# • Investigation into Controlled Vocabulary Authoring and Publishing

## • 1 Summary


- IMOS/EMII should avoid pursuing an internally developed ad-hoc solution and instead focus on integrating other provider's services and products to match needs due to limited development resources.
- LDP and SISSVoc should be considered complementary solutions rather than alternatives.
- It is not obvious that centralised registration management such as that provided by LDP necessarily matches IMOS/EMII organisational publishing requirements. Even LDP authors acknowledge that their project is contrary to the non-federated nature of RDF publishing - while SISSVoc deployments integrate well, even when independently hosted.
- Support for more sophisticated vocabulary authoring and development activity probably does not yet exist in the form of a web-based solution. For example, to support inferencing and consistency checking of OWL vocabularies already found in dedicated third-party vocabulary applications, or to verify the reciprocal relationship of broader/narrower SKOS terms.
- An investment has already been made in the development of a controlled vocabulary database. This includes supporting the appending of Geoserver parameter and unit information at WFS download and as a source of SKOS vocabulary for Geonetwork/Portal 123 for faceted search. Abandoning this effort would require the establishment of new work-flows as well as infrastructure changes.
- LDP is currently unfunded, lacking project oversight, technical support, points of contact or active discussion forums. It clearly presents an unrealistic technical-risk unless there is a consensus from multiple external groups on new funding and feature development.

## 2 Tasks undertaken

The following tasks were completed:

- A local instance of SISSVoc using pre-configured external SPARQL endpoints for various service providers was run-up under a Jetty webserver and demonstrated. [1]
- It was very difficult to glean an understanding of the functionality of LDP by reading source-code and internal development literature. Therefore a version of UKGovLD was also run-up on a private Nectar VM allocation. This followed prior unsuccessful attempts to provision a vagrant example of UKGovLD. [2] Attempting to generate a dummy configuration for LDP and login with administrative privileges was frustrated, due to an apparent bug in open-id redirection. However it was possible to explore and navigate the main menu-system of the application. The discussion of LDP features is therefore more limited in scope.
- Preliminary scripts to extract and encode the contents of the IMOS controlled vocabulary database as SKOS were written to gauge the difficulty to perform the required mapping. These were further modified to be suitable to input into Geonetwork for current development to support platform / parameter based faceted search. Such export feature would probably also be necessary for upload to a publishing provider such as ANDS or purl.org.
- Attempted to register an account with ANDS to establish a test registry of published vocabulary terms. There were some technical issues, requiring ANDS technical assistance which were unresolved.
- SISSVoc and LDP was assessed against the *High Level Functional Requirements For Vocabulary and Term Publishing* previously prepared by Project Officers.

[1] `http://github.com/jyucsiro/SISSVoc-runner`

[2] `https://github.com/UKGovLD/registry-deploy-poc` and `http://ukgovld-registry.s3.amazonaws.com/distribution/ukl-registry-0.0.1-SNAPSHOT-dist.tar.gz`

# 3 High Level Functional Requirements For Vocabulary and Term Publishing

## 3.1 Main function is to provide a resolvable endpoint for a vocabulary and its included terms (and details) using persistent identifiers.*

- For each term in the vocabulary persistent identifiers URI should be de-referenceable to the RDF item description.
- SISSVoc as a stand-alone application has no inherent support for managing persistent identifiers. The main SISSVoc paper describes a deployment [3], using a Persistent Identifier Service to map persistent URI resources to SISSVoc web service urls, but fails to give further details on the implementation or whether an external provider was chosen.
- ANDS controlled vocabulary services builds upon SISSVoc, although it appears their persistent identifier service may be a more general organisational capability. According to documentation ANDS has,

    - *[...] a simple HTTP-based interface, [which] ensures that identifier services can be integrated easily into existing data management work flows.*

  Furthermore, ANDS does undertake to persist the infrastructure required for keeping its identifiers online. Ideally this service would be integrated with other SISSVoc vocabulary management functionality, although this was not tested.
- Another alternative, would be to use a service such as `purl.org` which is a free provider of Persistent Uniform Resource Locators, that includes high-level administrative functionality including Users, Groups, Domains, Help etc. [4].
- It would probably also be a simple step to publish identifiers under an AODN or EMII DNS controlled url. In this context, Seegrid appear to have developed their own PID service used in conjunction with their own SISSVoc deployment [5]
- LDP is a resource registry management system, however it is unknown if this includes direct support for persistent identifiers. [6]

## 3.2 Resolvable content should be structured (or at least be able to be queried) using an RDF/SKOS encoding model. Content may however be adorned by other languages/metadata models (e.g. RDFS, OWL, Dublin Core).*

- SISSVoc provides a linked data API for publishing SKOS vocabularies. SKOS is a standard vocabulary for thesarui, classifications, taxonomies and controlled vocabularies using RDF.

[3] See Figure 3, SISSVoc: A Linked Data API for SKOS vocabularies. `http://www.semantic-web-journal.net/system/files/swj658.pdf`

[4] The most prominent instances of such schemes are PURLexternal link, which has been used by the National Library of Australia, and ARKexternal link, at the California Digital Library.

[5] https://www.seegrid.csiro.au/wiki/Siss/PIDService

[6] https://github.com/UKGovLD/ukl-registry-poc/wiki/Principles-and-concepts

- The SISSVoc web-service API supports URI patterns that are aligned with the SKOS vocabulary model. This includes access patterns for SKOS Concept, ConceptScheme and Collection. Further URI patterns are provided to discover broader and narrower terms in transitive and non-transitive forms and according to text based labels.
- SKOS can be decorated with other RDF based content and persisted to any underlying store independently of SISSVoc functionality. However, SISSVoc provides no URI patterns for the search and discovery of such content. Alternatively, a SPARQL interface does provide a query/search mechanism for non-SKOS content such as DC, RDFS or OWL classes. A local-instance of SISSVoc could be modified to use this SPARQL with support in the GUI if such functionality was deemed important.
- It is believed that LDP has no specific API support for SKOS resources.

### 3.3 It should be possible to access vocabularies and their terms via an (administratively) customisable Web-client interface and service interfaces.*

- SISSVoc provides a capable Web-client interface for read-only access vocabularies and their terms.
- In contrast to search, navigation and discovery, SISSVoc as a stand-alone application has no direct support for the creation and update of vocabulary terms. This is inherent to SISSVoc design, as a lightweight web-api implemented over a read-only SPARQL endpoint/interface.
- According to the ANDS handbook, ANDS provides web-based GUI support for editing SKOS but only at file level. [7] Some support for web-based versioning and author management at the file level is also available while permissions to make change are tied to authority roles. [8]
- For complex vocabulary authoring needs, SISSVoc authors suggest using an external vocabulary editor to maintain content that can also ensure that consistency of relationships between resources is maintained.

    - *Vocabulary content may be maintained using RDF editors (such as Protg 4 or TopBraid Composer 5 ), which ensure consistency of relationships between resources is maintained, and then generate RDF documents to transfer vocabulary content from the maintenance to publication environment, as outlined above. If a web-based vocabulary maintenance environment is required, then tools like TopQuadrants Enterprise Vocabulary Net 6 , and the PoolParty Thesaurus Server 7 are available.*

9

[7] See 3.4.3, Editing Vocabularies `http://www.ANDS.org.au/support/vocab-help-guide.pdf`

[8] There is a need to consider whether file level versioning and management is sufficiently fine-grained. Also how should this interact with existing registry management already used in IMOS `control_vocab_db`.

[9] 3.2. SISSVoc HTTP operations and REST behaviour discussed, `http://www.semantic-web-journal.net/system/files/swj658.pdf`

### 3.4 Most users require read only access to content.*

- SISSVoc provides an easy-to-use Web-client interface for read-only access to vocabularies and their terms.
- LDP provides a Web-client for fine grained registry management of term resources and is unlikely to be useful for read-only access to content.

### 3.5 There should ideally be a service interface that is REST-based* and a SPARQL service end-point.

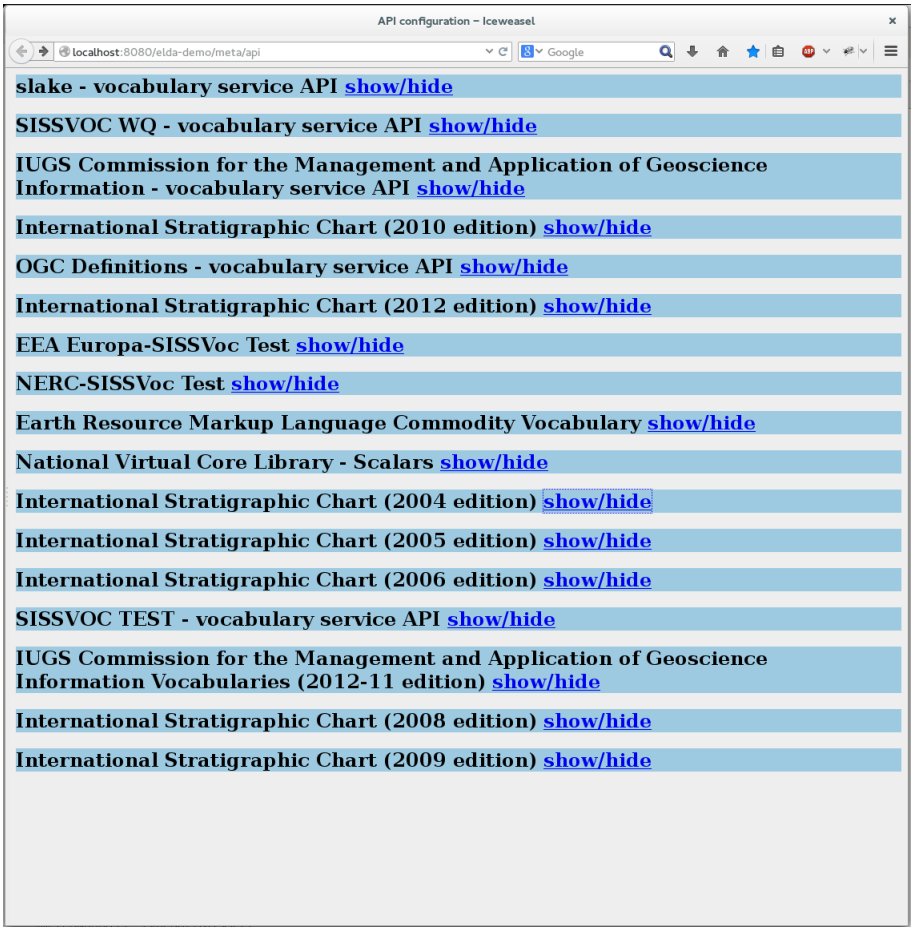
- SISSVoc is designed with a HTTP-based interface aligned with REST-based web services. The URI patterns facilitate SKOS discovery and access. However, as the authors of the system note, SISSVoc is not a full RESTful API, as it does not support HTTP operations for update and deletion of resources.
- LDP is designed with a view to RESTFUL management of registry resources. [10] It should be noted that references to SKOS in the discussion of the LDP api apply to registrars, not SKOS content maintained by particular registrants.
- SPARQL (Simple Protocol and RDF Query Language) is an RDF query language able to preserve, retrieve and manipulate data in RDF format and is a W3c standard. SPARQL abstracts the encoding model (XML, etc), and persistence layer and is adapted to the non-relational / graph structure of RDF.
- The SISSVoc RESTFUL URI patterns correspond closely with specific SPARQL queries which perform the substantial work. SISSVoc thus relies on a SPARQL endpoint as the mechanism to access RDF providing a strong separation of concerns. It would be feasible to map the already developed IMOS `control_vocab_db` to expose a SPARQL interface using a tool such as r2rml [11]. This has already been achieved in one controlled vocabulary instance [12]. The SPARQL interface would thus provide the endpoint for SISSVoc and replace the need to manage an additional persistence layer.
- It is expected that an external SISSVoc provider such as ANDS would be unlikely to expose such a low-level query capability.

### 3.6 The publishing and retrieval service should offer and receive re-direction(s) so that vocabularies or terms hosted on different domains (under differing content authorities) can still be accessed via the service (if desired). *

- SISSVoc supports seamless and consistent GUI based navigation between differing SISSVoc instances hosted on different domains.
- Alternatively, normal URI de-referencing allows navigation to non-SISSVoc vocabulary implementations and their terms (eg. BODC/NERC).
- Interestingly, due to the nature of the decoupled end-point design, SISSVoc can be configured to use multiple SPARQL endpoint providers. This unifies access to different vocabulary providers that expose SPARQL from the same SISSVoc instance.

[10] https://github.com/UKGovLD/ukl-registry-poc/wiki/Api
[11] `http://www.w3.org/TR/r2rml/`
[12] `http://www.seb-source.org/`

Figure 1: SISSVoc host demonstrating access to multiple SISSVoc domains from the same instance

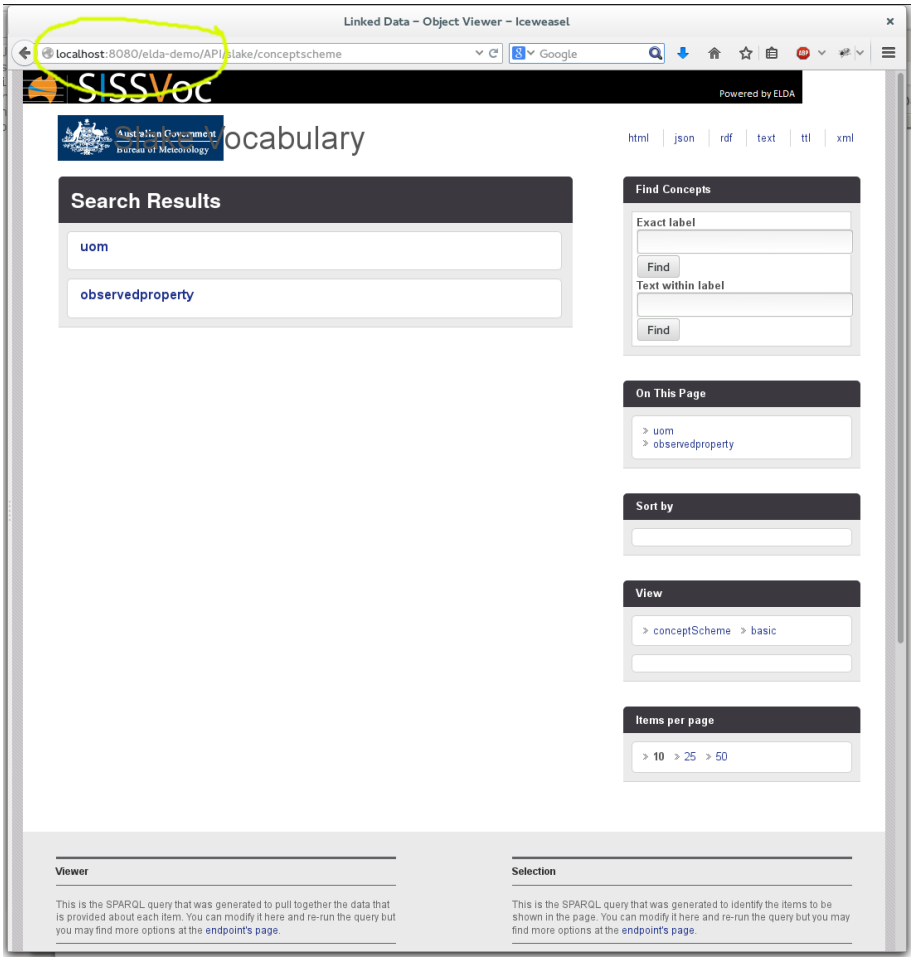



LDP GUI navigation capabilities are unknown, but assumed to be limited to registry management.

### 3.7 The Web-client should support some basic canned querying (e.g. free text search against concept, collection and scheme labels; traversing a named vocabulary via hypertext links to explore included terms, their details and any matches or mappings to other published vocabularies). *

- SISSVoc supports free-text searching against concept labels. It can list concept, collection and concept schemes. SISSVoc can also traverse links for associated broader and narrower terms.

Figure 2: SISSVoc Gui Concept search and results

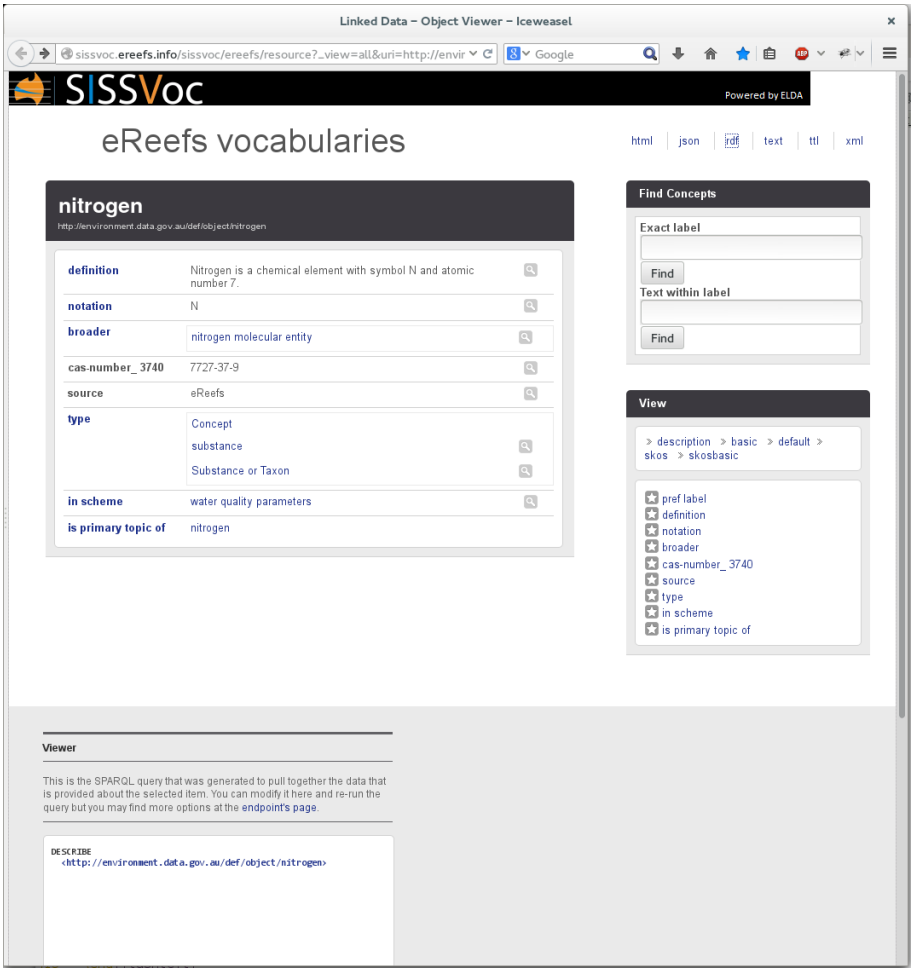



LDP capabilities for registrar searching are unknown.

### 3.8 The Web-Client should be able to display in a user-friendly way (e.g. using tables or forms) vocabulary and term details (e.g. scheme, collection and concept labels; alt labels, its type, description, membership, relations [including some Dublin core relations such as publisher and owner and revision info]). *

- The SISSVoc Web-Client supports user-friendly presentation of SKOS term details including preferred label, definition, broader, and membership relations. Also other metatadata terms such as DC.
- It appears that the simple html table layout supports showing all associated RDF properties if they are literals or RDFs:label types rather than URIs that need to be de-referenced.

Figure 3: SISSVoc Gui SKOS Resource




### 3.9 The Web-Client should be able to display categorized (classified) lists of discoverable content (e.g. all vocabs by provided by owner X; all terms in vocab Y)

- SISSVoc as a stand-alone application has no notion of term ownership semantics. There is a set of extensions for SKOS that appear to be designed to support management such as assigning author properties. [13]
- ANDS may have support since it combines some registry management with SISSVoc.
- LDP unknown, although conceivable since consistent with registry management function.

[13] `http://www.w3.org/TR/skos-reference/skos-xl.html`

### 3.10 The Web-Client should offer different formats in which to download vocabularies or their terms (e.g. RDF*, text*, json, html*)

- SISSVoc supports download in human and machine readable form including html, json, RDF, text, ttl and XML. However, the web-api has no support for downloading partial or complete SKOS vocabularies except via a manually crafted SPARQL request.
- LDPs encoding support is unknown.

### 3.11 The Web-Client should offer some statistics for users on the type and volume of content available (e.g., number of vocabularies that can be accessed and the number of terms in each vocabulary).

- SISSVoc has no inherent support for compiling vocabulary content statistics.
- However, using a SPARQL interface it ought to be trivial to construct queries to identify for example the number of SKOS concepts, schemes or collections available with varying search constraints applied.

### 3.12 The publishing and retrieval service should be capable of being configured to dynamically read one or more repository sources to access content that needs to be published. *

- As has been described, a stand-alone SISSVoc test-runner can be configured to read from multiple end-point data sources. Although the front-end GUI support is not very polished and uses styling inconsistent with the rest of the application.

### 3.13 Response times for retrieving queried content should be user-tolerable. *

- During basic user-interface testing, both the local and remote ANDS options assume appear to be responsive.

### 3.14 System should provide an administrative console/configuration files to enable simple maintenance and administration (e.g., small changes to Web-client interface displays and supported queries; to detect and fix broken links in client-based hypertext; detecting missing details in retrieved content indicating content needs moderating/validating; provide basic statistics on service usage).

- It's believed that neither SISSVoc or LDP offer any administrative control over configuration.
- A locally deployed SISSVoc would permit the normal configuration possibilities that comes with controlling the source code - such as css, ttl and JavaScript and xslt changes. Extended examples are provided for different branding options.
- For a local deployment, usage statistics could be compiled using awstats, in the same fashion as other IMOS web-applications.

### 3.15 There should be meaningful error messaging provided in response to service calls that cannot be satisfied (or which have been framed incorrectly). *

- SISSVoc and LDP appear to support basic error handling.

# 4 Appendium / Other Issues

- A further point in favor of maintaining our db. Harvesting netcdf and the opportunity to use the existing db connection to take advantage of parameter and unit during NetCDF attribute mapping . Allows AODN contributors to mark up NetCDF using IMOS/EMII controlled vocab, and record that information during harvest with possiblities for simpler generic harvesters using parameter mapping. Also much better that database data fields are controlled rather than dumb text strings.
- Importance of having a single authorative point of vocabulary. Use and exchange in many places - geoserver, geonetwork, sissvoc, pid. manual exchange of skos files is going to end has risk of inconsistency.
- R2R vocabularly management was unrelated to r2rml. Although SISSVoc author notes there is no reason existing realational db could not use sparql mapping system to use SISSVoc, and it's extensively specified by w3 standards.
- Important - for the PIP - sissvoc itself should be modified to serve - in accordance with the terms contained in the db - or checked by sparql