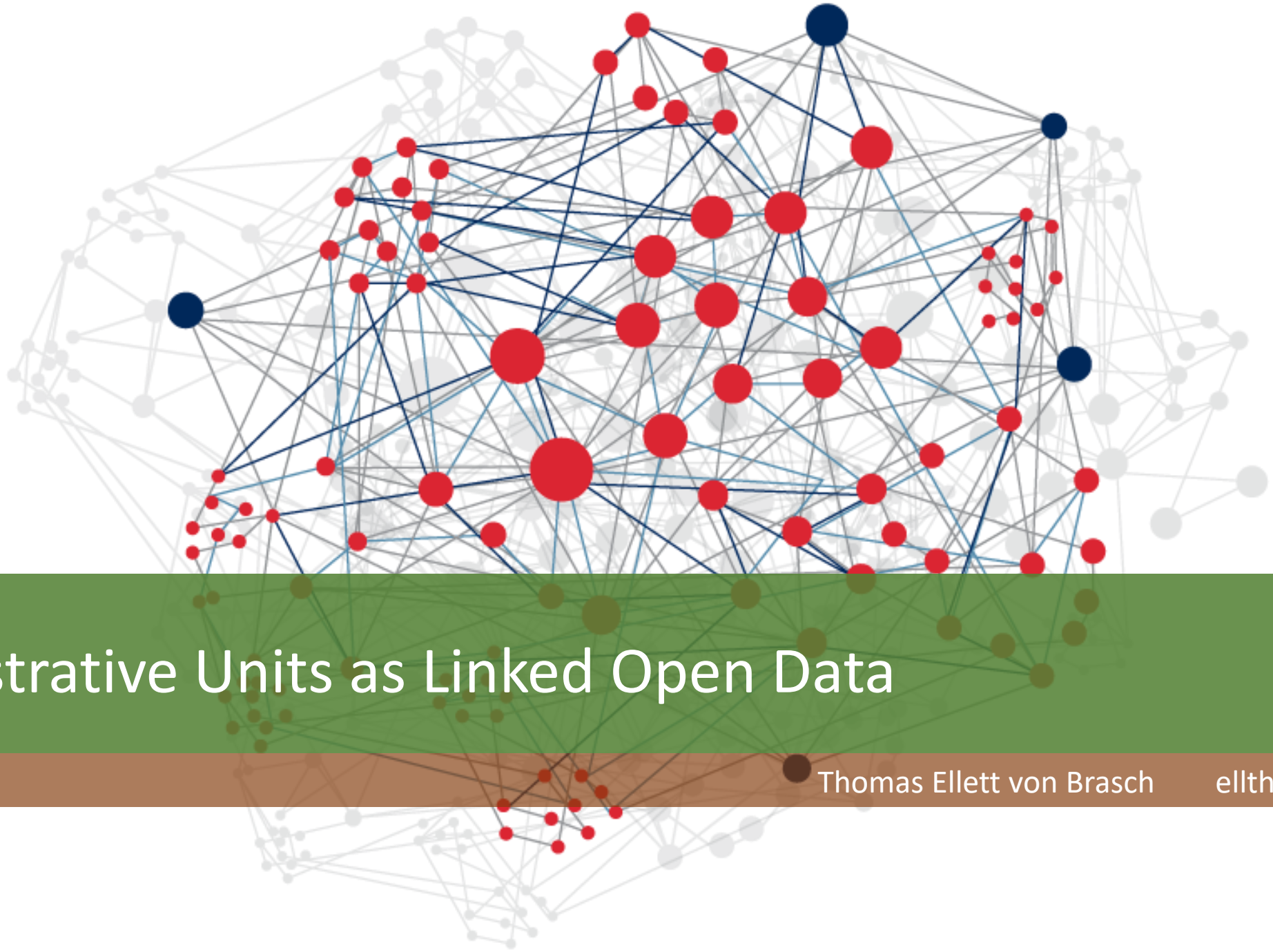


Administrative Units as Linked Open Data

Thomas Ellett von Brasch elltho@kartverket.no

Kartverket

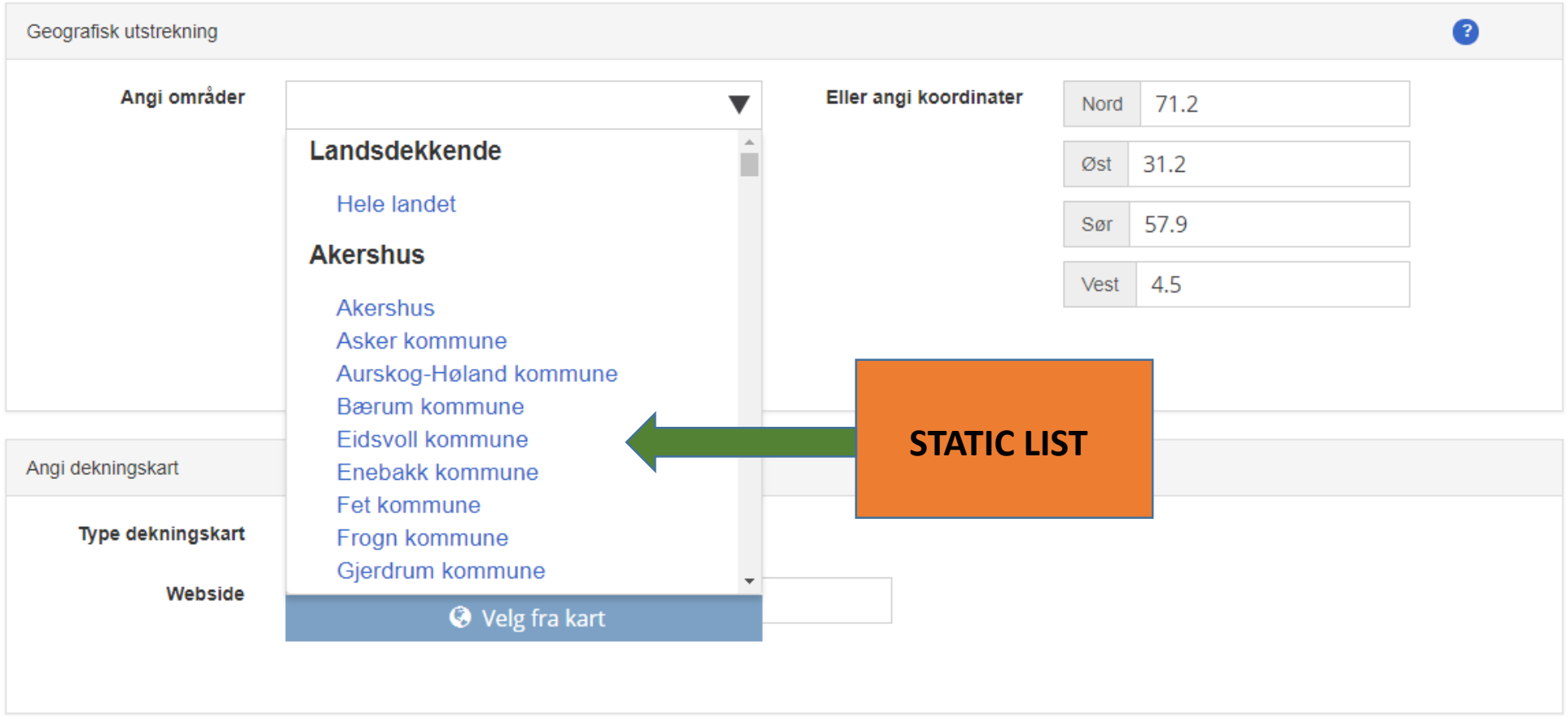
Contents

Project outline and objectives

THE PROJECT

Difi
Direktoratet for forvaltning og ikt

FELLES DATAKATALOG



THE PROBLEM

COUNTY AND DISTRICT MERGING PROJECT

THE SOLUTION

Linked

Open

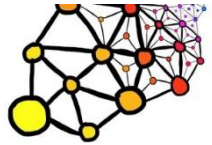
Data

Why should we use LOD?

Benefits

DATA INTEROPERABILITY

AUTHORITATIVE DATA Stored once but used many times



ONCE

http://data.geonorge.no/administrativeEnheter/fylke/id/173157

SEMANTIC UNDERSTANDING

DATA DRIVEN APPLICATIONS

NRK

BBC SPORT

Administrative Units from Kartverket

PROJECT PLAN

The original request came from DIFI, who requested a location based service from an authoritative source, so they didn't need to worry about updates and knew the data they were providing was always correct and up to date.

DIFI themselves are only interested in the URI's, so that they don't need to store any extraneous information that would need to be updated manually. At the same time they wanted the user to be able to input the URI in a normal browser and return human readable information for the subject.

We wanted to use the task as a pilot project and develop a methodology and system which we could use as a template for official RDF delivery from Kartverket.

Central Components

Simplified
Data Model

ONTOLOGY

Semantic
Understanding

http://data.geonorge.no/{namepace}/id/{localId}

Stable de-
referensable URI's

R D F

Data according
to a specification

REST API

Distribution/Access

Ontology

Ontologies and codelists

- Web accessable vocabularies which define which terms are available and how they are related to each other (terms can also be related to other terms in other onotlogies)
- All ontologies and codelists have URI's themselves: http://rdf.kartverket.no/onto/adm_enhet_4.0#
- All 'Classes', 'Object properties' and 'Data properties' also have URI's: http://www.opengis.net/ont/geosparql#ehContains
- RDFS and OWL ontologies give simple elements for building ontologies
- SKOS (simple knowledge organisation system) is used for codelists.

Ontologies and codelists

Reuse existing ontologies

Use multiple languages, or at least English.

BEST PRACTICES

‘Map’ to other ontologies

Simple is often better than heavy and complex

simple is better than complex

Creation of ontology

http://rdf.kartverket.no/onto/adm_enhet_4.0.owl

We originally wanted to go direct from the full UML model to a subset. This was extremely complicated and time consuming.

We ended up with creating our own ontology from scratch, but using the full model as a reference for data and object types. It was neccesary to import and reuse several other ontologies.

We used Protege from Stanford university, which is very powerful and very flexible, but a little unstable (especially combined with ontop)

The ontology is available on the net at rdf.kartverket.no/onto.

Experiences

1. Most important and most difficult part of the process. We don't have enough experience and knowledge to evaluate the methodology used here.

2. Easy to reuse other ontologies, just need to make sure that these are from an 'authoritative' source and vil continue to be maintained and available.

3. A simple model if much easier to use. The less hierarchy you have, the better.

4. Use as many relationships as possible to enrich the data.

URI

Dereferensable URI's (HTTP names/URL)

- All objects which are delivered as LOD must have a stable and unique URI so that when people begin to include them in their own data, they know that the link will always work and the can rely on the dataset.

- The URI's should be dereferensable HTTP URL's so that normal web clients can fetch information about the object.

- The URI's should also be unique globally. Therefore, its good practice to define a URI pattern og register it.

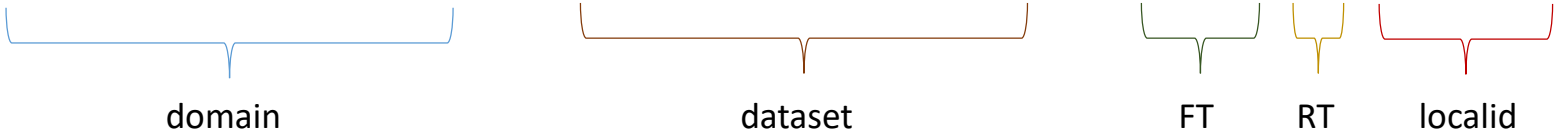
Dereferensable URI's (HTTP names/URL)

Example of a common pattern:

http://<domain>/<authority/dataset>/<featureclass>/<resourcetype>/<localid>

Example from Kartverket

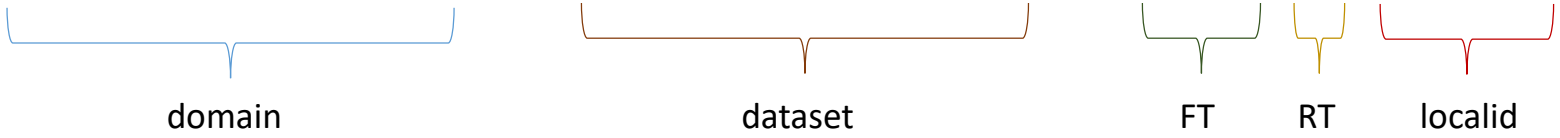
http://data.geonorge.no/administrativeEnheter/fylke/id/173157



Construction

Example from Kartverket

http://data.geonorge.no/administrativeEnheter/fylke/id/173157



Example from Ordnance Survey

http://data.ordnancesurvey.co.uk/id/4000000074577442

Example from Kadaster Netherlands

http://brk.basisregistraties.overheid.nl/id/kadastrale-grens/240128610

Experiences

http://data.geonorge.no/administrativeEnheter/fylke/id/173157

Not easy to create a pattern which covers all needs and that everyone in the organisation are agreed upon. It needs to handle uniqueness, persistence, versions and perhaps readability.

The biggest challenge was to decide on the dataset/featuretype sections. Should we follow the gml application schema, use names from the UML model or different names entirely etc.

The localid component was also difficult since the data originally comes from several different sources, with only a localid at the geometry level in the database. Therefore we need to use concatenation to create a localid per unit.

DATA CONVERSION AND LOADING

Data according to RDF specification (triples)

- RDF as a framework: 6 specifications

- RDF as a data model - triples:

```
uri://people#MikeSmith12        http://xmlns.com/foaf/0.1/knows        uri://people#JohnDoe45
```

Mike Smith Knows John Doe

- RDF as a data serialisation format – RDF/XML:

http://nnriap523:5000/#!/adminstrative_unit/get_describe_county

Data

Originally data in accordance with the national specification (SOSI)

Transformation with protege and the OnTop plugin

protégé

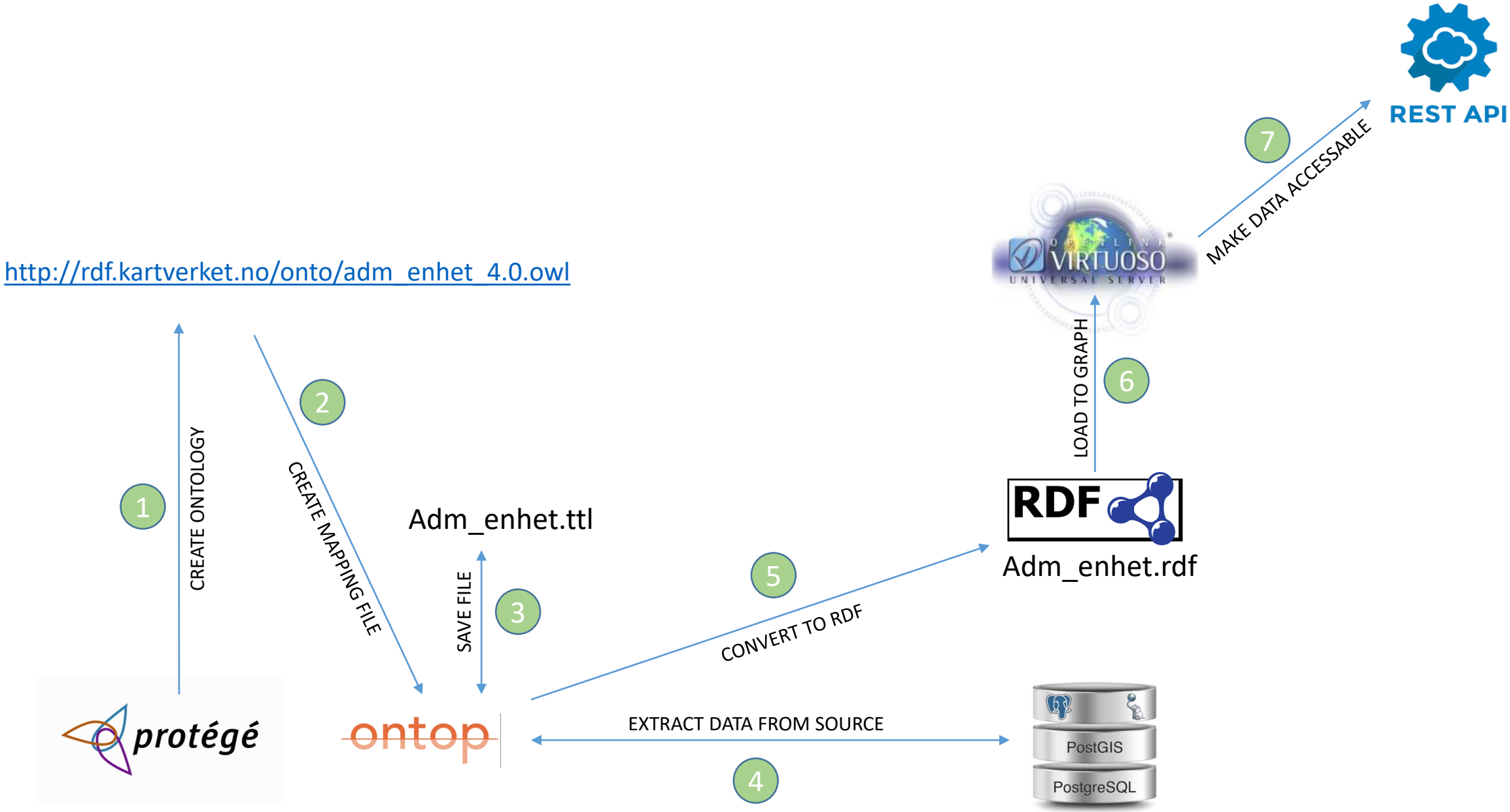
ontop

Data converted to RDF and loaded into Virtuoso

RDF

OPENLINK VIRTUOSO UNIVERSAL SERVER

Infrastructure



Experiences

Quite complex sql needed, especially to manage the relationships between the classes.

Ontop itself is fairly unstable and not that intuative to use. But once the mapping file is in place, the conversion process is very smooth and stable.

Loading the data into Virtuoso is very easy, but the setup of Virtusoso may need to be tweaked if you’re working with large data sets.

It is better to store the geometry in a seperate graph to the other data and object types to increase performace. We have also utilised our accompanying wfs service for spatial queries.

Tool Experiences

Very powerful, very flexible, quite unstable.

Flexible, powerful, stable, widely used
Triplestore + sparql endpoint + faceted search + bulk upload + powerful administration

Alternativer

Semafora

Strabon

Stardog

Parliament

Allegrograph

TopQuadrant

BrightstarDB

Oracle

ACCESS

Access

Sparql Endpoint

http://ng.hslayers.org:8890/sparql

Rest API's

'Doc' Representation

Access

Currently through a sparql endpoint, faceted search and rest API built over the top of the Sparql endpoint. We're also interested in creating a geocoding / resolver product to make it easier for people to use our URI's and include them in their own data.

Next stage is a full 'doc' representation which shows all the information about a subject in an html page. Currently experimenting with the Linked data theatre product (from Kadaster Netherlands), but will consider other solutions.

Started with administrative units, but we have begun working with placenames as well and will have those available before the end of the year. Adddresses as a product will also be considered after the general principles of the delivery are agreed within the organisation.

To infinity and beyond….

Whats Next?

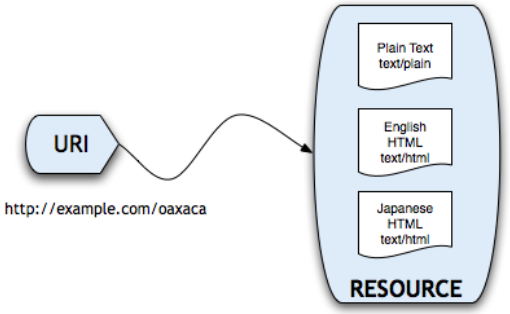
Add better descriptions in more languages

Ontology mapping - owl:equivalentProperty
Schema.org for ‘crawlability’

REST API

Customise for several datasets

New API for geocoding



Set up content negotiation

‘Doc’ representation– xslt or Linked Data Theatre or …

Better integration with existing products

Conclusions

Conclusions

1

A lot of work, have to create a whole parallel infrastructure to the traditional one.

2

Theres no ‘one’ way of doing it, technologies are changing all the time.

But….

3

Important to get the ontology correct to start with.

4

Maybe not that useful for the National Mapping Authorities if not all data is delivered in RDF

5

Extremely useful for society.

6

Data is saved once by the authoritative organisation, so money is saved on duplication and data quality increases.

7

Data is standardised so that all of society can access and use it instead of only a small niche

8

More connected, more stable and more flexible array of data products

Thankyou for listening!!!

Links

TOOLS

Protege - https://protege.stanford.edu/

Ontop - https://github.com/ontop/ontop/wiki

(Karma) - http://usc-isi-i2.github.io/karma/

Virtuoso - https://virtuoso.openlinksw.com/

Swagger - https://swagger.io/

Kartverket data

Ontology - http://rdf.kartverket.no/onto/adm_enhet_4.0.owl#

Sparql endpoint - http://rdf.kartverket.no//sparql?

REST API - http://rdf.kartverket.no/api/1.0/#/

Example subject - https://data.geonorge.no/administrativeEnheter/fylke/id/173156

Spatial Dat on the Web

SDW- https://www.w3.org/TR/sdw-bp/