



CSW - Schulung und Workshop

BSH Hamburg, 09. Juli 2011

Carsten Heidmann (disy Informationssysteme GmbH)

Agenda

- Einführung
 - Geplanter Tagesablauf
 - Erwartungen und Vorkenntnisse
 - Installation Testumgebung
- Grundlagen
 - Entwicklung Metadatenstandards
 - Entwicklung Katalogdienste
 - OGC Architektur
- CSW – Standards
 - Obsolete Versionen
 - CSW 2.0.2
 - AP ISO 1.0
 - INSPIRE
- OGC Filter Encoding für CSW
- NOKIS CSW im Detail
 - Überblick
 - Besonderheiten
 - Technik

Einführung

Installation Testumgebung

Auf USB-Stick von mir:

- aktuelles Java 1.6.0 (1.6.0_25) 32/64
- Preludio CSW-Klient
- Beispielanfragen
- Portable Notepad++
- Portable Firefox 4

Geplanter Tagesablauf

- 09:00 – 12:00 Block 1
 - Einführung
 - Grundlagen Katalogdienste
 - CSW-Standards
- 12:00 – 13:00 Mittag
- 13:00 – 17:00 Block 2
 - NOKIS CSW im Detail
 - Offene Fragen

- Übungen laufend zwischendurch
- Fragen sammeln wir und behandeln sie gleich bzw. separat am Ende

Erwartungen und Vorkenntnisse

- Warum habt ihr euch für diesen Workshop angemeldet?
- Was wollt ihr am Ende dieses Workshops können?
- Welche Erfahrungen habt ihr mit CSW?
- Welche Erfahrungen habt ihr mit der NOKIS-CSW?

Grundlagen

Entwicklung Metadatenstandards

- "Urform" Zettelkatalog
- Standards
 - MAB - Maschinelles Austauschformat für Bibliotheken
 - MARC - MACHine-Readable Cataloging
 - Dublin Core
 - Geodaten: FGDC CSDGM
 - Geodaten: ISO 19115, ISO 19119
- Bei Metadatenstandards handelt es sich normalerweise um ein Inhaltsmodell – nicht um ein Encoding!

Entwicklung Katalogdienste

- Notwendig erst durch Vernetzung
- "Urform": Austausch von Bibliotheksdaten: Z39.50 (seit 1984!)
- OGC: Catalog Interface 1.0 (1999)
- OGC: Catalog Service 2.0.1 (2004) → DE-Profil
- Opensearch (2005)
- OGC: Catalog Service 2.0.2 (2007) → AP ISO 1.0 → INSPIRE
- OGC: Catalog Service 3.0 (in Vorbereitung)

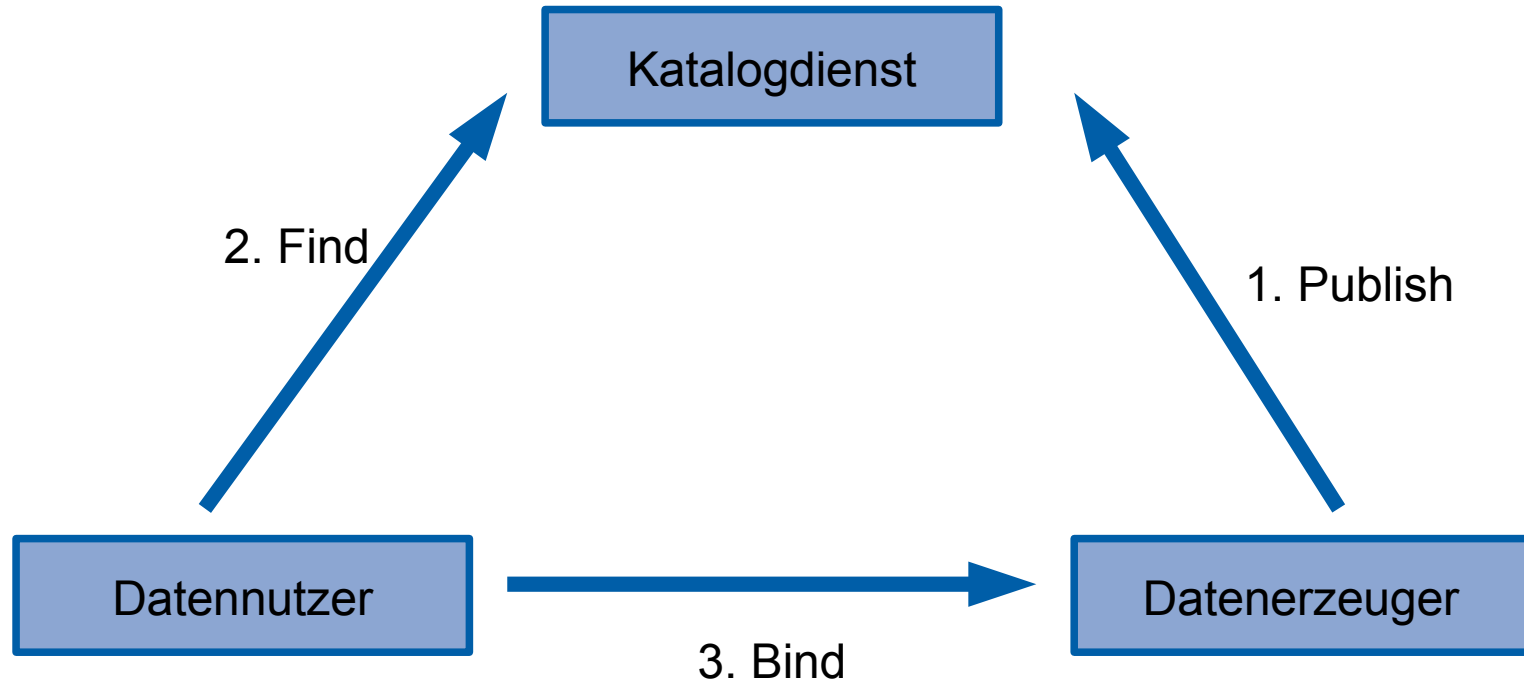
Warum Katalogdienste?

- Metadatenysteme alleine dienen organisationsinterner Information
- Übergreifende Informationen können entweder
 - Redundant oder
 - Dienstebasiert bereitgestellt werden
- Redundante Haltung **aller** Metadaten ("Harvesting") nicht sinnvoll
- Gemeinsame Schnittstelle notwendig
→ Schnittstelle = Gemeinsamer Dienstestandard + gemeinsamer Inhaltsstandard!
- Erst Katalogschnittstellen machen eine Einbettung in verschiedenen Kontexten möglich

Redundanz ("Harvesting") vs. Verteilte Suche

- Harvesting
 - + Schnelle Suche möglich
 - + Keine Probleme bei Netzwerkausfall oder Ausfall einzelner Komponenten
 - - Redundante Datenhaltung
 - - keine Garantie der Aktualität
- Verteilte Suche
 - + hohe Aktualität der Metadaten
 - + keine Redundanz in der Datenhaltung
 - + Daten bleiben bei datenführender Stelle
 - - Suche bricht bei Netzwerkausfall oder Ausfall einzelner Komponenten ganz oder teilweise zusammen
 - - Performanz ist **immer** schlechter als bei Harvesting

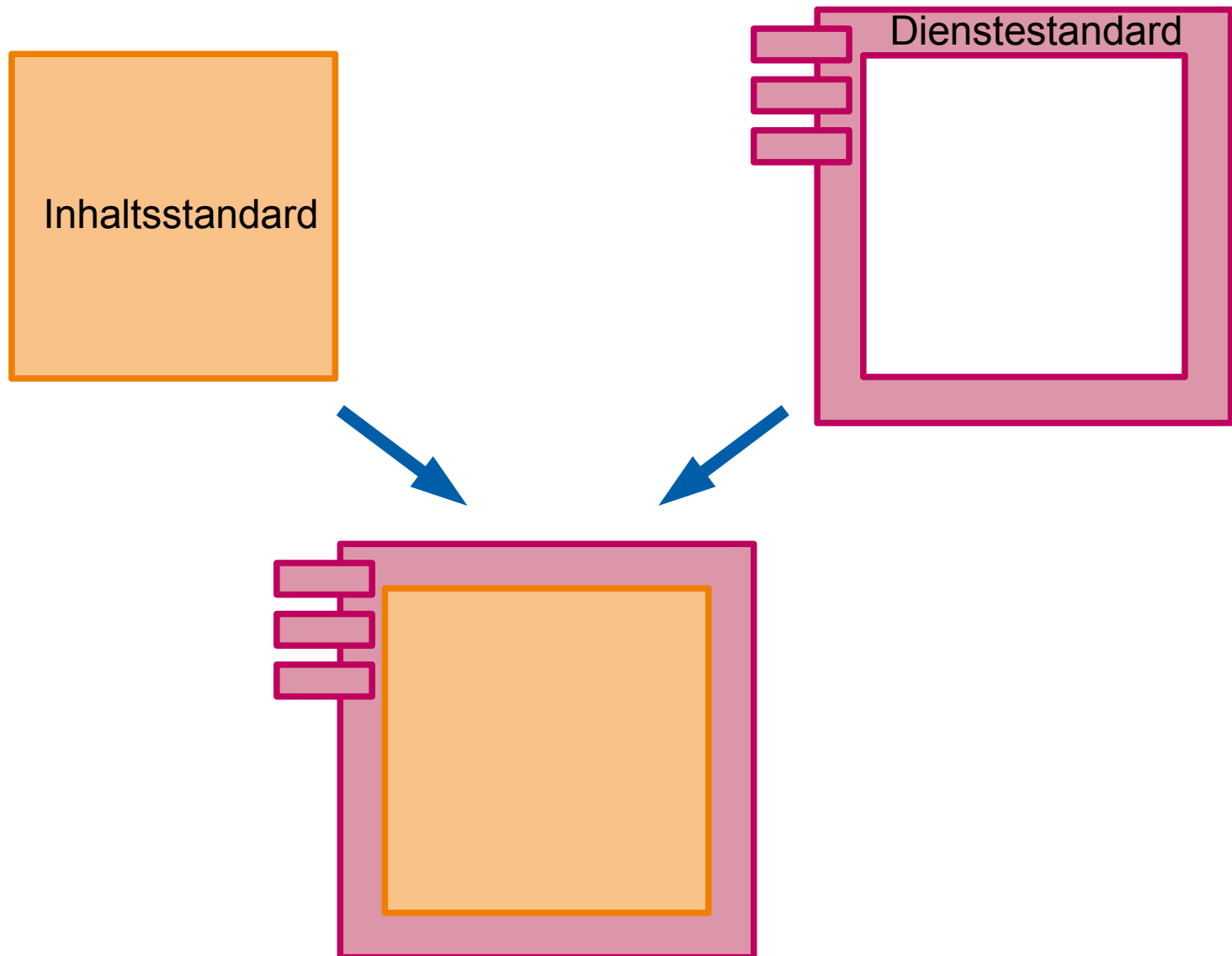
Das Publish-Find-Bind Pattern



OGC Referenzarchitektur

CSW – Standards

Dienstestandards vs. Inhaltsstandards



CSW-Dienstestandards im OGC-Umfeld

- **Obsolet**
 - CSW < 2.0
 - CSW 2.0.1 → DE-Profil (!)
- **Aktuell**
 - CSW 2.0.2 (→ AP ISO 1.0)
 - INSPIRE Discovery Service

Open Geospatial Consortium Inc.

Date: 2007-02-23

Reference number of this document: OGC 07-006r1

Version 2.0.2, Corrigendum 2 Release

Category: OpenGIS® Implementation Specification

Editors: Douglas Nebert, Arliss Whiteside, Panagiotis (Peter) Vretanos

OpenGIS® Catalogue Services Specification

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OpenGIS® Publicly Available Standard
Document subtype: Implementation Specification
Document stage: Proposed Revision
Document language: English

Bestandteile einer CSW-Dienstespezifikation

- Operationen
 - Anfrageformat
 - Antwortformat
- Abfragesprachen
 - OGC CQL
 - OGC Filter Encoding → CSW, WFS, SOS, ...
- Protocol Bindings
 - GET
 - POST
 - SOAP
- (Inhalte) → in CSW 2.0.2 wird kein Inhaltsschema vorgeschrieben!
- (Encodings) → Encoding ist stark vom Inhaltsmodell abhängig. Für ISO 19115 wird nur ein XML-Encoding bereitgestellt (ISO 19139)

Exkurs: Protocol Bindings

- GET vs. POST
- Was ist SOAP?
- POST XML vs. SOAP
- Mögliche CSW-Protocol-Bindings

GET vs. POST: Eigenschaften eines GET-Requests

- Spezifiziert im HTTP-Protokoll
- GET: "URL-Parameter"
- Beispiele:
 - <http://www.disy.net/index.php?id=222> → disy Homepage
 - <http://www.mein-server.de/suche.html?seite=42> → Beispiel für paginierte Suchseite
- Vorteile
 - Ausführung ist sehr einfach möglich (Browser, wget, curl, ...)
 - Parameter sind "lesbar" in der URL
- Nachteile
 - Komplexe Anfragen sind schwierig abzubilden (Längenbegrenzung URL, Sonderzeichen, ...)
 - Parameter sind "lesbar" in der URL
 - SSL-URLs werden verschlüsselt – im Serverlog tauchen aber die entschlüsselten URLs wieder auf!

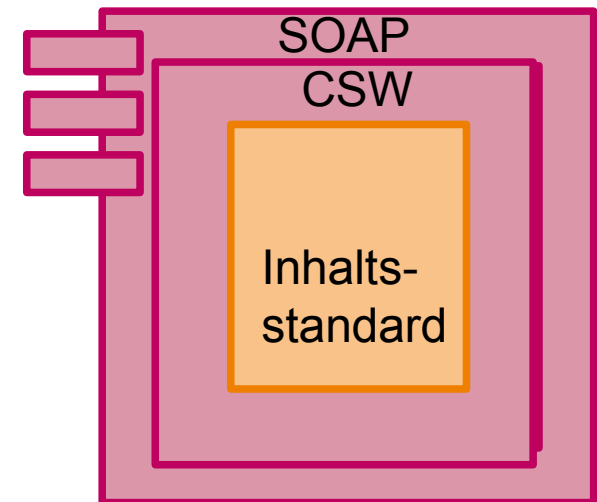
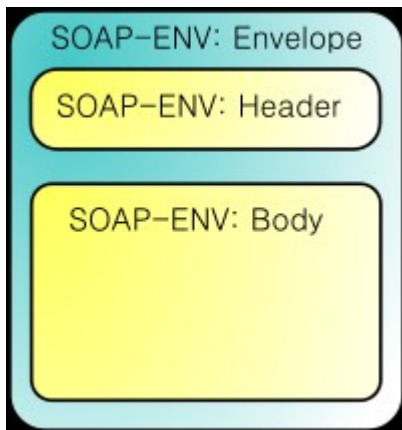
GET vs. POST: Eigenschaften eines POST-Requests

- Spezifiziert im HTTP-Protokoll
- POST: Daten sind nicht unmittelbar sichtbar im Request
- Beispiele: Upload einer Datei im Browser
- POST-KVP / POST-XML

- Vorteile
 - Keine Beschränkung der Größe (Achtung: Ausnahme möglich)
 - Keine/geringe Probleme beim Encoding von Sonderzeichen
 - Beliebige Inhalte möglich: Text, Bilder, Programmdateien, ...
 - Keine sichtbare Übertragung im RequestHeader
 - Durch SSL-Verschlüsselung sind die Daten bei der Übertragung gesichert
- Nachteile
 - Eine Anfrage lässt sich nicht ohne Add-Ons im Browser testen (z.B.FF: Tamper Data)

Was ist SOAP?

- SOAP (ursprünglich: Simple Object Access Protocol) ist ein **Netzwerkprotokoll**, mit dessen Hilfe **Daten** zwischen Systemen ausgetauscht und **Remote Procedure Calls** durchgeführt werden können (Wikipedia)
- Aufsatz auf POST-XML
- SOAP-Nachricht besteht aus
 - Envelope (immer) → "Verpackung"
 - Header (optional) → Metadaten
 - Body (meistens) → Payload



Vorteile und Nachteile von SOAP

- Vorteile
 - fügt Möglichkeiten zur erweiterten Authentifizierung und zur Autorisierung zum HTTP-Protokoll hinzu
 - Erlaubt weitergehende Verschlüsselungsmöglichkeiten
 - Erlaubt das Hinzufügen von Metadaten über die versendete Nachricht
- Nachteile
 - Weitere Komplexitätsstufe: SOAP-POST-CSW-Inhalt
 - Es wird ein eigenes Framework benötigt
 - Es können Kompatibilitätsprobleme auftreten, wenn verschiedene Frameworks benutzt werden

Mögliche CSW-Protocol-Bindings

- GET
 - KVP
- POST
 - XML
- SOAP
 - XML

CSW(T) ISO Operation	Request encoding
GetCapabilities	XML(POST+SOAP) and KVP (GET)
DescribeRecord	XML (POST+SOAP) and KVP (GET)
GetDomain	XML (POST+SOAP) and KVP(GET)
GetRecords	XML (POST+SOAP) and KVP(GET)
GetRecordById	XML (POST+SOAP) and KVP(GET)
Harvest	XML (POST+SOAP) and KVP(POST)
Transaction	XML (POST+SOAP)

Operationen in der CSW 2.0.2

- Service Operationen
 - GetCapabilities
- Discovery Operationen
 - GetDomain
 - DescribeRecord
 - GetRecordById
 - GetRecords
- Management Operationen
 - Transaction
 - Harvest

Ablauf einer Anfrage

- Tafelbild/Whiteboard

GetCapabilities – Request

- Parameter: SERVICE, REQUEST, VERSION

- GET-Request:

```
http://admin5.env.disy.net/nokis/servants/de/disy/preludio2/service/cat/csw/v_2_0_2/GetCapabilities?SERVICE=CSW&REQUEST=GetCapabilities&VERSION=2.0.2
```

- POST-Request-URL:

```
http://admin5.env.disy.net/nokis/servants/de/disy/preludio2/service/cat/csw/v_2_0_2/GetCapabilities$Post$XML
```

- SOAP-Request-URL:

```
http://admin5.env.disy.net/nokis/servants/de/disy/preludio2/service/cat/csw/v_2_0_2/GetCapabilities$Post$SOAP
```

- Request-Body (POST/SOAP):

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <csw:GetCapabilities
      xmlns:csw="http://www.opengis.net/cat/csw/2.0.2"
      service="CSW" xmlns="http://www.opengis.net/ows">
    </csw:GetCapabilities>
  </soap:Body>
</soap:Envelope>
```

- **Aufgabe: Selber machen!**

GetCapabilities - Response

- ServiceIdentification
 - Titel, Stichworte, Version, ...
- ServiceProvider
 - Ansprechpartner für den Dienst → Konfiguration in NOKIS
- OperationsMetadata
 - Unterstützte Operationen
 - Endpoints für die verschiedenen Protokolle
- Filter_Capabilities
 - Queryables
 - Unterstützte Operationen

GetDomain-Request und -Response

- GetDomain liefert eine Angabe über den tatsächlich verwendeten Wertebereich eines Queryables
- Beispiel: Stichwort
Eine GetDomain-Anfrage für das Queryable Stichwort (keyword) würde eine Liste der in allen Metadaten verwendeten Stichworten zurückliefern
- In Preludio (und in den meisten anderen Katalogen) nicht implementiert

DescribeRecord-Request

- Beschreibt die Struktur der in der Serverinstanz vorhandenen Metadatentypen
- Parameter
 - SERVICE (1..1) – Fester Wert: CSW
 - REQUEST (1..1) - Fester Wert: DescribeRecord
 - VERSION (1..1) – Fester Wert: 2.0.2
 - NAMESPACE (0..1) – Optionale Möglichkeit Namespaces zu definieren
 - TypeName (0..1) – Liste von qualifizierten Typbezeichnungen (z.B. gmd:MD_Metadata)
 - outputFormat (0..1) – MIME-Type, Default ist application/xml
 - SchemaLanguage (0..1) – Default-Wert ist XMLSCHEMA
- Response-Dokument ist ein XML-Schema für den angefragten Typen
- → **Aufgabe: DescribeRecord für nokis:project**

GetRecordById-Request und Response

- Liefert die Metadatenätze mit den angefragten Identifiern zurück
- Parameter
 - REQUEST (1..1) - Fester Wert: GetRecordById
 - ElementSetName (0..1) – Full, Summary, Brief, Default ist Summary
 - outputFormat (0..1) – MIME-Type, Default ist application/xml
 - outputSchema (0..1) – Referenz auf das bevorzugte Schema für die Antwort, Defaultwert hängt vom Server ab (z.B. <http://www.opengis.net/cat/csw/2.0.2>)
 - Id (1..*) - Liste der angefragten IDs (kommasepariert bei GET, wiederholbares Element bei POST-XML)
- Response-Dokument ist eine Liste von Metadatenätzen im angefragten Format
- → **Aufgabe: Abfrage des Servers mit einer GetRecordById-Anfrage per GET**
- → **Aufgabe: Abfrage des Servers mit einer GetRecordById-Anfrage per POST**

GetRecords-Request

- Zentrale Operation in einem nicht transaktionalen CSW-Service
- Komplexeste Operation aufgrund der Filtermöglichkeiten
- Parameter:
 - REQUEST, SERVICE, VERSION (Feste Werte, alle 1..1)
 - NAMESPACE (0..1) – Optionale Möglichkeit Namespaces zu definieren
 - resultType (0..1) – hits, results, validate, Defaultwert ist hits
 - requestId (0..1) – optionale Möglichkeit für den Klienten den Request mit einer ID zu versehen
 - outputFormat (0..1) – MIME-Type, Default ist application/xml → andere Formate sind möglich!
 - outputSchema (0..1) – Referenz auf das bevorzugte Schema für die Antwort, Defaultwert hängt vom Server ab (z.B. <http://www.opengis.net/cat/csw/2.0.2>)
 - startRecords (0..1) – Parameter zur Unterstützung des abschnittsweisen Abrufs von Metadaten
 - maxRecords (0..1) – maximale Anzahl der zurückzuliefernden Ergebnisse
 - ... Fortsetzung auf der nächsten Folie ...

GetRecords-Request (2)

- Parameter:
 - typeNameNames (1..1) – Liste von qualifizierten Typbezeichnungen (z.B. gmd:MD_Metadata)
 - ElementSetName (0..1) – Full, Summary, Brief, Default ist Summary
 - ElementName (0..1) – Möglichkeit, nur ganz bestimmte Elemente zurückzuliefern (XPath)
 - CONSTRAINTLANGUAGE (0..1) – CQL_TEXT oder FILTER
 - Constraint (0..1) – Filterausdruck
 - SortBy (0..1) – Angabe, nach welchem Attribut sortiert werden soll
 - DistributedSearch (0..1) – Boolean, gibt an, ob eine verteilte Suche durchgeführt werden soll
 - hopCount (0..1) – Anzahl der zu durchsuchenden Hierarchien
 - ResponseHandler (0..1) – Handler für asynchrone Requests
- Filter Encoding → wird später behandelt
- Aufgaben: in Zusammenhang mit Filter Encoding

CSW-T: Management-Operationen - Transaction

- Transaction-Operation erlaubt die Manipulation des Katalogs über die CSW-Schnittstelle
- Mögliche Operationsarten
 - Insert
 - Einfügen eines neuen Metadatensatzes in den Katalog
 - Für diese Operation kann eines der unterstützten Schemata verwendet werden
 - Es wird ein vollständiges Metadatum im Request spezifiziert
 - Update
 - Aktualisierung eines vorhandenen Metadatensatzes im Katalog
 - Für diese Operation kann eines der unterstützten Schemata verwendet werden
 - Request enthält ein Constraint-Element, das die zu aktualisierenden Datensätze angibt
 - Beispiel: Änderung der Freigabeebene (momentan einzige Property, die von Preudio unterstützt wird)
 - Delete
 - Löschen eines Metadatensatzes aus dem Katalog
 - Request enthält ein Constraint-Element, das die zu aktualisierenden Datensätze angibt

CSW-T: Management-Operationen - Harvest

- "Harvesten": Integrieren von Metadaten eines Katalogs in einen anderen
- Möglich ist
 - "manuelles" Harvesten, durch die Konfiguration eines Katalogs in einem übergeordnetem System
 - Automatisches Harvesten über die CSW-Operation Harvest
- CSW-T-Operation Harvest:
 - Synchrones Harvesten
 - Asynchrones Harvesten
- ... Ergebnisse der Projektarbeitsgruppe?

CSW 2.0.2 - AP ISO 1.0

- Anwendungsprofil für CSW 2.0.2
- Festlegung auf HTTP als Protokoll-Binding
- Festlegung des Inhaltsmodells (ISO 19115/19119)
- Festlegung der Operationen
- Erweiterungen des Inhaltsschemas
 - Handhabung von Diensten
 - Zusätzlicher ScopeCode application
- Zusätzliche Queryables

INSPIRE Metadatenregulierungen

- INSPIRE Metadatenprofil basiert auf AP ISO 1.0
- Zusätzliche Anforderungen im Inhaltsmodell
 - Schemabasiert
 - Regelbasiert (bedingte Anforderungen)
- Zusätzliche Queryables
- Wesentliche Dokumente:
 - Directive 2007/2/EC of the European Parliament and of the Council as regards metadata
 - INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119
 - Technical Guidance for the implementation of INSPIRE Discovery Services

Ausblick: CSW 3.0

OGC Filter Encoding für CSW

Filter Encoding: Einführung

- Filter Encoding ist die Abfragesprache für CSW, WFS, ...
- Vorschrift für das Encoding der CQL – Common Query Language des OGC
- Nicht spezifisch für ISO 19115/ISO 19119!
- Ein Filter besteht aus
 - Ausdrücken
 - PropertyName
 - Literal
 - Operatoren
 - Räumlich
 - Logisch
 - Vergleichend
- CSW 2.0.2 und AP ISO 1.0 benutzen nur eine Untermenge der Möglichkeiten, die die Filter Encoding Spezifikation bietet

Filter Encoding: Beispiel

```
<ogc:Filter xmlns:gml="http://www.opengis.net/gml">
```

```
<ogc:And>
```

Logischer Operator

```
<ogc:PropertyIsLike wildCard="*" singleChar="#" escapeChar="!">
```

```
<ogc:PropertyName>soap:Title</ogc:PropertyName>
```

```
<ogc:Literal>Bundeswasserstra*</ogc:Literal>
```

```
</ogc:PropertyIsLike>
```

Ausdruck: PropertyName

```
</ogc:And>
```

Ausdruck: Literal

```
</ogc:Filter>
```

Vergleichsoperator

Filter Encoding: Ausdrücke

- Eigenschaftsbezeichnungen: `PropertyName`
 - Logischer Name für eine Eigenschaft des Metadatums
 - Mögliche `PropertyNames` werden im `GetCapabilities`-Dokument angegeben
- Wert: `Literal`
 - Einzig möglicher Elementname für den Wert, gegen den verglichen wird
 - Hintergrund: `FilterEncoding` lässt auch arithmetische Operatoren und Funktionen zu:
`<Add>`, `<Sub>`, `<Mul>`, `<Div>`, `<Function>`
 - Die Formatierung des Inhalts des `Literal`-Elements wird nicht vom `Filter Encoding` spezifiziert!

Filter Encoding: Operatoren (Minimalanforderung AP ISO 1.0)

- Räumlich
 - Intersects
 - Disjoint
 - BBOX
- Logisch
 - And
 - Or
 - Not
- Vergleichend
 - EqualTo
 - NotEqualTo
 - LessThan
 - GreaterThan
 - LessThanOrEqualTo
 - GreaterThanOrEqualTo
 - Like
 - NullCheck

→ Für die Beschreibung der möglichen Operatoren siehe Filter Encoding Specification

GetRecords + Filter Encoding

NOKIS CSW im Detail

NOKIS CSW im Detail

- Überblick
 - Schnittstellen
 - Testklienten
- Besonderheiten
 - "internes" Schema
 - Servicebeschreibung
- Technik
 - Persistenzschicht
 - CSW-Queryables

CSW-Schnittstellen von NOKIS

- CSW 2.0.1 → veraltete Version!
 - CSW-Schnittstelle für das DE-Profil
 - CSW-Schnittstelle für das interne NOKIS-Format
 - Ausblick: Wenn wir die Rückmeldung bekommen, dass diese Version nicht mehr benutzt wird, dann wird die Schnittstelle in der nächsten NOKIS-Version entfallen
- CSW 2.0.2
 - CSW-Schnittstelle für AP ISO 1.0
 - CSW-Schnittstelle für die internen Formate
 - Ein gemeinsamer Endpoint für beide Dienste

CSW-Testkunden von NOKIS

- Webinterface
 - Unterstützt nur CSW 2.0.1
 - Falls der aktuelle CSW-Testclient nicht in die Oberfläche integriert werden kann, dann wird er in der nächsten NOKIS-Version aus der Oberfläche entfernt
- CSW-Testclient
 - Unterstützt nur CSW 2.0.2
 - Zahlreiche Testbeispiele
 - Benutzerdokumentation
 - Umfangreiche Loggingmöglichkeiten
 - Falls sich ein Partner zur Finanzierung findet, würden wir diesen Klienten gerne in die Oberfläche integrieren

Das "interne" Schema von NOKIS

- Warum ein "internes" Schema?
- Beispiel: Encoding des Elements CI_ResponsibleParty.OrganisationName
- ISO 19139:

```
<gmd:organisationName>  
  <gco:CharacterString>  
    disy Informationssysteme GmbH  
  </gco:CharacterString>  
</gmd:organisationName>
```
- NOKIS:

```
<base:organisationName>  
  disy Informationssysteme GmbH  
</base:organisationName>
```
- Probleme mit dem ISO-Format
 - "unnötige" Komplexität
 - Durch die Persistenzstrategie von NOKIS käme es zu einer massiven Erhöhung der Anzahl an Tabellen im Datenbankschema

Servicebeschreibung

- Aktuelle Situation in NOKIS:
 - Geometadatum → MD_Metadatum mit MD_DataIdentification
 - Servicebeschreibung → SV_ServiceIdentification
 - Geometadatum mit Dienstbeschreibung enthält MD_DataIdentification **und** SV_ServiceIdentification → konform zur Festlegung für Tightly Coupled Services in AP ISO 1.0
- Festlegung AK Metadaten
 - Geometadatum → MD_Metadatum mit MD_DataIdentification
 - Servicemetadatum → MD_Metadatum mit SV_ServiceIdentification
 - Verknüpfung über operatesOn und die ID von MD_DataIdentification (!) im Geometadatum
 - → Umgestaltung des Dienstprofils in NOKIS notwendig

Offene Fragen

Offene Fragen: Jörn

- a) Wenn es nicht gehen sollte, warum kann mdi-sh die Metadaten nicht lesen/erhalten?
- b) Wird die NOKIS Freigabe unterstützt?
- c) Welche Filtermöglichkeiten gibt es bei der Abfrage?
- d) Ist damit INSPIRE erfüllt oder was fehlt?

Offene Fragen: Gisbert

- Gibt es noch Einschränkungen, welche Felder abgefragt werden können?
Wenn ja, welche Felder können abgefragt werden?
- Gibt es andere CSW Bibliotheken als die Javabibliothek von Frank?
- Gibt es Performance Messungen zu CSW bei Nokis?
- Welche Outputs werden unterstützt (Brief, Full, Summary)?

Offene Fragen: Johannes Mielchen

- Aufbau des Referenzmodells für OGC Schnittstellen, anhand eines konkreten Szenarios (z.B. Abfrage von Schweinswalsichtungen beim NLWKN; wie müsste das Referenzmodell aufgebaut sein, was läuft dann technisch im Hintergrund ab, welche Funktion übernimmt der CSW?)
- Abfrage mit POST per Browser, ist dies möglich? wie wird das XML Dokument mit den Variablen übergeben?
- Abfrage einer CSW Schnittstelle mit Hilfe von POST und SOAP - Dokumentation

Offene Fragen: Christoph Wosniok/Michael Bauer

Das NOKIS Profil unterstützt mehrere Profile (Geometadaten, Projektmetadaten usw), für die unterschiedliche Elemente gelten. Diese Elemente sind alle in der (einen) getCapabilities aufgeführt. Eine Anfrage über die CS-W nach Elementen, die nicht für ein Profil gelten führt zu einem Fehler (Bsp: Abfrage von Abbreviation klappt nur bei Projekten, ist ja auch nur dafür in den getCap aufgeführt).

Zwei mögliche Lösungen:

- Für jedes NOKIS Profil (GeoMetadaten, Projektmetadaten, Dienste, Publikationen, Plattformen) einen eigenen Dienst anbieten.
- Namespaces in der getCap (z.B.
<queryable:nokisproject>abbreviation</queryable>

NOKIS CSW im Detail

- Überblick
 - Entwicklung der Schnittstellen und Inhalte
 - Vorhandene Funktionen und Werkzeuge rund um die NOKIS CSW
- Besonderheiten
 - Zusätzliche Metadatentypen
 - Parallelbetrieb von NOKIS (CSW 2.0.1), DE-Profil (CSW 2.0.1), NOKIS (CSW 2.0.2) und AP ISO 1.0 (CSW 2.0.2)
 - Queryables
- Technik
 - Persistenzschicht in NOKIS
 - Queryables in NOKIS
 - Welche Queryables gibt es?
 - Wie sieht die Implementierung dazu aus?

Vielen Dank für Ihre Aufmerksamkeit

Carsten Heidmann

disy Informationssysteme GmbH

Erbprinzenstr. 4–12

76133 Karlsruhe

Tel.: +49 721 1 6006-258

Fax: +49 721 1 6006-05

E-Mail: carsten.heidmann@disy.net

www.disy.net

