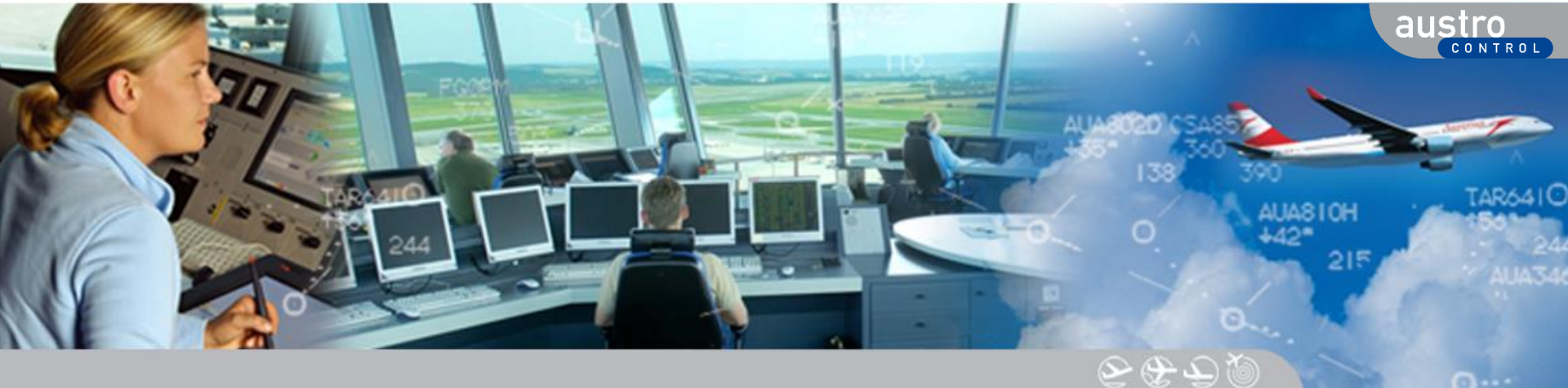
INSPIRE Workshop 2020

INSPIRE Netzdienste - ETF Validator, GeoServer, OGC API

Online, 17.12.2020

LFRZ

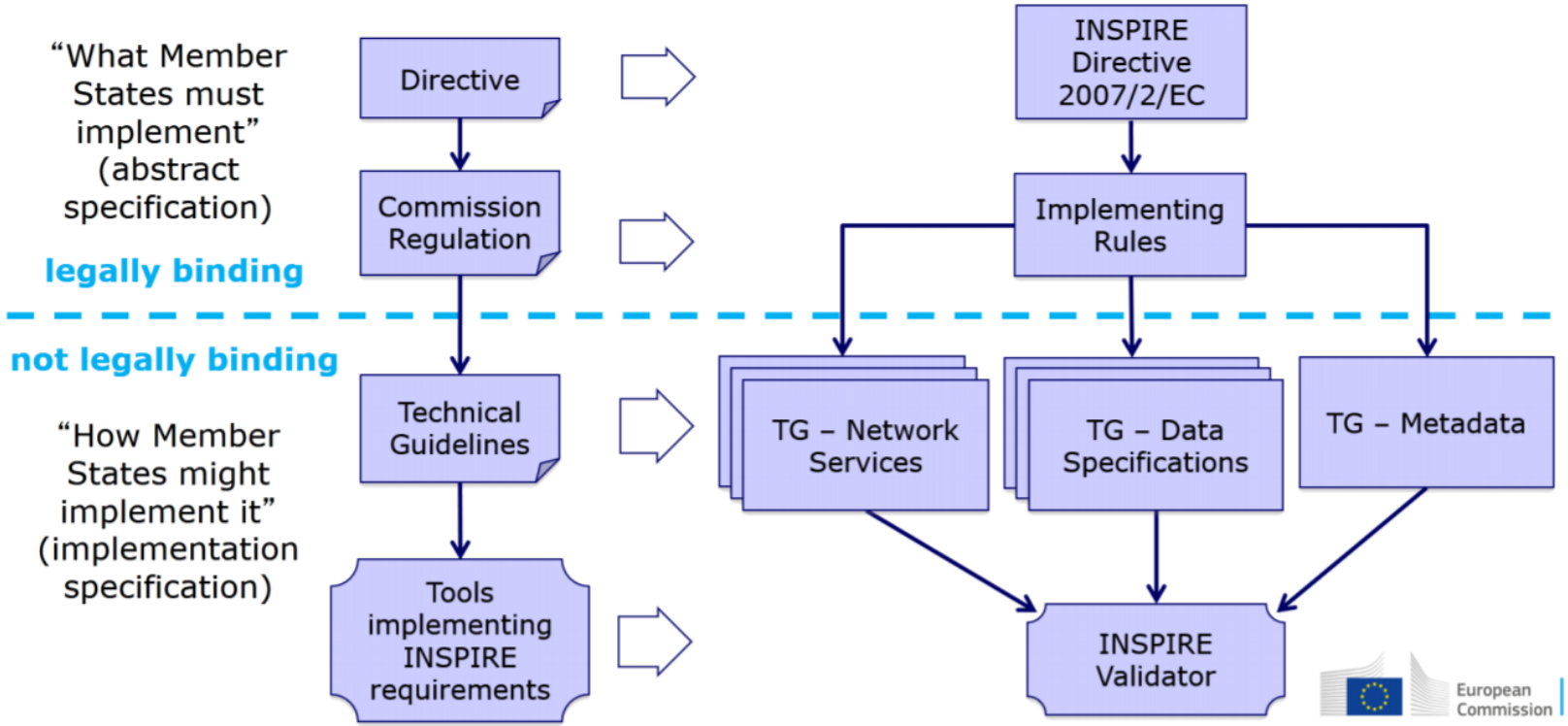


ETF Validator

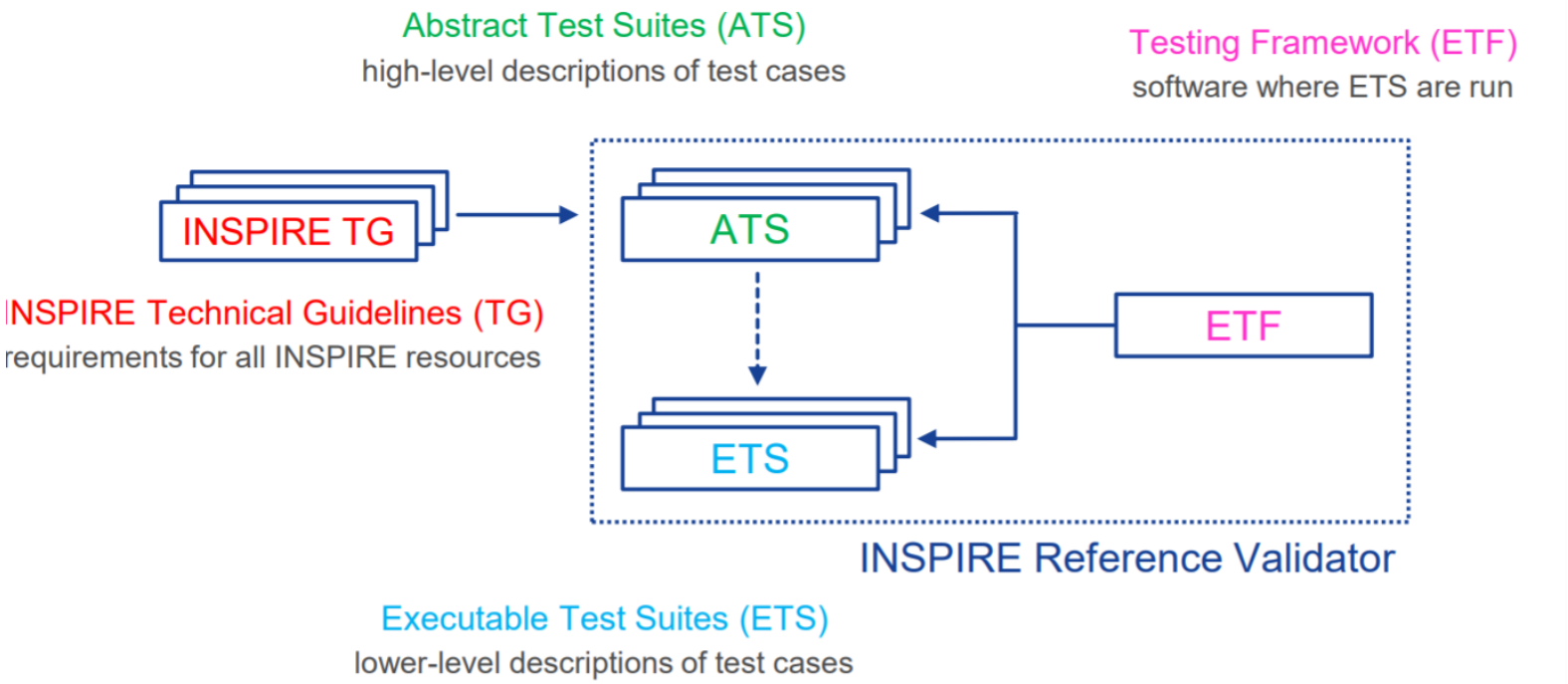
Inhalt

- Überblick
- Validator Docker Image Bestandteile
- Validator Docker Container Funktionsweise
- Ausblick

Einordnung des Validators



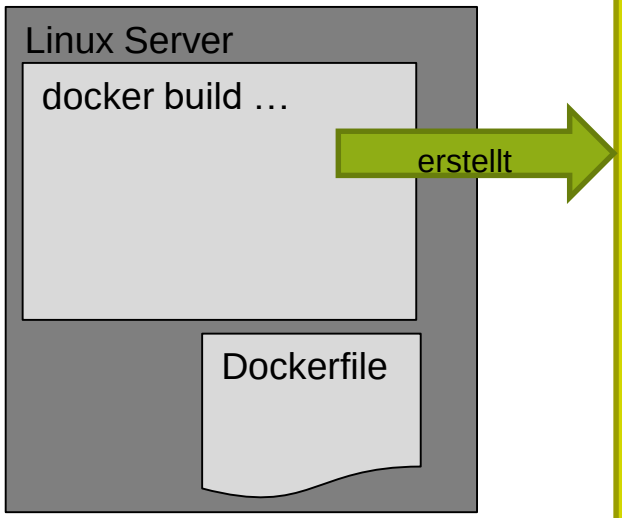
Bestandteile des INSPIRE Validators



Überblick INSPIRE ETF Validator

- Software zum Ausführen von ETS
  - Der INSPIRE Validator verwendet ETF und erweitert diese
    - ein Testframework zur Validierung von Daten, Metadaten und Webdiensten in SDIs
    - entwickelt seit 2010
    - Open Source unter EUPL v1.2
- Jedes ETF deployment besteht aus
  - einer Datenbank
  - einer oder mehrerer Test-Engines
  - einen Servlet-Container
- einfachster Weg um ETF bereitzustellen: Docker-Container

Docker Image - Bestandteile



erstellt

Validator Image

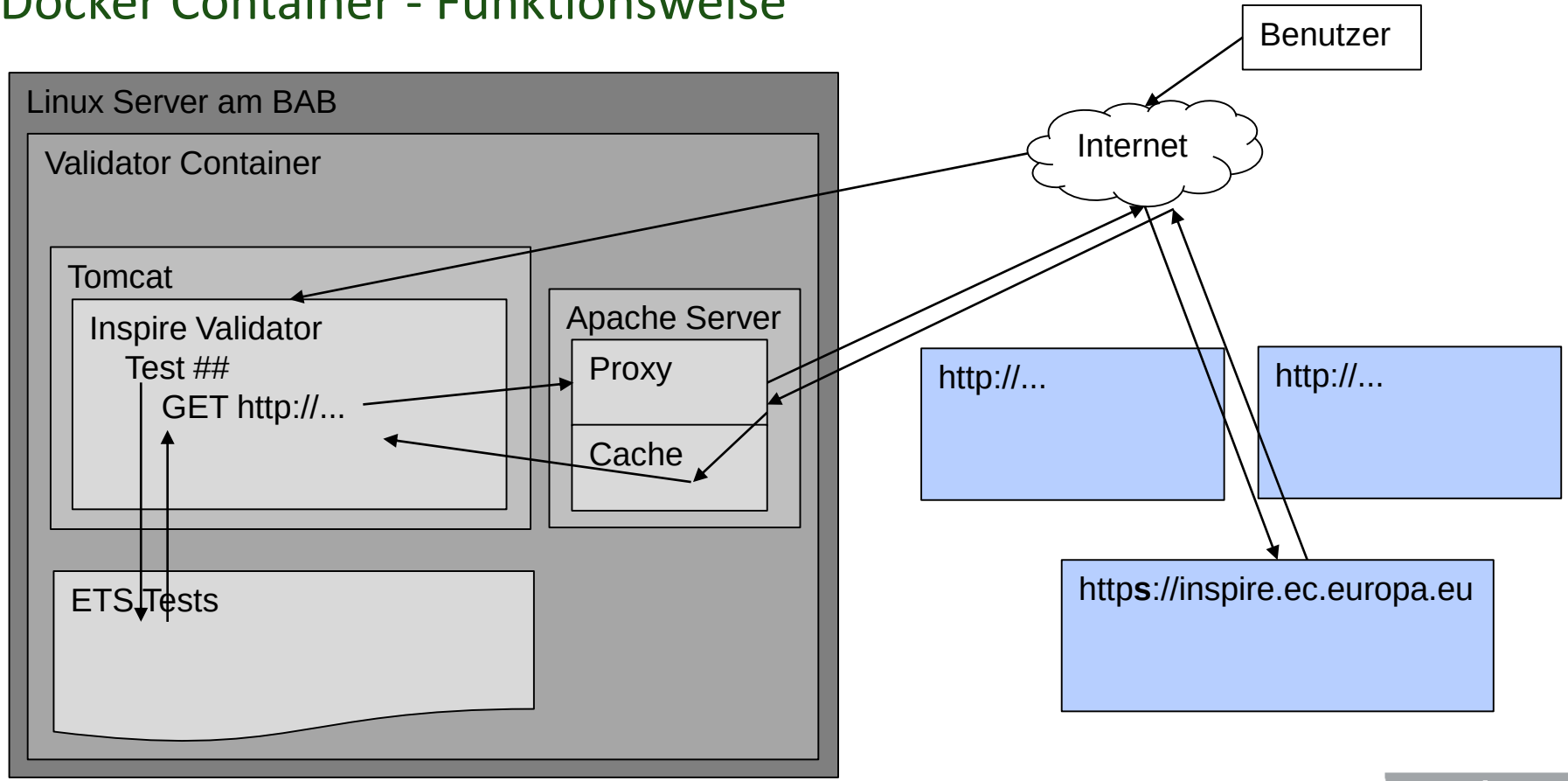
Basisimage:	OpenJDK (8u265-jdk-slim)
Tomcat:	v.8.5.58
JKS:	Zertifikat(e)
Tomcat:	Konfiguration (logs, Tomcat server, …)
Validator:	Konfiguration (Variables, dir's, scripts, …)
Apache:	Installation
Apache:	Konfiguration (proxy, web cache)
setup.sh:	Validator (Installation, ETF und ETS)
Validator :	Konfiguration (Ö spezifisch, Filegrößen)
Rechte:	Zugriffsrechte auf Verzeichnisse und Files
Apache:	Module aktivieren, Apache starten
entrypoint :	Tomcat mit Parametern starten

Unterschiede offizielles Image

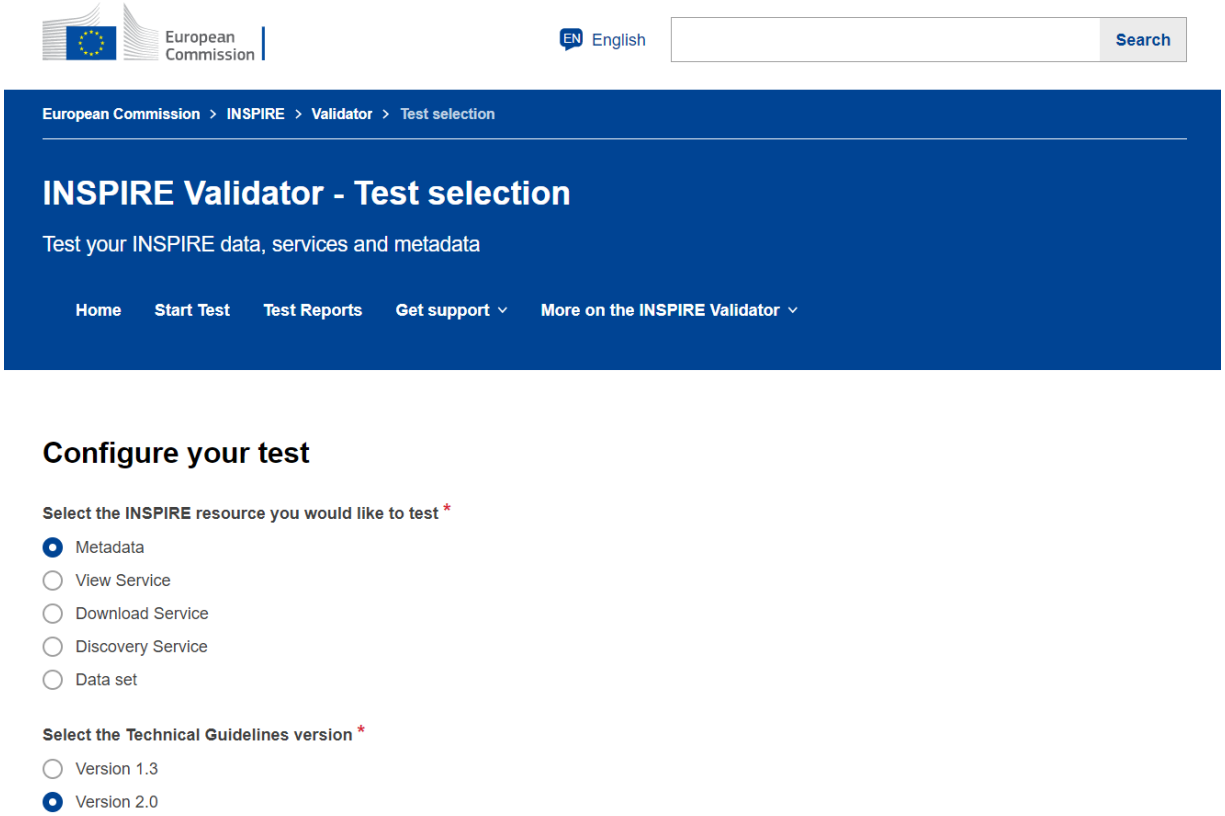
- Tomcat statt Jetty
- Squid (Caching) am Apache
- div. Configs

LFRZ austro CONTROL

Docker Container - Funktionsweise

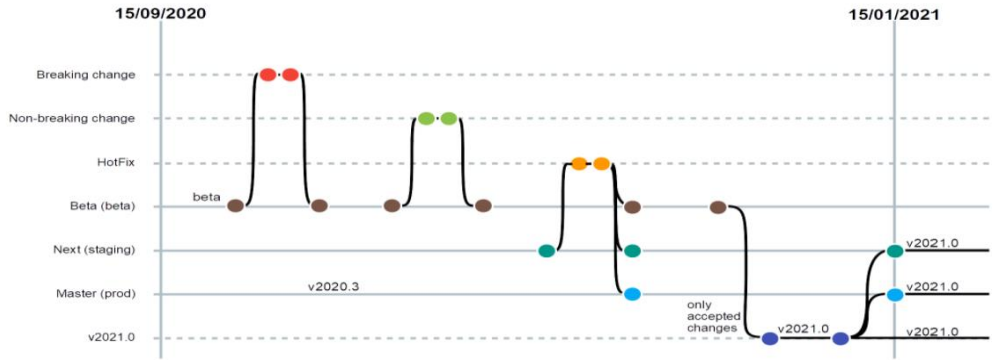


Vorschau neues GUI



Releases

- v2020.1 - 15/03/2020: enthält sowohl breaking als auch non-breaking Änderungen
- v2020.2 - 15/06/2020: enthält sowohl breaking als auch non-breaking Änderungen
- v2020.3 - 15/09/2020: enthält nur nicht non-breaking Änderungen, sodass jede INSPIRE-Ressource, die den Test in der vorherigen Version besteht, automatisch denselben Test in dieser Version besteht. (verwendet für Monitoring am 15.12.)
- v2021.b - 15/09/2020: Es enthält sowohl breaking als auch non-breaking Änderungen, die (zu Überwachungszwecken) im folgenden Jahr wirksam werden sollen.
- v2021.0 - 15/01/2021: Es enthält sowohl breaking als auch non-breaking Änderungen, einschließlich der in der Beta-Instanz des Vorjahres verfügbaren



Hilfreiches, Links

- ETF Validator: http://etf-validator.net/
- Validation Community github
  - https://github.com/inspire-eu-validation/community/
  - https://github.com/inspire-eu-validation/community/tree/master/release%20strategy
  - https://github.com/inspire-eu-validation/community/releases
  - https://github.com/inspire-eu-validation/community/tree/master/training%20material
  - https://github.com/inspire-eu-validation/community/tree/master/examples
- Validator Instanzen
  - AT (BAB): https://inspire.agrarforschung.at/validator/
  - JRC Prod: https://inspire.ec.europa.eu/validator/
  - JRC Staging: http://staging-inspire-validator.eu-west-1.elasticbeanstalk.com/etf-webapp/
- Tipps
  - Reports sollten heruntergeladen werden falls benötigt
  - dzt. noch kein persistenter Speicher im Docker (für Zukunft geplant)

GeoServer AppSchema

Inhalt

- Überblick App Schema
- Feature Chaining
- Direkte Assoziation mit multiplen Referenzen

GeoServer Application Schema Extension

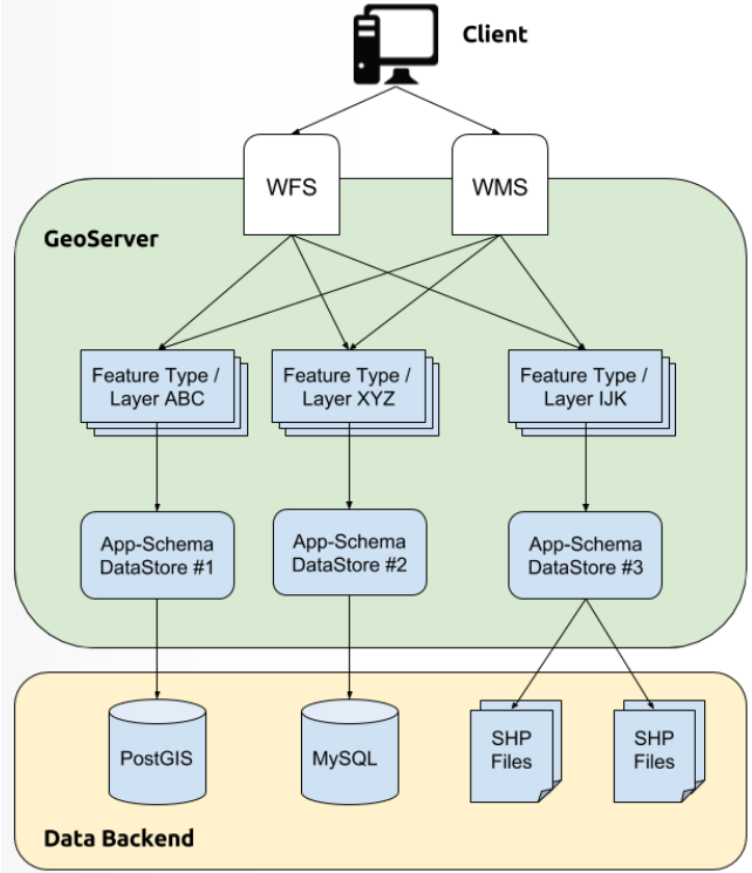
- GeoServer unterscheidet zwischen einfachen Features und komplexen Features
- Standardmäßig unterstützt GeoServer nur einfache Funktionen
- Im häufigsten Anwendungsfall ist GeoServer mit einer Datenquelle wie einer relationalen Datenbank verbunden, und jede Tabelle mit geografischen Daten wird automatisch einem einfachen Feature-Typ zugeordnet.
- Vorteile ist die einfache Implementierung und Leistung
- Nachteile: Da das Format der automatisch generierten XSD an das Datenbankschema gebunden ist, müssen User entweder dieselbe Datenbankstruktur verwenden, um den Informationsaustausch zu ermöglichen Schema oder müssen zwischen Schemata übersetzen

GeoServer Application Schema Extension

- durch Verwendung von App Schema im GeoServer können komplexe Features ein- oder mehrwertige Eigenschaften haben
- komplexe Features stellen Informationen nicht als XML-Ansicht einer einzelnen Tabelle dar, sondern als Sammlung verwandter Objekte unterschiedlichen Typs
- für die Erreichung von Interoperabilität notwendig
- Bsp.
  - Ein Tabelle enthält zwei Fremdschlüssel
  - simple Feature Ansatz
    - Erstellung einer großen view mit den 3 Tabellen
    - Darstellung als flaches XML
  - komplex Feature Ansatz
    - verschiedene Konzepte werden getrennt gehalten
    - Modellierung unterschiedlicher, miteinander verbundener Entitäten im Application Schema

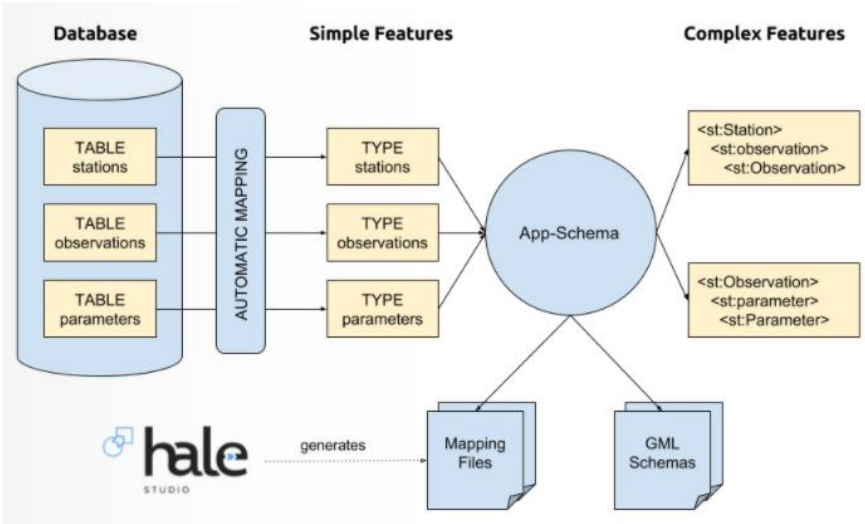
GeoServer Application Schema Extension

- Für GeoServer sieht das App-Schema-Modul wie jeder andere Datenspeicher aus und kann daher geladen und zur Bearbeitung von WFS- und WMS-Anforderungen verwendet werden.



GeoServer Application Schema Extension

- Tatsächlich ist der App-Schema-Datenspeicher ein Wrapper oder Adapter, der einfache Features aus einem oder mehreren einfachen Feature-Datenspeichern liest und sie basierend auf einer benutzerdefinierten Zuordnungskonfiguration zu komplexen Features zusammenfügt.
- Die Zuordnung funktioniert in beide Richtungen, sodass Abfragen für Eigenschaften komplexer Features unterstützt und nach Möglichkeit in native Abfragen der Sicherungsdatenquelle (z. B. SQL-Abfragen) übersetzt werden.



GeoServer Application Schema Extension

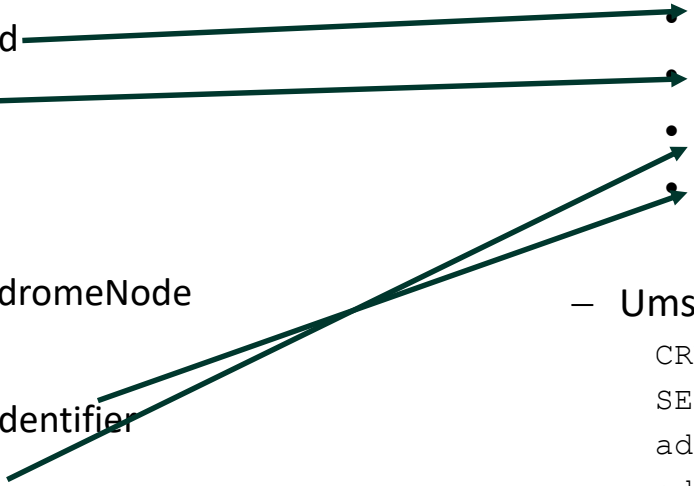
- Die Zuordnungskonfiguration wird in einer oder mehreren XML-Dateien angegeben
- Die wichtigsten Abschnitte einer Zuordnungsdatei sind:
  - Der Abschnitt mit den **Namespaces** listet alle in der Zuordnungsdatei verwendeten XML-Namespaces und die Präfixe auf, denen sie zugeordnet sind.
  - Der Abschnitt **sourceDataStores** enthält Konfigurationseinstellungen für jeden Datenspeicher, aus dem Eingabedaten gelesen werden sollen. Der Inhalt jedes DataStore-Eintrags besteht im Wesentlichen aus einer Liste benannter Verbindungsparameter.
  - Der Abschnitt **targetTypes** listet alle Anwendungsschemata auf, die zum Definieren der Zuordnung erforderlich sind. In der Regel ist nur eine erforderlich.
  - Der Abschnitt **typeMappings**, ist der wichtigste Teil des App-Schema-Moduls. Es definiert die Zuordnung von einfachen Features zur verschachtelten Struktur eines oder mehrerer komplexer Features. Es besteht aus einer Liste von FeatureTypeMapping-Elementen, die jeweils einen komplexen Ausgabe-Feature-Typ definieren.

Feature Chaining

- Ein Feature enthält mehrere Referenzen zu Features EINES anderen FeatureTypes (1..*)
- Beispiel:
  - Einem AerodromeType werden Referenzen mehrerer AerodromeNodes zugewiesen
  - Der AerodromeType = „aerodromeOnly“ enthält dann nur Referenzen auf AerodromeNodes dieses einen Typs
  - Referenz ist der gml_identifier des jeweiligen AerodromeNodes als href-Element
  - Der AerodromeType mit der gml_id=1 enthält dann die gml_identifier der Aerodromes (z.B: 6, 10, usw.).
- Vorgehensweise
  - Vorbereitung in der Datenbank mit einer eigenen View für ein interimistisches Feature
  - Umsetzung im AppSchema im GeoServer mit Hilfe des interimistischen Features

Feature Chaining – DB View für interim. Feature

- table: AerodromeType
  - columns
    - gml_id
    - typ
    - …
- table: AerodromeNode
  - columns
    - gml_identifier
    - typ
    - …

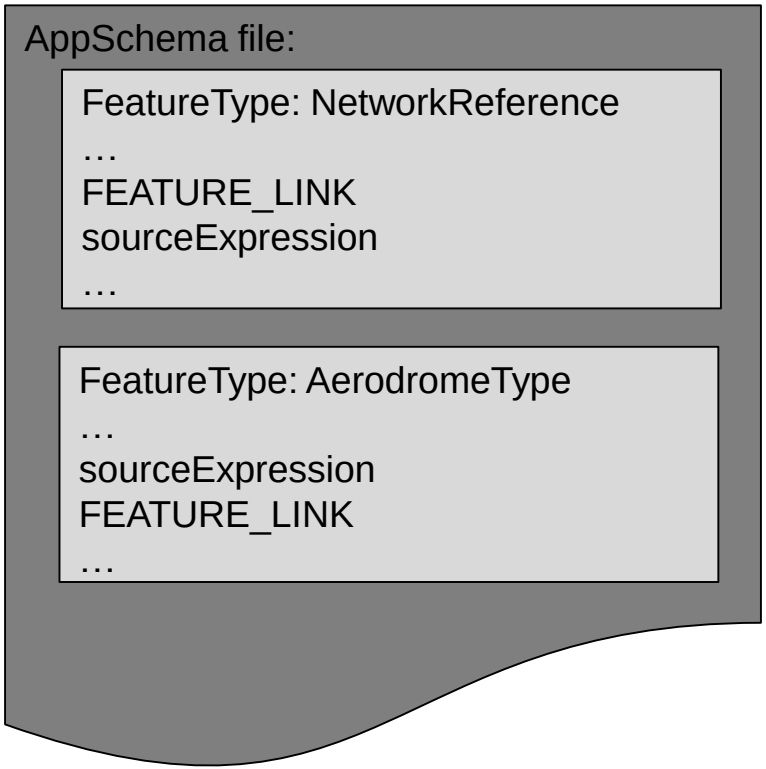


- view: NetworkReference
  - columns:
    - property_gml_id
    - propertytyp
    - elementtyp
    - elementhref
- Umsetzung mit SQL

```
CREATE OR REPLACE VIEW networkreference AS
SELECT adt.gml_id AS property_gml_id,
adt.typ AS propertytyp,
adn.typ AS elementtyp,
adn.gml_identifier AS elementhref
FROM aerodromenode adn JOIN aerodrometype adt
ON and.typ=adt.typ
```


LFRZ austro CONTROL

Feature Chaining – AppSchema (1)



- Interim. Feature im GS AppSchema
  - enthält FEATURE_LINK (fix) und
  - sourceExpression
    - Inhalt, der im zu erstellenden Feature eingefügt werden soll
    - merhmals da 1:n

- Im zu erstellenden Feature (hier: AerodromeType) wird das entsprechende Element über die sourceExpression angesprochen und der Teil aus dem interim. Feature eingefügt, der auf FEATURE_LINK zeigt

LFRZ austro CONTROL

Feature Chaining – AppSchema (2)

- view: NetworkReference
  - columns:
    - **property_gml_id**
    - propertytyp
    - elementtyp
    - **elementhref**

AppSchema file:

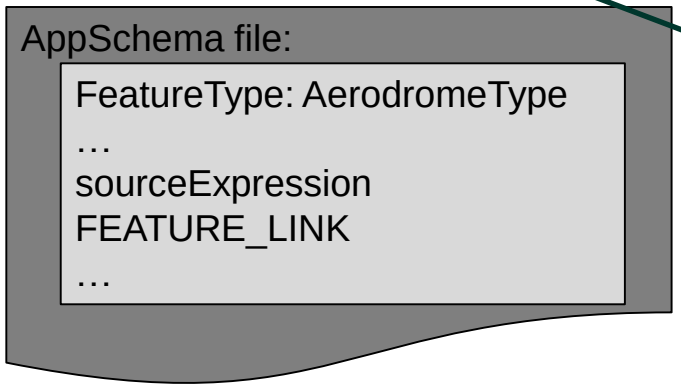
FeatureType: NetworkReference
…
FEATURE_LINK
sourceExpression
…

```
<FeatureTypeMapping>
    <sourceDataStore>dataStore</sourceDataStore>
    <sourceType>INSP_NETWORKREFERENCE</sourceType>
    <targetElement>net:NetworkReference</targetElement>
    <attributeMappings>
        <AttributeMapping>
            <targetAttribute>FEATURE_LINK</targetAttribute>
            <sourceExpression>
                <OCQL>PROPERTY_GML_ID</OCQL>
            </sourceExpression>
        </AttributeMapping>
        <AttributeMapping>
            <targetAttribute>net:element</targetAttribute>
            <encodeIfEmpty>true</encodeIfEmpty>
            <ClientProperty>
                <name>xlink:href</name>
                <value>ELEMENT_HREF</value>
            </ClientProperty>
        </AttributeMapping>
    </attributeMappings>
</FeatureTypeMapping>
```

FEATURE_LINK ist vom GeoServer fix vorgegeben
PROPERTY_GML_ID und **ELEMENT_HREF** kommen aus der DB View

Feature Chaining – AppSchema (3)

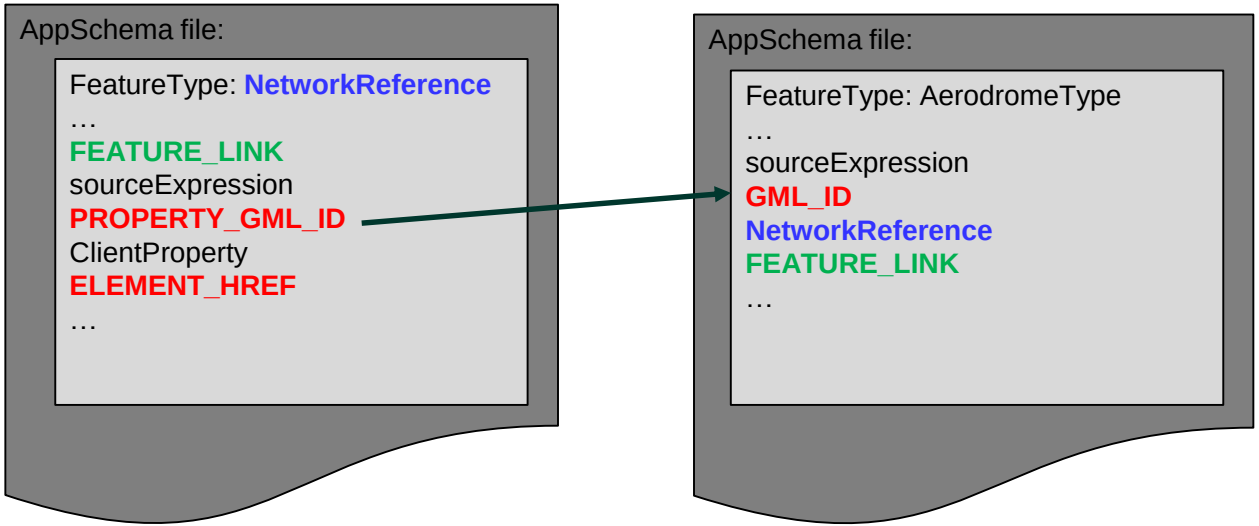
- table: AerodromeType
  - columns:
    - **gml_id**
    - typ
    - …




```
<FeatureTypeMapping>
    <sourceDataStore>dataStore</sourceDataStore>
    <sourceType>INSP_AERODROMETYPE</sourceType>
    <targetElement>tn-a:AerodromeType</targetElement>
    <attributeMappings>
        <AttributeMapping>
    …
        </AttributeMapping>
        <AttributeMapping>
        <targetAttribute>net:networkRef</targetAttribute>
            <sourceExpression>
                <OCQL>GML_ID</OCQL>
                <linkElement>net:NetworkReference</linkElement>
                <linkField>FEATURE_LINK</linkField>
            </sourceExpression>
        </AttributeMapping>
    </attributeMappings>
</FeatureTypeMapping>
```

- **FEATURE_LINK** ist vom GeoServer fix vorgegeben
- **GML_ID** kommt aus der DB table
- **NetworkReference** ist der Verweis auf das interim. Feature

Feature Chaining – AppSchema (4)



- Wenn GML_ID (vom AerodromeType) gleich dem PROPERTY_GML_ID ist, wird das Element im Feature AerodromeType durch ALLE Elemente mit derselben PROPERTY_GML_ID aus dem interim. NetworkReference Feature ersetzt
- Damit bekommt ein AerodromeType n Verweise auf die AerodromeNodes

Feature Chaining – Ergebnis (WFS)

```
    <wfs:member>
        <tn-a:AerodromeType gml:id="AT.0012.6bed1778-d6bf-11e8-9f8b-f2801f1b9fd1.tn-a.AerodromeType.1">
            <gml:identifier codeSpace="http://inspire.jrc.ec.europa.eu/ids">https://inspire.austrocontrol.at/AT.0012.6bed1778-d6bf-11e8-9f8b-f2801f1b9fd1/tn-a.AerodromeType.1</gml:identifier>
            <net:networkRef>
                <net:NetworkReference>
                    <net:element xlink:href="https://inspire.austrocontrol.at/AT.0012.6bed1778-d6bf-11e8-9f8b-f2801f1b9fd1/tn-a.AerodromeNode.6"/>
                </net:NetworkReference>
            </net:networkRef>
            <net:networkRef>
                <net:NetworkReference>
                    <net:element xlink:href="https://inspire.austrocontrol.at/AT.0012.6bed1778-d6bf-11e8-9f8b-f2801f1b9fd1/tn-a.AerodromeNode.10"/>
                </net:NetworkReference>
            </net:networkRef>
…
            <tn-a:aerodromeType xlink:href="http://inspire.ec.europa.eu/codelist/AerodromeTypeValue/aerodromeOnly"/>
        </tn-a:AerodromeType>
    </wfs:member>
```

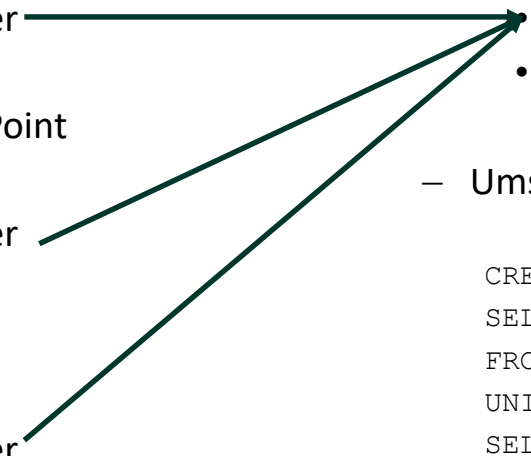

LFRZ austro CONTROL

Direkte Assoziation mit multiplen Referenzen

- Ein Feature enthält mehrere Referenzen zu mehreren anderen Features
- Beispiel:
  - Das Feature TransportNetwork soll Referenzen zu allen AerodromeNodes, DesignatedPoints, etc. (= Netzwerkfeatures) zugewiesen werden
  - Das Feature TransportNetwork (davon gibt es nur eines) enthält dann Referenzen auf alle AerodromeNodes, alle DesignatedPoints, etc.
  - Referenz ist der gml_identifier des jeweiligen AerodromeNodes, DesignatedPoints, etc.
- Vorgehensweise
  - Vorbereitung in der Datenbank mit einer denormalisierten View
  - Umsetzung im AppSchema im GeoServer mit Hilfe des “isMultiple” Flags

Direkte Assoziation – DB View denormalisiert

- table: AerodromeNode
  - columns:
    - gml_identifier
    - …
- table: DesignatedPoint
  - columns:
    - gml_identifier
    - …
- table: Navaid
  - columns:
    - gml_identifier
    - …

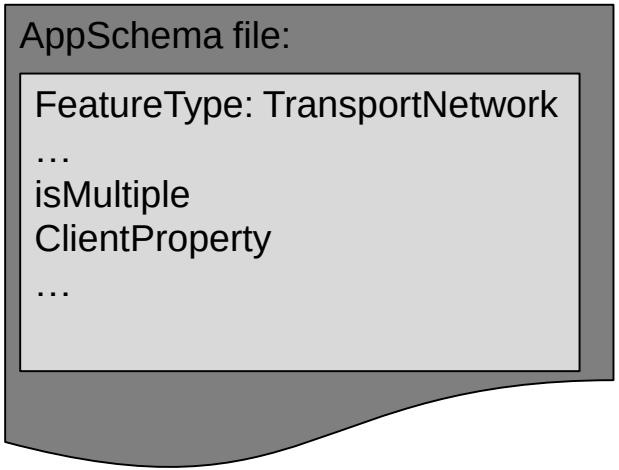


- view: TransportNetwork
  - columns:
    - element_href
    - …

- Umsetzung mit SQL und UNION verwenden:

```
CREATE OR REPLACE VIEW transportnetwork AS
SELECT gml_identifier AS element_href
FROM aerodromenode
UNION ALL
SELECT gml_identifier AS element_href
FROM designatedpoint
UNION ALL…
```

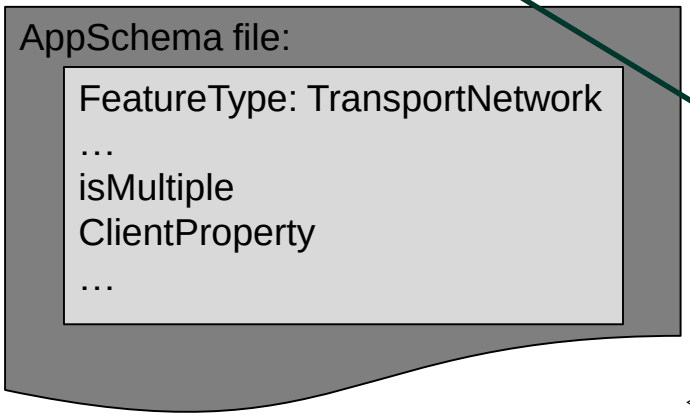
Direkte Assoziation – AppSchema (1)



- Das Flag “isMultiple” wird auf true gesetzt
- Damit erstellt GeoServer aus der DB View das Feature und gruppiert die im ClientProperty enthaltenen Attribute in das jeweilige Feature

Direkte Assoziation – AppSchema (2)

- view: NetworkReference
  - columns:
    - **element_href**
    - …




```
<typeMappings>
    <FeatureTypeMapping>
        <sourceDataStore>dataStore</sourceDataStore>
        <sourceType>INSP_TRANSPORTNETWORK</sourceType>
        <targetElement>tn:TransportNetwork</targetElement>
        <attributeMappings>
            <AttributeMapping>
                <targetAttribute>net:elements</targetAttribute>
                <encodeIfEmpty>true</encodeIfEmpty>
                <isMultiple>true</isMultiple>
                <ClientProperty>
                    <name>xlink:href</name>
                    <value>ELEMENT_HREF</value>
                </ClientProperty>
            </AttributeMapping>
        </attributeMappings>
    </FeatureTypeMapping>
</typeMappings>
```

- **isMultiple** Flag auf **true** setzen
- **ELEMENT_HREF** kommt aus dem DB View

LFRZ austro CONTROL

Direkte Assoziation – Ergebnis (WFS)

```
…
    <wfs:member>
        <tn:TransportNetwork gml:id="AT.0012.243b6e36-d14a-11e8-a8d5-f2801f1b9fd1.tn.TransportNetwork.1">
            <gml:identifier codeSpace="http://inspire.jrc.ec.europa.eu/ids">https://inspire.austrocontrol.at
/AT.0012.243b6e36-d14a-11e8-a8d5-f2801f1b9fd1/tn.TransportNetwork.1</gml:identifier>
…
            <net:elements xlink:href="https://inspire.austrocontrol.at/AT.0012.6bed1778-d6bf-11e8-9f8b-
f2801f1b9fd1/tn-a.DesignatedPoint.1173"/>
            <net:elements xlink:href="https://inspire.austrocontrol.at/AT.0012.6bed1778-d6bf-11e8-9f8b-
f2801f1b9fd1/tn-a.DesignatedPoint.805"/>
…
            <net:elements xlink:href="https://inspire.austrocontrol.at/AT.0012.6bed1778-d6bf-11e8-9f8b-
f2801f1b9fd1/tn-a.AerodromeNode.6"/>
            <net:elements xlink:href="https://inspire.austrocontrol.at/AT.0012.6bed1778-d6bf-11e8-9f8b-
f2801f1b9fd1/tn-a.AerodromeNode.61"/>
…
            <net:elements xlink:href="https://inspire.austrocontrol.at/AT.0012.6bed1778-d6bf-11e8-9f8b-
f2801f1b9fd1/tn-a.Navaid.ILS-GP_4"/>
…
            <tn:typeOfTransport>air</tn:typeOfTransport>
        </tn:TransportNetwork>
    </wfs:member>
```


LFRZ austro CONTROL

OGC API Features

Inhalt

- Überblick
- GeoServer
  - Installation
  - Admin GUI
  - Landing Page
  - Collections
  - REST API

OGC API

- Die OGC API-Standardfamilie wird entwickelt, um es jedem zu erleichtern, Geodaten für das Web bereitzustellen.
- Die Standards bauen auf OGC-Webdienststandards (WMS, WFS, WCS, WPS usw.) auf
- definieren ressourcenzentrierte APIs, die moderne Webentwicklungspraktiken nutzen
- Die einzelnen APIs werden nicht nur durch die Anforderungen der spezifischen Standards definiert, sondern durch Interoperabilitäts-Prototyping und -Tests im OGC-Innovationsprogramm
- Paradigmenwechsel mit der Einführung der OGC APIs
- Vereinfachung des Zugriffs auf verteilte Geodaten, sowie die einfachere Integrierbarkeit in beliebige Webanwendungen und Prozesse

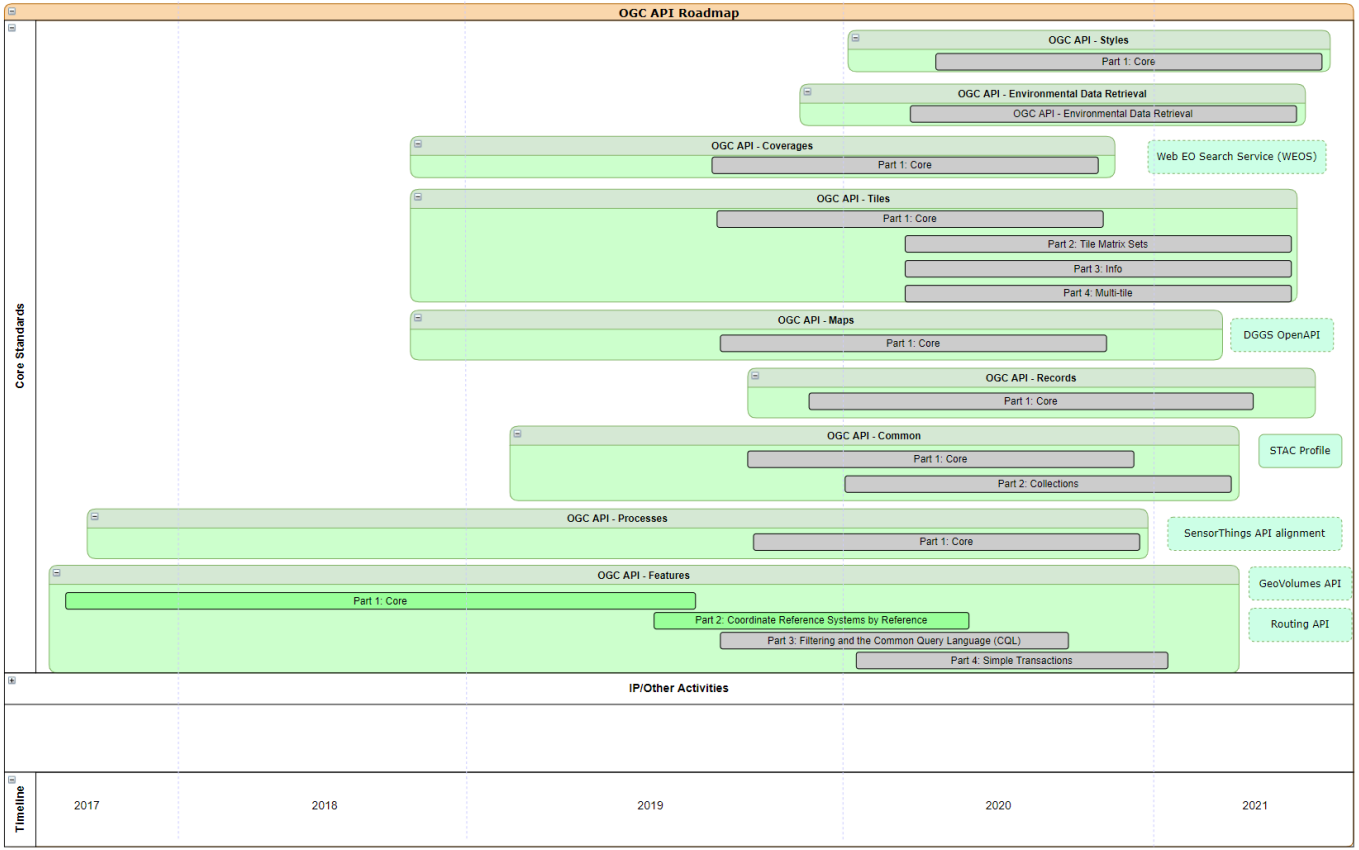
OGC API Standards

- Common (OWS Common Nachfolger)
- Features (WFS Nachfolger)
- Maps (WMS Nachfolger)
- Records (CSW Nachfolger)
- Processes (WPS Nachfolger)
- Coverages (WCS Nachfolger)
- Tiles (WMTS Nachfolger)
- Environmental Data Retrieval API
- zukünftig: Styles, Routing

OGC API Features

- OGC-API Features ist ein mehrteiliger Standard, der die Möglichkeit bietet, räumliche Daten im Web zu erstellen, zu ändern und abzufragen.
- Der Kernteil der Spezifikation beschreibt die obligatorischen Funktionen, die jeder implementierende Dienst unterstützen muss, und beschränkt sich auf den Lesezugriff auf räumliche Daten.
- Zusätzliche Funktionen, die auf bestimmte Anforderungen zugeschnitten sind, werden in zusätzlichen Teilen angegeben.
- Zu den geplanten zukünftigen Funktionen gehören beispielsweise die Unterstützung beim Erstellen und Ändern von Daten, komplexere Datenmodelle, umfangreichere Abfragen und zusätzliche Koordinatenreferenzsysteme.
- Teile:
  - OGC API - Features - Part 1: Core (Standard seit 2019/10)
  - OGC API - Features - Part 2: Coordinate Reference Systems by Reference (Standard seit 2020/11)
  - OGC API - Features - Part 3: Filtering and the Common Query Language (CQL)
  - OGC API - Features - Part 4: Simple Transactions

OGC API Roadmap



OGC API Roadmap

Progress of Official OGC Standards OGC & Community Standards Community 2020-12-17

Proposed Standards	SWG Work / Work Item	OAB Review	OGC-NA Review	Public Review	Prepare for Approval	TC Approval to Vote	TC Vote	PC Vote	Public Release		
OGC API - Common 19-072	✓	✓	✓	✓	✓	?					
OGC API - Coverages	?										
OGC API - Environmental Data Retrieval 19-086	✓	✓	✓	✓	✓	?					
OGC API - Features - Part 1: Core 17-069	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
OGC API - Features - Part 2: CRS by Reference 18-058	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
OGC API - Features - Part 3: Common Query Language 19-079	?										
OGC API - Features - Part 4: Simple Transactions 20-002	?										
OGC API - Features - Part 5: OpenAPI 3.1	?										
OGC API - Maps	?										
OGC API - Processes	✓	✓	✓	✓	✓	?					
OGC API - Records	?										
OGC API - Styles	?										
OGC API - Tiles	?										

LFRZ austro CONTROL

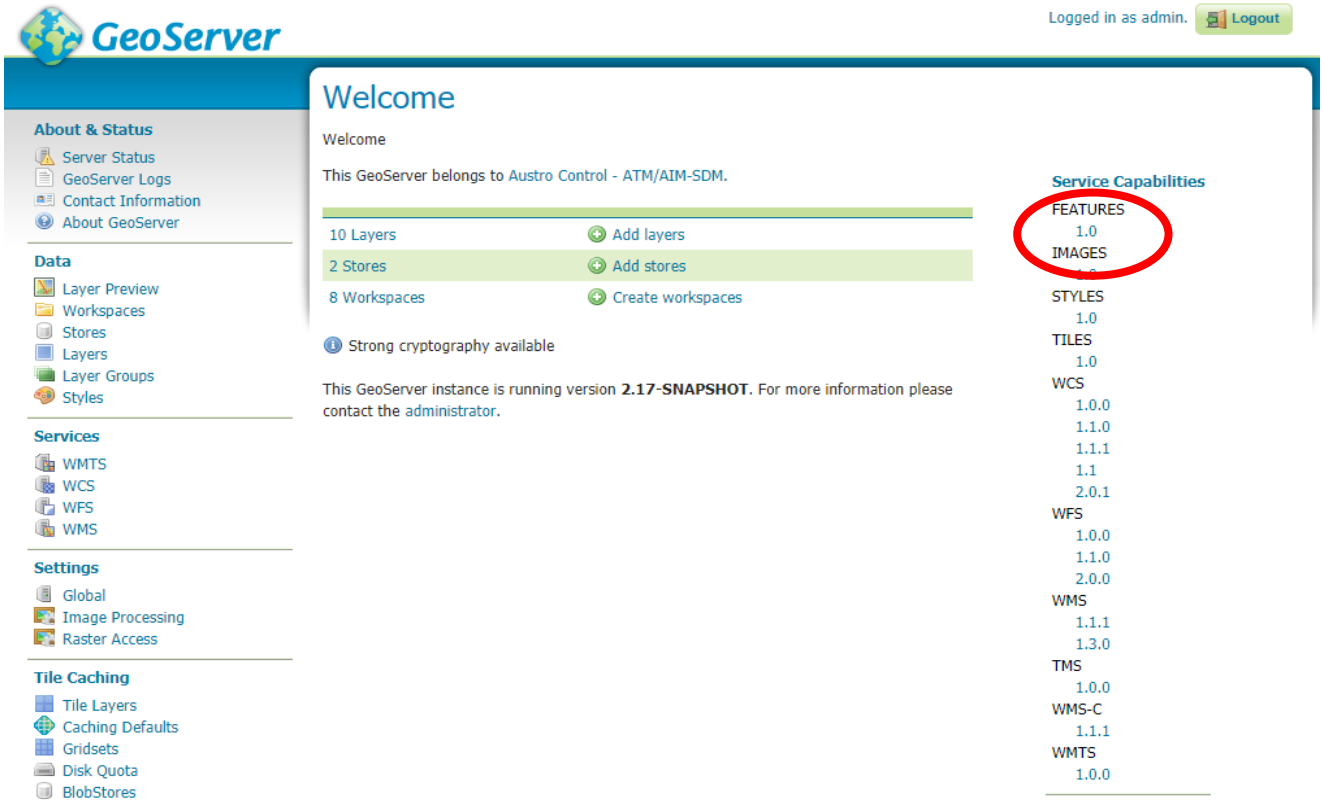
GeoServer – OGC API

- Voraussetzungen
  - AppSchema muss vorhanden sein
  - OGC API muss installiert sein
- OGC API installieren
  - Community Extension installieren
    - https://docs.geoserver.org/latest/en/user/community/ogc-api/index.html
    - „geoserver-2.19-SNAPSHOT-ogcapi-plugin.zip“
  - ODER Nightly build installieren
    - https://build.geoserver.org/geoserver/master/
    - „geoserver-master-latest-war.zip“
- Versionsnummer der Extension muss mit GS übereinstimmen
- OGC API in Geoserver noch in Entwicklung

GeoServer – OGC API

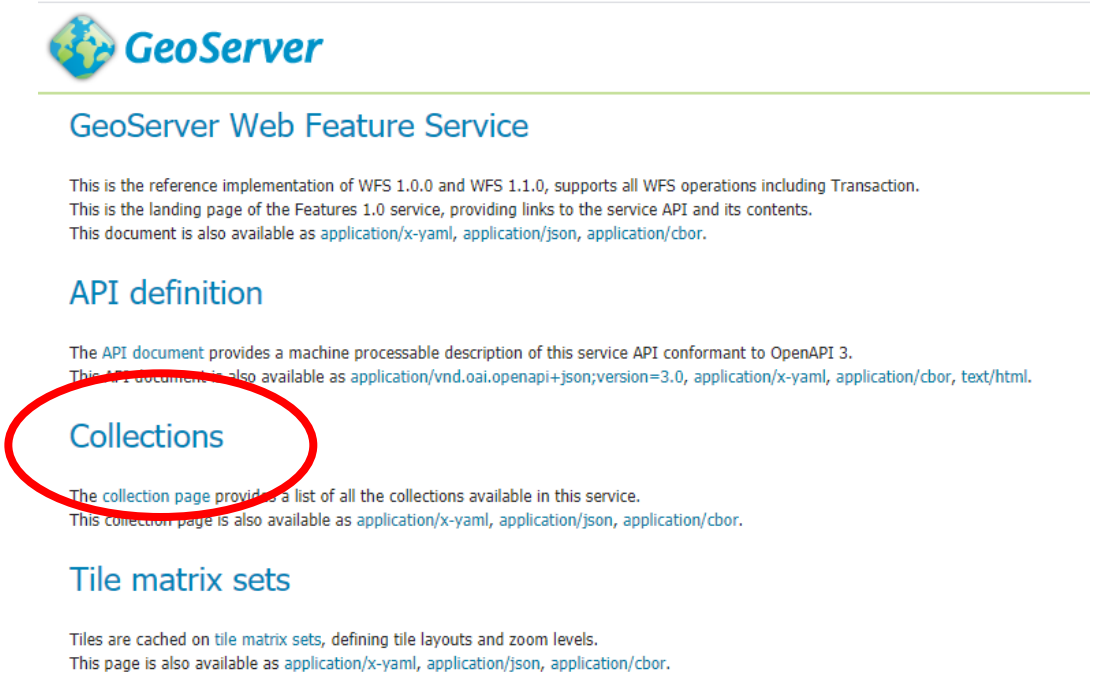
- Das Paket enthält
  - OGC Features API (inkl. draft CQL-Filtererweiterung und draft Multi-CRS-Erweiterung)
  - OGC-Tiles-API, für Vektor- und Rasterkacheln
  - OGC-Styles-API
  - zusätzliche APIs, basierend auf technischen Berichten von Testbed 15
- Konfiguration
  - keine zusätzliche Konfiguration
  - hängen von der Konfiguration der klassischen OGC-Dienste ab, die sie ersetzen möchten
  - könnte sich in Zukunft ändern

GeoServer – OGC API – Admin GUI



GeoServer – OGC API – Landing page

- OGC API – Landing page
  - http(s)://meinServer:Port/geoserver/ogc/features



GeoServer – OGC API – Collections

GeoServer

GeoServer Feature Collections

This document lists all the collections available in the Features service.
This document is also available as application/x-yaml, application/json, application/cbor.

tn-a:AerodromeCategory

- **Title**: Aerodrome Category
- **Description**: Aerodrome category concerning the scope and importance of the air traffic services offered from and to it.
- **Geographic extents**:
  - 8, 46, 18, 49.
- Data as HTML. Collection items are also available in the following formats: --Please choose an option--
- Queryables as HTML.

tn-a:AerodromeNode

- **Title**: Aerodrome Node
- **Description**: A node located at the aerodrome or heliport reference point. An aerodrome or heliport reference point is the designated geographical location of an aerodrome, heliport or landing location. An aerodrome reference point shall be established for an aerodrome. A heliport reference point shall be established for a heliport or a landing location not collocated with an aerodrome. The aerodrome or heliport reference point shall be located near the initial or planned geometric centre of the aerodrome, heliport or landing location and shall normally remain where first established. An aerodrome is a defined area on land or water (including any buildings, installations and equipment) intended to be used either wholly or in part for the arrival, departure and surface movement of aircraft. A heliport is an aerodrome or a defined area on a structure intended to be used wholly or in part for the arrival, departure and surface movement of helicopters. A landing location is a marked or unmarked area that has the same physical characteristics as a visual heliport final approach and take-off area (FATO). The attribute "code_type" distinguishes aerodromes 'AD' from heliports 'HP' but also markes aerodromes containing heliports 'AH'. The aerodrome elevation is the vertical distance between the highest point of the landing area of an aerodrome and mean sea level and is the combination of the attributes "val_elev" (this is the value of the field eleveation itself), "uom_dist_ver" (which is the unit of measurement) and the "txt_ver_datum" (either 'ADRIA' meaning above mean sea level based on Adria or 'OTHER:AGM08' which is the elevation above the Austrian Geoid Model 2008). This dataset covers Austrian territory only.
- **Geographic extents**:
  - 8, 46, 18, 49.
- Data as HTML. Collection items are also available in the following formats: --Please choose an option--
- Queryables as HTML.

tn-a:AerodromeType

- **Title**: Aerodrome Type
- **Description**: A code specifying the type of aerodrome.
- **Geographic extents**:
  - 8, 46, 18, 49.
- Data as HTML. Collection items are also available in the following formats: --Please choose an option--
- Queryables as HTML.

LFRZ austro CONTROL

GeoServer – OGC API – REST API

- REST Abfrage
  - http(s)://meinServer:Port/geoserver/ogc/features/collections/tn-a:AerodromeNode/items?f=application/application%2Fgeo%2Bjson&limit=50
  - liefert ein GeoJSON des angefragten Features:


```
{
    "type": "FeatureCollection",
    "features": [
        {
            "type": "Feature",
            "id": "AT.0012.6bed1778-d6bf-11e8-9f8b-f2801f1b9fd1.tn-a.AerodromeNode.1",
            "geometry": {
                "type": "Point",
                "coordinates": [
                    15.21555556,
                    48.41805556
                ]
            },
…
```


LFRZ austro CONTROL

DANKE

Ing. Klaus Gäbler, MAS(GIS), MSc
klaus.gaebler@austrocontrol.at
+43 51703 2424

Austro Control GmbH
Wagramer Straße 19
1220 Wien

Manuel Illmeyer, MSc
manuel.illmeyer@lfrz.gv.at
+43 1 33176-414

Land-, forst- und wasserwirtschaftliches Rechenzentrum GmbH
Hintere Zollamtsstraße 4
1030 Wien