

Open European Location Services (Open ELS)

WFS Central Prototype

Deliverable OpenELS-2.1

Author: Jari Reini, Frank Arntsen

Date: November 9, 2018

Version: 1.0



Co-financed by the European Union
Connecting Europe Facility

Change Summary

Version	Date	Author/Editor	Change Summary
0.1	04/09/2018	Jari Reini	First proposal for document structure
0.5	22/10/2018	Jari Reini	Updates in most of the chapters
1.0	09/11/2018	Jari Reini	Final edits

References

Ref.	Title/Version/Publication Date/Author
[1]	

Acronyms

ELF	European Location Framework
ELS	European Location Services
GML	Geography Markup Language
INSPIRE	Infrastructure for Spatial Information in Europe
NMCA	National Mapping and Cadastral Agency
OGC	Open Geospatial Consortium
WFS	Web Feature Service
WMS	Web Map Service

Contents

1	Introduction.....	5
1.1	General.....	5
1.2	Prototype architecture.....	5
1.3	New data model.....	6
1.4	Database installation.....	8
1.5	Geoserver	9
2	Tools Configuration	9
3	ETL (Extract-Transform-Load) Process.....	9
4	WFS 3.0.....	10
5	Results and conclusions.....	11
6	Annex I	12
6.1	Addressed XML schema	12
6.2	Data Definition Files to create a database	16

I Introduction

I.1 General

The ELF project implemented a European spatial information infrastructure, where INSPIRE-based services are provided by each NMCA and are presented through Showcase Application. Users were managed through a centralized security manager and were offered the opportunity to acquire either a test or development license for interface services.

WFS services were implemented by compiling the services into one access point called 'cascading' architecture. This decentralized architecture has caused quality deviations and major differences in service response times.

The Open ELS Program Board decided that the project could test a centralized architecture where the data (GML files) are uploaded to a centralized database and are offered as a WFS interface. At the same time, it was also tried to simplify the data model so that services are easier to implement and also easier to use.

This document describes the architecture, implementation and conclusions of the solution.

I.2 Prototype architecture

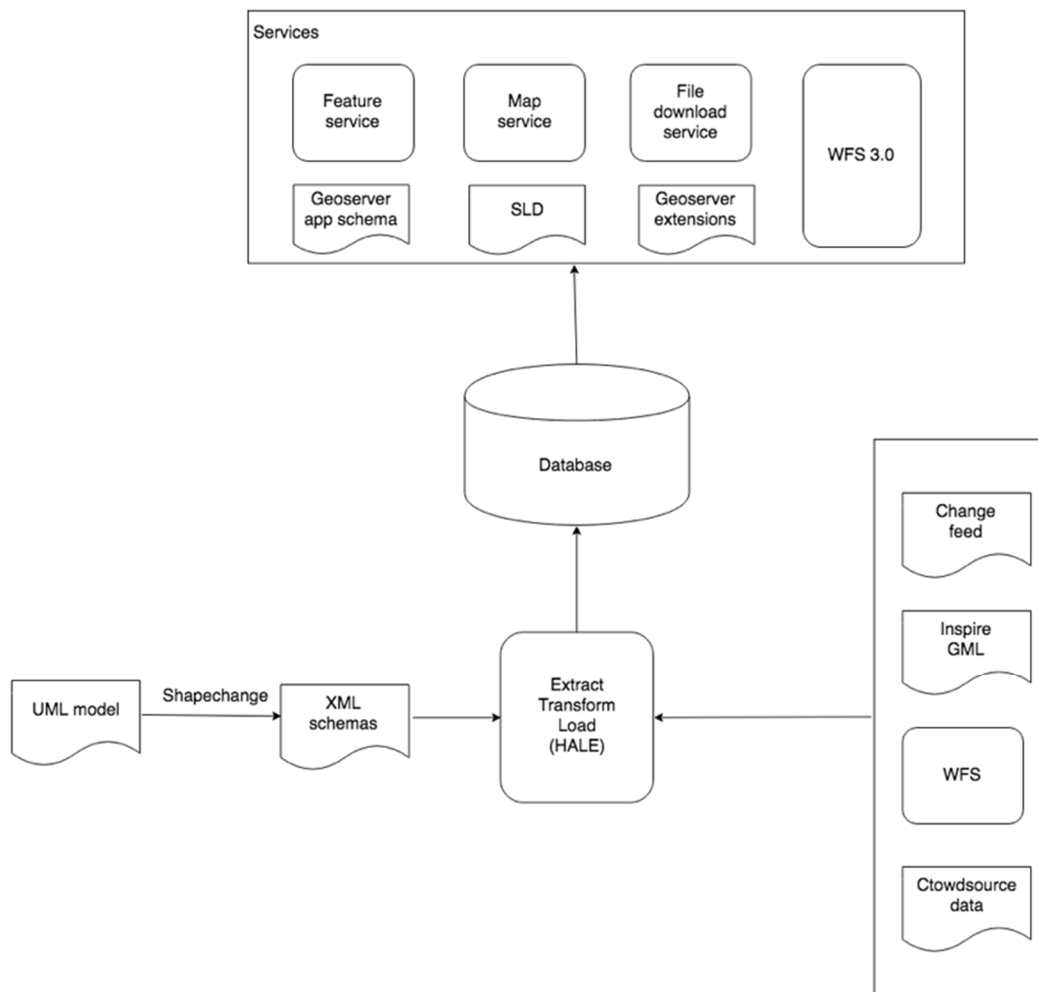
In architecture, the basic idea is to upload country-specific GML files to a centralized PostGIS data store and distribute the data through OGC interfaces.

The data model is based on the UML model (Sparx Enterprise Architect) described in Section 1.3. The XML schema (xsd) and DDL commands were created for the database to be initialized.

GML data was read and converted to the database model by wetransform.to's Open Source HALE tool. The modified data was uploaded to the database using HALE's JDBC connection to the PostGIS database. The architecture also supports the use of change data and other data models and formats than the INSPIRE GML but were not tested in this prototype.

OGC interfaces were implemented using an open source Geoserver that connects to the PostGIS database and shares the data with the WFS standard. In addition, Geoserver extension was installed that also allows sharing content as Geopackage files. Geoserver also allows a WMS interface, but is not configured in this prototype.

The architecture is described in the picture below:

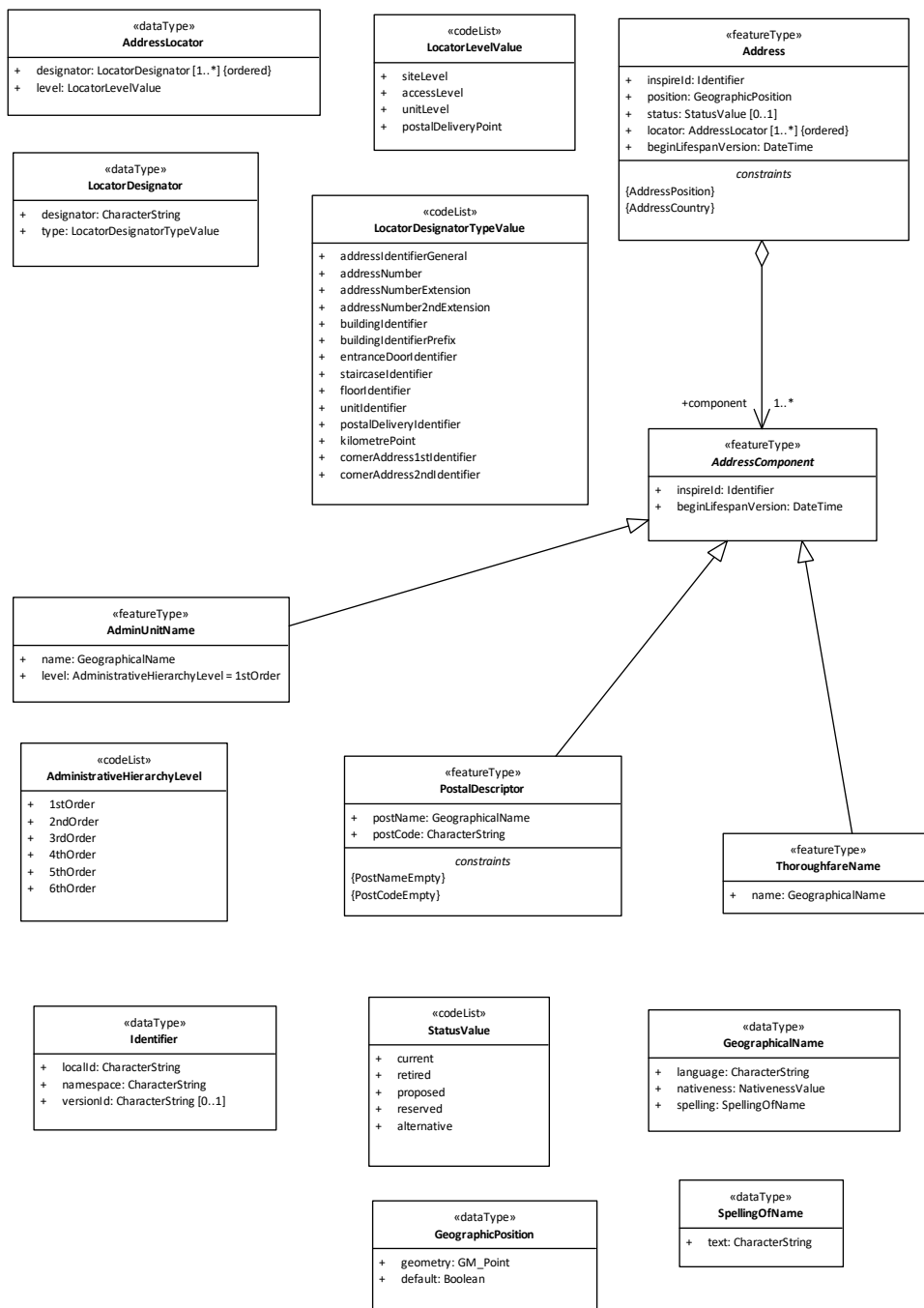


Picture. Architecture of prototype

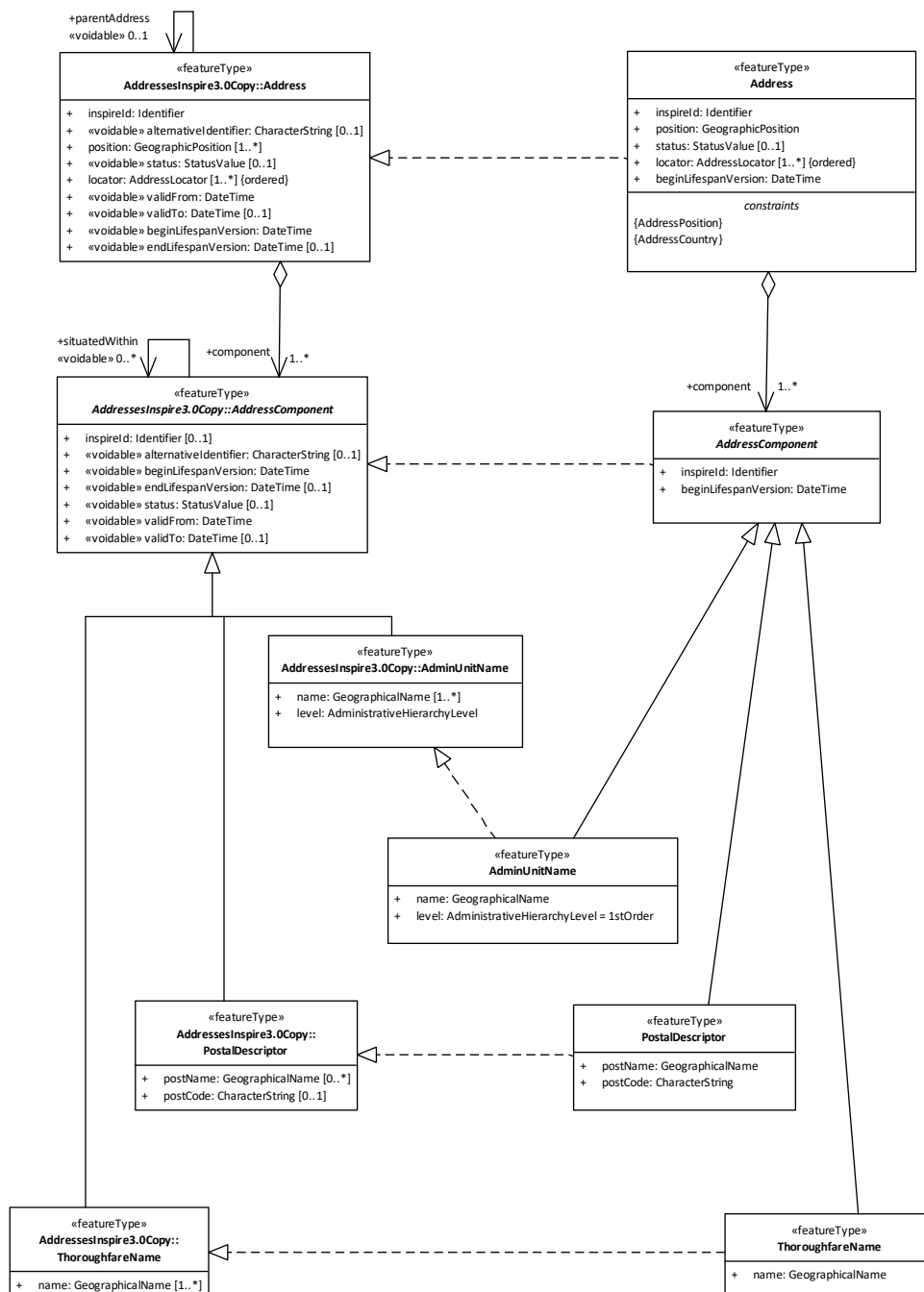
1.3 New data model

Goal is to reduce options and unused constructs, but still keep the Inspire structure, and generate database and WFS-services that generate Inspire-compliant GML. No flattening should be needed.

The following UML model defines the data model in principle, not all details revealed in the document.



The UML-model was constructed by stripping most voidable and optional elements from a copy of Inspire (3.0!) model, and make some central elements mandatory (meaningful to have queries on). The changes are documented in diagrams with UML realizations.



Schemas and example datasets are available at:

http://skjema.geonorge.no/INSPIRE_COPY/openelsAddresses/0.1/examples

I.4 Database installation

In first place PostgreSQL database and PostGIS extension were installed.

One may find more information about installation i.e. here:

<http://www.postgresql.com/journal/archives/362-An-almost-idiots-guide-to-install-PostgreSQL-9.5,-PostGIS-2.2-and-pgRouting-2.1.0-with-Yum.html>

1.5 Geoserver

The prototype uses GEOServer version 2.14 and the following extensions:

- geoserver-2.14.0-app-schema-plugin.zip
- geoserver-2.14.0-ogr-wps-plugin.zip

The first extension (app-schema) provides support for INSPIRE's complex schemas. Another extension offers alternative distribution formats such as Geopackage.

More information about Geoserver can be found here: <http://geoserver.org/release/stable/>

2 Tools Configuration

WFS configuration (app-schema) requires an extensive knowledge about data model and web services, but GISTools from Arkitektum (<https://www.arkitektum.no/produkter/gistools-for-uml/>) is able to create configuration files required by Geoserver.

We made the first versions of the app-schema configuration using the Arkitektum tool and continued the configuration manually. The summaries of configurations can be found in Annex I.

3 ETL (Extract-Transform-Load) Process

The ETL process is implemented in accordance with the architecture of paragraph 1.2.

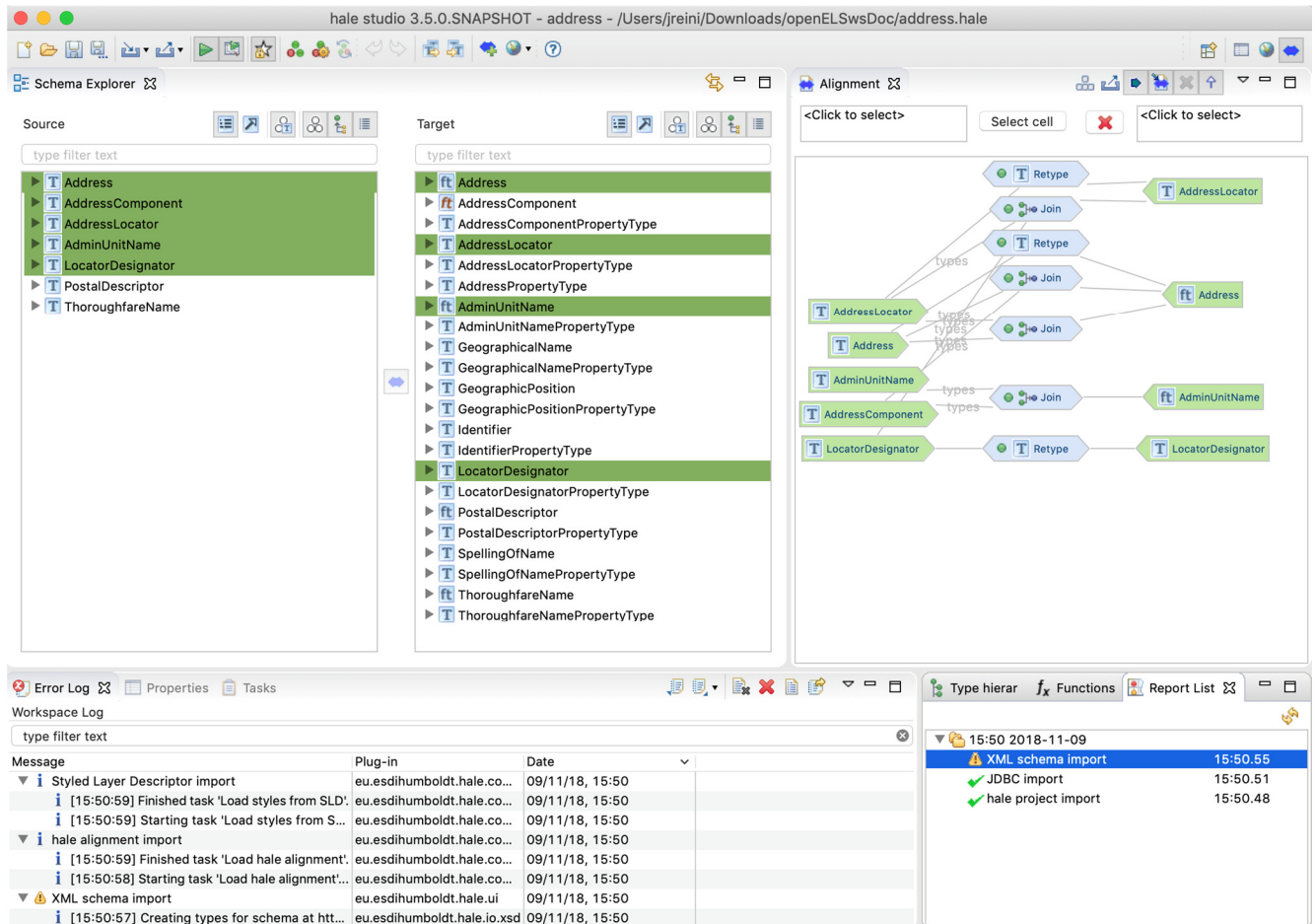
The process is as follows:

- 1) Create a Data Model (UML)
- 2) A data model is created for schemas and database definition statements
- 3) The HALE tool reads the target schema and the GML file to be read
- 4) The HALE tool is used to create mapping between input data and the target model

5) The data of the target model is downloaded to the PostGIS database

Finally, the schema to be provided is defined in Geoserver

Below is a screenshot of HALE used in mapping and uploading content to the database.



Picture. Data mapping using HALE

The left side column of HALE shows the input data model, in the middle one can find a target schema and on the right hand side mapping between input and output.

4 WFS 3.0

“All versions of the WFS standard have so far used a Remote-Procedure-Call-over-HTTP architectural style using XML for any payloads as it was state-of-the-art in the late 1990s and early 2000s, when WFS was originally designed. The current WFS 2.x draft standard made a significant start towards modernization by introducing a REST binding, but was still encumbered by legacy design decisions.

The next version (WFS 3.0) is intended to break free of this legacy and will specify a modernized service that aligns with the current architecture of the Web and the Spatial Data on the Web Best Practices. “

More information can be found from here:

https://github.com/opengeospatial/WFS_FES/blob/master/overview.md

During this prototype, WFS3 standard functionality was tested such that WFS3 is proxy on WFS2 service. The Interactive Instruments Idproxy product was used in the test. Learn more about Idproxysta: <https://github.com/interactive-instruments/Idproxy>

During the first test, it was found that WFS3 standard material can be provided on WFS2 standard or WFS3 can be implemented directly into the database. For more information on WFS3 functionality and demo, please contact jari.reini@eurogeographics.org

5 Results and conclusions

This work was done with the exception of a small preparation, as a one-day hackathon, with participants from Kartverket, National Land Survey of Finland and Eurogeographics. The way of operation was found to be very good and tedious to create a prototype in a very short time.

Centralized architecture offers many benefits but also challenges. It is easier for a centralized database and service to offer homogeneous quality, but on the other hand, much more is needed for computing power and capacity. In a decentralized architecture there may be several defects, but in a centralized architecture the service either works or does not work. The functionality of the service can be guaranteed eg. clustering of databases and web services.

Reading and mapping individual country data to a centralized database requires time and expertise, but on the other hand, once the configuration is first made, the process can easily be automated.

6 Annex I

6.1 Addressed XML schema

```
<?xml version="1.0" encoding="UTF-8"?><schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ad="http://openels.eu/schemas/Addresses/0.1" xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:gmlx="http://www.opengis.net/gml/3.3/exr" elementFormDefault="qualified"
targetNamespace="http://openels.eu/schemas/Addresses/0.1" version="0.1">

  <import namespace="http://www.opengis.net/gml/3.2"
schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>

  <import namespace="http://www.opengis.net/gml/3.3/exr"
schemaLocation="http://schemas.opengis.net/gml/3.3/extdEncRule.xsd"/>

  <!--XML Schema document created by ShapeChange - http://shapechange.net/-->

  <element name="Address" substitutionGroup="gml:AbstractFeature" type="ad:AddressType"/>

  <complexType name="AddressType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="inspireId" type="ad:IdentifierPropertyType"/>
          <element name="position" type="ad:GeographicPositionPropertyType"/>
          <element minOccurs="0" name="status" type="gml:ReferenceType">
            <annotation>
              <appinfo>
                <targetCodeList xmlns="http://www.opengis.net/gml/3.3/exr">StatusValue</targetCodeList>
              </appinfo>
            </annotation>
          </element>
          <element maxOccurs="unbounded" name="locator" type="ad:AddressLocatorPropertyType"/>
          <element name="beginLifespanVersion" type="dateTime"/>
          <element maxOccurs="unbounded" name="component">
            <annotation>
              <appinfo>
                <targetElement
xmlns="http://www.opengis.net/gml/3.2">ad:AddressComponent</targetElement>
              </appinfo>
            </annotation>
            <complexType>
              <complexContent>
                <extension base="gml:AbstractFeatureMemberType">
                  <sequence/>
                  <attributeGroup ref="gml:AssociationAttributeGroup"/>
                </extension>
              </complexContent>
            </complexType>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

```

        </extension>
    </complexContent>
</complexType>
<complexType name="AddressPropertyType">
    <sequence minOccurs="0">
        <element ref="ad:Address"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element abstract="true" name="AddressComponent" substitutionGroup="gml:AbstractFeature"
type="ad:AddressComponentType"/>
<complexType abstract="true" name="AddressComponentType">
    <complexContent>
        <extension base="gml:AbstractFeatureType">
            <sequence>
                <element name="inspireId" type="ad:IdentifierPropertyType"/>
                <element name="beginLifespanVersion" type="dateTime"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="AddressComponentPropertyType">
    <sequence minOccurs="0">
        <element ref="ad:AddressComponent"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="AddressLocator" substitutionGroup="gml:AbstractObject" type="ad:AddressLocatorType"/>
<complexType name="AddressLocatorType">
    <sequence>
        <element maxOccurs="unbounded" name="designator" type="ad:LocatorDesignatorPropertyType"/>
        <element name="level" type="gml:ReferenceType">
            <annotation>
                <appinfo>
                    <targetCodeList
xmlns="http://www.opengis.net/gml/3.3/exr">LocatorLevelValue</targetCodeList>
                </appinfo>
            </annotation>
        </element>
    </sequence>
</complexType>
<complexType name="AddressLocatorPropertyType">
    <sequence>
        <element ref="ad:AddressLocator"/>
    </sequence>
</complexType>

```

```

    </sequence>
</complexType>
<element name="AdminUnitName" substitutionGroup="ad:AddressComponent" type="ad:AdminUnitNameType"/>
<complexType name="AdminUnitNameType">
  <complexContent>
    <extension base="ad:AddressComponentType">
      <sequence>
        <element name="name" type="ad:GeographicalNamePropertyType"/>
        <element name="level" type="gml:ReferenceType">
          <annotation>
            <appinfo>
              <targetCodeList
xmlns="http://www.opengis.net/gml/3.3/exr">AdministrativeHierarchyLevel</targetCodeList>
            </appinfo>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="AdminUnitNamePropertyType">
  <sequence minOccurs="0">
    <element ref="ad:AdminUnitName"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="GeographicPosition" substitutionGroup="gml:AbstractObject"
type="ad:GeographicPositionType"/>
<complexType name="GeographicPositionType">
  <sequence>
    <element name="geometry" type="gml:PointPropertyType"/>
    <element name="default" type="boolean"/>
  </sequence>
</complexType>
<complexType name="GeographicPositionPropertyType">
  <sequence>
    <element ref="ad:GeographicPosition"/>
  </sequence>
</complexType>
<element name="GeographicalName" substitutionGroup="gml:AbstractObject"
type="ad:GeographicalNameType"/>
<complexType name="GeographicalNameType">
  <sequence>
    <element name="language" type="string"/>
    <element name="nativeness"/>
    <element name="spelling" type="ad:SpellingOfNamePropertyType"/>

```

```

    </sequence>
</complexType>
<complexType name="GeographicalNamePropertyType">
    <sequence>
        <element ref="ad:GeographicalName"/>
    </sequence>
</complexType>
<element name="Identifier" substitutionGroup="gml:AbstractObject" type="ad:IdentifierType"/>
<complexType name="IdentifierType">
    <sequence>
        <element name="localId" type="string"/>
        <element name="namespace" type="string"/>
        <element minOccurs="0" name="versionId" type="string"/>
    </sequence>
</complexType>
<complexType name="IdentifierPropertyType">
    <sequence>
        <element ref="ad:Identifier"/>
    </sequence>
</complexType>
<element name="LocatorDesignator" substitutionGroup="gml:AbstractObject"
type="ad:LocatorDesignatorType"/>
<complexType name="LocatorDesignatorType">
    <sequence>
        <element name="designator" type="string"/>
        <element name="type" type="gml:ReferenceType">
            <annotation>
                <appinfo>
                    <targetCodeList
xmlns="http://www.opengis.net/gml/3.3/exr">LocatorDesignatorTypeValue</targetCodeList>
                </appinfo>
            </annotation>
        </element>
    </sequence>
</complexType>
<complexType name="LocatorDesignatorPropertyType">
    <sequence>
        <element ref="ad:LocatorDesignator"/>
    </sequence>
</complexType>
<element name="PostalDescriptor" substitutionGroup="ad:AddressComponent"
type="ad:PostalDescriptorType"/>
<complexType name="PostalDescriptorType">
    <complexContent>
        <extension base="ad:AddressComponentType">
            <sequence>
                <element name="postName" type="ad:GeographicalNamePropertyType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```
        <element name="postCode" type="string"/>
    </sequence>
</extension>
</complexContent>
</complexType>
<complexType name="PostalDescriptorPropertyType">
    <sequence minOccurs="0">
        <element ref="ad:PostalDescriptor"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
<element name="SpellingOfName" substitutionGroup="gml:AbstractObject" type="ad:SpellingOfNameType"/>
<complexType name="SpellingOfNameType">
    <sequence>
        <element name="text" type="string"/>
    </sequence>
</complexType>
<complexType name="SpellingOfNamePropertyType">
    <sequence>
        <element ref="ad:SpellingOfName"/>
    </sequence>
</complexType>
<element name="ThoroughfareName" substitutionGroup="ad:AddressComponent"
type="ad:ThoroughfareNameType"/>
<complexType name="ThoroughfareNameType">
    <complexContent>
        <extension base="ad:AddressComponentType">
            <sequence>
                <element name="name" type="ad:GeographicalNamePropertyType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<complexType name="ThoroughfareNamePropertyType">
    <sequence minOccurs="0">
        <element ref="ad:ThoroughfareName"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attributeGroup ref="gml:OwnershipAttributeGroup"/>
</complexType>
</schema>
```

6.2 Data Definition Files to create a database

```
CREATE SCHEMA addresses AUTHORIZATION postgres;
```



```
CREATE TABLE addresses."Address" (  
  "localId" text NOT NULL,  
  -- format "Address.12345"  
  "namespace" text NOT NULL,  
  "versionId" text DEFAULT NULL,  
  "geometry" geometry(Point, 4258) NOT NULL,  
  "default" boolean NOT NULL,  
  "status" text DEFAULT NULL,  
  -- locator AddressLocator has FK  
  "beginLifespanVersion" timestamp with time zone NOT NULL  
  -- UTC?  
  --component AddressComponent has FK  
);  
ALTER TABLE addresses."Address" ADD PRIMARY KEY ("localId");  
ALTER TABLE addresses."Address" ADD CONSTRAINT enforce_dims_geometry CHECK (st_ndims(geometry) = 2);  
ALTER TABLE addresses."Address" ADD CONSTRAINT enforce_srid_geometry CHECK (st_srid(geometry) = 4258);  
-- hvor i modellen kommer dette fra?  
;  
  
CREATE TABLE addresses."AdminUnitName" (  
  "localId" text NOT NULL UNIQUE,  
  -- format "AdminUnitName.12345"  
  "namespace" text NOT NULL,  
  "versionId" text DEFAULT NULL,  
  "beginLifespanVersion" timestamp with time zone NOT NULL,  
  -- UTC?  
  "language" text NOT NULL,  
  "nativeness" text DEFAULT 'http://inspire.ec.europa.eu/codelist/NativenessValue/endonym',  
  "text" text NOT NULL,  
  "level" text DEFAULT 'http://inspire.ec.europa.eu/codelist/AdministrativeHierarchyLevel/1stOrder'  
);  
  
CREATE TABLE addresses."PostalDescriptor" (  
  "localId" text NOT NULL UNIQUE,  
  -- format "PostalDescriptor.12345"  
  "namespace" text NOT NULL,  
  "versionId" text DEFAULT NULL,  
  "beginLifespanVersion" timestamp with time zone NOT NULL,  
  -- UTC?  
  "language" text NOT NULL,  
  "nativeness" text DEFAULT 'http://inspire.ec.europa.eu/codelist/NativenessValue/endonym',  
  "text" text NOT NULL,  
  "postCode" text NOT NULL
```

```
);
```

```
CREATE TABLE addresses."ThoroughfareName" (  
  "localId" text NOT NULL UNIQUE,  
  -- format "ThoroughfareName.12345"  
  "namespace" text NOT NULL,  
  "versionId" text DEFAULT NULL,  
  "beginLifespanVersion" timestamp with time zone NOT NULL,  
  -- UTC?  
  "language" text NOT NULL,  
  "nativeness" text DEFAULT 'http://inspire.ec.europa.eu/codelist/NativenessValue/endonym',  
  "text" text NOT NULL  
);
```

```
CREATE TABLE addresses."AddressLocator" (  
  objid serial NOT NULL,  
  -- designator has FK  
  "level" text,  
  "address_fk" text  
);  
  
ALTER TABLE addresses."AddressLocator" ADD PRIMARY KEY ("objid");  
  
ALTER TABLE addresses."AddressLocator" ADD FOREIGN KEY ("address_fk") REFERENCES  
addresses."Address"("localId") MATCH SIMPLE ON UPDATE NO ACTION ON DELETE CASCADE;
```

```
CREATE TABLE addresses."LocatorDesignator" (  
  objid serial NOT NULL,  
  "designator" text NOT NULL,  
  "level" text NOT NULL,  
  "seq" int NOT NULL,  
  "addressLocator_fk" int  
);  
  
ALTER TABLE addresses."LocatorDesignator" ADD PRIMARY KEY ("objid");  
  
ALTER TABLE addresses."LocatorDesignator" ADD FOREIGN KEY ("addressLocator_fk") REFERENCES  
addresses."AddressLocator"("objid") MATCH SIMPLE ON UPDATE NO ACTION ON DELETE CASCADE;
```

```
CREATE TABLE addresses."AddressComponent" (  
  "localId" text NOT NULL,  
  -- format "AddressComponent.12345"  
  "namespace" text NOT NULL,  
  "versionId" text DEFAULT NULL,  
  "beginLifespanVersion" timestamp with time zone NOT NULL,  
  -- UTC?  
  "address_fk" text NOT NULL,
```

```
-- FK back to "Address"
"componentAdminUnitName_fk" text DEFAULT NULL,
"componentPostalDescriptor_fk" text DEFAULT NULL,
"componentThoroughfareName_fk" text DEFAULT NULL
);

ALTER TABLE addresses."AddressComponent" ADD PRIMARY KEY ("localId");
ALTER TABLE addresses."AddressComponent" ADD FOREIGN KEY ("address_fk") REFERENCES
addresses."Address"("localId") MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE addresses."AddressComponent" ADD FOREIGN KEY ("componentAdminUnitName_fk") REFERENCES
addresses."AdminUnitName"("localId") MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE addresses."AddressComponent" ADD FOREIGN KEY ("componentPostalDescriptor_fk") REFERENCES
addresses."PostalDescriptor"("localId") MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION;
ALTER TABLE addresses."AddressComponent" ADD FOREIGN KEY ("componentThoroughfareName_fk") REFERENCES
addresses."ThoroughfareName"("localId") MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION;
;
```