

Approach to persistent identifiers and data-service-coupling in the German Spatial Data Infrastructure

Martin Seiler¹, Marcus Walther², Jürgen Walther²

¹Coordination Office Spatial Data Infrastructure; ²Federal Agency for Cartography and Geodesy

1 Introduction

The *Geodateninfrastruktur Deutschland* (GDI-DE) is the German initiative to build a national Spatial Data Infrastructure (SDI). Its goal is to offer standard-based access to all spatial data from public administrations through web services. Furthermore it is open to participants from the private sector and the general public. Additionally the GDI-DE sees itself as part of a European SDI as defined through the INSPIRE directive.

The GDI-DE is realized through a distributed architecture of OGC Web Services (OWS). In this architecture metadata are replicated in several nodes.

In this position paper we briefly describe one of the most pressing issues in SDIs: The data-service-coupling, which relies on unique and persistent identifiers for resources. We present our approach to solve this issue making use of a national registry service.

2 Resource description and discovery in the SDIs

To realize the use cases of the publish-find-bind pattern in SDIs, metadata on resources are accessible through two distinguishable concepts:

- a) through catalogue services providing search interfaces on metadata documents describing datasets, dataset series or services;
- b) through capability documents describing web service interfaces regarding the supported functionality and offered data.

In a) resources are described following the metadata model defined in ISO 19115/19119 and made available for search and retrieval through OGC CSW 2.0.2 ISO AP 1.0 compliant interfaces. Results are then returned in XML encoding as defined in ISO 19139.

In b) OWS interfaces are described as defined in the relevant OGC specifications. OGC web services provide their capabilities documents in response to *GetCapabilities*-requests in XML encoding.

The INSPIRE directive deviates partly from the ISO and OGC standards/specifications and adds additional requirements regarding metadata.

Organizations collecting, updating and providing data are commonly not the same as the organizations providing service interfaces in a SDI. One dataset can be provided through many services, a service can provide access to many datasets. So the data provider does not necessarily have knowledge about services offering its data. Therefore metadata describing datasets/dataset series (type a) do not contain information on the services providing access to them. But metadata on

services (both types) must contain references to the datasets they operate on (Fig. 1).

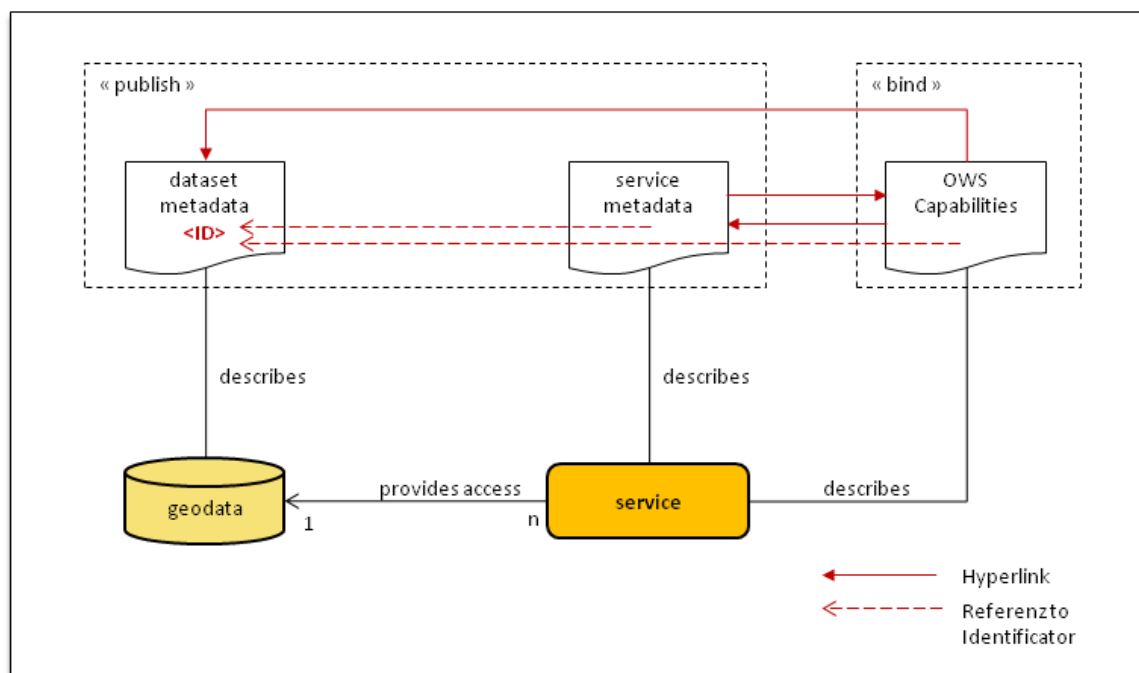


Figure 1: Data-Service-Coupling in a SDI

GDI-DE (2014)

To realize the <<find>> use case in an SDI the user/application has to query the CSW for **service** metadata containing a reference to the dataset in question. This service metadata set then contains the endpoint of the OWS offering access to the dataset.

3 Identification and coupling of resources in SDIs/INSPIRE

In SDIs each resource shall be identifiable through at least one unique identifier. ISO 19115 provides the element

XPath: `identificationInfo[1]/*/citation/*/identifier/MD_Identifier/code`

for this purpose. In ISO 19115 the *MD_Identifier* data type, with attributes for *authority* and *code*, is defined as a “value uniquely identifying an object within a namespace”. Additionally the standard defines *RS_Identifier* as inheritor of *MD_Identifier* with additional attributes for *codeSpace* and *version* as “identifier used for reference systems”.

The INSPIRE Technical Guidelines on metadata (INSPIRE, 2013) suggest to use either *code* in *MD_Identifier* or to construct the unique identifier for a resources using *code* + *codespace* as *RS_Identifier*.

To document the reference to the dataset in the service metadata record ISO 19119 provides the element

```
XPath: identificationInfo[1]/*/operatesOn
```

INSPIRE (2013) mandates here the use of “a unique resource identifier or locator (URL) of the *MD_DataIdentification* object” of the metadata record describing the dataset.

In the example provided in the Technical Guidelines a *GetRecordById* request (returning the XML metadata record of a dataset) is used. To reference the *MD_DataIdentification* object an anchor is added to the request (*#code*):

<http://SERVICEURL/csw?SERVICE=CSW&VERSION=2.0.2&REQUEST=GetRecordById&ID=fileidentefierOfMetadataSet&OUTPUTSCHEMA=http://www.isotc211.org/2005/gmd&ELEMENTNAME=full#code>

4 Issues in practical application

INSPIRE encourages the use of unique identifiers for metadata in the form of an URL. Additionally there should be a service returning exactly the referenced metadata record by using the HTTP GET method. In contrast to this request in Germany the federal structure of the administration led to diverging implementations. Most of our stakeholders already use URLs as prefixes for their identifiers (partially split into “code” and “codespace”). But often the URLs are non-resolvable and may change over time. Additionally other concepts exist.

The use of *GetRecordById*-requests is problematic here, as it implies that in

```
XPath: identificationInfo[1]/*/citation/*/identifier/MD_Identifier/code
```

the entire *GetRecordById* request has to be provided and updated in case of changes in the architecture.

5 Solution

In order to provide stable, unique and resolvable identifiers we suggest the use of a central registry service. Identifiers are built from the following pattern: <https://registry-URL/namespace/ID>, e.g.

<https://registry.gdi-de.org/de.bund/bkg/abcd-1234-efgh-5678>

The providers of metadata simply register at the service. Afterwards they are allowed to maintain their own namespace. Namespaces may be organized hierarchically in order to reflect federal structures (as shown in the example). There is no need to use a UUID as ID, but the ID has to be unique within its namespace.

Furthermore the provider also has to define a redirect pattern for accessing its original catalogue service. By defining a *GetRecords* request the registry is enabled to send an appropriate HTTP redirect to the user.

Example identifier:

[https://registry.gdi-de.org/de.bkg/test/\[\[ID\]\]](https://registry.gdi-de.org/de.bkg/test/[[ID]])

redirects to:

[http://gdk.gdi-de.org/gdi-de/srv/csw?REQUEST=GetRecords&SERVICE=CSW&VERSION=2.0.2&OUTPUTSCHEMA=http://www.isotc211.org/2005/gmd&constraintLanguage=CQL TEXT&constraint=ResourceIdentifier like '\[\[ID\]\]'&constraint language version=1.1.0&typenames=csw:Record&resulttype=results&elementsetname=full#\[\[ID\]\]](http://gdk.gdi-de.org/gdi-de/srv/csw?REQUEST=GetRecords&SERVICE=CSW&VERSION=2.0.2&OUTPUTSCHEMA=http://www.isotc211.org/2005/gmd&constraintLanguage=CQL TEXT&constraint=ResourceIdentifier like '[[ID]]'&constraint language version=1.1.0&typenames=csw:Record&resulttype=results&elementsetname=full#[[ID]])

We need to use *GetRecords* here instead of the *GetRecordById* method because the latter is based on the value of the *fileIdentifier* of the metadata record – not the resource it describes. This differs from the value of *RS_Identifier* or *MD_Identifier*.

To handle metadata record access without knowing the *fileIdentifier*, the redirect pattern inside the registry returns a *GetRecords* request. The result is returned inside a *GetRecordsResponse* envelope. Because “#code” is added as anchor to the request, the result for the client is identical to a *GetRecordById* request.

A potential problem with our current implementation is that OGC CSW 2.0.2 ISO AP 1.0 defines only POST requests for the *GetRecords* operation as mandatory. However most implementations used in Germany support the optional GET request in a proper manner.

6 Conclusion & Outlook

Making use of the registry service we can support the definition of persistent, unique and dereferenceable identifiers in our SDI.

Regarding the data-service-coupling the INSPIRE TG on metadata (INSPIRE, 2013) show a *GetrecordsById* request inside the *operatesOn* element as outstanding example. In case the service URL changes all metadata records have to be updated. Using the registry redirect service the update is only necessary once for the redirect pattern inside the registry in order to point to the new location. This should eventually lead to more data being accessible through OWS in SDIs.

We plan to use our registry service at least for the following purposes:

1. Namespaces
2. Codelists
3. Automated INSPIRE-Monitoring
4. CRS-parameters

Using a registry service and a resolver may enable the use of URN patterns, e.g. “urn:sdi:de.bund/bkg/**ID**”.

7 References

GDI-DE (2014): Architektur der GDI-DE v3 – Technik (unpublished)

INSPIRE (2013):

http://inspire.jrc.ec.europa.eu/documents/Metadata/MD_IR_and_ISO_20131029.pdf