



## **Updated metadata governance tools report**

## Document Metadata

Property	Value
Date	18/05/2017
Version	1.00
Authors	Alexandru Droscariu – PwC EU Services Ana Fernández de Soria Risco – PwC EU Services Emidio Stani – PwC EU Services Ioana Novacean – PwC EU Services
Reviewed by	Nikolaos Loutas – PwC EU Services Susanne Wigard – European Commission
Approved by	Susanne Wigard – European Commission

**This study was prepared for the ISA Programme by:**

*PwC EU Services*

**Disclaimer:**

The views expressed in this report are purely those of the authors and may not, in any circumstances, be interpreted as stating an official position of the European Commission.

The European Commission does not guarantee the accuracy of the information included in this study, nor does it accept any responsibility for any use thereof.

Reference herein to any specific products, specifications, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favouring by the European Commission.

All care has been taken by the author to ensure that s/he has obtained, where necessary, permission to use any parts of manuscripts including illustrations, maps, and graphs, on which intellectual property rights already exist from the titular holder(s) of such rights or from her/his or their legal representative.

## Table of Contents

<b>GLOSSARY</b>	<b>IV</b>	
<b>1. INTRODUCTION</b>	<b>5</b>	
1.1.	OBJECTIVES & SCOPE	5
1.2.	APPROACH	5
	<i>Selection criteria</i>	6
1.3.	STRUCTURE	7
<b>2. REQUIREMENTS</b>	<b>8</b>	
2.1.	CREATE, USE AND EXTEND	8
2.2.	CHANGE MANAGEMENT	9
2.3.	USER MANAGEMENT	9
2.4.	RELEASE	9
2.5.	CREATE MAPPINGS	9
2.6.	MANAGE QUALITY	10
2.7.	COMMUNICATE	10
<b>3. ANALYSIS OF TOOLS</b>	<b>11</b>	
3.1.	CALLIMACHUS	11
3.2.	GINCO	17
3.3.	RE3GISTRY	21
3.4.	REGISTRY CORE	26
3.5.	SKOSMOS	31
3.6.	TEMATRES	35
3.7.	VOCBENCH	41
<b>4. CONCLUSIONS</b>	<b>46</b>	

## List of Figures

Figure 1: Editing a concept in Callimachus.....	15
Figure 2: Global scores for Callimachus.....	15
Figure 3: Editing a term in Ginco .....	20
Figure 4: Global scores for Ginco.....	20
Figure 5: Example of code list ("Access Restriction") which can be published in different formats in Re3gistry .....	24
Figure 6: Global scores for Re3gistry.....	24
Figure 7: "All properties" tab of a code list in the Environment Registry.....	29
Figure 8: Global scores for Registry Core .....	30
Figure 9: Example of a vocabulary term presented in Skosmos.....	34
Figure 10: Global scores for Skosmos.....	34

Figure 11: TemaTres search terms suggestion / autocomplete .....	38
Figure 12: Example of vocabulary term in TemaTres .....	39
Figure 13: Example of vocabulary term in TemaTres, metadata view .....	39
Figure 14: Global scores for TemaTres .....	40
Figure 15: Relationships tab in VocBench .....	44
Figure 16: Global scores for VocBench .....	45
Figure 17: Global scores .....	55
Figure 18: Scores per tool and per category .....	56

## List of Tables

Table 1: Selection criteria, motivation, and measurement for tool selection.....	6
Table 2: Creation, usage and extension requirements matrix .....	46
Table 3: Change management requirements matrix.....	49
Table 4: User management requirements matrix .....	49
Table 5: Release requirements matrix.....	50
Table 6: Mappings creation requirements matrix .....	51
Table 7: Quality management requirements matrix .....	52
Table 8: Communication requirements matrix .....	52

## GLOSSARY

Term	Definition
<b>Authority list</b>	Controlled vocabulary of descriptive terms designed to facilitate retrieval of information
<b>Code list</b>	List of values in a predefined set that can be used in metadata and which help metadata creators in selecting from a set of descriptors.
<b>Concept</b>	In this context, a word or code in a code list.
<b>Ontology</b>	Formal naming and definition of the types, properties, and relationships of the entities in a domain.
<b>Term</b>	A word or code in a code list.
<b>Thesaurus</b>	A type of controlled vocabulary seeking to dictate semantic manifestations of metadata.
<b>Vocabulary</b>	List of terms in a particular domain and pertaining definitions.

## 1. INTRODUCTION

This document is part of TASK-06 “Tools and Methodologies” of [ISA<sup>2</sup>](#) Action 2016.07 “Promoting semantic interoperability amongst EU Member States”, commonly known as [SEMIC](#). This task aims to provide updates to the tools and methodologies developed by the SEMIC action. The current report’s purpose is to provide practical guidance to less experienced organisations on the selection of the most appropriate tools for reference data management in general and code lists in particular.

This document will build on previous work carried out under the SEMIC action, which focused on the governance and management of data models as well as for tools for managing those<sup>1</sup>. Deliverable “D06.01 – Guidelines for the use of code lists”, under the same specific contract, also concerns code list management, but from a different point of view, as such the two reports complement each other. While D06.01 guides code list publishers and consumers on the subject of code list management and governance in a manner that is tool-agnostic, the present deliverable provides guidance on how to choose a suitable tool in line with the needs of each organisation.

### 1.1. Objectives & scope

The objective of this document is to provide guidance to owners and publishers of code lists on the selection of an appropriate tool for code list management depending on their needs or the requirements of their organisations. By code list management, we understand the entire lifecycle of a code list, including design, release, change management, extension, mapping, quality management, communication, etc.

As the needs of organisations vary greatly by area of activity, size, and purpose of code lists, the guidance offered by this report does not aim to provide a definitive approach to the selection of a code list management tool. It does however provide an overview of the main features such a tool should provide. While the findings of the report could apply to almost any organisation, the analysis considers public administration representatives as its main stakeholders.

When an organisation, a developer or any other person needs to work with code lists, they have to do so through various points of the code list lifecycle: design, release, change management, extension, retirement, etc. Even more, if they want to exchange information based on a code list, they might need to perform mappings with other code lists or transform the codes to an agreed-to format, etc.

### 1.2. Approach

This section defines the approach followed for the development of this report, which included:

- Determining the appropriate selection criteria for a solution to be included in the analysis;
- Using the aforementioned selection criteria to draw a list of solutions;

---

<sup>1</sup> <https://www.slideshare.net/SEMICEu/semic-governance-and-management-of-data-specifications>

- Determining which features would be the focus of the analysis after the selection of the solutions;
- Evaluating the solutions against the pre-determined list of features;
- Summarising the findings of the analysis and drawing appropriate conclusions.

### Selection criteria

This sub-section explains the criteria defined for the selection of tools.

In order to make a relevant selection of code list management tools that can be used by public administrations in the Member States of the EU and EU institutions, a number of criteria act as pre-conditions for a tool to be considered for the report. The motivation for selecting these criteria and the indicators used for measuring the compliance of a tool with the criteria are provided in Table 1.

**Table 1: Selection criteria, motivation, and measurement for tool selection**

Criterion	Motivation	Measurement
Proven use of the tool by a public administration	To make sure the tool is relevant to the target audience of the report.	Mention of a public administration among the known users of a solution.
Open Source Software	To avoid waste of public funds, reduce the risk of vendor lock-in, and support interoperability.	Licensing information of the software should mention an Open Source licence.
Maintenance activity &	To avoid inactive solutions and enjoy the benefits of an active user community.	Number of users, repository activity, publications about the tool.

The proven **use of the tool by a public administration** can provide the additional advantage of reduced costs (as a result of reusing a tool developed by/for another public administration), or of having a community of practice that includes other public administration representatives. The [Sharing and Reuse Framework](#)<sup>2</sup>, a European Commission guideline on the improvement of public IT services through sharing, reuse and collaborative development of IT solutions, encourages the exchange of information among public administrations, in addition to reusing or sharing software.

The focus on **Open Source** tools is also in line with the Sharing and Reuse Framework's specifications for how public administrations can improve their service delivery, and with one of the underlying principles of the [European Interoperability Framework](#)<sup>3</sup>. Additionally, since some organisations have either an obligation to use

<sup>2</sup> [https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/2017-10/sharing\\_and\\_reuse\\_of\\_it\\_solutions\\_framework\\_final.pdf](https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/2017-10/sharing_and_reuse_of_it_solutions_framework_final.pdf)

<sup>3</sup> [https://ec.europa.eu/isa2/sites/isa/files/eif\\_brochure\\_final.pdf](https://ec.europa.eu/isa2/sites/isa/files/eif_brochure_final.pdf)

Open Source software or follow a “comply or explain” policy<sup>4</sup> in this regard, delving into commercial solutions would have limited impact and usefulness to the target audience.

### 1.3. Structure

The remainder of this report is structured as follows:

- Section 2 delves into the functional requirements specifically examined through the analysis;
- Section 3 contains the solutions analysis based on the requirements identified in section 2;
- Section 4 summarises the findings of the work.

---

<sup>4</sup> The “comply or explain” policy refers to the policy adopted in certain Member States of the EU that requires public administrations to use Open standards in all public administration IT solutions, or explain why they did not. One example of such a policy is that of The Netherlands: [https://www.noraonline.nl/wiki/Beleid\\_open\\_standaarden](https://www.noraonline.nl/wiki/Beleid_open_standaarden)



## 2. REQUIREMENTS

This section contains an overview of the requirements and expectations public administrations might have regarding solutions for code list management. These have been collected over the years through the interactions of SEMIC with publishers of code lists in the EU institutions and Member States.

In the interest of supporting public administrations seeking guidance in the selection of an appropriate code list management tool to best serve their needs, the approach selected for this section focuses on the features of the tools and user experience aspects such as multilingual interfaces and ease of use, coupled with the different typical steps involved in the management of a code list.

Each sub-section features the relevant features from important points in the management and governance of a code list. The aim is to prepare a concrete overview of these required features to evaluate the solutions and determine their suitability in section 3.

### 2.1. Create, use and extend

The following features can be important to code list management software users:

- Installation process: the existence of a clearly identified and explained installation process for a tool (applicable to tools not available as a service);
- Web applications vs. stand-alone applications: Web applications usually provide the potential for collaborative work, and could be seen as easier to access than stand-alone applications, in addition to facilitating the sharing of existing resources;
- Multilingual user interface: as in most cases the users of the tool will not be English native speakers. In some countries, it is also required to support several national languages;
- Multilingual code list support: the tool should be able to support the management of multilingual code lists, such as the Named Authority Lists of the Publications Office. The support of standards such as SKOS-XL<sup>5</sup>, OASIS XLIFF<sup>6</sup>, or W3C's ITS<sup>7</sup>, which support multilingual labels for codes can cover this aspect;
- Search: simple or advanced search, auto-completion, metadata search function, support for SPARQL, SQL or other advanced query language, dropdown of applicable terms;
- Data import/export: export in different formats, preferably SKOS, according to compatibility with vocabulary metadata; import content from other tools in multiple formats, ideally at least SKOS/RDF and XML. API<sup>8</sup> access, for example for importing code lists from other applications, or for exporting, is also important;

---

<sup>5</sup> <https://www.w3.org/TR/skos-reference/skos-xl.html>

<sup>6</sup> <https://www.oasis-open.org/committees/xliff/>

<sup>7</sup> <https://www.w3.org/TR/its20/>

<sup>8</sup> <https://techterms.com/definition/api>

- User support: guides, tutorials, usage methodologies, wikis, FAQs, etc.;
- Usability: No limitation on the number of terms or concepts, user friendliness and capacity for personalisation, usefulness of menus and complexity of creating or modifying content, look and feel, etc.;
- Visualisation of code lists: visual editing, visible hierarchy, tree structures, etc. for presenting and browsing code lists.

## 2.2. Change management

The following features can be important to code list solution publishers and consumers:

- Change request management: ticket collection and issue tracking, etc.;
- Change synchronisation: use of Web services to synchronise code list versions by pushing code lists directly to applications;
- Notify changes: ability to notify code list consumers of changes to a code list;

## 2.3. User management

- User profile management;
- Credential-based authentication: Distinguishing users through individual usernames and passwords;
- Role-based access: different authorization for users based on their roles, e.g. only admin can delete a concept, editors can only add a concept, etc.

## 2.4. Release

The following features can be important to code list solution users:

- Documentation: availability of sufficiently thorough documentation supporting at least the current version of the software;
- Versioning: showing the version number of a concept or entire code list;
- History: providing information about the progression of a code or the entire code list, support for a release calendar;
- Status attribution: the ability to mark a concept, group of concepts or entire code list as active, superseded or retired by attributing a status label;
- Providing licensing information: the feature of displaying licensing information at concept- or list-level as a way to encourage reuse and maintain legal certainty;
- The ability to retire or delete individual terms;
- The ability to retire or delete groups of terms or entire code lists.

## 2.5. Create mappings

The following features can be important to potential code list solution users:

- Managing relations between a code in one code list and a term from another vocabulary;
- Defining and creating relations between codes in a given code list and another Web resource such as a DPpedia dataset (this feature only applies for those users interested in the Semantic Web<sup>9</sup>);
- Defining different types of relationships between terms.

## 2.6. Manage quality

The following features can be important to code list solution users:

- Quality control: metadata quality is an important aspect of facilitating access to information and search<sup>10</sup>, both at code list level and for each term in the code list. This aspect could be ensured by the presence of a validator or entry field-level validation during the creation or updating of a code. A high level of metadata quality can make information more easily findable by specifically providing certain types of metadata such as description or date attributes;
- (Dis)allow duplicate terms: ability to disable the possibility of entering duplicate terms into a code list.
- Consistency control: test for erroneous relations and duplicate concepts within a single language;

## 2.7. Communicate

The following features can be important to code list solution users:

- Forums: Web pages where code list managers (publishers, editors, consumers) and solution owners (code list management software developers) can discuss features and bugs, future developments, etc.
- Support requests: the existence of one or more ways for solution users to request support from the developer or the user community;
- RSS feed: a way for solution owners to push news about the solution to interested parties that subscribe to the feed.
- Publications: blogs, use cases, mailing lists, videos, etc.

---

<sup>9</sup> <https://www.w3.org/standards/semanticweb/>

<sup>10</sup> <http://dri.ie/sites/default/files/files/metadata-quality-control.pdf>

### 3. ANALYSIS OF TOOLS

This section analyses a list of tools that fulfil the basic selection criteria, as described in section 1.2, against the features listed in section 2. Each tool fulfils the three basic criteria of this analysis (being Open Source Software, having been used by a public administration entity, and having some degree of presence in the market and/or activity around it). The information was collected by means of desk research.

Each sub-section provides an overview of the main features of the solutions, as well as aspects related to its ownership, licensing information, and possible room for improvement or gaps.

#### 3.1. Callimachus



[Callimachus](http://callimachusproject.org/)<sup>11</sup> is a content management system which enables content publishing via web pages, giving the possibility to export metadata in RDF. This Open Source solution is released under Apache License 2.0<sup>12</sup>. It is regularly updated on GitHub<sup>13</sup>, where it has 5 contributors, 19 watchers, 79 favourites and 19 forks. In the public sector, it has been used by the US Environmental Protection Agency<sup>14</sup>.

Callimachus enables the creation of different types of content, including SKOS concepts. A concept can have the following properties:

- Label;
- Alternate label;
- Definition;
- Example;
- Scope;
- History;
- Related concept;
- Narrower concept;
- Image;
- Change notes.

Concepts can then be organized in folders, which can act as code lists within its metadata description. Users can delete a single concept or an entire code list (folder).

---

<sup>11</sup> <http://callimachusproject.org/>

<sup>12</sup> <https://www.apache.org/licenses/LICENSE-2.0>

<sup>13</sup> <https://github.com/3-Round-Stones/callimachus>

<sup>14</sup> <http://callimachusproject.org/videos/0.17/epa-success-story.xhtml?view>

<b>Create, use and extend</b>	
Installation process	✓
Web application	✓
Stand-alone application	x
Create new code list	✓
Edit existing code list	✓
Browse existing code list	✓
Multilingual interface	x
Multilingual vocabularies	✓
Search	✓
Data import formats	RDF, TTL, JSON, XML
Data export formats	RDF/XML, Turtle, JSON-LD
API access	✓ (RESTful API integration)
User support	✓
Usability	✓
Visualisation	x

<b>Manage changes</b>	
Change request management	x
Change synchronisation	x
Change notification	x

Manage users	
User management	✓
Credential-based authentication	✓
Role-based access	✓

Release	
Documentation	✓
Version number attribution	✗ (possibility to indicate it through change notes)
History	✓
Status attribution	✗
Provide licensing information	✗
Retire individual terms	✓
Retire groups of terms	✗
Retire entire code list	✓

Create mappings	
Manage relationships between concepts in a code list	✓
Define relationships between concepts in a code list and those from another Web resource	✓
Different types of relationships	✓ (only narrower and related)

## Manage quality

Quality control	x
(Dis)allow duplicate terms	x
Consistency control	x

## Communicate

Forums	✓ (as discussion)
Support requests	✓
RSS feed	x (developers can create RSS <sup>15</sup> )
Publications	✓ (videos, blog posts, etc.)

---

<sup>15</sup> [http://callimachusproject.org/docs/1.5/callimachus-for-web-developers.docbook?view#Create\\_an\\_RSS\\_feed\\_from\\_two\\_named\\_Atom\\_feeds](http://callimachusproject.org/docs/1.5/callimachus-for-web-developers.docbook?view#Create_an_RSS_feed_from_two_named_Atom_feeds)

Callimachus

localhost:8585 / concepts / c++ View Edit Discussion Describe History

## C++

<b>Label</b> <input type="text" value="c++"/>	<b>Alternate label</b> <input type="button" value="+"/>	<p>The primary label should be recognizable</p> <p>Other labels can be provide to assist concept lookup</p> <p>The definition, example, and scope can be provide to clarify usage and meaning</p> <p>Links to other related concepts may also be provided</p> <p>To add an image, click the browse icon to open a file upload dialogue</p>
<b>Definition</b> <input type="text" value="programming language"/>	<b>Example</b> <input type="text"/>	
<b>Scope</b> <input type="text"/>	<b>History</b> <input type="text"/>	
<b>Related</b> <input type="text" value="java"/>	<b>Narrower</b> <input type="text"/>	
<b>Image</b> <input type="button" value="📁"/>		

**Change note**

Figure 1: Editing a concept in Callimachus

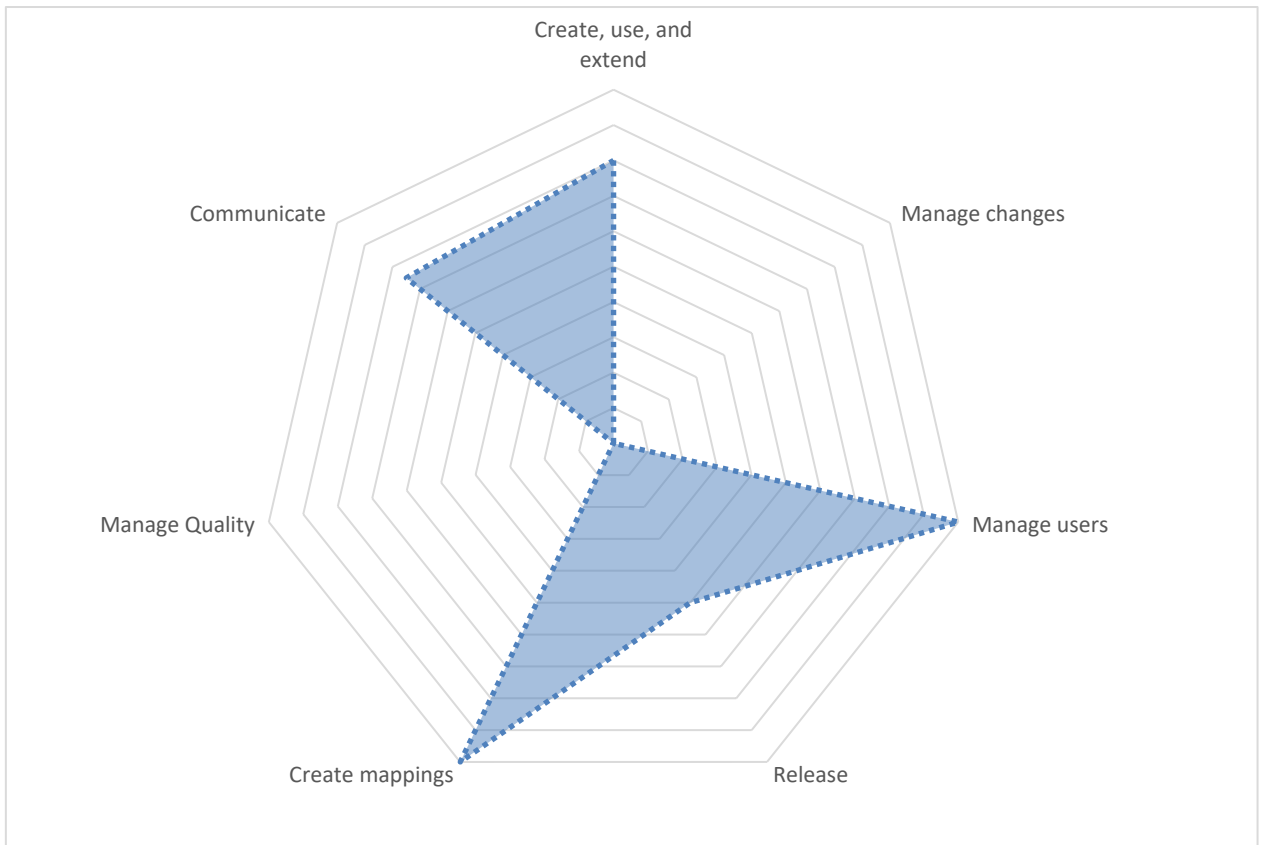


Figure 2: Global scores for Callimachus

Figure 2 presents a global view of how Callimachus scores for the different feature categories, each score as a proportion of the maximum score possible for its category.



As the figure shows, this tool is stronger for creating mappings and managing users, but lacks features related to quality management and change management.

### 3.2. Ginco



[Ginco](#) (Gestion Informatisée de Nomenclatures Collaboratives et Ouvertes<sup>16</sup>) is an application which allows creating and managing scientific and technical vocabularies (authority list, thesaurus, etc.). The application relies on the ISO 25964-1:2011 standard for the definition of the terms such as Concept and Term and it allows to export them in SKOS format.

Ginco is released as Open Source software under CeCiLL v2 license<sup>17</sup>. It is updated on GitHub<sup>18</sup>, where it has 8 contributors, 21 watchers, 31 favourites and 16 forks. In the public sector, it has been developed by the Ministry of Culture and Communication of France<sup>19</sup>, which publishes vocabularies online<sup>20</sup>.

Create, use and extend	
Installation process	✓ (relatively complex, but thoroughly explained by documentation)
Web application	✓
Stand-alone application	✓
Create new code list	✓
Edit existing code list	✓
Browse existing code list	✓ (code lists are stored as file)
Multilingual interface	✓ (yes, default is French)
Multilingual vocabularies	✓
Search	✓
Data import formats	SKOS, GINCO XML
Data export formats	RDF, TXT, GINCO XML

<sup>16</sup> "Computerized Management of Collaborative and Open Nomenclatures" in English

<sup>17</sup> [http://www.cecill.info/licences/Licence\\_CeCiLL\\_V2-en.html](http://www.cecill.info/licences/Licence_CeCiLL_V2-en.html)

<sup>18</sup> <https://github.com/culturecommunication/ginco>

<sup>19</sup> <https://github.com/culturecommunication/ginco#what-is-ginco->

<sup>20</sup> <http://data.culture.fr/thesaurus/>

### Create, use and extend

<b>API access</b>	✓ (SOAP API access)
<b>User support</b>	✓
<b>Usability</b>	✓ (customisable)
<b>Visualisation</b>	✗

### Manage changes

<b>Change request management</b>	✗
<b>Change synchronisation</b>	✗
<b>Change notification</b>	✗

### Manage users

<b>User management</b>	✓
<b>Credential-based authentication</b>	✓
<b>Role-based access</b>	✓ (administrator, expert, etc.)

### Release

<b>Documentation</b>	✓
<b>Version number attribution</b>	✓ (for code list)
<b>History</b>	✓
<b>Status attribution</b>	✓
<b>Provide licensing information</b>	✗
<b>Retire individual terms</b>	✓

## Release

Retire groups of terms	✓
Retire entire code list	✓

## Create mappings

Manage relationship between concepts in a code list	✓
Define relationships between concepts in a code list and those from another Web resource	✓
Different types of relationships	✓

## Manage quality

Quality control	✓
(Dis)allow duplicate terms	✓
Consistency control	✓

## Communicate

Forums	✓ (through suggestions)
Support requests	✓
RSS feed	✗
Publications	✗

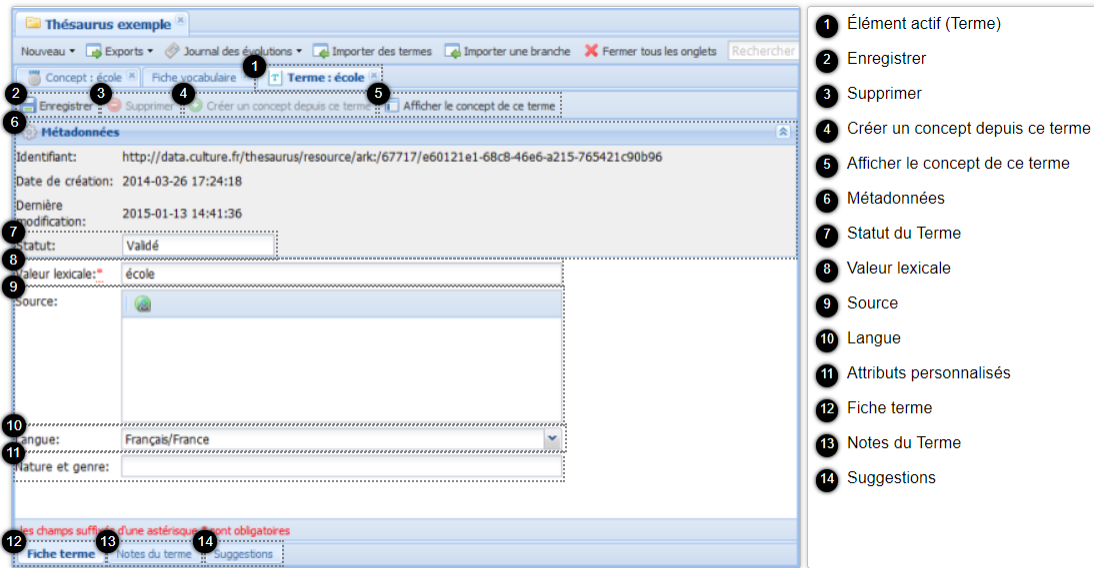


Figure 3: Editing a term in Gincó

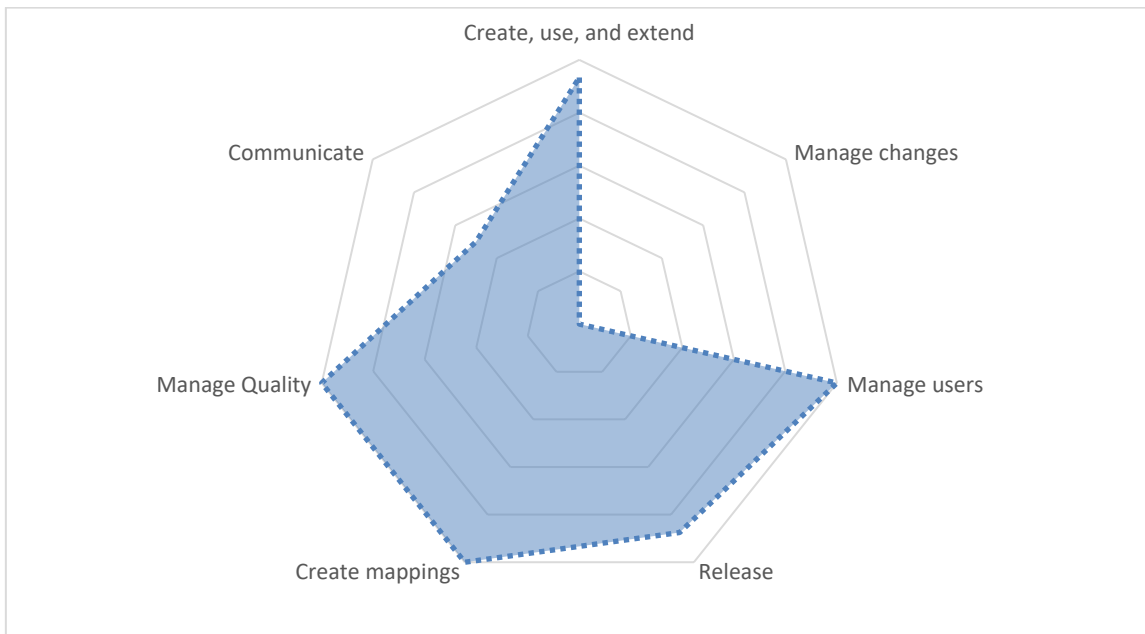


Figure 4: Global scores for Gincó

Figure 4 presents a global view of how Gincó scores for the different feature categories, each score as a proportion of the maximum score possible for its category. The scores indicate this tool is suitable for most aspects, with the exception of change management and, to a lesser degree, communication aspects.

### 3.3. Re3gistry

## Re<sup>3</sup>gistry

The [Re3gistry](https://ec.europa.eu/isa2/solutions/re3gistry_en)<sup>21</sup> is an Open Source software released under European Union Public Licence - [EURL](https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl1.1.-licence-en_0.pdf) v.1.1<sup>22</sup> and created to serve as a tool for managing and sharing reference codes in the context of [INSPIRE](http://inspire.ec.europa.eu/)<sup>23</sup>. It helps ensure that concepts are correctly referenced in INSPIRE, but also other sectors. The solution has been used extensively by public administrations, including the European Commission and governmental organisations from Austria, Finland, France, Italy, Slovakia, etc.<sup>24</sup>

Create, use and extend	
Installation process	✓ (relatively complex, but thoroughly explained by documentation)
Web application	✓
Stand-alone application	✓
Create new code list	✓
Edit existing code list	✓
Browse existing code list	✓
Multilingual interface	✓
Multilingual vocabularies	✓
Search	✓
Data import formats	CSV
Data export formats	HTML, XML, JSON, RDF, Atom, CSV
API access	✓ (RESTful API access)

<sup>21</sup> Re3gistry: [https://ec.europa.eu/isa2/solutions/re3gistry\\_en](https://ec.europa.eu/isa2/solutions/re3gistry_en)

<sup>22</sup> EURL: [https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl1.1.-licence-en\\_0.pdf](https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl1.1.-licence-en_0.pdf)

<sup>23</sup> INSPIRE Knowledge Base: <http://inspire.ec.europa.eu/>

<sup>24</sup> [https://ec.europa.eu/isa2/solutions/re3gistry\\_en](https://ec.europa.eu/isa2/solutions/re3gistry_en)

Create, use and extend	
User support	✓
Usability	✓ (customisable)
Visualisation	x

Manage changes	
Change request management	x
Change synchronisation	x
Change notification	x

Manage users	
User management	✓
Credential-based authentication	✓
Role-based access	✓ (admin, user, etc.)

Release	
Documentation	✓
Version number attribution	✓
History	✓
Status attribution	✓
Provide licensing information	✓
Retire individual terms	✓
Retire groups of terms	✓

Release	
Retire entire code list	✓

Create mappings	
Manage relationship between concepts in a code list	✓
Define relationships between concepts in a code list and those from another Web resource	✓
Different types of relationships	✓

Manage quality	
Quality control	✓
(Dis)allow duplicate terms	✗
Consistency control	✓

Communicate	
Forums	✓
Support requests	✓ (bug reporting, support requests)
RSS feed	✓ (Related to all INSPIRE initiatives)
Publications	✓ (news items, videos, demos)

Authentication occurs either through Apache SHIRO (static user authentication using a simple configuration file containing the list of users) or through the EU Login system.

When importing data, the files need to be compressed in .zip format.



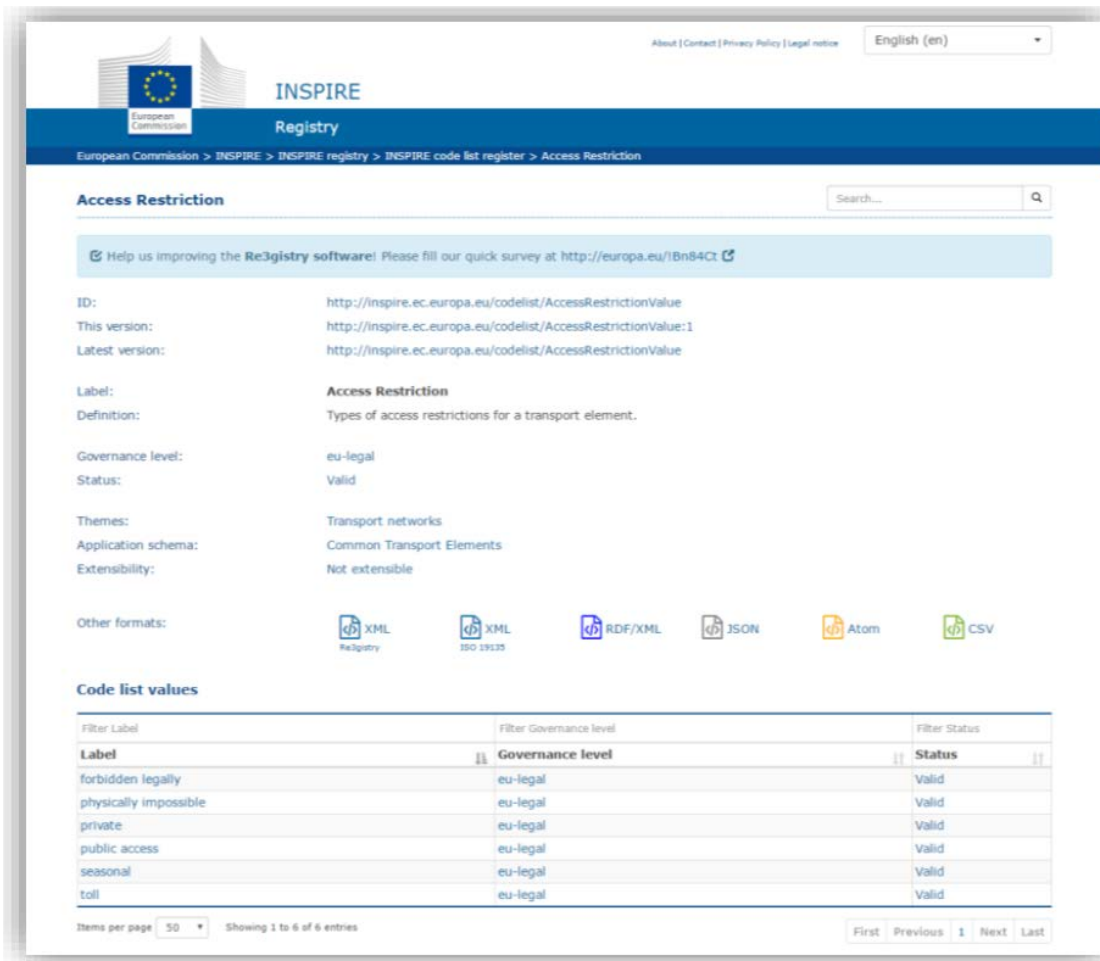


Figure 5: Example of code list (“Access Restriction”) which can be published in different formats in Re3gistry

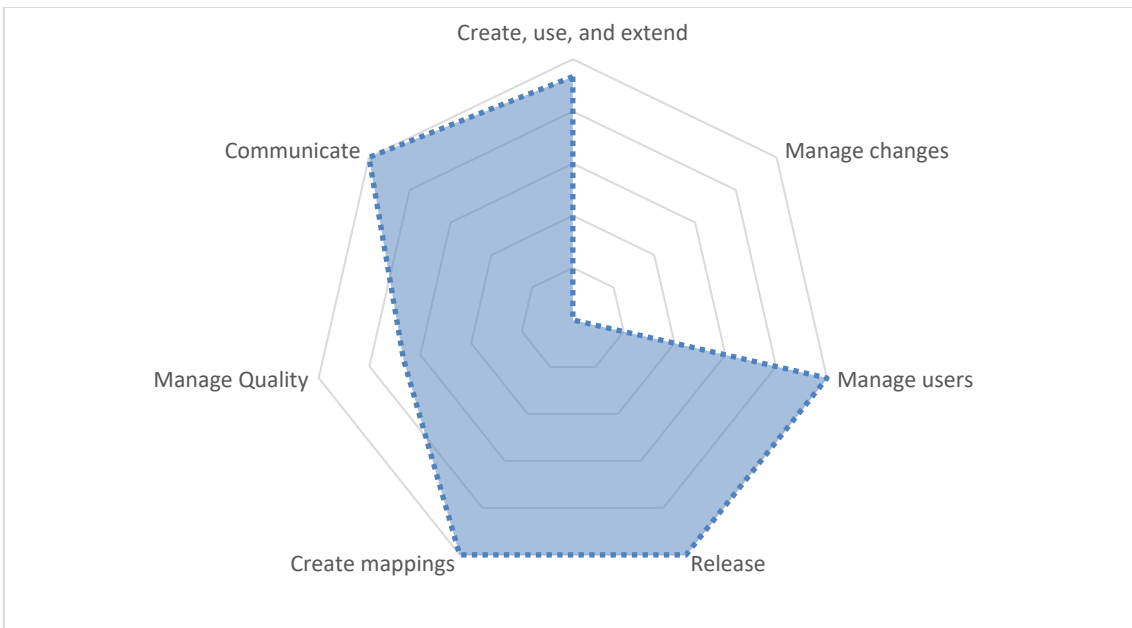


Figure 6: Global scores for Re3gistry

Figure 6 presents a global view of how Re3gistry scores for the different feature categories, each score as a proportion of the maximum score possible for its category. This tool’s strongest aspects are user management, mapping creation, release

management, and communication. It also has a relatively high quality management score, but is lacking in the change management area.

### 3.4. Registry Core

[Registry Core](#) is an Open Source solution the UK government contracted. It serves as the basis for the Core Reference Data Management Platform provided by Epimorphics, which supports and documents Open Source standards itself. The GitHub<sup>25</sup> repository for this solution shows constant activity, with 10 watchers, 9 favourites and 5 forks. The solution is released under Apache license and it can be installed following a relatively complex process<sup>26</sup>, or implementations can be contracted, for instance through the UK government's Digital Marketplace<sup>27</sup>.

The solution has different implementations by public administrations. For instance, the [Environment Registry](#)<sup>28</sup> is a Web application available online, featuring user management. Users that are simply interested in accessing the existing code lists do not need any credentials, but those wishing to add or change entries do need a username and password.

Code lists existing within the solution display as a simple Web page containing important details such as a code list's URI, a description of it, and a table featuring the name, notation, description, types and status of the concepts in the code list. Remarkably, change management is covered by the possibility to label the concepts in a code list as "experimental" or "stable". These labels can also apply to entire code lists or registers. The "All properties" tab for a register, for instance, brings up a table containing information like description, label, owner, but also versioning information and date when it has last been modified, as shown in Figure 7.

Create, use and extend	
Installation process	✗ (not required)
Web application	✓
Stand-alone application	✗
Create new code list	✓
Edit existing code list	✓
Browse existing code list	✓
Multilingual interface	✗
Multilingual vocabularies	✗
Search	✓

<sup>25</sup> <https://github.com/UKGovLD/registry-core>

<sup>26</sup> <https://github.com/UKGovLD/registry-core/wiki/Installation>

<sup>27</sup> <https://www.digitalmarketplace.service.gov.uk/g-cloud/services/991341327991934>

<sup>28</sup> [https://environment.data.gov.uk/registry/\\_def](https://environment.data.gov.uk/registry/_def)

Create, use and extend	
Data import formats	RDF syntaxes Turtle and RDF/XML, JSON-LD, etc.
Data export formats	RDF, RDF/XML, JSON-LD, CSV, etc.
API access	✓
User support	✓ (e-mail, online ticketing)
Usability	Somewhat complex to install without help of contractor.
Visualisation	✓

Manage changes	
Change request management	✗
Change synchronisation	✗
Change notification	✗

Manage users	
User management	✓
Credential-based authentication	✓
Role-based access	✓

Release	
Documentation	✓
Version number attribution	✓
History	✓
Status attribution	✓

Release	
Provide licensing information	✓
Retire individual terms	✓
Retire groups of terms	✓
Retire entire code list	✓

Create mappings	
Manage relationship between concepts in a code list	✓
Define relationships between concepts in a code list and those from another Web resource	✓
Different types of relationships	✓

Manage quality	
Quality control	✓
(Dis)allow duplicate terms	x
Consistency control	x

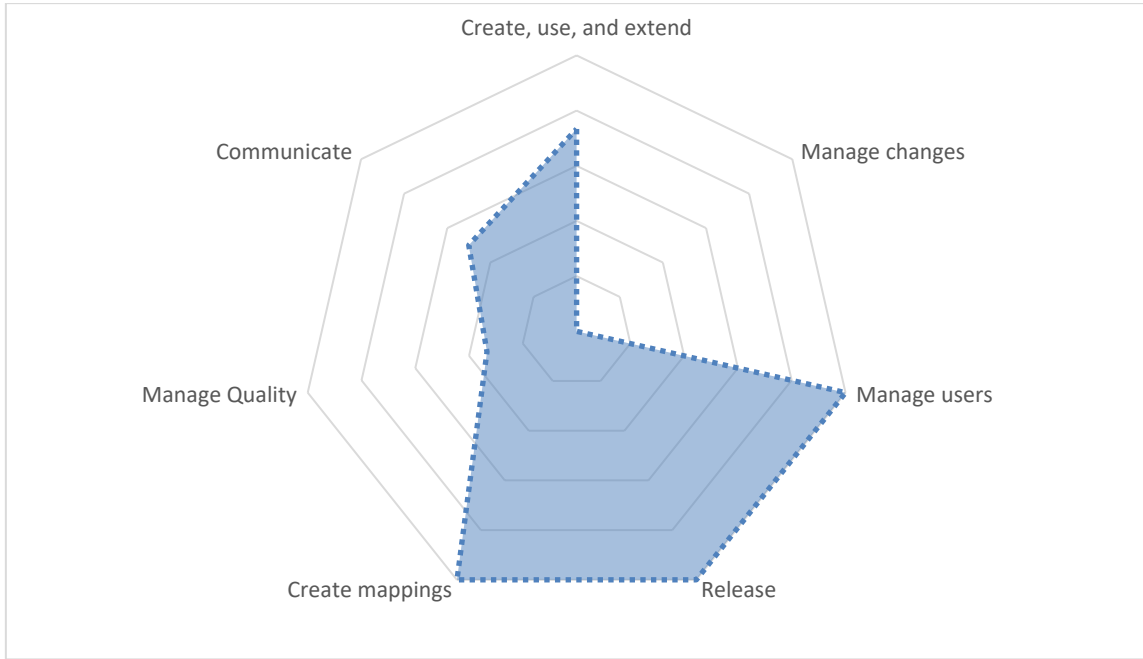
Communicate	
Forums	✓
Support requests	✓
RSS feed	x
Publications	x

All properties of the register					
<b>category</b>	<a href="#">Water</a>				
<b>description</b>	The river basin is the geographical area from which all surface run-off flows through a sequence of streams, rivers and, possibly, lakes into the sea at a single river mouth or estuary.   The river basin district is the main unit for management of river basins under the Water Framework Directive. River basin districts were designated by Member States. A river basin district can consist of one or more river basins.				
<b>entity type</b>	<a href="#">regions and habitats</a>				
<b>label</b>	River Basin Districts				
<b>license</b>	<a href="http://www.nationalarchives.gov.uk/doc/open-government-licence/version/2/">http://www.nationalarchives.gov.uk/doc/open-government-licence/version/2/</a>				
<b>member</b>	<a href="#">South West</a>   <a href="#">Solway Tweed</a>   <a href="#">Severn</a>   <a href="#">Northumbria</a>   <a href="#">North West</a>   <a href="#">Thames</a>   <a href="#">South East</a>   <a href="#">Scotland</a>   <a href="#">Western Wales</a>   <a href="#">Humber</a>   <a href="#">Dee</a>   <a href="#">Anglian</a>				
<b>membership predicate</b>	<a href="#">member</a>				
<b>modified</b>	5 Mar 2014 17:24:26.359				
<b>notation</b>	RiverBasinDistrict				
<b>owner</b>	<a href="#">environment agency</a>				
<b>publisher</b>	<a href="#">environment agency</a>				
<b>rights</b>	<table border="1"> <tr> <td><b>attribution</b></td> <td>Contains public sector information</td> </tr> <tr> <td><b>text</b></td> <td>licensed under the Open Government Licence v2.0.</td> </tr> </table>	<b>attribution</b>	Contains public sector information	<b>text</b>	licensed under the Open Government Licence v2.0.
<b>attribution</b>	Contains public sector information				
<b>text</b>	licensed under the Open Government Licence v2.0.				
<b>type</b>	<a href="#">Register</a>   <a href="#">Container</a>				
<b>version info</b>	1				

Figure 7: "All properties" tab of a code list in the Environment Registry

Individual concepts in a code list receive their own Web page and assigned URI. Each concept page contains information about the concept, such as label, notation, same as, and type. The "same as" field suggests the possibility to map the relationships between concepts in one code list and those in other vocabularies. Concept pages also feature a tab called "History", which contains information about the versions through which a term has passed, and its validity (both start date and end date if applicable).

Other features include the possibility to use an API or perform SPARQL queries and checking whether a URI is registered, both accessible without requiring users to register or log in to the system.



**Figure 8: Global scores for Registry Core**

Figure 8 presents a global view of how Registry Core scores for the different feature categories, each score as a proportion of the maximum score possible for its category. As the figure indicates, this tool is most useful for user management, release management, and mapping creation. It is most lacking in change management and quality management.

### 3.5. Skosmos

[Skosmos](http://skosmos.org/)<sup>29</sup> is an Open Source software for publishing controlled vocabularies, distributed under the MIT licence<sup>30</sup>. The National Library of Finland is the institution behind it. It has an active community on GitHub<sup>31</sup>, where it is frequently updated and followed by 20 watchers, favourited 92 times and forked 29 times.

Interested developers can install Skosmos and run their own instance, but the installation involves a relatively complex procedure, assuming they have a Linux machine with Apache and PHP installed and enabled. It also assumes the developer has a SPARQL endpoint with SKOS vocabulary data. However, typical users do not need to install the developer version, and can access Skosmos through a browser.

Create, use and extend	
Installation process	✓ (not required for browsing)
Web application	✓
Stand-alone application	✓
Create new code list	✓
Edit existing code list	✓
Browse existing code list	✓
Multilingual interface	✓
Multilingual vocabularies	✓
Search	✓
Data import formats	SKOS
Data export formats	RDF/XML, TURTLE, JSON-LD
API access	✓
User support	✓
Usability	✓

<sup>29</sup> <http://skosmos.org/>

<sup>30</sup> <http://skosmos.org/publishing-skos-vocabularies-with-skosmos.pdf>

<sup>31</sup> <https://github.com/NatLibFi/Skosmos>



**Create, use and extend**

<b>Visualisation</b>	✓
----------------------	---

**Manage changes**

<b>Change request management</b>	x
<b>Change synchronisation</b>	x
<b>Change notification</b>	x

**Manage users**

<b>User management</b>	✓
<b>Credential-based authentication</b>	x
<b>Role-based access</b>	x

**Release**

<b>Documentation</b>	✓
<b>Version number attribution</b>	✓
<b>History</b>	✓
<b>Status attribution</b>	✓
<b>Provide licensing information</b>	x
<b>Retire individual terms</b>	✓
<b>Retire groups of terms</b>	✓
<b>Retire entire code list</b>	✓

Create mappings	
Manage relationship between concepts in a code list	✓
Define relationships between concepts in a code list and those from another Web resource	✓
Different types of relationships	✓

Manage quality	
Quality control	x
(Dis)allow duplicate terms	x
Consistency control	x

Communicate	
Forums	✓
Support requests	✓
RSS feed	x
Publications	✓ (demo, scientific articles)

Other features:

- Structured concept display;
- Visualised concept hierarchy;
- Multilingual user interface: English, French, Italian, Finnish, Swedish, German, Spanish, Polish, Norwegian, Japanese, etc.;
- Linked Data access to underlying vocabulary data.

The screenshot shows the Skosmos interface for the 'STW Thesaurus for Economics'. The main content area displays the concept 'Indigenous people' with the following details:

- PREFERRED TERM:** Indigenous people
- CONCEPT TYPE:** Descriptor
- BROADER CONCEPT:** Ethnic group, N.03 Demography and Population
- NARROWER CONCEPTS:** Aboriginal Australians, American Indians, Inuit, Maori (People), Sami people, Samoyedes
- ALTERNATIVE LABEL:** Aboriginal people
- SCOPE NOTE:** Ethnic groups, who are native in the country where they live.
- IN OTHER LANGUAGES:** Indigene Völker (German), Eingeborener, Naturvölker, Ureinwohner, Eingeborene
- URI:** http://zbw.eu/stw/descriptor/19643-5
- Download this concept:** RDF/XML TURTLE
- EXACT MATCH:** indigenous peoples (en) (AGROVOC)

Figure 9: Example of a vocabulary term presented in Skosmos

Figure 9 shows an example of what a concept might look like in a code list in the Skosmos solution. As shown, the user can choose between the hierarchical and the alphabetical view for the full code list, and a significant amount of information is presented regarding related terms, related terms from other vocabularies, etc.

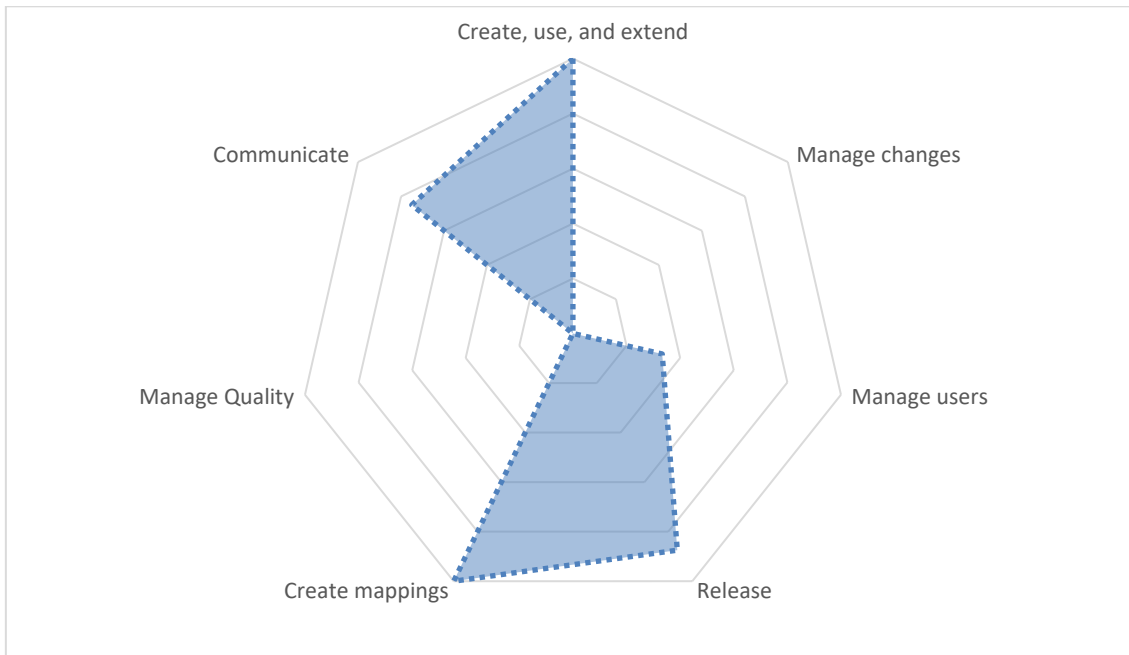


Figure 10: Global scores for Skosmos

Figure 10 presents a global view of how Skosmos scores for the different feature categories, each score as a proportion of the maximum score possible for its category.

The figure indicates Skosmos is particularly suitable for creating, using, and extending code lists, but also for creating mappings. It is however not a suitable tool for user management, quality management, nor change management.

### 3.6. TemaTres



[TemaTres](#)<sup>32</sup> is an Open Source application to manage and exploit controlled vocabularies, thesauri, taxonomies, and other types of formal representations of knowledge. TemaTres was originally developed by the Library and the University of Buenos Aires in Argentina and released under the GNU General Public License 2.0<sup>33</sup>. The solution has an active GitHub [page](#)<sup>34</sup> with 6 contributors, 11 watchers, 41 favourites and 29 forks.

TemaTres supports distributed management models, ensuring consistency and integrity of data and relationships between terms. Its features cover data traceability and quality control in the context of a controlled vocabulary.

There are more than 400 known cases of TemaTres being used to manage, publish, share, reuse vocabularies, thesauri, glossaries, code lists, etc. Many of these use cases come from public administrations such as: the Australian Government's Interactive Functions Thesaurus (AGIFT), the Argentinian Government's controlled [vocabulary](#) on the policy of the Labour Ministry, The National Agricultural Thesaurus of [Brazil](#), The Monument Thesaurus [Wales](#), and many others.

TemaTres is ideally suited for those publishers managing code lists on the web and those that need a long-term perspective and a dedicated management system. TemaTres is described as having a simple, functional user interface and sophisticated search capabilities.

The tool features a multilingual interface, specifically in English, Spanish, Italian, German, French, Portuguese, Dutch, Catalan, Chinese.

Regarding its usability, it is relatively simple and straightforward, but it does seem aimed at more experienced users, who can understand its alerts<sup>35</sup>.

#### Create, use and extend

<b>Installation process</b>	✓
-----------------------------	---

<sup>32</sup> <http://www.vocabularyserver.com/index.html>

<sup>33</sup> <http://aims.fao.org/activity/blog/tematres-22-open-source-web-tool-manage-formal-representations-knowledge>

<sup>34</sup> <https://github.com/tematres/TemaTres-Vocabulary-Server>

<sup>35</sup> <http://accidental-taxonomist.blogspot.be/2016/02/free-taxonomy-management-software.html>

<b>Create, use and extend</b>	
<b>Web application</b>	✓
<b>Stand-alone application</b>	✗
<b>Create new code list</b>	✓
<b>Edit existing code list</b>	✓
<b>Browse existing code list</b>	✓
<b>Multilingual interface</b>	✓
<b>Multilingual vocabularies</b>	✓
<b>Search</b>	✓
<b>Data import formats</b>	TXT, CSV, SKOS
<b>Data export formats</b>	XML, TXT, SKOS-Core, JSON, JSON-LD, PDF
<b>API access</b>	✓
<b>User support</b>	✓
<b>Usability</b>	✓
<b>Visualisation</b>	✓

<b>Manage changes</b>	
<b>Change request management</b>	✓
<b>Change synchronisation</b>	✓
<b>Change notification</b>	✓

Manage users	
User management	✓
Credential-based authentication	✓
Role-based access	✓

Release	
Documentation	✓
Version number attribution	x
History	✓
Status attribution	✓
Provide licensing information	x
Retire individual terms	✓
Retire groups of terms	✓
Retire entire code list	✓

Create mappings	
Manage relationship between concepts in a code list	✓
Define relationships between concepts in a code list and those from another Web Resource	✓
Different types of relationships	✓

Manage quality	
Quality control	✓
(Dis)allow duplicate terms	✓
Consistency control	✓

Communicate	
Forums	✓
Support requests	✓
RSS feed	✗
Publications	✓ (demo, examples, blog posts)

Additional features:

- SPARQL endpoint (SPARQL Protocol and RDF Query Language);
- Meta-terms; define facets, collections or arrays of terms;
- Powerful web services to expose vocabularies;
- Web services allowing for integration with other platforms;
- Search terms suggestion, as shown by Figure 11.

## TemaTres



Figure 11: TemaTres search terms suggestion / autocomplete

Figure 12 and Figure 13 show an example of a term in TemaTres, illustrating features such as the relationships between a term and others, the inclusion of versioning and history information, etc.

# marital status

Home / marital status

Term   Notes **1**   Metadata

**marital status**

More specific terms

- [NT1 ↓ cohabitation](#)
- [NT1 ↓ marriage migration](#)

Related terms

- [RT ⇄ child marriage](#)
- [RT ⇄ early marriage](#)
- [RT ⇄ family care](#)
- [RT ⇄ family responsibilities](#)
- [RT ⇄ forced marriage](#)
- [RT ⇄ marital rape](#)

Figure 12: Example of vocabulary term in TemaTres

# marital status

Home / marital status

Term   Notes **1**   Metadata

Date of creation	19-Feb-2018
Modified	01-Jan-1970
<b>Accepted term</b>	<b>19-Feb-2018</b>
Descendant terms	2
More specific terms	2
Alternative terms	0
Related terms	6
Notes	1

Metadata

[BS8723-5](#)
[DC](#)
[MADS](#)
[SKOS-Core](#)
[VDEX](#)
[XTM](#)
[Zthes](#)
[JSON](#)
[JSON-LD](#)


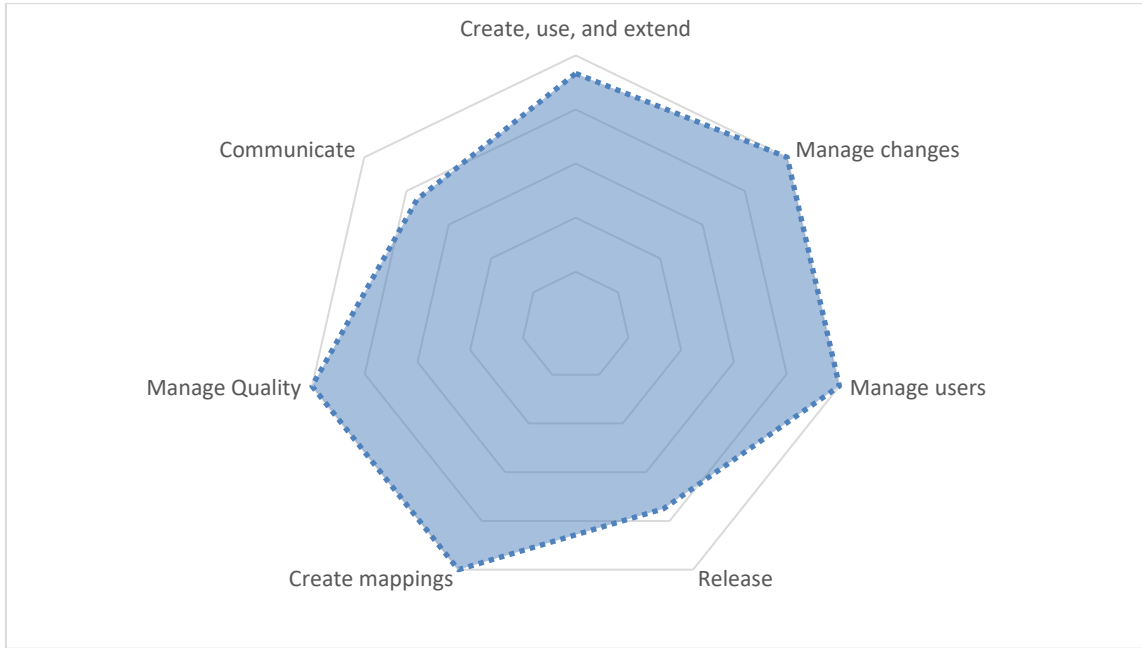
Search 

Figure 13: Example of vocabulary term in TemaTres, metadata view





**Figure 14: Global scores for TemaTres**

Figure 14 presents a global view of how TemaTres scores for the different feature categories, each score as a proportion of the maximum score possible for its category. The figure indicates almost perfect scores in most categories, albeit to a lesser degree for release management and communication.

### 3.7. VocBench



[VocBench](#)<sup>36</sup> is an editing and workflow tool that manages thesauri, authority lists and glossaries using SKOS. There are two active versions: VocBench 2 and VocBench 3. VocBench 3 enriched VocBench 2 with significant additional features, including versioning<sup>37</sup>. VocBench is being used by various public sector organisations, including the European Commission, the Scottish Government and the Italian Government<sup>38</sup>.

Create, use and extend	
Installation process	✓ (detailed instructions)
Web application	✓
Stand-alone application	✓
Create new code list	✓
Edit existing code list	✓
Browse existing code list	✓
Multilingual interface	✓
Multilingual vocabularies	✓
Search	✓
Data import formats	RDF (SKOS, SKOS-XL), OWL
Data export formats	RDF (SKOS, SKOS-XL), OWL
API access	✗
User support	✓

<sup>36</sup> <http://vocbench.uniroma2.it/>

<sup>37</sup> Towards VocBench 3: Pushing Collaborative Development of Thesauri and Ontologies further beyond: <http://ceur-ws.org/Vol-1937/paper4.pdf>

<sup>38</sup> <http://vocbench.uniroma2.it/aboutus/>

### Create, use and extend

Usability	✓
Visualisation	✓

### Manage changes

Change request management	x
Change synchronisation	x
Change notification	x

### Manage users

User management	✓ (Administration, group management)
Credential-based authentication	✓
Role-based access	✓

### Release

Documentation	✓
Version number attribution	✓
History	✓
Status attribution	✓
Provide licensing information	x
Retire individual terms	✓
Retire groups of terms	x

Release	
Retire entire code list	✓

Create mappings	
Manage relationship between concepts in a code list	✓
Define relationships between concepts in a code list and those from another Web resource	✓
Different types of relationships	✓

Manage quality	
Quality control	✓
(Dis)allow duplicate terms	x
Consistency control	✓

Communicate	
Forums	✓
Support requests	✓
RSS feed	x
Publications	✓ (scientific articles, guides)

Additional features:

- SPARQL endpoint Collaborative;
- Validation and publication workflow;
- Maintenance, validation, quality assurance;
- Extensive support infrastructure;

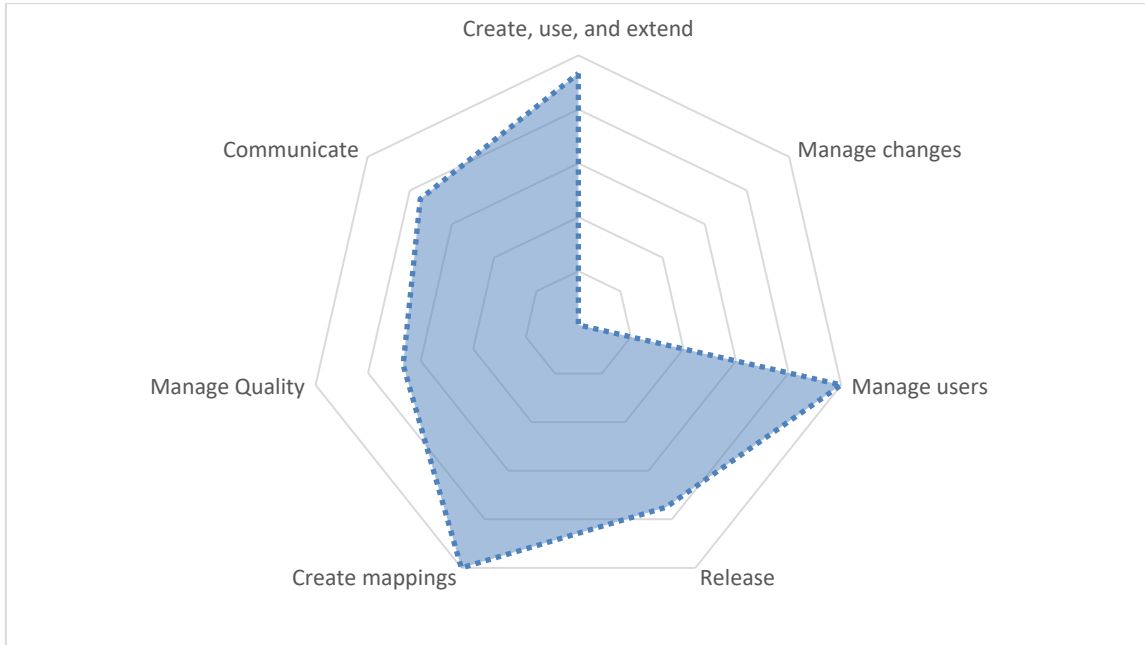
- Detailed instructions for download, installation, configuration;
- User manual details all the operations that users can perform on the application;
- System administrator manual deals with various installation options and all the advanced features related to system customisation/reconfiguration that need to be performed under the hood;
- Developer manual provides behind-the-scenes information for VocBench developers, third party developers and anybody interested in how VocBench works;
- SKOS development pages providing insights on less-known aspects of SKOS development, best practices and typical modelling issues and how they are dealt with in VocBench.



Figure 15: Relationships tab in VocBench

Figure 15 shows the relationship section corresponding to a term as seen in VocBench<sup>39</sup>. The figure also illustrates some other VocBench features, such as History.

<sup>39</sup> <https://www.slideshare.net/Keizer/the-vocbench-project>



**Figure 16: Global scores for VocBench**

Figure 16 presents a global view of how VocBench scores for the different feature categories, each score as a proportion of the maximum score possible for its category. VocBench scores relatively high in all categories, except for change management.

## 4. CONCLUSIONS

This section summarises the analysis performed in the previous sections, giving an overview of the tools and their features as seen from the point of view of the requirements a user might have at different points in the lifecycle of a code list.

Table 2 takes into account general requirements, including those regarding the actual usage of a code list and the possibility of extending it. Here, we see some solutions have some sort of required installation process, with a few of them having a relatively complex one for which organisations may require external help. However, many solutions differentiate between a developers' version of the solution, and a more basic one, for end-users simply interested in using the solution to manage code lists, and which is often available after a basic installation process and/or directly in the browser. Several solutions enable basic browsing of code lists without the need to actually install any software. Most solutions offer API access, opening the possibility to integrate them into other systems. A strong majority of solutions allow for multilingual vocabularies, and most of them also feature multilingual interfaces, enabling users to perform their work in a language they can comfortably use.

**Table 2: Creation, usage and extension requirements matrix**

	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
<b>Installation process</b>	✓	✓ (relatively complex, but thoroughly explained by documentation)	✓ (relatively complex, but thoroughly explained by documentation)	✗ (not required)	✓ (not required for browsing)	✓	✓ (detailed instructions)
<b>Web application</b>	✓	✓	✓	✓	✓	✓	✓
<b>Stand-alone application</b>	✗	✓	✓	✗	✓	✓	✓

Updated Metadata Governance Tools Report

	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
<b>Create new code list</b>	✓	✓	✓	✓	✓	✓	✓
<b>Edit existing code list</b>	✓	✓	✓	✓	✓	✓	✓
<b>Browse existing code list</b>	✓	✓ (code list are stored as file)	✓	✓	✓	✓	✓
<b>Multilingual interface</b>	✗	✓ (yes, default is French)	✓	✗	✓	✓	✓
<b>Multilingual vocabularies</b>	✓	✓	✓	✗	✓	✓	✓
<b>Search</b>	✓	✓	✓	✓	✓	✓	✓
<b>Data import formats</b>	RDF, TTL, JSON, XML	SKOS, GINCO XML	CSV	RDF syntaxes Turtle and RDF/XML, JSON-LD, etc.	SKOS	TXT, CSV, SKOS	RDF (SKOS, SKOS-XL), OWL



	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
<b>Data export formats</b>	RDF/XML, Turtle, JSON-LD	RDF, TXT, GINCO XML	HTML, XML, JSON, RDF, Atom, CSV	RDF, RDF/XML, JSON-LD, CSV, etc.	RDF/XML, TURTLE, JSON-LD	XML, TXT, SKOS-Core, JSON, JSON-LD, PDF	RDF/XML, TURTLE, JSON-LD
<b>API access</b>	✓ (RESTful API integration)	✓ (SOAP API access)	✓ (RESTful API access)	✓	✓	✓	✗
<b>User support</b>	✓	✓	✓	✓ (e-mail, online ticketing)	✓	✓	✓
<b>Usability</b>	✓	✓ (customisable)	✓ (customisable)	✓ (Somewhat complex to install and configure)	✓ (customisable)	✓	✓
<b>Visualisation</b>	✗	✗	✗	✓	✗	✓	✓

Table 3 presents the situation regarding change management requirements users might have. It is one of the areas where the solutions tend to be lacking, with TemaTres being the only one that offers all three foreseen feature requirements. That said, change request management and

change notification are relatively easy to manage by code list publishers by using external solutions. Change synchronisation is likely to be lacking due to a relative technical complexity, in addition to code list consumers being less likely to accept automatic code list updates.

Table 3: Change management requirements matrix

	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
Change request management	x	x	x	x	x	✓	x
Change synchronisation	x	x	x	x	x	✓	x
Change notification	x	x	x	x	x	✓	x

Table 4: User management requirements matrix

	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
User management	✓	✓	✓	✓	✓	✓	✓ (Administration, group management)
Role-based access	✓	✓	✓	✓	x	✓	✓
Credential-based authentication	✓	✓ (administrator, expert, etc.)	✓ (admin, user, etc.)	✓	x	✓	✓

Table 5 presents features related to release management. Particularly important here would be history, versioning, and status attribution. All analysed solutions offer some form of history records for each concept. Versioning is also provided by most solutions, along with status attribution. The possibility to feature licensing information for concepts is less common, potentially seen as less important because code list owners can simply provide licensing information covering the whole code list, not just a concept or group of concepts. Table 5 also provides information about the retirement features of the analysed solutions. As expected from such solutions, all of them feature the possibility to retire individual terms in a code list or the entire code list. More interesting is the aspect of deleting or retiring groups of terms, which is not possible in some solutions, like Callimachus or VocBench.

Table 5: Release requirements matrix

	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
<b>Documentation</b>	✓	✓	✓	✓	✓	✓	✓
<b>Version number attribution</b>	x (possibility to indicate it through change notes)	✓ (for code list)	✓	✓	✓	x	✓
<b>History</b>	✓	✓	✓	✓	✓	✓	✓
<b>Status attribution</b>	x	✓	✓	✓	✓	✓	✓
<b>Provide licensing information</b>	x	x	✓	✓	x	x	x
<b>Retire individual terms</b>	✓	✓	✓	✓	✓	✓	✓

	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
<b>Retire groups of terms</b>	x	✓	✓	✓	✓	✓	x
<b>Retire entire code list</b>	✓	✓	✓	✓	✓	✓	✓

Table 6 contains information regarding the relationships between concepts, or between code lists and other vocabularies. All solutions feature some form of relationship management between the terms in a code list, even if it relates simply to designating narrower or broader terms or a basic “related” status.

**Table 6: Mappings creation requirements matrix**

	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
<b>Manage relationship between concepts in a code list</b>	✓	✓	✓	✓	✓	✓	✓
<b>Define relationships between concepts in a code list and those from another Web resource</b>	✓	✓	✓	✓	✓	✓	✓
<b>Different types of relationships</b>	✓ (only narrower and related)	✓	✓	✓	✓	✓	✓

Table 7 contains information regarding quality management requirements. Issues like quality control are relatively well covered by most solutions, but features like the possibility to (dis)allow duplicate terms are rarer. Consistency control is a welcome feature for larger code lists, but only 3 out of 7 solutions mention it. Overall, quality management features are some of the less developed aspects of code list management solutions.

**Table 7: Quality management requirements matrix**

	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
<b>Quality control</b>	x	✓	✓	✓	x	✓	✓
<b>(Dis)allow duplicate terms</b>	x	✓	x	x	x	✓	x
<b>Consistency control</b>	x	✓	✓	x	x	✓	✓

Table 8 contains an overview of the communication features of the analysed solutions. The existence of forums where community members can debate and/or support each other, report issues or discuss difficulties encountered while using the solutions exist for almost all solutions. Alternatively, most solutions have some manner of accepting support requests, even if it is simply by reporting bugs or issues on GitHub or SourceForge. RSS feeds as a way to communicate with users are increasingly rare and unused, but most solutions mention a mailing list of some form.

**Table 8: Communication requirements matrix**

	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
<b>Forums</b>	✓ (as discussion)	✓ (through suggestions)	✓	✓	✓	✓	✓

	Callimachus	Ginco	Re3gistry	Registry Core	Skosmos	TemaTres	VocBench
<b>Support requests</b>	✓	✓	✓ (bug reporting, support requests)	✓	✓	✓	✓
<b>RSS feed</b>	✗ (developers can create RSS)	✗	✓ (Related to all INSPIRE initiatives)	✗	✗	✗	✗
<b>Publications</b>	✓ (videos, blog posts, etc.)	✗	✓ (news items, videos, demos)	✗	✓ (demo, scientific articles)	✓ (demo, examples, blog posts)	✓ (scientific articles, guides)

To sum up, organisations interested in creating or editing code lists can select an appropriate solution by examining the features of each solution against their own requirements. By selecting solutions which, as a baseline, are Open Source, active, and have previously been used in the public sector, this analysis ensures that almost any of the tools analysed here could be used for code list management with reasonable success. However, organisations using this report as guidance in selecting the most appropriate code list management software for their needs, should take into account the specific characteristics most relevant to their specific situation. For instance, an organisation lacking the technical skills to implement a more complex solution, might prefer avoiding those solutions which involve a complicated installation and configuration process. Alternatively, an organisation planning to create a code list that would have a lot of stakeholders and a lot of changes to implement to that code list would need to pay closer attention to the change management aspect.

### **Global scores**

Figure 17 below illustrates the overall results of the tools' evaluation, based on collecting the total number of fulfilled requirements out of the total of 39 quantified features. The top three solutions, TemaTres, Re3gistry and Ginco, all fulfil more than 30 of the requirements, with TemaTres standing out as the solution meeting most requirements. Although some solutions earn better scores overall, this does not necessarily indicate that a higher-scoring solution is the optimal one to use for all organisations. Since organisations have different needs depending on whether they are code list publishers or consumers, whether their code lists are frequently updated, whether external change requests are common, the score itself does not make a definitive statement about the case-by-case suitability of a solution for the work of a given organisation. Needs, technical context, existing resources vary, and therefore an interested organisation would be better off examining whether a solution fulfils their individual requirements as well, not just whether it has a high score.

Figure 18 represents a juxtaposition of each tool's scores per category. This chart allows for easy comparison of the features provided by each of the tools analysed, in particular for users looking for software to support them for a specific area of code list management. This figure indicates a relative dearth of tools that enable change management, while most do very well in the areas of code list creation, use, and extension, as well as in user management, release management, and mapping creation.

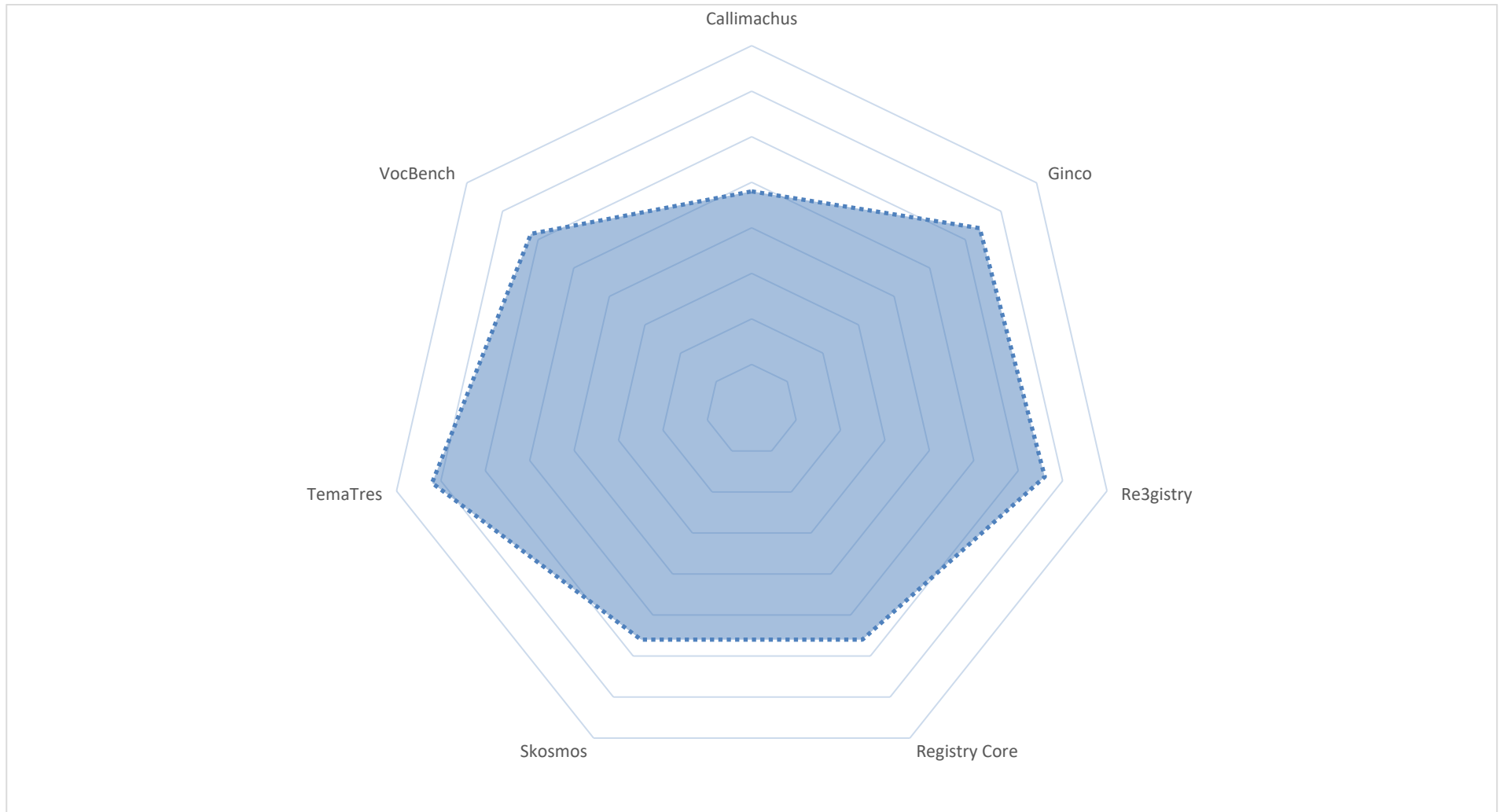


Figure 17: Global scores



# Updated Metadata Governance Tools Report

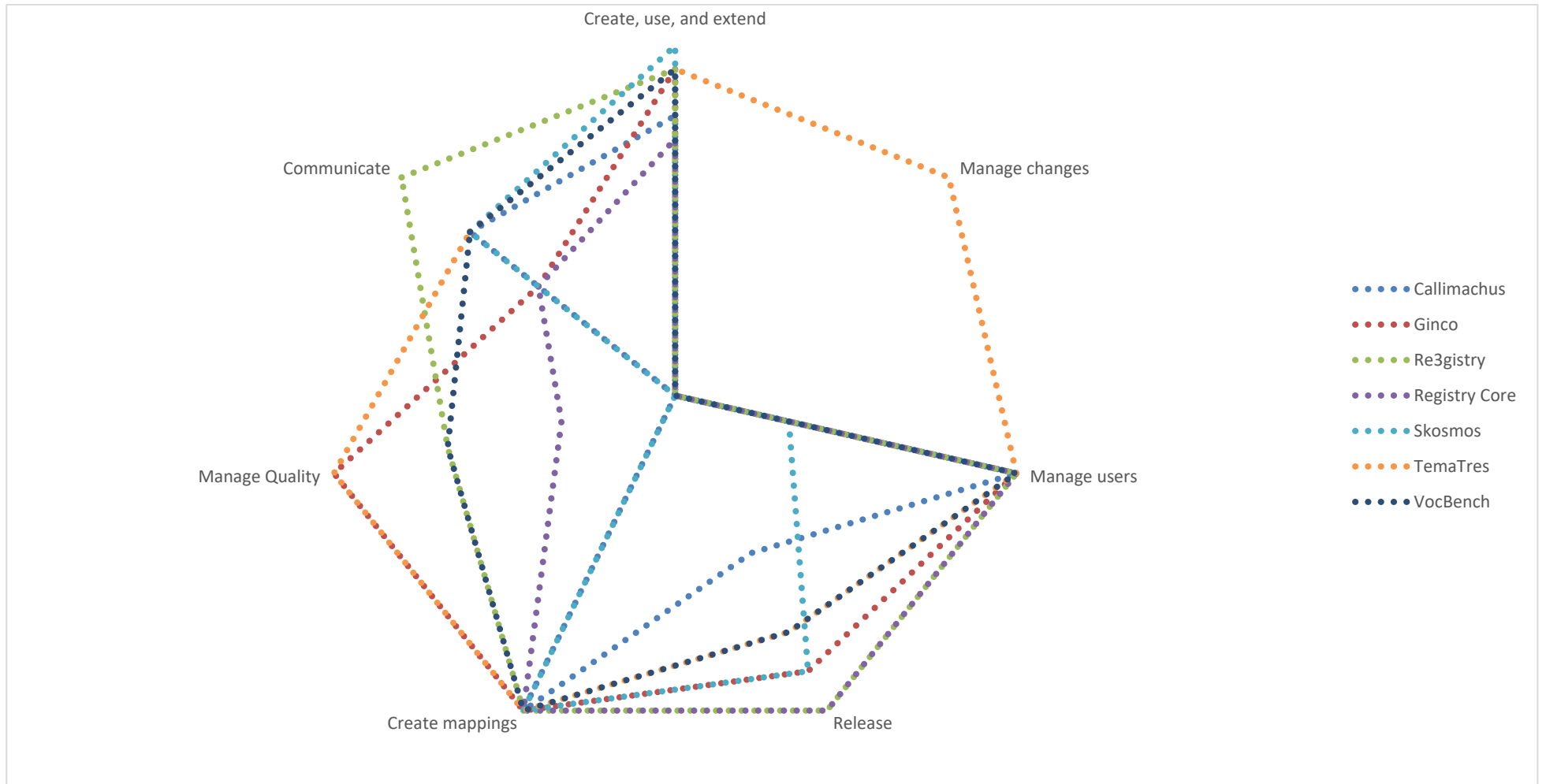


Figure 18: Scores per tool and per category