CLARINO+ WP2
Rapport

9.12.2022

D2.5.2 Best practice in metadata descriptions, v1.0

Oddrun Pauline Ohren (National Library of Norway)

Summary

This document aims to guide Norwegian CLARIN centers (CLARINO) in creation and management of good quality metadata for language resources. The guide starts with a brief description of the CLARIN metadata infrastructure (CMDI), followed by an outline of metadata profiles and tools to be used in CLARINO. The main part of the document explains how to go about describing language resources, including general approach, scoping and granularity of resources, as well as practical advice about filling in the chosen metadata profile. Some words about metadata exposure to discovery services are also included. In the last chapter, sources to more help and information are outlined.

List of abbreviations and acronyms

CCR	CLARIN Concept Registry
CMDI	Component Metadata Infrastructure
LR	Language resource
NCLR	Norwegian catalogue of language resources
OAI PMH	Open Archive Initiative Protocol for Metadata Harvesting
PID	Globally unique persistent identifier
TEI	Text Encoding Initiative
VLO	Virtual Language Observatory

Content

1 Introduction

The purpose of this document is to advice language resource (LR) providers about metadata issues, that is which metadata to create and how to go about creating them.

In the CLARIN context, metadata about the provided resources are essential for visibility and access, even more so than for traditional library catalogues. While the latter often describe human-readable resources like books and articles, that is generally not the case for LRs, - a type of resources dominated by data-sets intended for interpretation and processing by machines rather than humans. Hence, metadata constitute the main source of information about the LRs, a fact that in itself emphasises the importance of good quality.

The 2 main uses of metadata we should aim for, are that the metadata should

- make the resources *findable*, that is, contain information that makes them show up in discovery services
- enable the user to *decide on the suitability* of a specific resource for his/her purpose. This means, it is not enough to include information that brings the resource into the search result list, the metadata must also be rich and detailed enough to determine with reasonably high probability whether it can be used by the specific user in a specific project.

1.1 A few words about quality (of metadata)

A commonly used definition of quality is “fitness for purpose”; something has good quality if it is suitable for what it is intended. However, while metadata may have many different purposes, in practice, most quality evaluation frameworks described in the literature take a holistic perspective. Thus, most authors rely on the idea of a purpose-independent (or multi-purpose) notion of metadata quality, defining general purpose quality parameters. The well-known metadata quality framework by Bruce and Hillman (2004) includes 7 quality aspects to be evaluated: completeness, accuracy, provenance, conformance to expectations, logical consistency and coherence, timeliness and accessibility.

- ***Completeness***: Applies both to the metadata model used, and the actual metadata. Completeness of the metadata model involves considering whether the element set of the model provides constructs for expressing all important aspects of the target objects (LRs). In our context, this is something for the metadata modeler (profile authors) to consider, not the metadata creator. Completeness of metadata (according to a model) concerns the extent to which the relevant elements in the metadata model are actually used to describe the target objects (LRs)
- ***Accuracy***: To which degree the information/facts conveyed by the metadata should be correct and have a correct form
- ***Provenance***: Whether life-cycle information about the metadata is included. Examples of such data metadata creator, creation method (e.g. manual vs. automatic creation), modifications, etc

- ***Conformance to expectations***: How well the metadata satisfy the *expectations* from relevant communities or target groups. In our case, researchers/end users, CLARIN and other service providers are important target groups to consider. For example, catalogue services like The Norwegian catalogue of language resources[1] (NCLR, maintained by the National Library of Norway) and Virtual Language Observatory[2] (VLO) by CLARIN, may have specific expectations to metadata, to be able to expose the described resources favourably.
- ***Logical consistency and coherence***: To which degree the metadata elements are applied consistently across resources, comply with their definitions in the metadata model and are coherent with concepts used in related communities and subject domains.

To support LR providers in creating metadata with sufficient quality according to the aspects above, CLARINO has developed metadata profiles, offering guidance on which metadata elements to include, which values to record, etc.

2 The metadata infrastructure

In the following, the basics of the metadata infrastructure as proposed by CLARIN is outlined. In our context, the expression “metadata infrastructure” is meant to comprise data model and metamodel, metadata profiles (schemas) as well as metadata tools.

2.1 CMDI – the CLARIN approach to metadata

As part of CLARIN, CLARINO uses metadata profiles compliant with the Component Metadata Infratsructure[3] (CMDI) provided by CLARIN. The main goal behind CMDI is to offer flexibility in metadata descriptions, while at the same time catering for interoperability between descriptions. This is not done by offering a set of fixed metadata schemas, but a standard way of building one’s own metadata schemas, - in other words, a shared *metamodel* for resource descriptions. More concretely, CMDI enables users to build reusable metadata components to be combined into larger components, and eventually into whole profiles. The profiles can then be loaded into editors and applied for metadata creation. All components and profiles are stored and managed in the CLARIN Component Registry[4]. A typical component combines information elements and other components that represent entities related to a resource or aspects of a resource, while a profile typically combines elements and components suitable for describing resources of a certain type. For example, the profile corpusProfile may be applied to describe resources of type corpus, whereas toolProfile may be used to describe tools. Both profiles include the component licenceInfo, to hold information about the licence governing the described resource.

[1] https://www.nb.no/sprakbanken/en/resource-catalogue/

[2] https://vlo.clarin.eu

[3] https://www.clarin.eu/content/component-metadata

[4] https://catalog.clarin.eu/ds/ComponentRegistry

Interoperability between profiles is obtained by relating elements and components in CMDI profiles to concepts in CCR[5], denoting their meaning. Thus, if an element *Title* in one profile and an element *Name* in another profile both are connected to the same CCR concept resource title, the conclusion is that Title and Name "mean" the same thing, and can be handled similarly in most applications processing federated data.

2.2 CMDI profiles for CLARINO

In general, metadata authors in CLARINO will not need to concern themselves about the inner workings of CMDI as described above. A handful of profiles are already developed specifically for CLARINO, which should suffice for most resources. Therefore, the tasks of metadata creators are to

1. Select the CLARINO profile which best suits the resource you are about to describe
2. Fill in the information asked for in the profile, to the best of one's knowledge, see chapter 4 for more details.

The current CLARINO metadata profiles are as follows (Component Registry identifier in parenthesis):

- corpusProfile (clarin.eu:cr1:p_1407745711925): To describe a corpus running text/speech, treebanks, ngrams and more. The content may be represented in any medium. The profile Includes facilities to describe distinct parts of the resource separately, for example if one part is audio and another part contains text or images.
- lexicalProfileRev1 (clarin.eu:cr1:p_1548239945780): To describe lexical resources, i.e. typically terminological resources organized according to their lexical/conceptual units. Examples of lexical resources are dictionaries, glossaries, taxonomies, thesauri, lexica, wordnets and ontologies.
- teiProfile (clarin.eu:cr1:p_1422885449322): To be used for resources already encoded using TEI[6] and TEI Header. The teiProfile is the result of adding the generic component resourceCommonInfo to the teiHeader component, to make the resulting records fit NCLR.
- toolProfile (clarin.eu:cr1:p_1562754657363): To describe (resources that are) language tools, for example taggers, parsers, translators and search services

2.2.1 resourceCommonInfo – the component for general information

All the CLARINO profiles listed above include the component resourceCommonInfo, designed to contain information applicable to resources of any kind. Such information include:

- name/title, identifier and textual description
- resource type
- information about availability, e.g. licence and access method

[5] https://www.clarin.eu/content/clarin-concept-registry
[6] https://tei-c.org/

- information about versioning and validation
- contact information
- resource provenance: information about how the resource was created, as well as who created its metadata
- reference to any other documentation about the described resource

By gathering general information into one specific component, the effort of modelling such information elements is done “once and for all”, and creators of future CLARINO profiles need only remember to include this component in their new profiles.

2.2.2 Metadata components targeted to specific resource types

The information elements requested by resourceCommonInfo is usually not sufficient to represent the resource fully. In addition, information elements specific for the type of resource may be needed to give the user a reliable idea of its potential usefulness.

Such information elements vary across resource and media types, typically those that represent technical features specific for some, but not all the types. For example, information about *annotation* is generally relevant for corpora, but not for lexical resources. On the other hand, information about which *languages* that are represented is relevant for lexical resources as well as corpora, while not necessarily relevant when describing tools.

2.2.3 When the CLARINO profiles are not enough

While our recommendation is to use one of the profiles above, new profiles can be developed if needed. However, inexperienced CMDI users are not advised to create new components and profiles on their own, please approach the CLARINO metadata contact for help, see Chapter 6.

Note that any new profile must follow the CLARINO rule, as explained below.

2.2.4 The CLARINO rule for metadata profiles

All profiles to be used in CLARINO shall include the component resourceCommonInfo (clarin.eu:cr1:c_1396012485126), designed to host general information applicable to all resource types. This rule enables consistency in core metadata across Norwegian providers, both in VLO and in our national metadata registry NCLR.

3 Tools for editing metadata

There are basically 3 levels of support for creating CMDI metadata.

1. Using a general, CMDI-agnostic XML editor, for example oXygen, for typing in the information. Instructions on how to create a CMDI file with oXygen can be found in the CMDI FAQ[7]. This approach is not recommended in CLARINO, as it leaves the responsibility for metadata correctness and quality entirely by the user.

[7] https://www.clarin.eu/faq/how-do-i-create-new-cmdi-metadata-file

2. Using a non-CMDI metadata scheme and tool for creating the metadata, converting the result to CMDI afterwards. This approach is relevant for providers storing their resources in a repository with their own metadata handling facilities and requirements. Note that the original metadata should be converted to one of the CLARINO profiles.
3. Using a CMDI-supporting metadata editor. COMEDI[8] , developed within CLARINO, is an extensive metadata editor with good editing facilities. COMEDI is not limited to any particular profile, - any CMDI profile can be loaded into the editor and used for creating metadata.

Best practice for creating metadata in CLARINO is to use a CMDI-supporting metadata editor, in practice, to use COMEDI.

3.1 Using COMEDI

While COMEDI may be installed locally, the recommendation is to use it from its main site: https://clarino.uib.no/comedi. Be sure to read the documentation before using it for the first time. Also, defining a group representing your CLARINO centre may prove convenient, especially if more than one person plan to create metadata.

4 Creating metadata

This chapter describes the structure and content of a CMDI metadata record, and how to go about creating it.

4.1 CMDI metadata records – structure and content

Below is a description of the main parts of a metadata record/file created based on a CLARINO profile. For a better overview, the record structure is visualized in Figure 1.

1. Header
   - The header is included in, and structurally identical for all CMDI records, and contains key information about the CMDI file as such. Although only one element is required (MdProfile), it is highly recommended to fill in this part conscientiously, as the header is important for interoperability, especially in federated services. The following information should be given:
     - Creator of the metadata (MdCreator): If using COMEDI, MdCreator is automatically set to the COMEDI user.
     - Date of creation (MdCreationDate) of metadata file: Automatically set by COMEDI.
     - The profile used (MdProfile): The identifier of the CMDI profile. Automatically set by COMEDI. Exactly one profile must be given. Set automatically by COMEDI in editing mode.

[8] https://clarino.uib.no/comedi

        - Reference to this metadata file (MdSelfLink): A link in the form of a PID or URL to this metadata record in its home residence.
        - Name of collection to which the resource belongs (MdCollectionDisplayName): Although not a controlled label, it is useful for display purposes to keep related resources together in a user interface.

2. List of resources
    - A specification of the dataset(s), file(s) etc that constitute the language resource you are about to describe. Each item in this list is referred to as a *resource proxy* (ResourceProxy), and the specification itself as ResourceProxyList.
3. List of relations between resource files (ResourceRelationList)
    - A specification of significant binary relations between the individual resources listed in ResourceProxyList. For example, if one of the files contains annotations to the content in another file, this fact may be expressed by generating a relation (ResourceRelation) *annotates* between the two.
4. General metadata
    - General metadata about the resource, as required by the resourceCommonInfo component
5. Specific metadata
    - Metadata as required by any other components in the applied profile, typically specific to the type of the resource in question.

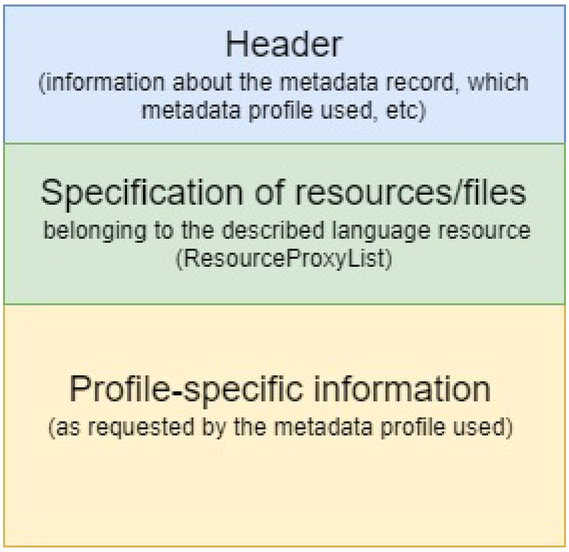
Header
(information about the metadata record, which metadata profile used, etc)
Specification of resources/files
belonging to the described language resource (ResourceProxyList)
Profile-specific information
(as requested by the metadata profile used)

Figure 1 The 3 main parts of a CMDI metadata record. Only the lowest part depends on the selected profile.

4.2 The main metadata process

There is no mandatory order or sequence in which the above metadata parts must be created. Even so, there are more or less intuitive ways of going about it, - below is described an approach that should be a good starting point. While the proposed *main* sequence is independent on which metadata tools are used, that is not the case when it comes to the *details* of the metadata record. Your choice of tool (metadata editor) has great influence on the level of support offered during the editing process. The proposed metadata authoring workflow can be outlined as:

1. Select and activate a suitable CLARINO profile
2. Fill in the header information
3. Define, delimit, and specify your language resource
4. Insert the essential general information (fill in the resourceCommonInfo)
5. Insert profile specific information

Below each step is described in more detail

4.3 Select and activate a suitable CLARINO profile

When creating a new metadata record, the first thing to do is to select and activate a metadata profile to use for describing the resource in question, preferably a CLARINO profile. If using COMEDI, there is a good chance that the appropriate profile is already available in the tool. If not, or if using another tool, you might have to look up the profiles in the CLARIN Component Registry, filter by status *Development* and sort by column Group (*CLARINO*). Select a profile and load it into COMEDI.

You should now have a skeleton metadata record, ready for you to fill in. It should contain profile-specific fields as well as fields present in all records, irrespective of the chosen profile.

4.4 Fill in the Header information.

If using COMEDI, only MdSelfLink and MdCollectionDisplayName are given by the user, the rest is provided automatically by COMEDI. The collection name should be a string as you would like your collection to be displayed to end users.

4.5 Define, delimit, and specify your language resource

It is vital to be clear on exactly which items (datasets, documents/files, compressed archives a.o.) that make up the language resource to be represented by this metadata record. This is to a certain extent a question of policy, and by no means self-evident. However, once a clear idea of the structure and configuration of your resource has been obtained, everything else should fall reasonably easy into place.

4.5.1 What is a resource?

Starting out with a set of text and/or other media files that are to be offered for general use to other researchers and developers, it is not always easy to decide whether the files/data sets in question should be published en bloc as one resource, or organized into multiple resources, each with their own metadata. Although there is no one correct answer to this, doing some careful thinking on which level of granularity to adopt in a repository of language resources, is well worth the effort, as it may have strong effect on the usability of your resources.

If you are in doubt whether your data should be considered as one or multiple resources, ask yourself the following questions:

- Is there an obvious way to subdivide the data into multiple resources?
- Can each identified part be used independently of the other parts? And are there probable use cases for such usage?

If the answers of the above are in the affirmative, you might consider partitioning your resource into multiple resources, and create a metadata record for each one. If a holistic view is needed in addition, you might create a metadata record for the total resource, in which the parts are listed as single resources.

4.5.2 Specifying your resource in ResourceProxyList

Once the basic structure of your resource is established, its content parts/files are to be listed as ResourceProxy elements in the ResourceProxyList. Each ResourceProxy should be described by the following information elements:

- A *reference to the item* (a ResourceRef) in the form of a PID (preferred) or a URL.
- A *local identifier* for the item, a string, unique within this metadata record. To be referenced in metadata elements which applies only to this specific resource proxy
- The *resource type* of the item (see below)
- The *media type* (aka mime type) of the file.

Note that the ResourceProxyList element is included in all metadata files, irrespective of which metadata profile is used. That said, the content of ResourceProxyList, including the nature of its ResourceProxy elements, very much depends on how your language resource is structured as well as the infrastructure in which your resources reside.

4.5.3 Granularity and composite language resources

Below three ways of representing a composite language resource LR#1 in the ResourceProxyList are described and illustrated. Assume that an initial analysis has concluded that the resource may be subdivided into three logical parts. CMDI allows for several ways of configuring the resource specification (ResourceProxyList) for LR#1, as outlined in the following by Alternative 1-3. In the accompanying figures, "List of resources" and "ResourceFile" correspond to ResourceProxyList and ResourceProxy, respectively.

Alternative 1: Represent the whole resource LR#1 as one, disregard partitioning.

In this case it is not considered necessary, nor particularly useful to represent each resource part by separate metadata records. Then one metadata record for the whole resource is generated, and all files making up the resource are listed in its ResourceProxyList, as visualized in Figure 2 below.

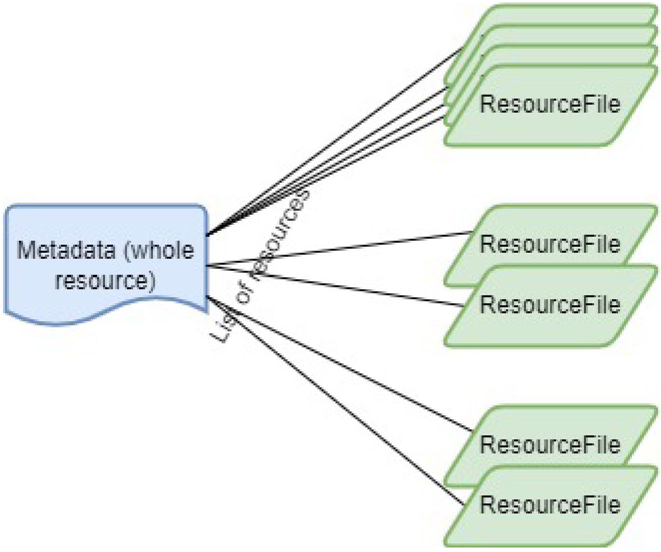

Figure 2 LR#1 is represented as a whole, with all reseource files connected to the metadata record

Alternative 2: Represent only the parts of LR#1

In this case, representing each part as individual resources is considered more important than keeping them together. If so, the notion of LR#1 as one resource may be disregarded altogether, leaving us with 3 completely separate resources, each with their resource files, as illustrated in Figure 3.

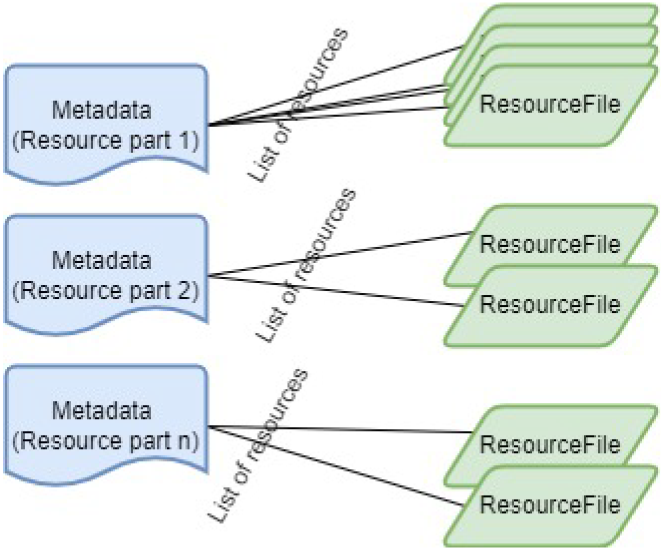

Figure 3 LR#1 as a whole is represented implicitly by 3 independent resources

Alternative 3: Represent each part, as well as the whole LR#1, as separate resources.

In this case the items to list in the ResourceProxyList of LR#1 will be the metadata files of its parts, whereas the actual resources should be listed in the metadata record for the part to which they belong. If needed, the inverse hierarchical relation may be recorded in the metadata records of each part, as an IsPartOf relation.

This is the most complete (and complex) way of representing a language resource with distinct parts, see Figure 4 for visualization.

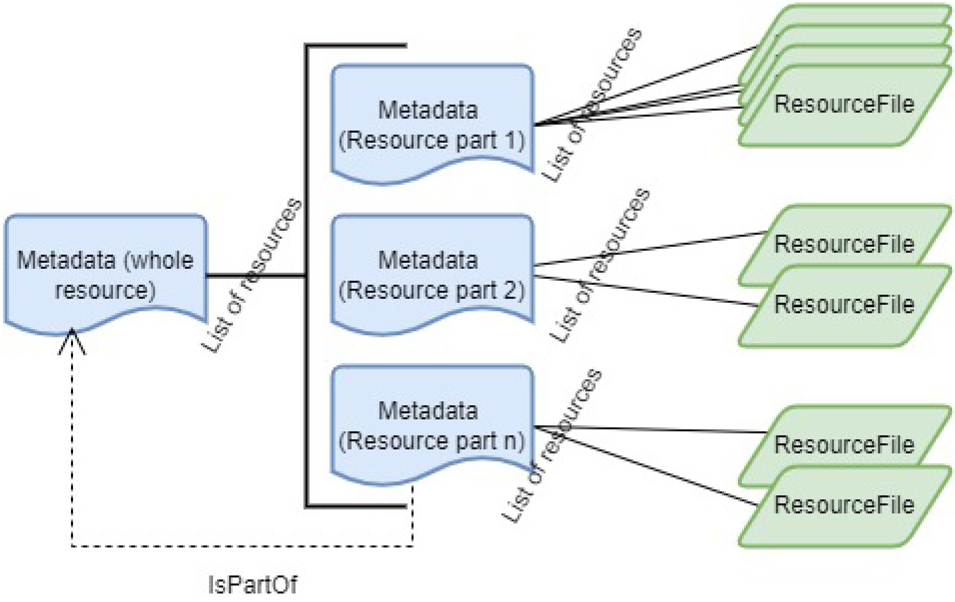

Figure 4 LR#1 is subdivided into 3 separate resources, each with their own metadata record

4.5.3.1 Choosing between the alternatives

It is good practice to choose the simplest alternative that is "good enough". While alternative 3 is richer and as such expresses the structure of your resource in a more complete way than the other two, it is also more demanding to maintain. Changes in any of the three parts may cause need for update also in the parent resource. Ultimately a complete restructuring may be needed.

Another issue to consider is consistency in how your resource collection is represented by metadata. Whenever possible, it is good practice to adopt the approximate same granularity level throughout your collection of language resources.

4.5.4 Assigning types and media types to ResourceProxy entries

In all examples visualised above, the resources listed are meant to be interpreteded as actual data files. However, CMDI also allows for more indirect ways of referring to your content files. This is expressed by decalring that your resources are of certain *types*.

All resources listed in the ResourceProxyList *must* be assigned a type and *should* be assigned a media type.

4.5.4.1 Resource types

In CMDI the following types are recognised as valid for ResourceProxy entities:

- *Resource*: “Real” data that are available directly from this metadata record, for example text documents, media files or tools. All resources in the examples visualized above (the green parallellograms) are meant to represent resources of type *Resource*.
- *Metadata*: Reference to another CMDI record. To be used for items described separately in their own metadata records. In such cases it is best practice to list the resource proxies by their metadata records (using their MdSelfLink), not by their data files. An example of resources of type Metadata is given in Figure 4.
    - The media type of metadata resource should be set to application/x-cmdi+xml.
- *SearchPage*: Points to a web page at which end-users may search your resource. It is best practice to provide no more than 1 SearchPage.
- *SearchService*: Points to a web service that can be called from dedicated applications to query your resource. It is best practice to provide no more than 1 SearchService.
- *LandingPage*: Points to a web page that provides the original context of your resource, for example by showing core metadata up front together with links into a repository system. It is best practice to provide no more than one landing page.

Below the resource types SearchPage, SearchService and LandingPage are illustrated.

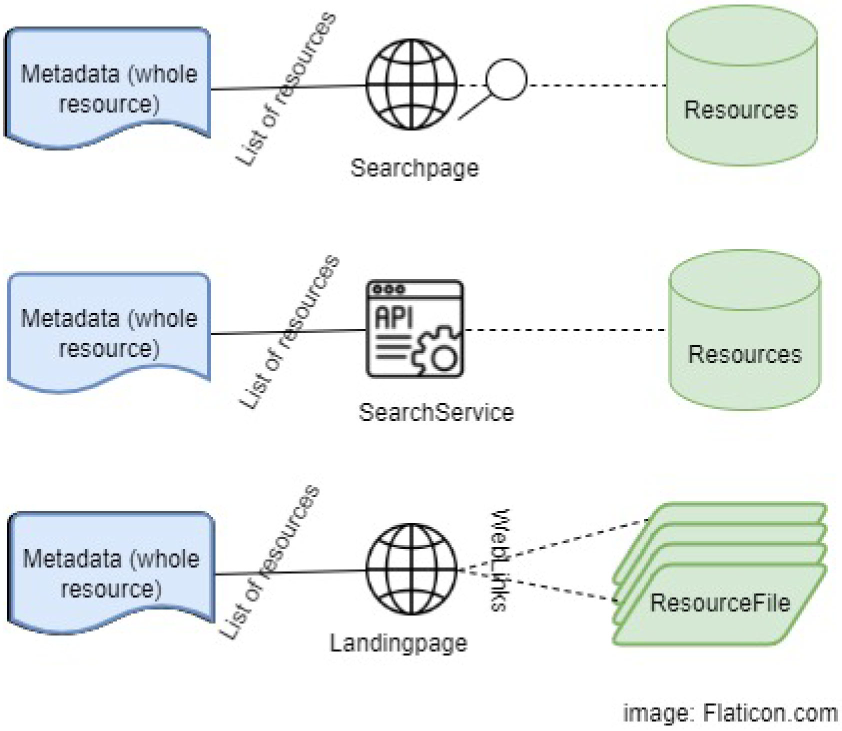

Figure 5 ResourceProxy of the types SearchPage, SearchService and LandingPage

4.6 Insert the essential general information – fill in the resourceCommonInfo

While CLARINO handles a wide range of LR types, with a great variety of features, there still exist some generic information elements, applicable to resources of any type, and particularly essential when it comes to findability of the resource. Such elements include

- Identifying information elements, such as resource title or name, identifiers, homepage and a natural language description
- Information about licence, rights and conditions of use
- Information about how to access and use the resource
- Information about version and validation
- Information about agents (persons or organisations) possessing key roles vis a vis the resource, such as creators, responsible agents, funders, metadata creators and, most importantly, contact agents.

4.7 Insert (other) profile specific information

This involves information required by the rest of the specific profile used. Be sure to assign values to all required elements.

COMEDI offers several shortcut and copying facilities which makes the metadata work less tedious. Note that rich and correct metadata gives the best guarantee that your resource shows up to advantage in relevant discovery services like NCLR and VLO.

4.7.1 Language used in metadata

It is best practice to provide all “readble” metadata in at least English and Norwegian (either bokmål, nynorsk or both). Metadata in other languages may be included as deemed appropriate for the individual resource or collection.

5 Language Resource exposure

Most data providers in CLARIN want to expose their language resources to a broadest possible user group. If so, information in the form of metadata about language resources owned by individual centres should be shared throughout the CLARIN community. The most convenient way of doing this, is to make your metadata available to relevant discovery services. Norwegian providers should at least aim for the national NCLR and CLARIN VLO, - in addition to any centre-specific service.

Making metadata available for NCLR and VLO involves establishing an OAI-PMH server containing your metadata and governed by appropriate updating procedures, from which the said services can harvest your CMDI data. Note: For harvesting to take place, your centre as well as link to your OAI server must be registered in the Centre Registry[9].

[9] https://centres.clarin.eu/.

6 Help and more information

The CLARIN website[10] is a rich source of information and assistance with issues pertaining to metadata, especially the Component Metadata page[11], which is organised into topics, including indroduction and general overview; instructional examples and data sets, as well as services related to creation, validation and usage of CMDI metadata.

Among the many resources referenced here, the chapter Component Metadata Infrastructure (Windhouwer and Goosen 2022), the current CMDI specification (CMDI Taskforce 2016) and the CMDI best practices Guide (CMDI Taskforce and Metadata Curation Taskforce 2017) are important. The latter was also presented at CLARIN Annual Conference 2017, of which a video[12] is available.

For help with metadata issues specifically related to CLARINO, email CLARINO's metadata contact[13].

7 References

Bruce, T. R. and D. I. Hillman (2004). The continuum of metadata quality: Defining, Expressing, Exploiting. Metadata in Practice. D. I. Hillman and E. L. Westbrooks. Chicago, Illinois, American Library Association**:** 238-256.

CMDI Taskforce (2016). CMDI 1.2 specification.

CMDI Taskforce and Metadata Curation Taskforce (2017). CMDI Best Practices.

Windhouwer, M. and T. Goosen (2022). Component Metadata Infrastructure. CLARIN. F. Darja and W. Andreas. Berlin, Boston, De Gruyter**:** 191-222.

[10] Clarin.eu

[11] https://www.clarin.eu/content/component-metadata

[12] http://videolectures.net/clarinannualconference2017_windhouwer_practices/

[13] mailto:sprakbanken@nb.no

8 Revision history

Date	What	Who
07.07.2021	Preliminary version	Oddrun Ohren
3.12.2021	Version 0.8	Oddrun Ohren
9.12.2022	Version 1.0	Oddrun Ohren