



Statens vegvesen

Jan Kristian Jensen  
Norwegian Road Administration  
FME World Tour 2014, May 6th, Oslo  
[jan.kristian.jensen@vegvesen.no](mailto:jan.kristian.jensen@vegvesen.no)



 @LtGlahn



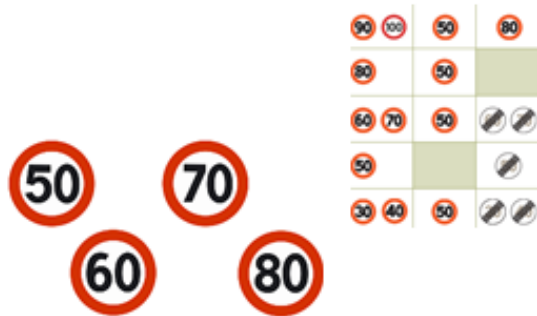
Fetch data from Nasjonal  
Vegdatabank (NVDB) with FME

# Norwegian Road database (NVDB)



Statens vegvesen

238.253.944m

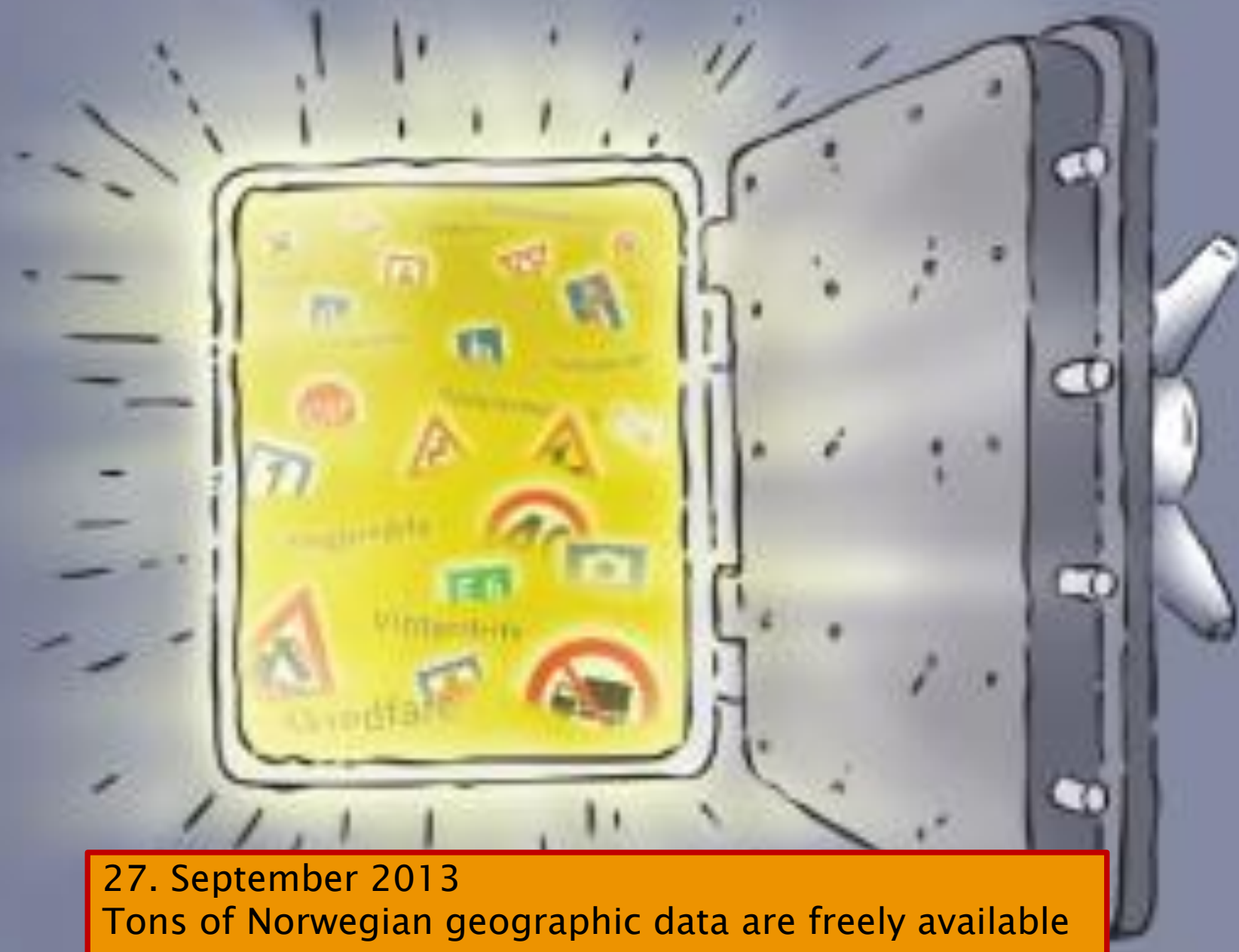


Repository of Norwegian road network

... and road related objects  
– 370 different object types

Very flexible schema  
Object definition in separat data catalogue





27. September 2013

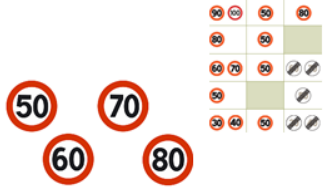
Tons of Norwegian geographic data are freely available

# REST api, Nasjonal vegdatabank



Statens vegvesen

238.253.944m



Datacatalogue – definition and structure

<https://www.vegvesen.no/nvdb/api/datakatalog/>

Generic, powerful search function

<https://www.vegvesen.no/nvdb/api/sok/>

The different objects

<https://www.vegvesen.no/nvdb/api/vegobjekter>



Functionality to calculate x,y  $\Leftrightarrow$  road network

<https://www.vegvesen.no/nvdb/api/vegreferanse>

Formats: XML, JSON.

Do specify MediaType in http header.

*Accept:application/vnd.vegvesen.nvdb-v1+json*

*Accept:application/vnd.vegvesen.nvdb-v1+xml*

*Per 2014: Only road related objects, not the road network)*





Special cortecy [@KenAtSafe](#)



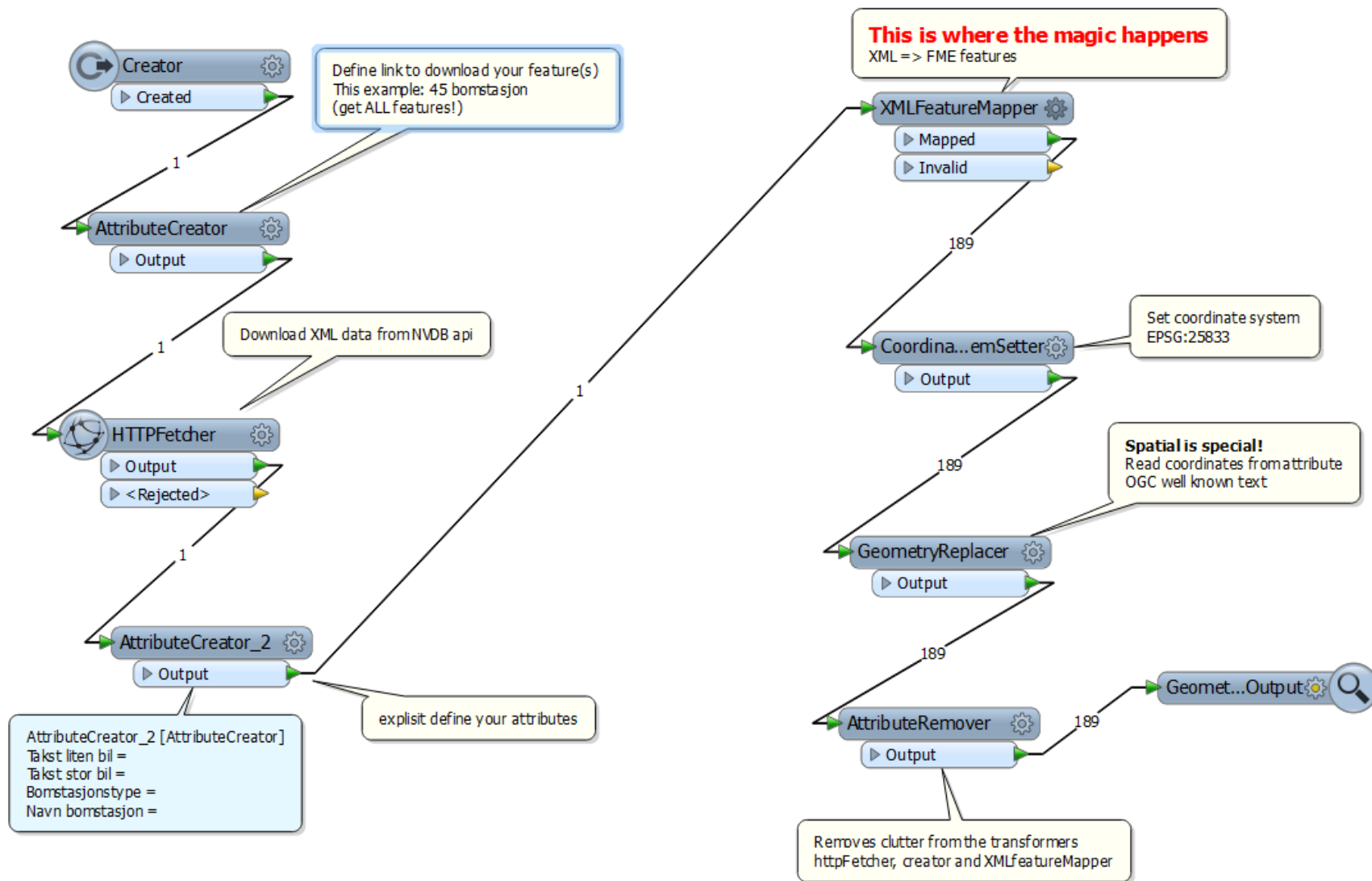
Statens vegvesen

Our development blog:  
**vegdata.no**

<http://www.vegdata.no/2014/02/19/a-little-note-to-our-international-fans/>

[https://github.com/LtGlahn/vegdatalabs\\_fme](https://github.com/LtGlahn/vegdatalabs_fme)







# Looking good...

View 6 (189)

inspector [FFS] (189)

GeometryReplacer\_Output (1...

Features Selected: 1 of 1

Property	Value
Properties	
Feature Type	GeometryReplacer_Output
Coordinate System	EPSG:25833
Dimension	2D
Number of Vertices	1
Bounding Box	
Minimum Values	290809.634208956, 6698820.40748...
Maximum Values	290809.634208956, 6698820.40748...
Attributes	
22 attribute(s)	
Geometry	
IFMEPoint	(290809.634208956, 6698820.4074...
x	290809.634208956
y	6698820.40748208

Table View

	Takst liten bil	Takst stor bil	Bomstasjonstype	Navn bomstasjon	modifisert	objektId	objektTypeId	objektTypeNavn	politiDistrikt	region	geometriUtm33	kommune
1	20.0	100.0	AutoPASS + manuell	Svinesund (Sonvikv.)	2014-04-03T16:...	92260258	45	Bomstasjon	Østfold	ØST	POINT (285647.5639...	HALDEN
2	23.0	46.0	AutoPASS + automat...	Raukerud	2014-02-14T10:...	92269108	45	Bomstasjon	Østfold	ØST	POINT (256959.3803...	MOSS
3	14.0	28.0	AutoPASS + automat...	HOVINMOEN	2013-06-25T07:...	261942611	45	Bomstasjon	Romerike	ØST	POINT (287779.4799...	ULLENSAKER
4	15.0	30.0	AutoPASS + automat...	Andelva	2013-01-08T09:...	305830577	45	Bomstasjon	Romerike	ØST	POINT (286849.0907...	EIDSVOLL
5	11.0	22.0	AutoPASS + automat...	Tømte	2013-01-07T10:...	305830476	45	Bomstasjon	Romerike	ØST	POINT (290809.6342...	EIDSVOLL
6	31.0	93.0	AutoPASS + automat...	E6 Europavegen	2014-03-18T22:...	86574511	45	Bomstasjon	Oslo	ØST	POINT (266083.0947...	OSLO
7	31.0	93.0	AutoPASS + automat...	E6 Djupdalsveien v/ ...	2014-03-18T22:...	86574510	45	Bomstasjon	Oslo	ØST	POINT (266993.0358...	OSLO
8	31.0	93.0	AutoPASS + automat...	R190 Ulven	2014-03-18T22:...	209148809	45	Bomstasjon	Oslo	ØST	POINT (266729.4320...	OSLO
9	31.0	93.0	AutoPASS + automat...	Rampe E6 Europaveg...	2014-03-18T22:...	209148808	45	Bomstasjon	Oslo	ØST	POINT (266108.9362...	OSLO
10	17.0	34.0	AutoPASS + automat...	Kolomoen	2013-06-25T07:...	262052940	45	Bomstasjon	Hedmark	ØST	POINT (297359.8593...	STANGE
11	22.0	44.0	AutoPASS + automat...	Tingberg	2013-09-04T16:...	370897162	45	Bomstasjon	Gudbrandsdal	ØST	POINT (253478.0189...	ØYER
12	10.0	30.0	AutoPASS + automat...	Klett-E6	2014-03-19T09:...	265182535	45	Bomstasjon	Sør-Trøndelag	MIDT	POINT (266320.3602...	TRONDHEIM
13	8.0	24.0	AutoPASS + automat...	E6 Tonstad	2014-03-19T09:...	487415246	45	Bomstasjon	Sør-Trøndelag	MIDT	POINT (269087.9660...	TRONDHEIM
14	33.0	66.0	AutoPASS + automat...	Ranheim	2014-03-05T09:...	264832263	45	Bomstasjon	Sør-Trøndelag	MIDT	POINT (278231.6486...	TRONDHEIM
15	17.0	34.0	AutoPASS + automat...	Hommelvik	2014-03-05T09:...	264832262	45	Bomstasjon	Sør-Trøndelag	MIDT	POINT (291655.1534...	MALVIK

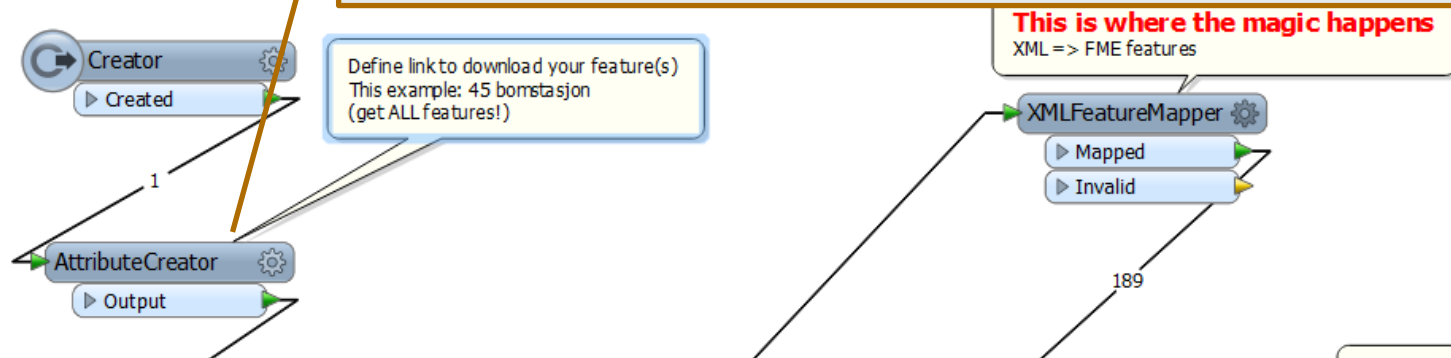
Q

in any column

1 selected / 189 row(s)

<https://www.vegvesen.no/nvdb/api/vegObjekter/{Id}>

Essentially: Give me ALL (or a number of) objects  
Optionally: Whithin bbox..



Let's try the SEARCH function!

VERY powerful: Search for objects with:

- Road network properties
- Attribute values
- bbox, region, county, municipality

*Plus a bit more...*

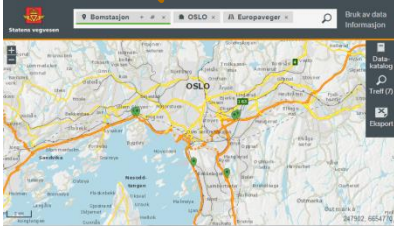
<https://www.vegvesen.no/nvdb/api/sok?>



Vegkart example

FME workspace modification example

A fancy  
Vegkart-search

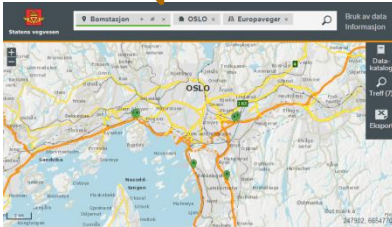


URL  
decoding



Template for  
creating of valid  
Search-URI's

## A fancy Vegkart-search

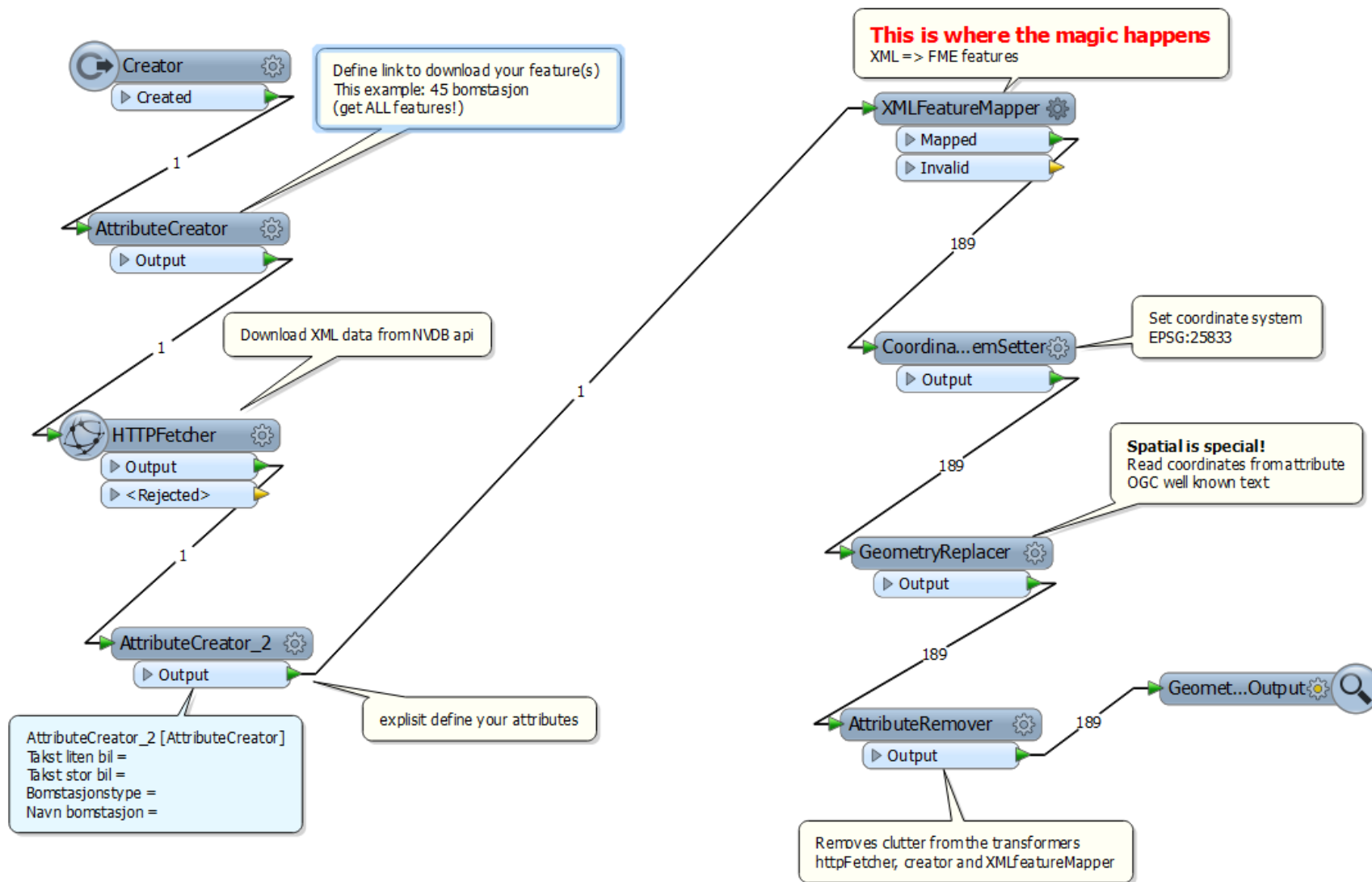


[https://www.vegvesen.no/vegkart/vegkart/#!//vegreferanse:296070.0829735:6732223.668614/sok:{\"lokasjon\":{\"kommune\":\[301\]},\"objektTyper\":{\"id\":45,\"antall\":\"1 000\",\"filter\":{\"type\":\"vegreferanse\",\"operator\":\"=\",\"verdi\":\[\"EV\"\]}}}\]](https://www.vegvesen.no/vegkart/vegkart/#!//vegreferanse:296070.0829735:6732223.668614/sok:{\)

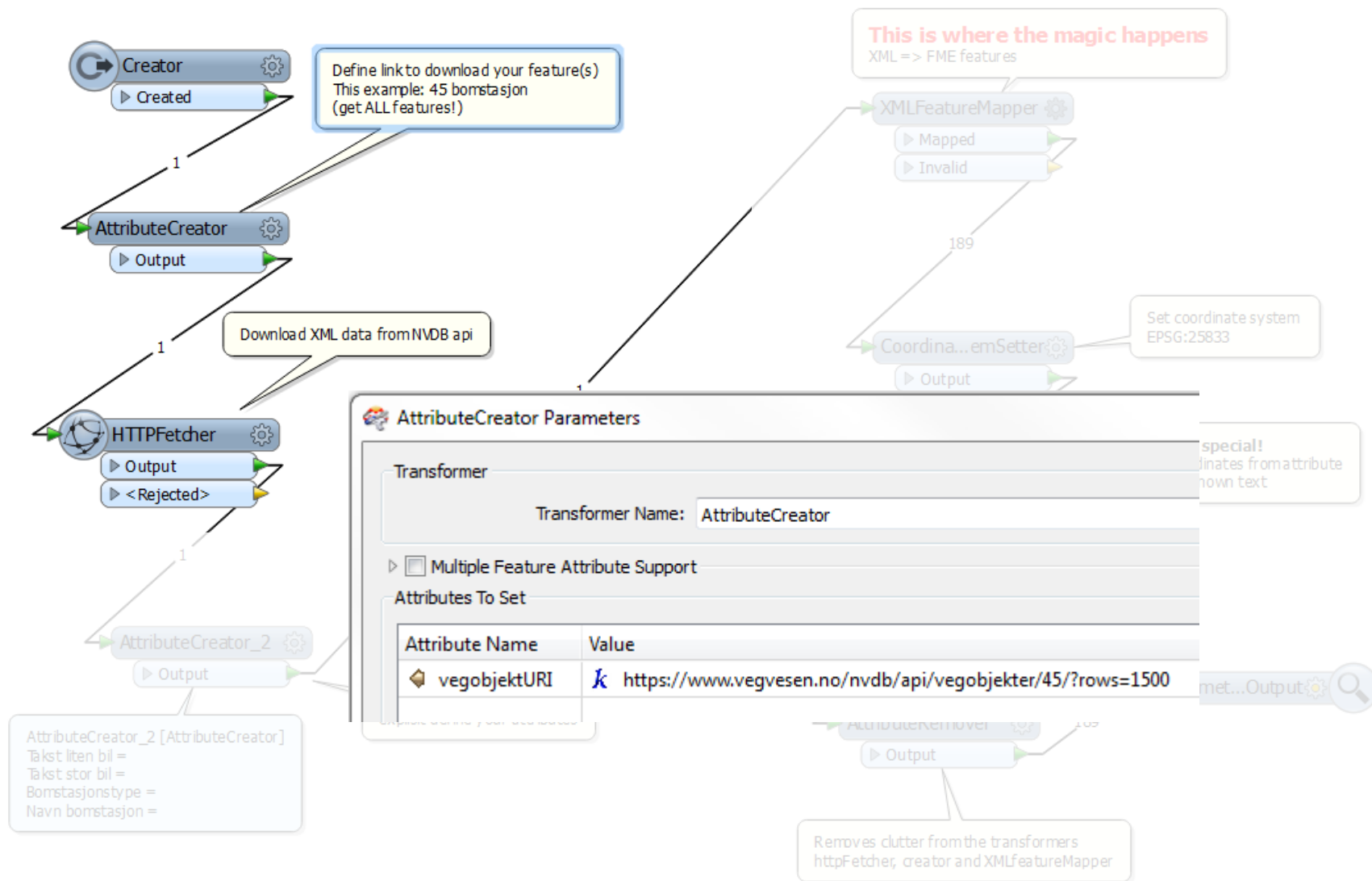
sok:{\"lokasjon\":{\"kommune\":[301]},\"objektTyper\":{\"id\":45,\"antall\":\"1 000\",\"filter\":{\"type\":\"vegreferanse\",\"operator\":\"=\",\"verdi\":[\"EV\"]}}}]

[https://www.vegvesen.no/nvdb/api/sok/?kriterie={\"lokasjon\":{\"kommune\":\[301\]},\"objektTyper\":{\"id\":45,\"antall\":\"1 000\",\"filter\":{\"type\":\"vegreferanse\",\"operator\":\"=\",\"verdi\":\[\"EV\"\]}}}\]](https://www.vegvesen.no/nvdb/api/sok/?kriterie={\)

## A little more detail



<https://www.vegvesen.no/nvdb/api/vegobjekter/45/?rows=1500>



# NVDB api – http headers

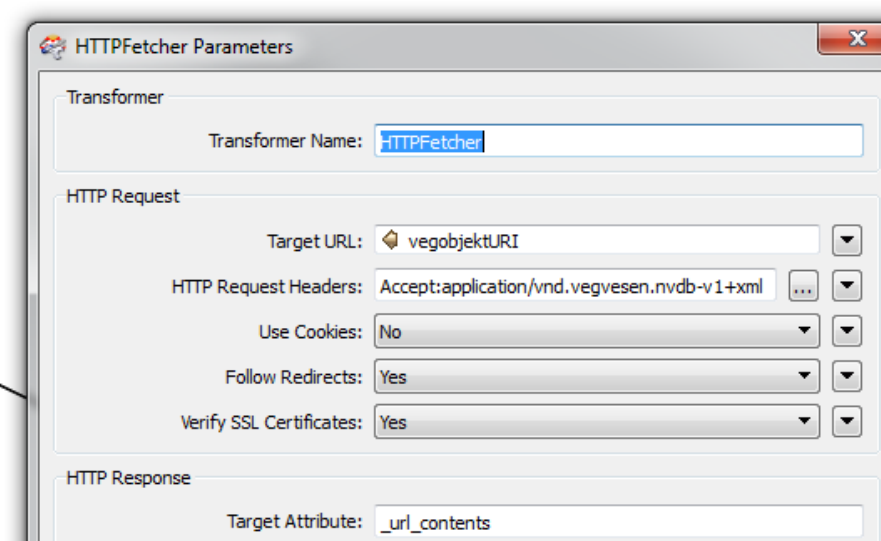
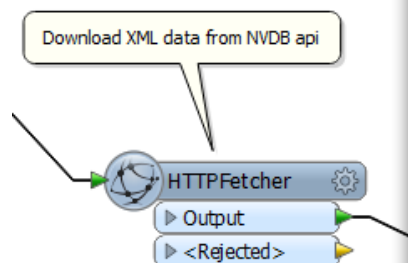
<https://www.vegvesen.no/nvdb/api/dokumentasjon>

## Mediatyper

REST-APIet støtter kommunikasjon basert på xml og json gjennom egendefinerte media-typer.

Klienter angir ønsket MediaType-representasjon fra serveren ved å sette "Accept"-header i request.

Type	NVDB REST-API MediaType
json	application/vnd.vegvesen.nvdb-v1+json
xml	application/vnd.vegvesen.nvdb-v1+xml







Statens vegvesen

# NVDB REST-API BETA

Tilgjengelige ressurser

Programmatisk ressurs-  
utledning

Media-typer

Geometrier

Feilhåndtering

Retningslinjer for klient-  
utvikling

NVDBs REST-API gir deg tilgang til alle vegobjekter som er lagret i NVDB samt tilhørende metadata om vegobjektene (datakatalogen)

## Tilgjengelige ressurser

VERB	URI	Beskrivelse	Dokumentasjon
GET	/	Gir en oversikt over tilgjengelige ressurser på rot-nivå, med tilhørende URI. Ressursene vil være datakatalog og vegobjekter (se under)	
GET	/datakatalog	En beskrivelse av hvilke vegobjekt-typer nvdb inneholder, og hvilke egenskaper disse har  En klient vil normalt først aksessere denne tjenesten for å få oversikt over hvilke data som kan hentes ut. Kjennskap til innholdet i datakatalogen er nødvendig for å kunne gjøre spørringer og hente ut data om vegobjekter.	dokumentasjon

# NVDB api/vegobjekter

<https://www.vegvesen.no/nvdb/api/dokumentasjon/vegobjekter>

## Tilgjengelige ressurser

Alle ressursene er relative til /vegobjekter.

VERB	URI	Beskrivelse
GET	/	Gir en oversikt over tilgjengelige ressurser under /vegobjekter , med tilhørende URI. Ressursene vil være tilgjengelige vegobjekter-av-gitt-type og spesifikk-vegobjekt
GET	/ {vegobjekt-type-id}	Gir en liste over alle tilgjengelige vegobjekter av en gitt type. Dette søket kan begrenses med query-restriksjoner; se "Resultatbegrensning" under.
GET	/objekt/ {vegobjekt-id}	Returnerer detaljerte data for et gitt vegobjekt

Example: All toll stations (Bomstasjoner)

<https://www.vegvesen.no/nvdb/api/vegobjekter/45/?rows=1500>

*(If you omit the «rows» part you get 100 items... )*

# NVDB api/vegobjekter: Explaining the «rows» part ...

<https://www.vegvesen.no/nvdb/api/dokumentasjon/vegobjekter>

## Resultatbegrensning

Antall resultatobjekter kan begrenses ved å angi "rows" som parameter.

<https://www.vegvesen.no/nvdb/api/vegobjekter/45/?rows=1500>

*(If you omit the «rows» part you get 100 items... )*

# NVDB api/vegobjekter **BoundingBox**

<https://www.vegvesen.no/nvdb/api/dokumentasjon/vegobjekter>

## Geografisk begrensning

Resultatet kan filtreres på geografisk område, dette gjøres ved å angi en "bounding box" med koordinater gitt i enten UTM33 eller WGS84 (Lat/Lon)

## Format

- UTM33: bbox={Xmin, Ymin, Xmax, Ymax}&srid=UTM33
- WGS84: bbox={longitudeMin, latitudeMin, longitudeMax, latitudeMax}&srid=WGS84

## Eksempel

- UTM33: /vegobjekter/45/?bbox=-43991.2,6715654.7,-26539.2,6726862.5&srid=UTM33
- WGS84: /vegobjekter/45/?bbox=5.159,60.211,5.441,60.333&srid=WGS84"

Dersom ikke srid angis, vil UTM33 bli brukt

Easy to implement in FME! *(left as excercise to the reader...)*

# NVDB api/vegobjekter

<https://www.vegvesen.no/nvdb/api/dokumentasjon/vegobjekter>

## Tilgjengelige ressurser

Alle ressursene er relative til /vegobjekter.

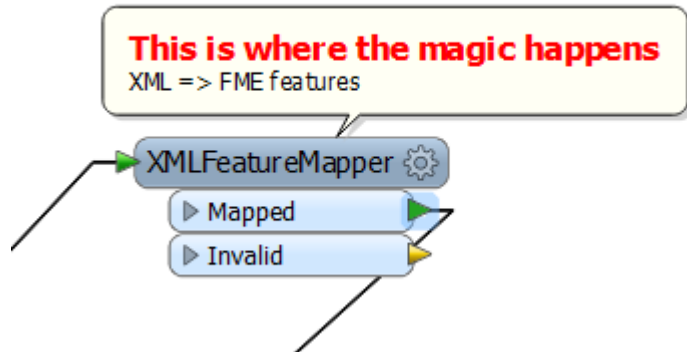
VERB	URI	Beskrivelse
GET	/	Gir en oversikt over tilgjengelige ressurser under /vegobjekter , med tilhørende URI. Ressursene vil være tilgjengelige vegobjekter-av-gitt-type og spesifikk-vegobjekt
GET	/ {vegobjekt-type-id}	Gir en liste over alle tilgjengelige vegobjekter av en gitt type. Dette søket kan begrenses med query-restriksjoner; se "Resultatbegrensning" under.
GET	/objekt/ {vegobjekt-id}	Returnerer detaljerte data for et gitt vegobjekt

Or you could ask for a specific objekt (by ID)

<https://www.vegvesen.no/nvdb/api/vegobjekter/objekt/487458624>

*We'll do some nifty cool things with this function!*

# What's inside XMLfeatureMapper ?



We take a look inside the xfmap definition

**XMLFeatureMapper Parameters**

**Transformer**

Transformer Name: XMLFeatureMapper

**XML Source**

XML Source Type: Attribute with XML document

XML File: <Unused>

XML Attribute: \_url\_contents

**Configuration**

Configuration Type: Single xfMap file

xfMap File: e:\WT\xfmapDef\_simplestExample\_v02.xml

xfMap Script: <Unused>

xfMap Attribute: <Unused>

xfMap File(s): <Unused>

XRS File: <Unused>

**Optional**

XML Source Name: \_xml\_source\_

XML Source Value: <Unused>

Feature Count Attribute: \_xml\_feature\_count\_

Feature Type Attribute: \_xml\_feature\_type\_

Merge Attributes From Input Feature: No

☒ Expose Attributes

Help Defaults OK Cancel



# Inside XMLfeatureMapper **xfsmap** definition

Telling xfmap to construct a feature each time it finds «vegObjekt»

```
<?xml version="1.0"?>
<xfMap>
  <feature-map multi-feature-construction="true">
    <mapping match="vegObjekt">
      <feature-type> <literal expr="vegObjekt"/> </feature-type>
    </mapping>
  </feature-map>
  <feature-content-map>
```

## The cool part: Extract values from a list of *egenskaper*

```
<!-- Generic, beautiful treatment of a dynamic list of properties
The "Egenskaper" element, that has 1 to N members -->
<mapping match="vegObjekt/egenskaper/egenskap">
  <attributes>
    <attribute>
      <name>    <extract expr="./navn"/> </name>
      <value>  <extract expr="./verdi"/> </value>
    </attribute>
  </attributes>
</mapping>
```

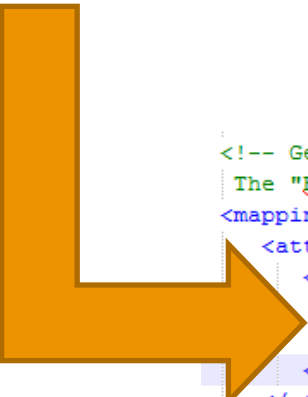
# How does that work?

← → ↻ <https://www.vegvesen.no/nvdb/api/vegobjekter/objekt/487458624>

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<vegObjekt>
  <assosiasjoner/>
  <definisjon uri="/datakatalog/objekttypen/45" typeNavn="Bomstasjon" typeId="45"/>
  ▼<egenskaper>
    ▼<egenskap>
      <definisjon uri="/datakatalog/egenskapstype/1078"/>
      <id>1078</id>
      <navn>Navn bomstasjon</navn>
      <verdi>Tungasletta</verdi>
    </egenskap>
    ▼<egenskap>
      <definisjon uri="/datakatalog/egenskapstype/1819"/>
      <enhet kortNavn="Kr" navn="Kroner" id="19"/>
      <id>1819</id>
      <navn>Takst stor bil</navn>
      <verdi>24.0</verdi>
    </egenskap>
    ▼<egenskap>
      <definisjon uri="/datakatalog/egenskapstype/1820"/>
      <enhet kortNavn="Kr" navn="Kroner" id="19"/>
      <id>1820</id>
      <navn>Takst liten bil</navn>
      <verdi>8.0</verdi>
    </egenskap>
  </egenskaper>
</vegObjekt>
```

Attribute name = «Takst liten bil»  
Attribute value = 8



```
<!-- Generic, beautiful treatment of a dynamic list of properties
The "Egenskaper" element, that has 1 to N members -->
<mapping match="vegObjekt/egenskaper/egenskap">
  <attributes>
    <attribute>
      <name> <extract expr="./navn"/> </name>
      <value> <extract expr="./verdi"/> </value>
    </attribute>
  </attributes>
</mapping>
```

# Explicit treatment of some standard elements (at root level, or within the «lokasjon» element)

```
<modifisert>2014-03-19T09:46:21+01:00</modifisert>
<objektId>487458624</objektId>
<objektTypeId>45</objektTypeId>
<objektTypeNavn>Bomstasjon</objektTypeNavn>
<self uri="/vegobjekter/objekt/487458624" typeNavn="Bomstasjon" typeId="45"/>
<startDato>2014-03-10</startDato>
▼<lokasjon>
  <fylke nummer="16" navn="SØR-TRØNDELAG"/>
  <geometriUtm33>POINT (273611.525299889 7040653.669017122)</geometriUtm33>
  <geometriWgs84>POINT (10.462096644243022 63.42219196530382)</geometriWgs84>
  <kommune nummer="1601" navn="TRONDHEIM"/>
  <kontraktsomrader/>
  <politiDistrikt nummer="20" navn="Sør-Trøndelag"/>
  <region nummer="4" navn="MIDT"/>
  <riksvegruter/>
  <vegAvdeling nummer="16" navn="Sør-Trøndelag"/>
```

```
<!--  
Matching some straightforward elements at root level.  
These all have the <local-name>Value</local-name> structure  
-->
```

```
<mapping match="modifisert objektId objektTypeId objektTypeNavn  
sluttDato startDato strekningslengde">  
  <attributes>  
    <attribute>  
      <name> <matched expr="local-name"/> </name>  
      <value> <extract expr="."/> </value>  
    </attribute>  
  </attributes>  
</mapping>
```

Attribute names & values  
are part of XML-tag  
*<tagname>value<tagname>*

```
<modifisert>2014-03-19T09:46:21+01:00</modifisert>  
<objektId>487458624</objektId>  
<objektTypeId>45</objektTypeId>  
<objektTypeNavn>Bomstasjon</objektTypeNavn>  
<self uri="/vegobjekter/objekt/487458624" typeNavn="Bomstasjon" typeId="45"/>  
<startDato>2014-03-10</startDato>  
▼<lokasjon>  
  <fylke nummer="16" navn="SØR-TRØNDELAG"/>  
  <geometriUtm33>POINT (273611.525299889 7040653.669017122)</geometriUtm33>  
  <geometriWgs84>POINT (10.462096644243022 63.42219196530382)</geometriWgs84>  
  <kommune nummer="1601" navn="TRONDHEIM"/>  
  <kontraktsomrader/>  
  <politiDistrikt nummer="20" navn="Sør-Trøndelag"/>  
  <region nummer="4" navn="MIDT"/>  
  <riksvegruter/>  
  <vegAvdeling nummer="16" navn="Sør-Trøndelag"/>
```

```
<!-- Politidistrikt, kommune and region -->  
<mapping match="vegObjekt/lokasjon/politiDistrikt  
vegObjekt/lokasjon/region  
vegObjekt/lokasjon/kommune">  
  <attributes>  
    <attribute>  
      <name> <matched expr="local-name"/> </name>  
      <value> <extract expr="@navn"/> </value>  
    </attribute>  
  </attributes>  
</mapping>
```

Attribute names & values  
are part of XML-tag  
*<tagname navn="value">*

# Treatment of lists

```
<!-- Some location attributes are lists -->
<mapping match="vegObjekt/lokasjon/kontraktsomrader/kontraktsomrade
vegObjekt/lokasjon/riksvegruter/riksvegrute">
  <attributes>
    <attribute type="list">
      <name>    <matched expr="local-name"/> </name>
      <value>  <extract expr="."/> </value>
    </attribute>
  </attributes>
</mapping>
```

```
▼<kontraktsomrader>
  <kontraktsomrade navn="0201 ASKER OG BÆRUM 2013-2018"/>
</kontraktsomrader>
<politiDistrikt nummer="10" navn="Asker og Bærum"/>
<region nummer="1" navn="ØST"/>
▼<riksvegruter>
  <riksvegrute nummer="5C" navn="RUTE5C"/>
</riksvegruter>
```

(Example from

<https://www.vegvesen.no/nvdb/api/vegobjekter/objekt/225760360> )



# Hardwiring list of linear references

```
<!-- Hard-wiring RoadLink linear reference -->
<mapping match="vegObjekt/lokasjon/veglenker/veglenke">
  <attributes>
    <attribute type="list">
      <name>    <literal expr="veglenke_Id"/>    </name>
      <value>  <extract expr="@id"/>              </value>
    </attribute>
    <attribute type="list">
      <name>    <literal expr="veglenke_til"/>    </name>
```

```
▼<vegReferanser>
  ▼<vegReferanse>
    <fraMeter>0</fraMeter>
    <fylke>2</fylke>
    <hp>7</hp>
    <kategori>E</kategori>
    <kommune>0</kommune>
    <nummer>18</nummer>
    <status>V</status>
    <tilMeter>4580</tilMeter>
  </vegReferanse>
</vegReferanser>
▼<veglenker>
  <veglenke til="0.621406" sidepos="NULL" id="444232" fra="0.13197" direction="WITH"/>
</veglenker>
```

Enter Values for Attributes to Expose

ER_DEL_AV{}
politiDistrikt
region
geometriUtm33
kontraktsomrade{}
riksveggrute{}
vegreferanse_fraMeter{}
vegreferanse_tilMeter{}
vegreferanse_fylke{}
vegreferanse_hp{}
vegreferanse_kategori{}
vegreferanse_kommune{}
vegreferanse_nummer{}
vegreferanse_status{}
veglenke_Id{}
veglenke_til{}
veglenke_fra{}
veglenke_sidepos{}
veglenke_felt{}
veglenke_direction{}

+ - ▲ ▼

OK Cancel

# The übercool treatment of associations

Composition (consist of)  
Also called *daughter objects*



```
<assosiasjoner>
▼<assosiasjon>
  <relasjon uri="/vegobjekter/objekt/225760347" typeNavn="Temperaturmåler" typeId="463" rel="BESTAR_AV" kardinalitet="EN_TIL_MANGE"/>
</assosiasjon>
▼<assosiasjon>
  <relasjon uri="/vegobjekter/objekt/225760310" typeNavn="Lysmåler" typeId="459" rel="BESTAR_AV" kardinalitet="EN_TIL_MANGE"/>
</assosiasjon>
▼<assosiasjon>
  <relasjon uri="/vegobjekter/objekt/225760327" typeNavn="Telefonsentral" typeId="473" rel="BESTAR_AV" kardinalitet="EN_TIL_MANGE"/>
</assosiasjon>
▼<assosiasjon>
  <relasjon uri="/vegobjekter/objekt/225760269" typeNavn="Tunnel" typeId="581" rel="ER_DEL_AV" kardinalitet="MANGE_TIL_EN"/>
</assosiasjon>
</assosiasjoner>
```

Composition (is part of)  
Also called *Mother*

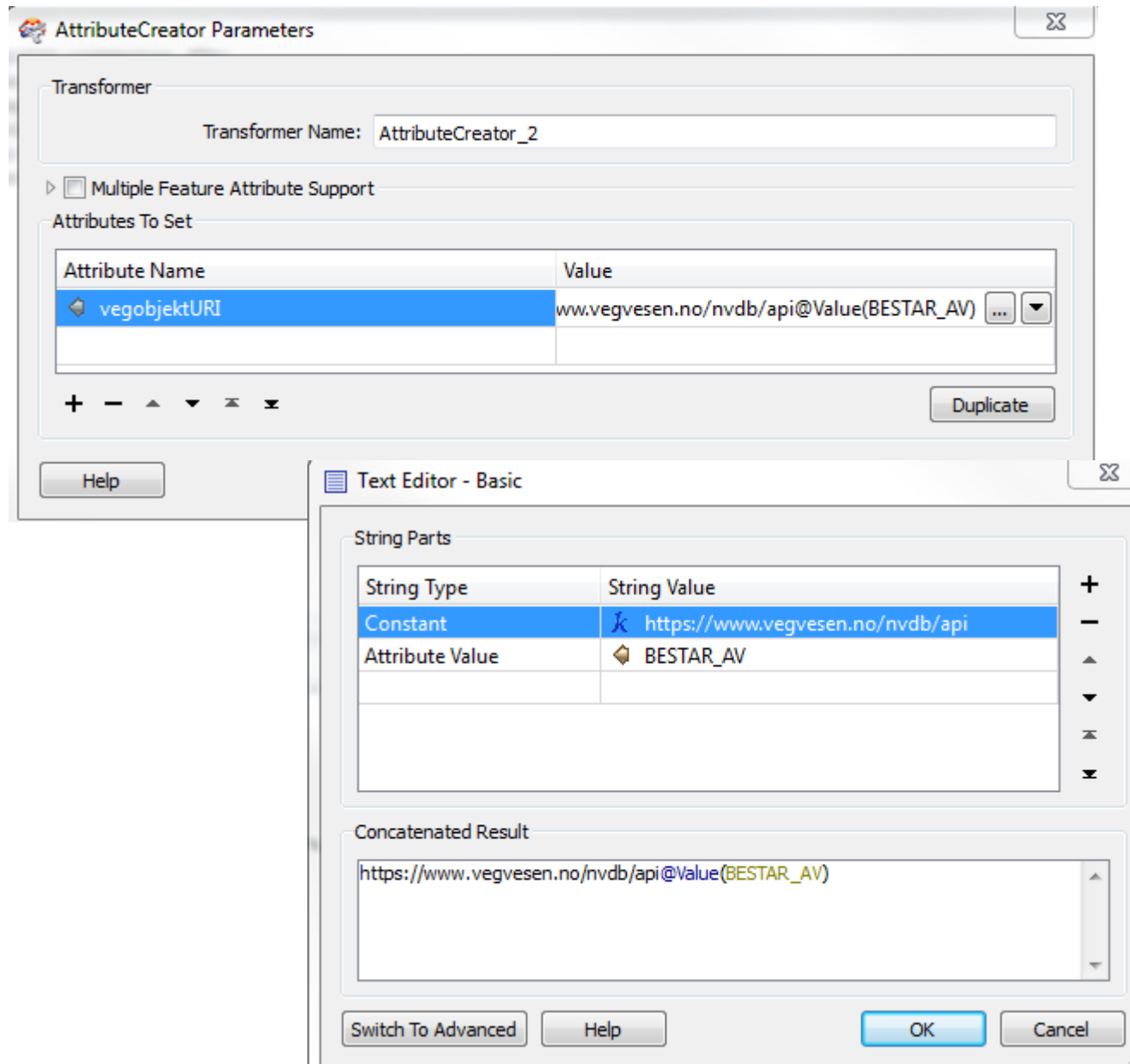


```
<mapping match="vegObjekt/assosiasjoner/assosiasjon/relasjon">
  <attributes>
    <attribute type="list">
      <name> <extract expr="@rel"/> </name>
      <value> <extract expr="@uri"/> </value>
    </attribute>
  </attributes>
</mapping>
```

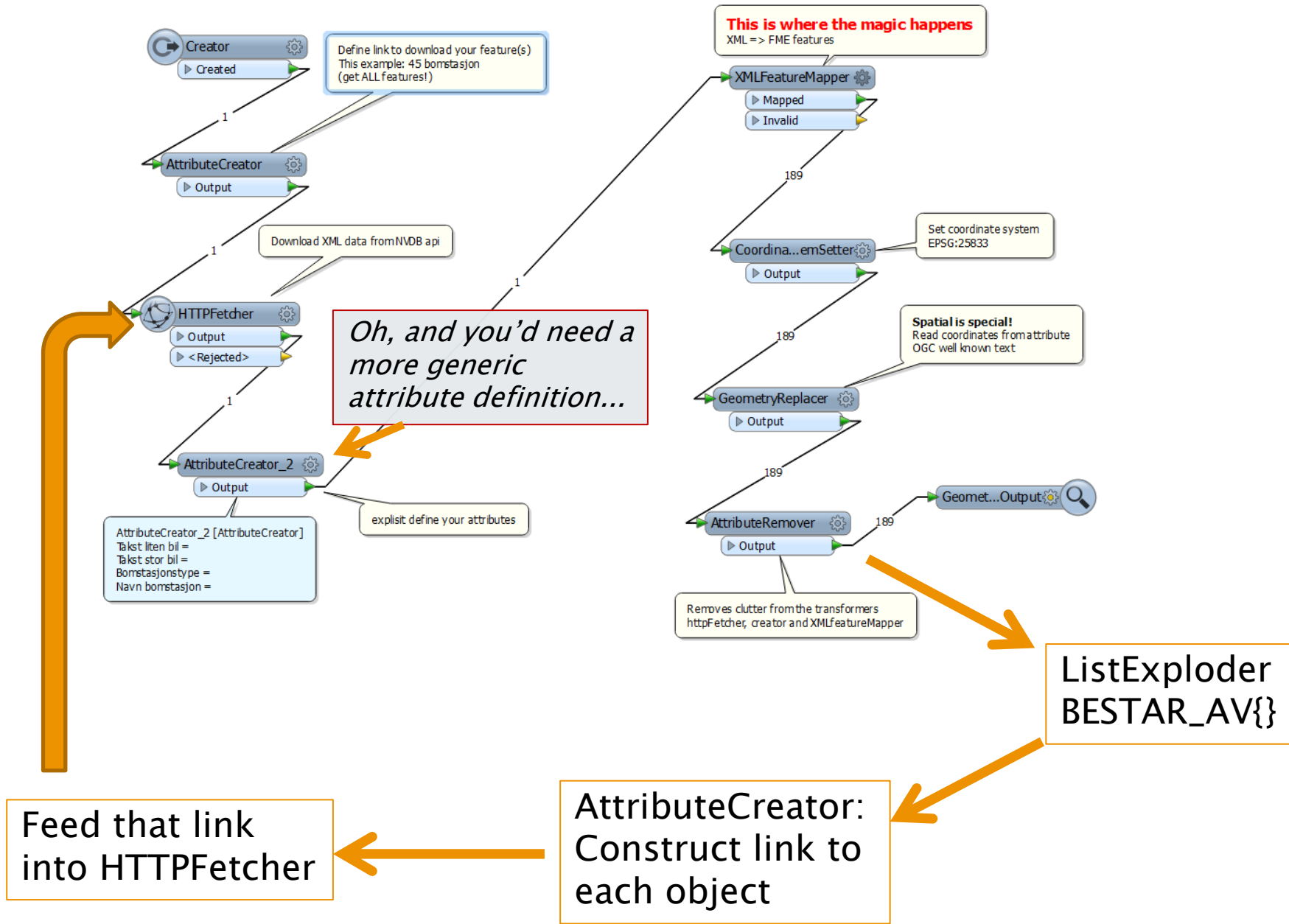
Produces lists  
BESTAR\_AV{}  
ER\_DEL\_AV{}



# Making valid URI to each object is supereasy

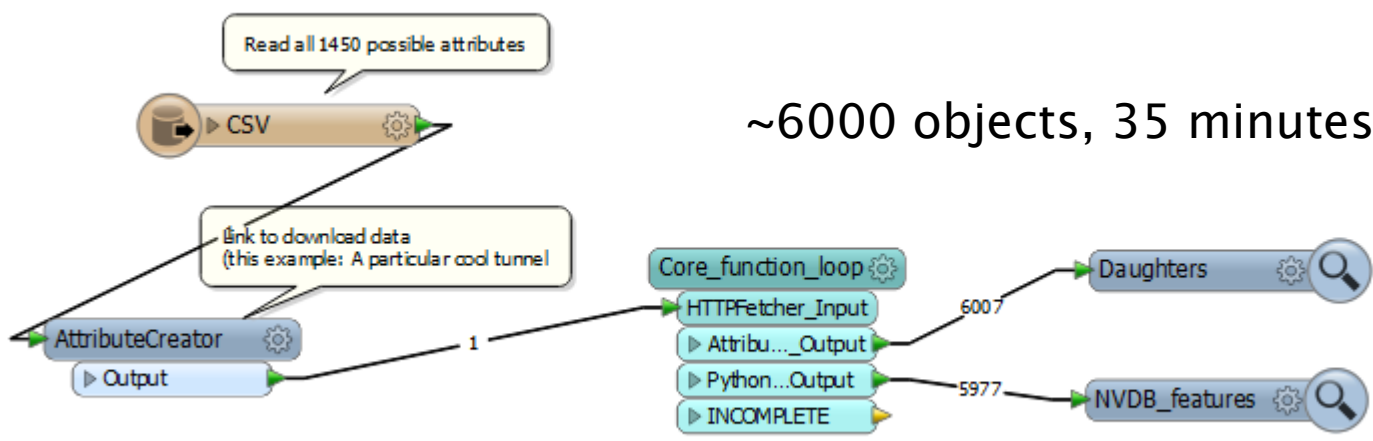
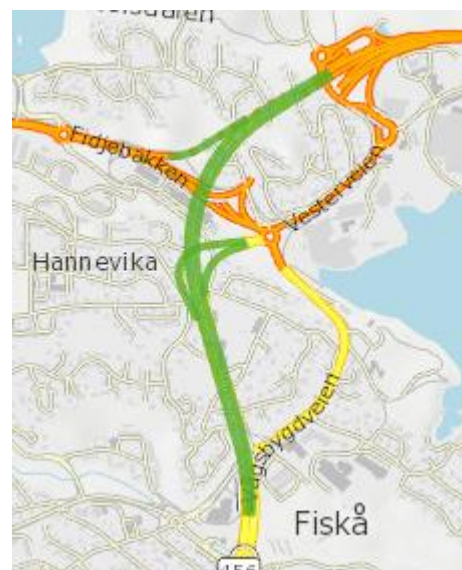


# Get all «daughter» and «granddaughter» objects recursively



# A rather hairy recursive example... Vågsbygdporten tunnel

<https://www.vegvesen.no/nvdb/api/vegobjekter/objekt/359603347>



~6000 objects, 35 minutes



# The uncool parts

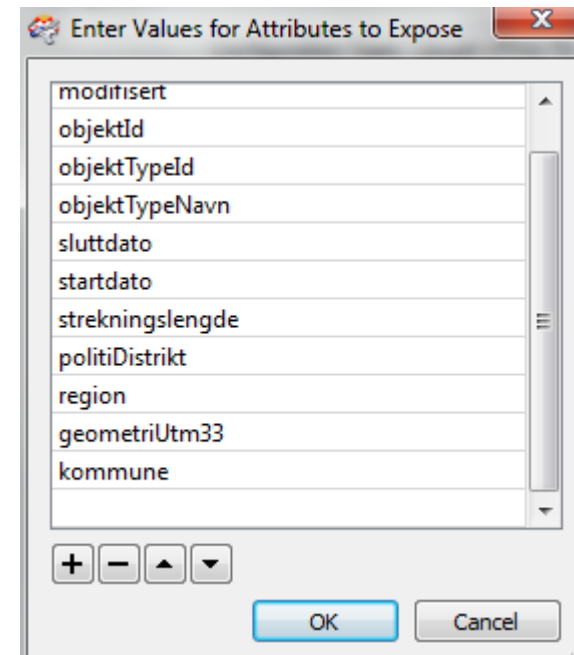
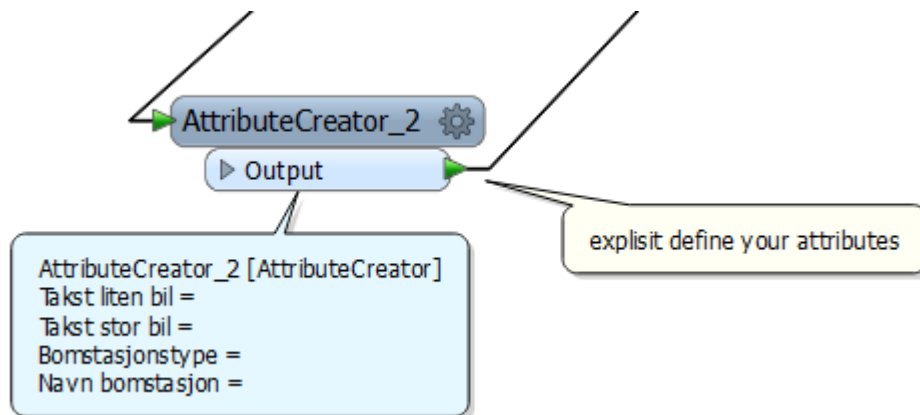




# 370 different objects

## 1750 different attributes

*(not counting the «standard» elements (road reference, region, police district, county, ... )*



**This gets old really fast!**

FME attribute names must be known to the workspace definition if you want to do cool stuff with them.

Besides the standard *attribute Creation/exposing* – methods, there are more or less cludgy ways of making attributes spring into existence

- *FeatureReader? Dynamically write feature type definition to a file (with python?) WorkspaceRunner?*

I've yet to see a nice, elegant way of true dynamic way of creating schema definitions

- *But I do have hopes for FME schema reader...*

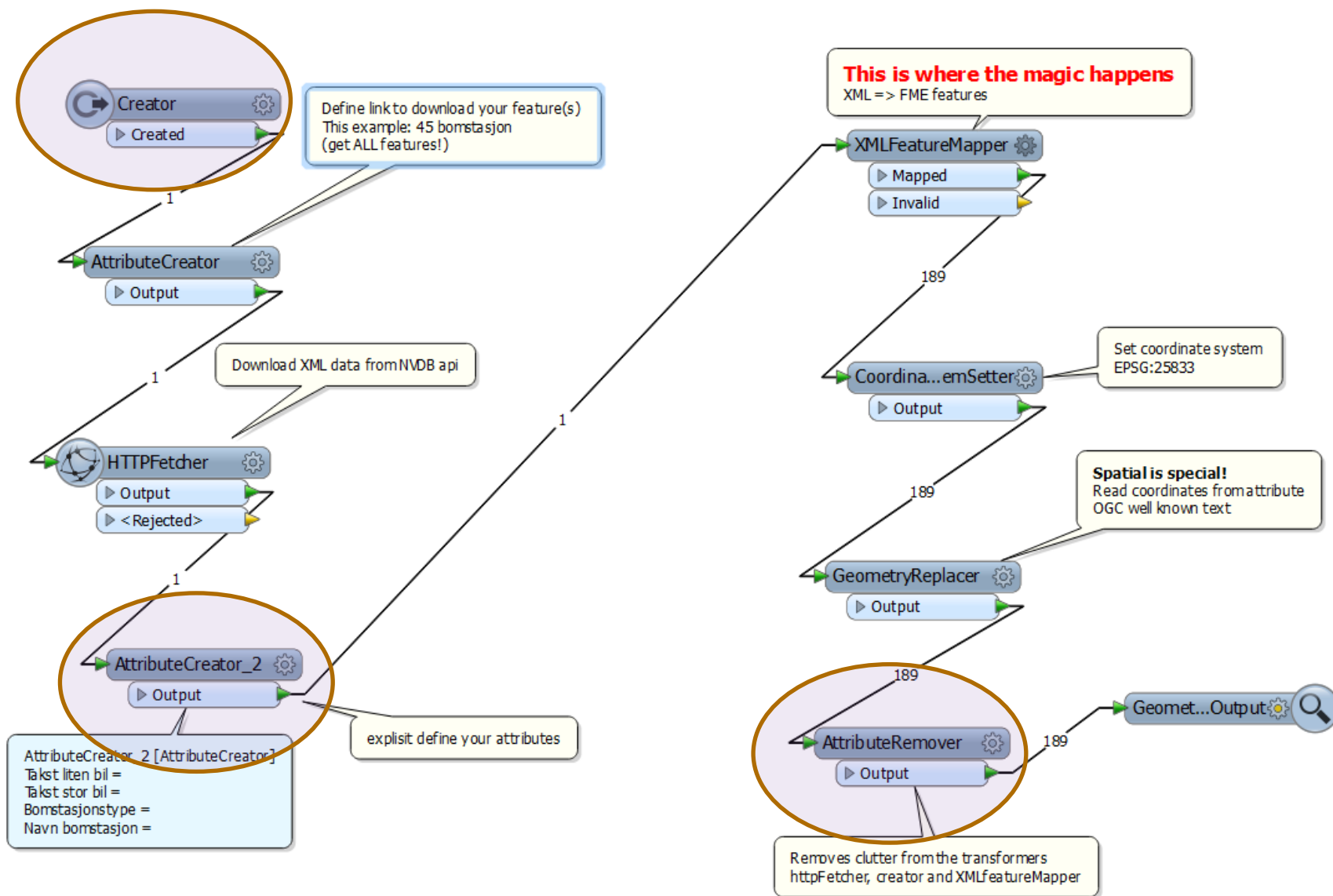


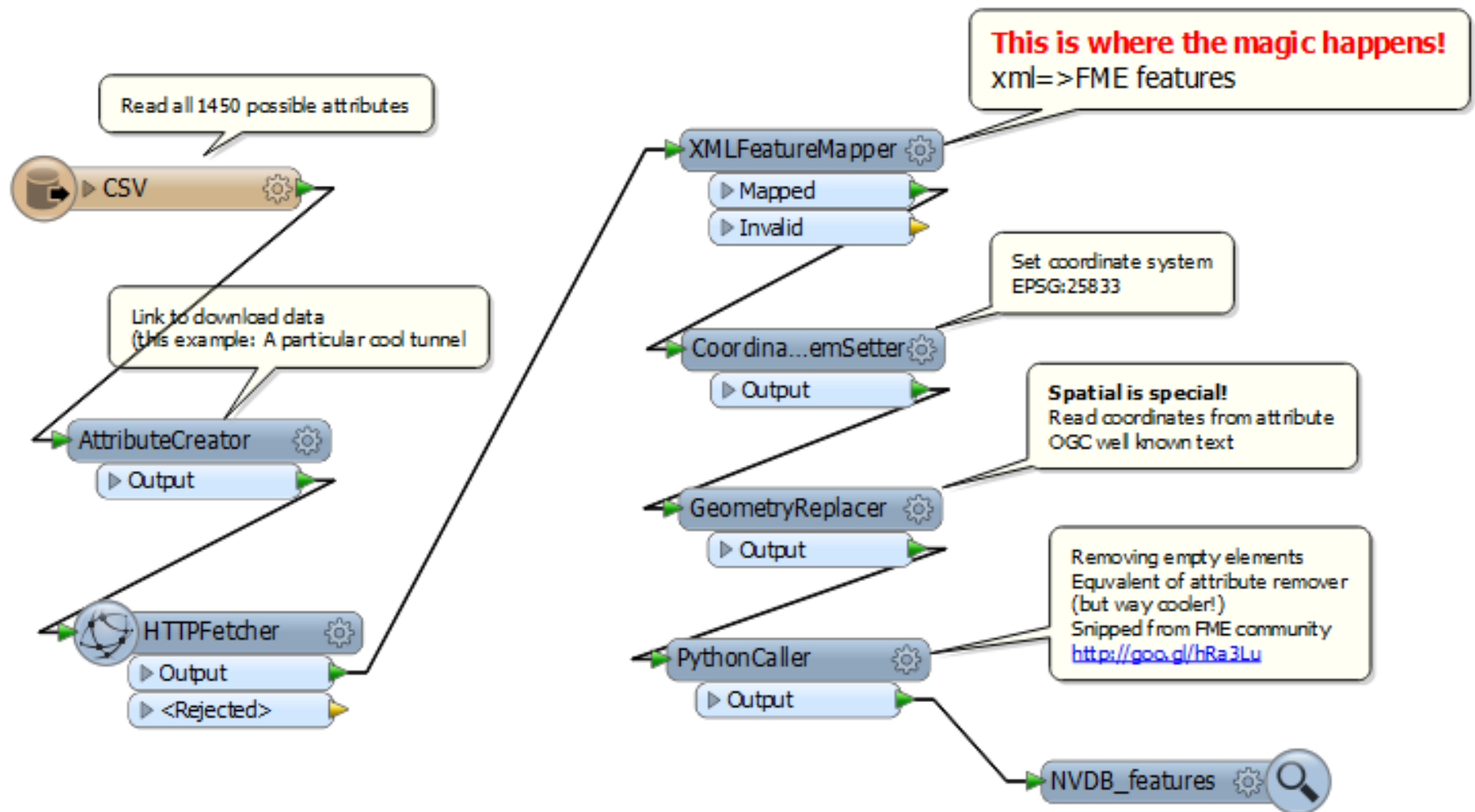


## My own half-cludgy way around these limitations

1. Python routine:  
*Read data catalogue definition*  
<https://www.vegvesen.no/nvdb/api/datakatalog>  
*Write attributes to CSV file*
2. Read that definition into FME
3. Do the XMLFeatureReader-magic
4. Get rid of empty attributes (python caller)  
Example from FME community answers  
<http://goo.gl/hRa3Lu>  
*(Easy way to remove all Empty attributes)*







## Recommendation 1:

- NVDB is data catalogue driven.
  - Data catalogue updated about 4–5 times/year
  - You should verify / update feature definition accordingly  
Best solution: Feature definition created dynamically based on <https://www.vegvesen.no/nvdb/api/datakatalog>

*Any tips on how to achieve this is greatly appreciated!*



## Data catalogue resources

- The official data catalogue

<http://www.vegvesen.no/Fag/Teknologi/Nasjonal+vegdatabank/Datakatalogen>

- Data catalogue through NVDB api

<https://www.vegvesen.no/nvdb/api/dokumentasjon/datakatalog>

... *which again is used by these implementations:*

- Data catalogue embedded in Vegkart

<https://www.vegvesen.no/vegkart/vegkart/>

- Our own playground example

<http://labs.vegdata.no/nvdb-datakatalog/>





## Recommendation 2:

- **KISS – as simple as possible**
- If you're happy with a (very) limited subset of object types and/or attributes from NVDB
  - my «simplest possible» workspace should be a good starting point
  - *FMEWT\_simplestExample.fmw*
  - *FMEWT\_apisearch.fmw*



## Recommendation 3:

- **... but not simpler**

- When you're tired of «attributeCreator» the more generic workspace should provide fertile starting ground.
  - Far from perfect (cludgy, in fact), but easy to improve upon & adapt
  - *FMEWT\_genericExample.fmw*



## Recommendation 4:

# Learn python

- It is super fun – and it will greatly enhance your FME skills!
- <https://www.pythonanywhere.com/> is a truly great place to start
  - » And NVDB api is white listed, so you can access it from a free account!  
<https://www.pythonanywhere.com/forums/topic/1290/>



## Recommendation 5:

# These examples provide messy attribute names

- UTF-8 encoding, white space, non-ascii Norwegian characters (Æ, Ø, Å).
- Not all writers and formats are too happy about that...
  - » *AttributeRenamer is your friend...*
- The «NVDB» way is to always refer to the ID of the attribute definition. (The name is supplementary).

Have fun!



Statens vegvesen

<http://vegdata.no>

<= our blog

<http://labs.vegdata.no>

<= experimental stuff

[https://github.com/LtGlahn/vegdata labs\\_fme](https://github.com/LtGlahn/vegdata labs_fme)

