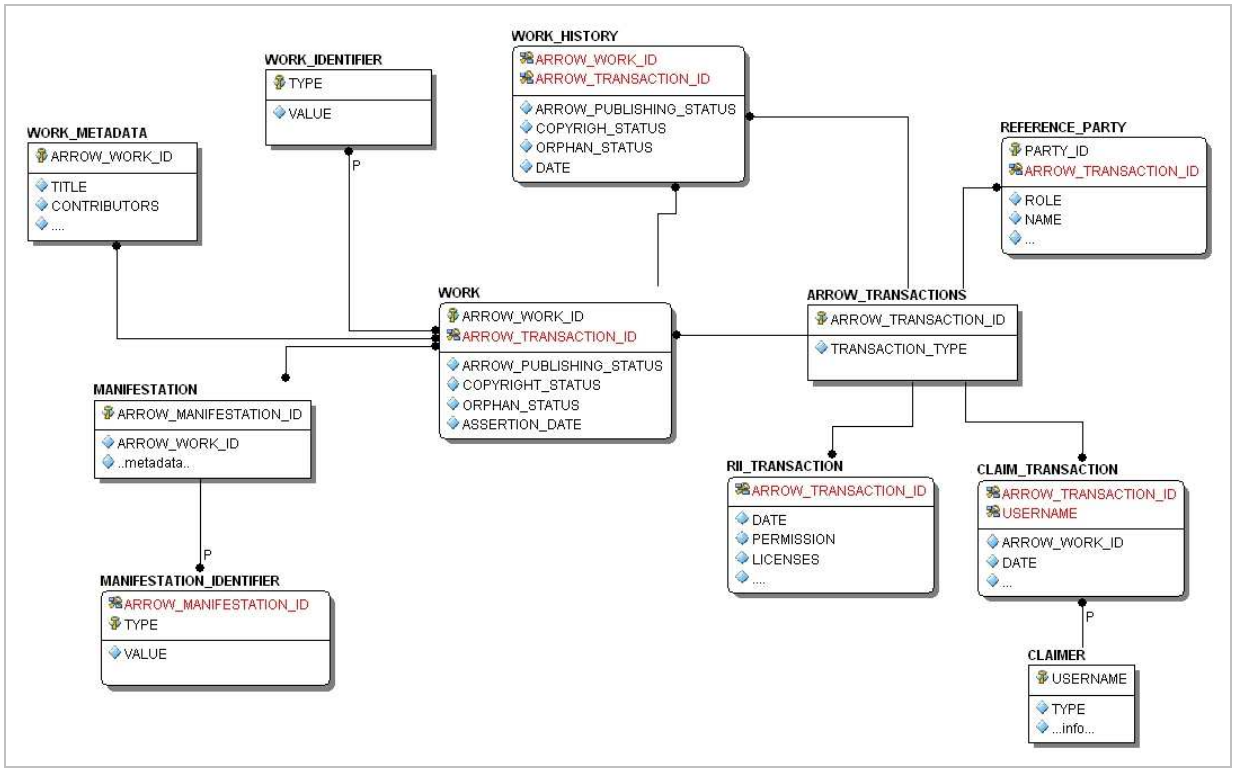


Figure 27: AWR/ROW Software Components

Component Name	Description	Hosted at
AWR/ROW Web (B2C)	This component represents the user interaction layer. It contains all the 'views' that renders the model into a form suitable for user interaction.	CINECA
AWR/ROW Web Service (B2B)	This component implements all the services exposed to automatic external systems, for example, functionalities related to the claim service.	CINECA
AWR/ROW Service	This component is the most important one. It contains all the business logic needed to manipulate the data in the application, including specific algorithms. It represents the core of the application and its main sub components are: Feeding Handler, Identification handler, History manager, Claiming manager.	CINECA
AWR/ROW Model	This component collects all the business objects used in the application. For example, the logic representation of the work with its properties and functionalities strictly related to it. Furthermore, the objects maintain connections with each other in a complex net of relationships.	CINECA
AWR/ROW DAO	This component represents the persistence layer. Everything that needs to be persisted passes through it. It implements a well-known design pattern to make available functionalities in order to access and manage data from and to the database.	CINECA

4.9. AWR/ROW Data Model

The following figure represents a logic view of some of the possible relations considered in designing the ROW, according to the requirements for the ROW gathered so far and can be extended with additional entities and relation if necessary.



A brief description of the model is provided starting from the WORK entity which is the ROW main one. This entity contains an ARROW_WORK_ID which uniquely identifies it in the AWR. Each work is also related to a list of identifiers of the same work (WORK_IDENTIFIER entity). Such list will initially contain the TEL Work ID and may further contain the ISTC assigned to the work. Other attributes of the WORK entity are the PublishingStatus, CopyrightStatus, OrphanStatus. They correspond to the last work status values obtained for that work as a consequence of an ARROW transaction. Such transaction is modeled by the ARROW_TRANSACTIONS entity and the relation between them is enabled by the ARROW_TRANSACTION_Id.

Work status and metadata may change due to different workflows such as: RII workflow handling library requests or claiming workflow on orphan works upon validation of in charge organizations. In order to express this scenario the ARROW_TRANSACTIONS entity is designed as a generalization of a RII transaction or claim transaction by the means of the TRANSACTION_TYPE attribute. This choice allows to scale to further workflows that may be added to the ARROW system influencing the above work data.

REFERENCE_PARTY entity models the role of the different organizations playing a role in the ARROW system, related both to the RII and to the ROW. For example it contains the necessary information for identifying an RRO or a BIP as well as a ROW Manager. A relation between the

ARROW_TRANSACTION entity and the REFERENCE_PARTY entity exists since the ARROW System always needs to know the responsible organization for the particular transaction. For example the RRO declaring work status data (RII_TRANSACTION) or confirming claiming data (CLAIM_TRANSACTION).

In order to maintain the work history, the WORK_HISTORY entity is provided. This entity contains an ARROW_WORK_ID, the work status information, the ARROW_TRANSACTION_ID and a DATE. Each time an ARROW_TRANSACTION is performed, this entity is enriched with the previous work information (the ones represented by the WORK entity). The WORK entity itself is updated with the most recent information. There is obviously a relation between the WORK entity and the WORK_HISTORY entity that enables to trace the changes that a work has undergone in time. The WORK_HISTORY also maintains the ARROW_TRANSACTION_ID which is a very important attribute that enables to retrieve the workflow (and relative details) that dictated such change. The DATE attribute traces the date in which a particular work change was performed.

The CLAIMER entity models the necessary information for identifying and describing claimers.

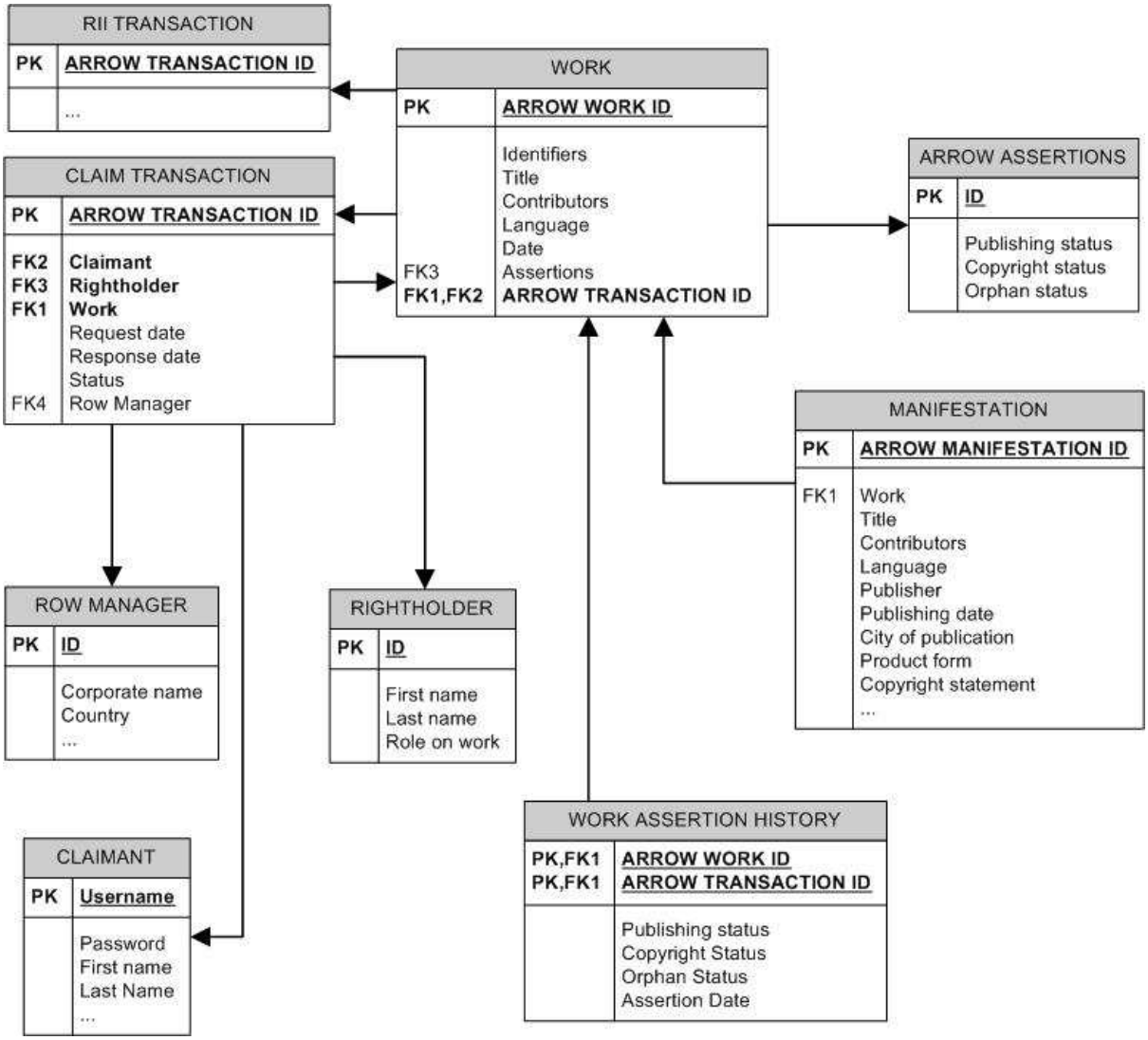
CLAIM_TRANSACTION entity uniquely identifies a claiming ARROW transaction. The relation between these two entities is necessary to trace different claims performed by a claimer as well as to link its data with the claimed work.

RII_TRANSACTION entity models the RII part[21].

4.10. ARROW AWR/ROW Repository

A schematic representation of the AWR/ROW database is presented in the figure below:

[21] For more information on RII_TRANSACTION entity model, see D6.1 *Rights Information Infrastructure*


RII TRANSACTION
PK ARROW TRANSACTION ID
...
WORK
PK ARROW WORK ID
Identifiers
Title
Contributors
Language
Date
FK3 Assertions
FK1,FK2 ARROW TRANSACTION ID
ARROW ASSERTIONS
PK ID
Publishing status
Copyright status
Orphan status
CLAIM TRANSACTION
PK ARROW TRANSACTION ID
FK2 Claimant
FK3 Rightholder
FK1 Work
Request date
Response date
Status
FK4 Row Manager
MANIFESTATION
PK ARROW MANIFESTATION ID
FK1 Work
Title
Contributors
Language
Publisher
Publishing date
City of publication
Product form
Copyright statement
...
ROW MANAGER
PK ID
Corporate name
Country
...
RIGHTHOLDER
PK ID
First name
Last name
Role on work
WORK ASSERTION HISTORY
PK,FK1 ARROW WORK ID
PK,FK1 ARROW TRANSACTION ID
Publishing status
Copyright Status
Orphan Status
Assertion Date
CLAIMANT
PK Username
Password
First name
Last Name
...

5. Arrow system deploy

The following figure displays the deployment of the entire ARROW system with all the involved components.

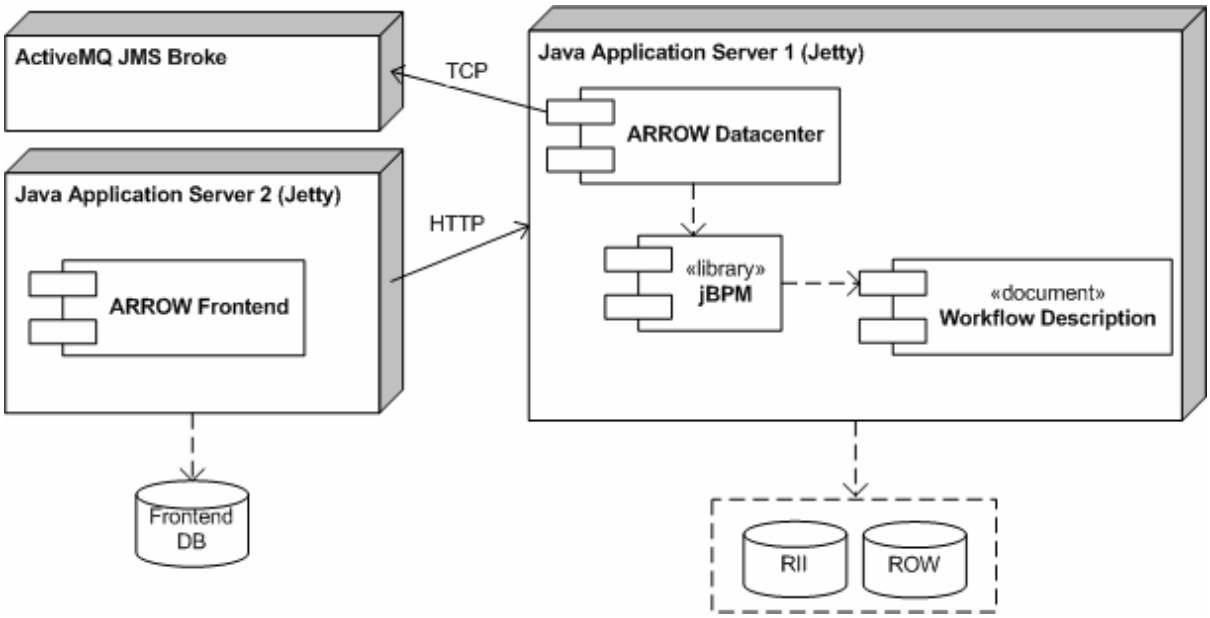


Figure 28: ARROW System Deploy

The ARROW Frontend is deployed as a web application into its own jetty instance. It communicates with DataCentre over HTTP protocol. The ARROW DataCentre is also deployed as a web application into its own jetty instance accepting requests from outside over HTTP protocol.

ActiveMQ is a standalone service accessible via TCP protocol. It offers us an administration console too.

The relational DBMS used for the ARROW repository is Oracle.

Conclusions and Future Work

Right management in the mass digitization programs has high costs (in terms of resources -human and economic - as well as in terms of time) mainly due to the difficulty to seek and find all the necessary information.

In order to facilitate the right management, the ARROW project aimed to set up a Rights Information Infrastructure (RII) and a Registry of Orphan Works (ROW).

RII is a distributed system for facilitating rights information management in any digitisation programme, scalable to further applications.

The current implementation of RII facilitates the diligent search necessary for identifying the rights status of any textual work that a library wishes to digitise and make available online. It is now ready to be used and validated by libraries that wish to digitise books published in the following countries: Germany, United Kingdom, Spain and France.

RII is standard oriented, scalable and flexible:

- Standards orientation can be seen from different perspectives; for example, the system allows the encoding of data according to the most widely spread metadata formats adopted in the different domains involved: MARC 21 XML in library domain, ONIX for Books in BiP domain, ONIX-Licensing Terms in the RRO one. The data exchange with different data providers is performed by using Web Services that are platform-independent and language-independent. This standard oriented approach provides the interoperability among all the involved resources.

- Flexibility can be seen from different perspectives; for example easy integration and handling of other standard metadata formats (i.e. UNIMARC), easy workflow modification and extension in order to adapt the system to different national scenarios.

- Scalable as the system has the capability to handle growing amounts of work in a graceful manner. For example, further countries or further data providers can be easily integrated in the system without heavily impacting system performance.

RII is currently in open Beta release and is already used for demonstrating the system to interested parties and stakeholders of different countries.

Regarding the second objective of the ARROW project not only a Registry of Orphan Works has been set up, but also a more comprehensive registry of works (AWR) has been created.

In the current release of the AWR/ROW, the registries get fed from the results and information collected during the RII workflow.
The AWR is important since if forms the basis for the Registry of Orphan Works (ROW) as well as constitutes an added value to the system since the set of works and related information can be exploited in several scenarios such as: supporting different players of the book value chain that are interested to obtain information not only at "manifestation level" but also at "work level" or fostering the adoption of the ISTC ISO standard.

The Registry of Orphan Works is not just a mere registry but a registry at the top of which several services have been implemented. The scope of these services is to enable the consult and the management of the orphan works. The consult service comprises the access to the registry for searching and browsing works and manifestations. It is important since it allows eventual rightholders to find own works. In this case rightholders/agents can claim rights by using the claiming service. The management service enables the eventual ROW Manager to browse, view and evaluate pending claiming requests.

The current release of the AWR/ROW (Beta) was designed and implemented following the key principles provide by the HLEG, for the set up and management of databases of Orphan Works and of related Rights Clearing Centres. The system architecture was designed in order to be flexible enough to be adapted to different operative solutions (Interoperable National ROWs and/or Centralised ROW) that may emerge as soon as a definition of the legal framework concerning Orphan Works will be established in Europe.

The ARROW system and its algorithms will be further enhanced based on the guidance and requirements that came out during the Validation phase (D7.2 Validation Report).

Since the success of ARROW depends on it being used, at present, the ARROW Management Board is analysing the possibility to exploit the ARROW System within several National Initiatives emerging in Europe. In Germany and France for instance, stakeholders are finalising agreements to handle rights clearance for digitisation plans, while in United Kingdom a private charity is working on a digitisation

plan that includes a full diligent search. To serve these particular initiatives (use cases) that may arise from different legal framework, business models etc…, some customisation of the ARROW system may be required in order to comply with different national scenarios.

The required system enhancements and adaptations (to be accomplished during the ARROW Plus project) aim to reach a system version that may be shipped for revenue and that can be employed in future digital market applications where the rights management is crucial.

List of Annexes:

D6.4_ANNEX_I_110228_ARROW_Technical Infrastructure