Statens vegvesen

Cool things with Norwegian National road data base (NVDB)

Jan Kristian Jensen
jan.kristian.jensen@vegvesen.no

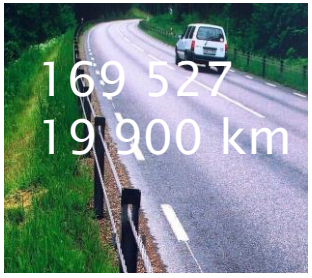
@LtGlahn
@nvdbapi

NVDB

All roads > 50m *(for motor vehicles)*

253.000 km

Primary source for road network data, navigation +++



Inventory – physical equipment

Data coverage:
Mainly along major roads

E, R, F categories (Our responsibility)

Communal roads

more spotty

191.000 km.

Administrative + statistics

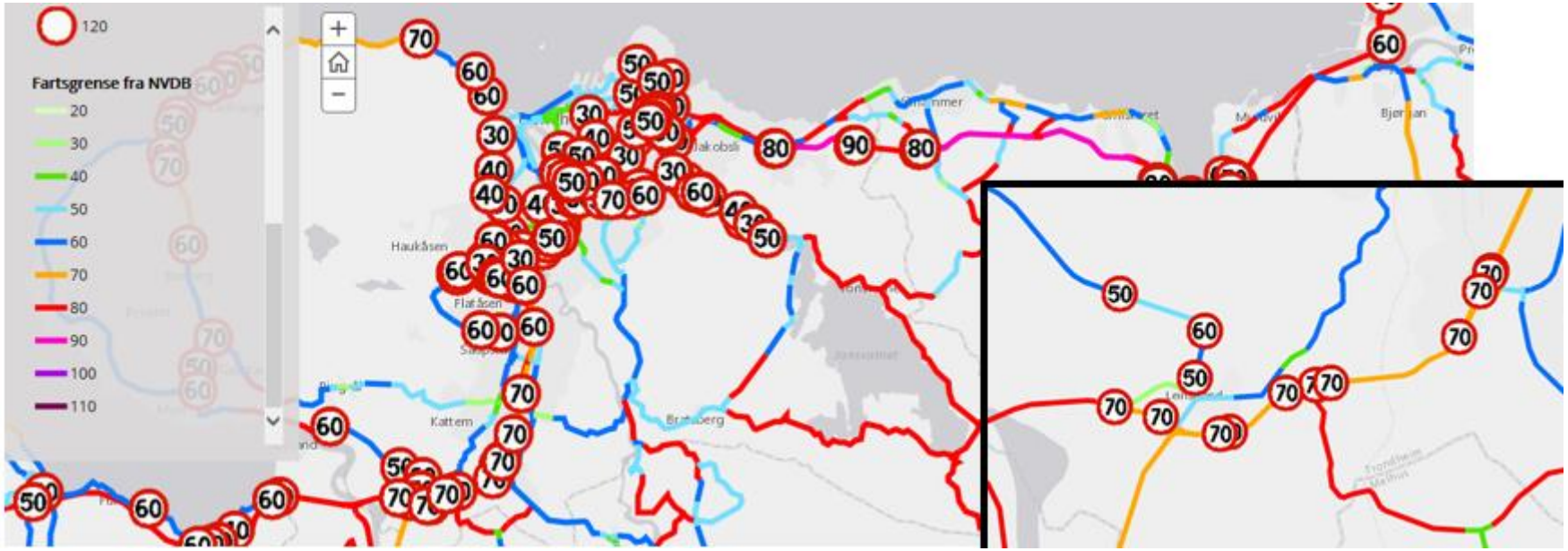
Who maintains the road?

How important is it?

Regulations

Traffic accidents

How much traffic?



Topopogical road network

385 data types, defined in data catalogue

NVDB – data served in:

- **Different services**
- **Different flavors**
- **Different formats**

Are you interested in…?

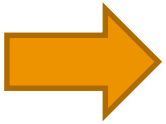
Navigation?

- Optimum route?
- Travel distance / time?
- Topological data
- Road network analysis

Are you interested in…?

Navigation?

- Optimum route?
- Travel distance / time?
- Topological data
- Road network analysis



Pre-made navigation data
- *ELVEG*
- *Route planner data set*

Elveg

Geonorge ▸ Kartkatalogen ▸ Elveg

https://kartkatalog.geonorge.no/search?text=elveg
Official road data + adresses. Sosi format.

https://www.vegdata.no/2013/08/08/hvor-finner-jeg-vegnettsdata-til-navigasjon/
.gdb, sqlite format => used with pgrouting

Official product

Elveg – road network for navigasjon

- Versjoned, 10 times each year
- File based, SOSI format
  - Road network with restrictions
  - Selected NVDB data (speed limits m.m.)
  - Adresses
  - Road names

More info:
https://www.vegdata.no/2014/08/15/vegdata-for-navigasjon/

Cant read SOSI???

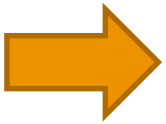
- SOSI =Ascii text format, Norwegian standard, from 1987
- Vegvesen route planning application API
  - Used in http://vegvesen.no/trafikk
  - http://data.norge.no/data/statens-vegvesen/api-ruteplantjeneste-bil
- Open data policy: What we use => we publish!
  - https://www.vegdata.no/2013/08/08/hvor-finner-jeg-vegnettsdata-til-navigasjon/
  - .gdb and spatialLite – format
  - Produced in parallell with Elveg (x10/year)

Are you interested in…?

Navigation?

- Optimum route?
- Travel distance / time?
- Topological data
- Road network analysis

Pre-made navigation data
ELVEG
Route planner data set



Route planner API

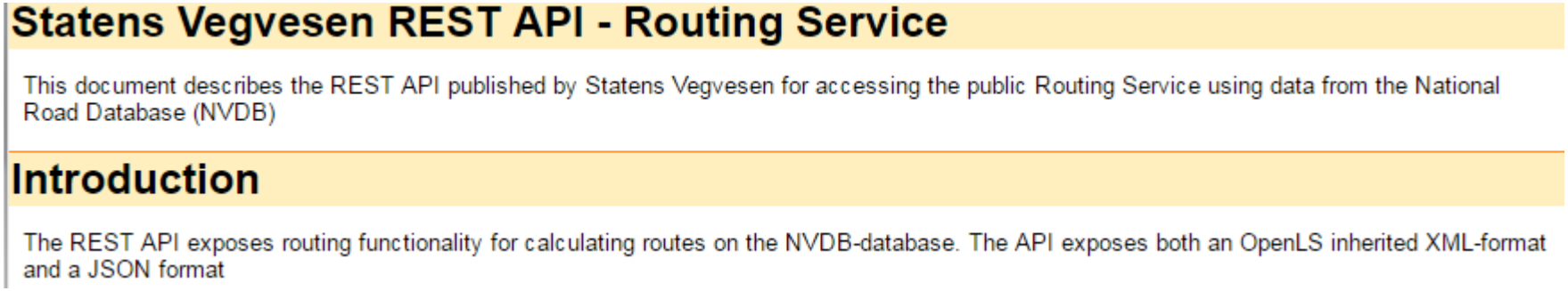
http://data.norge.no/data/statens-vegvesen/api-ruteplantjeneste-bil
REST api for A => B navigation, up to 8 wp'ts. (via-points).
JSON, XML

Route planner API

Open API, but you need an account…

2500 req/day.

http://data.norge.no/data/statens-vegvesen/api-ruteplantjeneste-bil



Statens Vegvesen REST API - Routing Service

This document describes the REST API published by Statens Vegvesen for accessing the public Routing Service using data from the National Road Database (NVDB)

Introduction

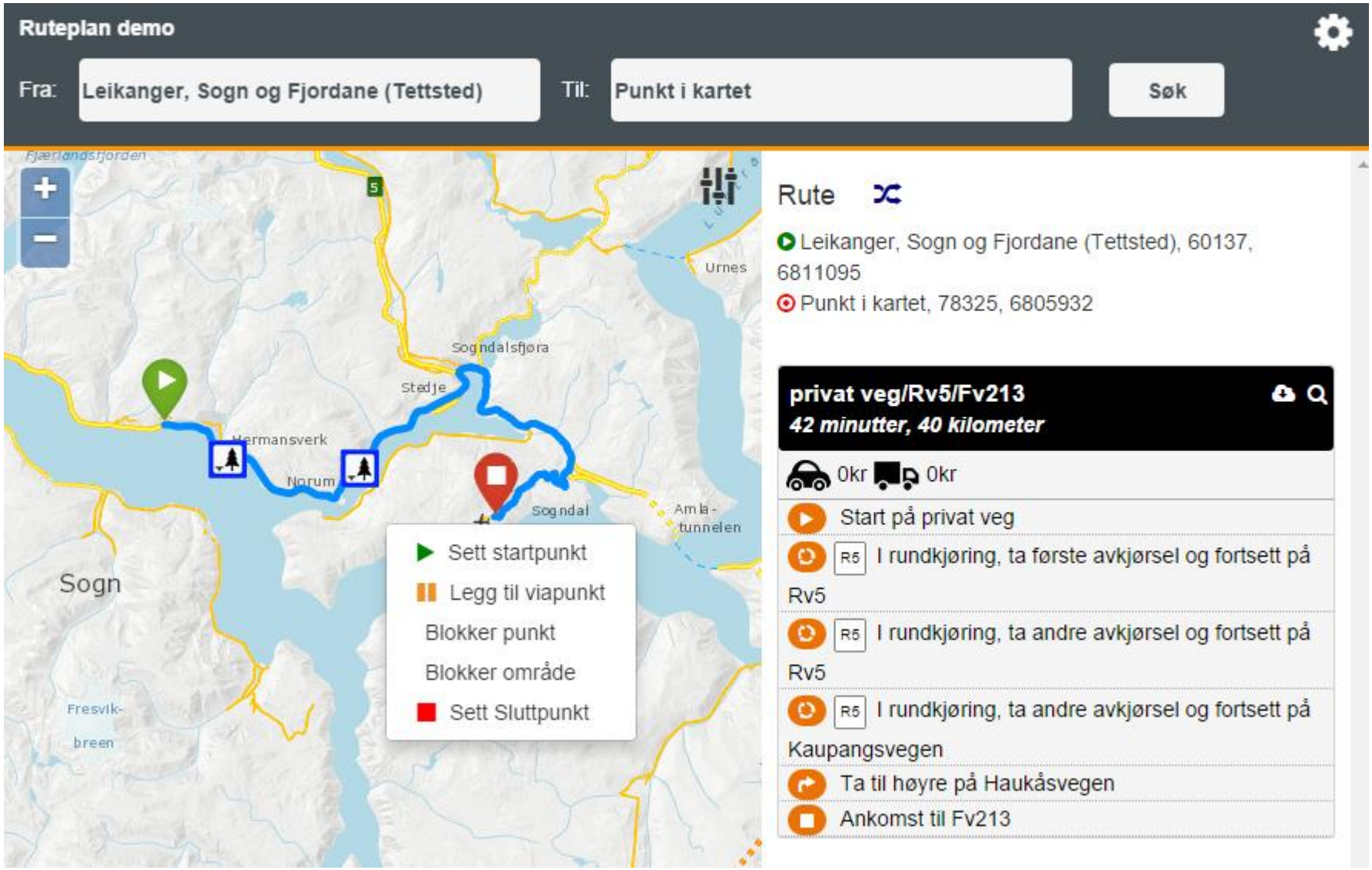
The REST API exposes routing functionality for calculating routes on the NVDB-database. The API exposes both an OpenLS inherited XML-format and a JSON format

Good examples in the documentation –straigthforward REST API

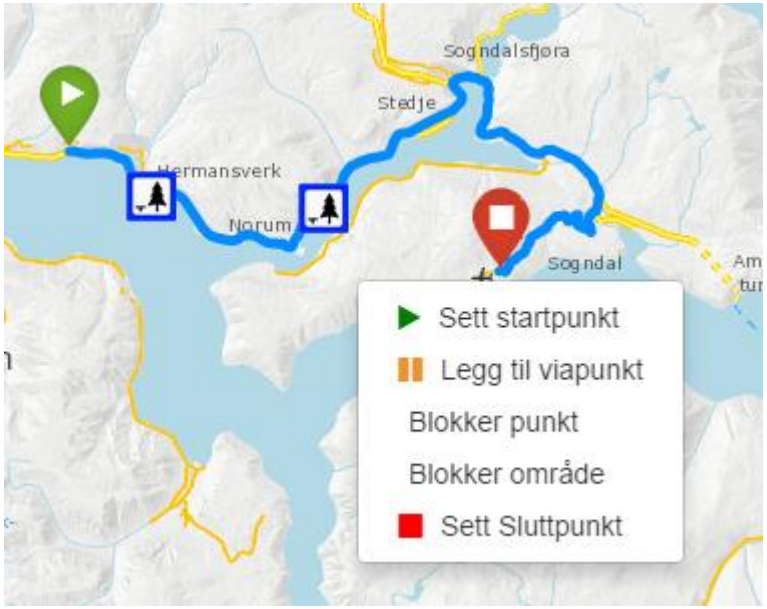
Basic http auth, not very JS / CORS – friendly

Test/demo client, Route planner API

http://svvruteplan.azurewebsites.net/#/



Test/demo client, Route planner API



https://github.com/petlof/svv-ruteplan-demo

More information:
https://www.vegdata.no/2014/09/18/vegvesenets-forste-apen-kildekode-prosjekt/

«Code dump», not really FOSS…

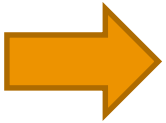
Are you interested in…?

Navigation?

- Optimum route?
- Travel distance / time?
- Topological data
- Road network analysis

Pre-made navigation data
ELVEG
Route planner data set

Route planner API



NVDB API/vegnett
Topological network

https://www.vegvesen.no/nvdb/apidokumentasjon/#/get/vegnett

NVDB api – Topological network

https://www.vegvesen.no/nvdb/apidokumentasjon/#/get/vegnett

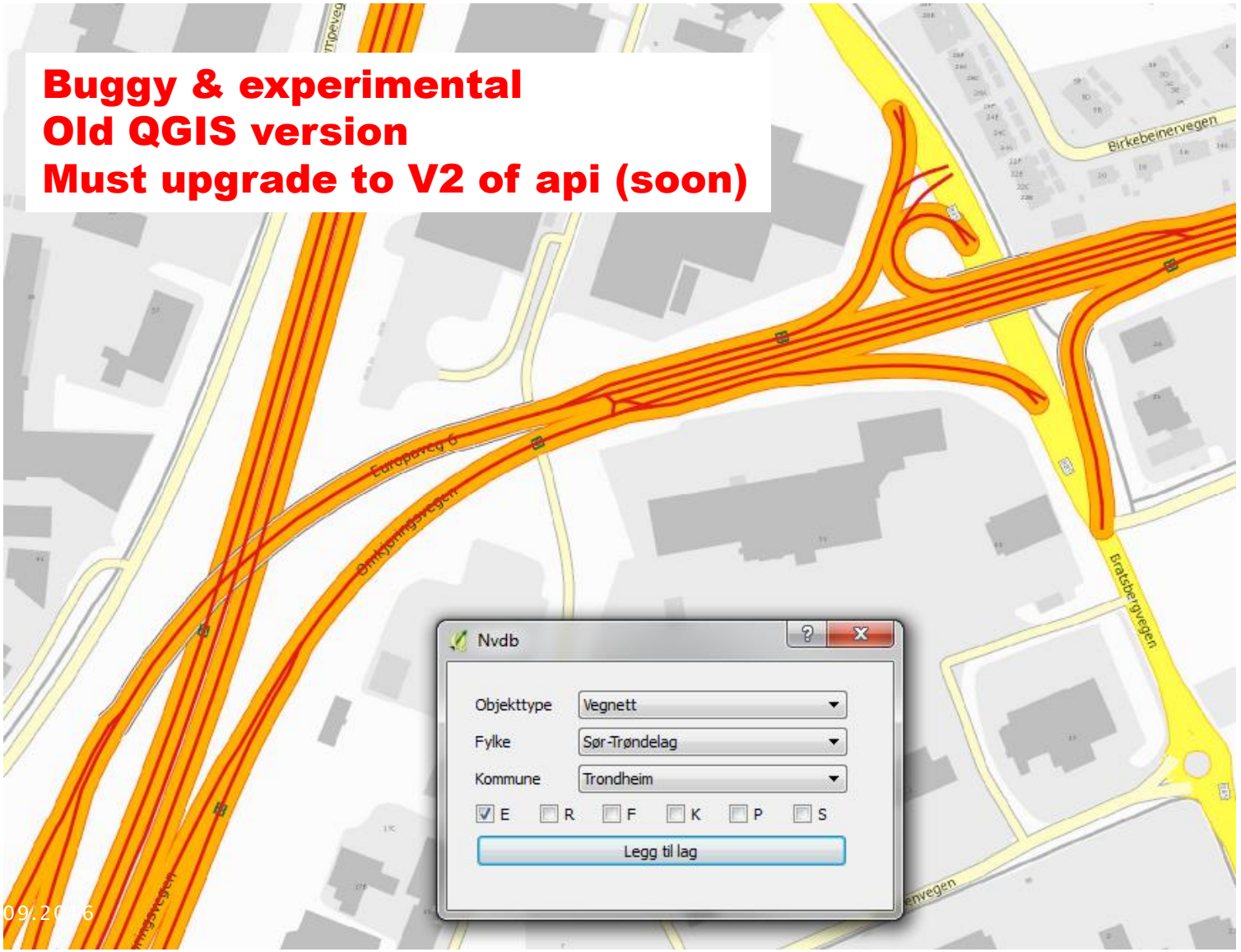
Only the links – but nodes are referenced
Enough to reverse-engineer topology
No turn restrictions etc => **not suitable for navigation**

```
GET https://www.vegvesen.no/nvdb/api/v2/vegnett/lenker
```


```
{
    "objekter": [
        {
            "id": 1000,
            "href": "https://www.vegvesen.no/nvdb/api/v2/vegnett/lenker/1000",
            "metadata": {
                "startdato": "1950-01-01",
            },
            "fra_posisjon": 0,
            "til_posisjon": 1,
            "startnode": "34674-1",
            "sluttnode": "1272325-1",
            "felt": "1#2",
            "medium": "T",
            "vegtype": 7001,
            "konnekteringslenke": false,
            "topologinivå": 0,
            "topologinivå_tekst": "Vegtrasé",
            "vegsegment": {
                "geometri": {
                    "wkt": "LINESTRING (147939.8 6514243.8 27.1, 147944.1 6514241.5
```

QGIS prototype – road network plugin

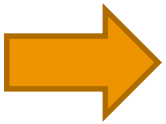
https://github.com/nvdb-vegdata/nvdb-qgis



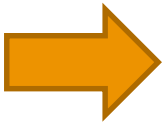
Are you interested in…?

Navigation? Road network data?

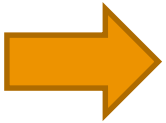
- Optimum route?
- Travel distance / time?
- Topological data
- Road network analysis



Pre-made navigation data
ELVEG
Route planner data set



Route planner API
Rest API
A=>via C => B



NVDB API/vegnett
Topological network

Are you interested in…?

~~Navigation?~~ Everything else!

- 385 object types
- NVDB api is excellent!
  - https://www.vegvesen.no/nvdb/apidokumentasjon/
- Get friends with NVDB using Vegkart
  - http://vegkart.no

V2 of NVDB api

https://www.vegvesen.no/nvdb/apidokumentasjon/

Nasjonal vegdatabank API

Statens vegvesen tilbyr et REST-basert API som kan benyttes for å få tilgang til informasjonen som befinner seg i Nasjonal vegdatabank (NVDB).

Tips for å komme raskt igang

1. Besøk Vegkart for å få innblikk i hvilke data som er tilgjengelig.
2. Les Om Nasjonal vegdatabank for å bli kjent med NVDBs datastruktur.
3. Utforsk endepunktene i APIet gjennom nettleseren, og bli motivert til å fortsette.
4. Les Bruk av NVDB API for å lære hvordan APIet er ment å brukes.
5. Om du skriver i Java, kan du bruke bruke NVDB API Client som er skrevet som åpen kildekode.

NVDB API versjon 2

Versjon 2 er gjeldende versjon av NVDB API, og er dokumentert på dette nettstedet.

NVDB API v2 er tilgjengelig på følgende adresse:

```
https://www.vegvesen.no/nvdb/api/v2
```

V2 of NVDB api

https://www.vegvesen.no/nvdb/apidokumentasjon/

Developer friendly

More REST-full

Powerful **filters**

More intiutive
- object type definiton ⇔ Data queries
- Filtering

Versioning / change log (changes after…)

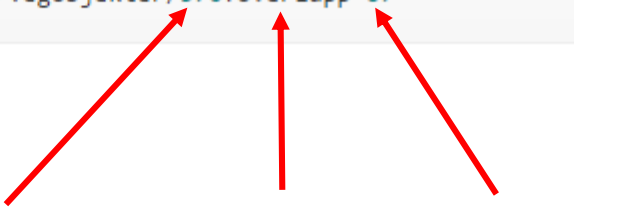
Easy-to-use pagination

Overlap filter

https://www.vegvesen.no/nvdb/apidokumentasjon/#/parameter/overlappfilter

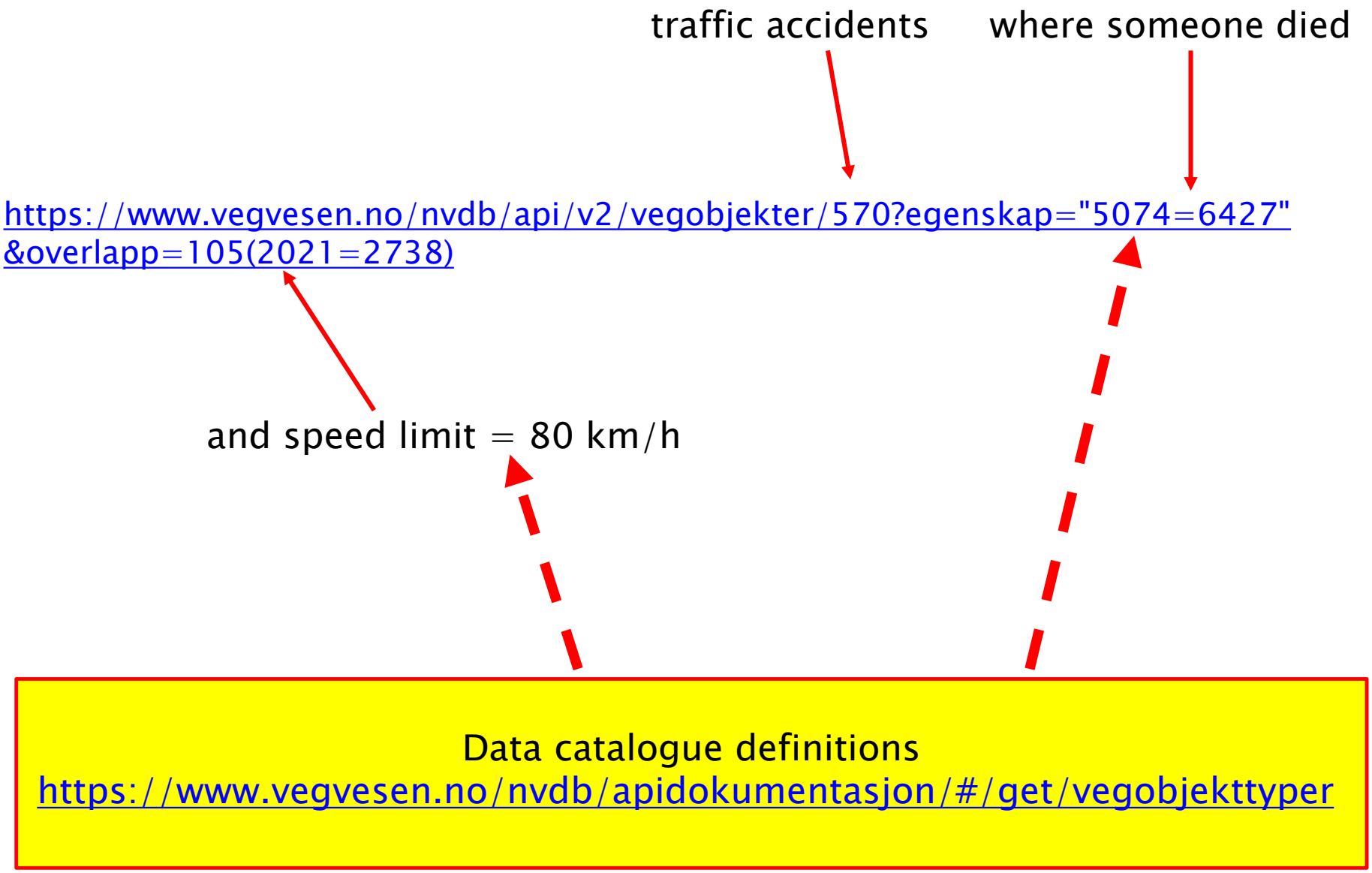
Eksempel: Trafikkulykker på samme sted som tunnelløp

```
https:/www.vegvesen.no/nvdb/api/v2/vegobjekter/570?overlapp=67
```

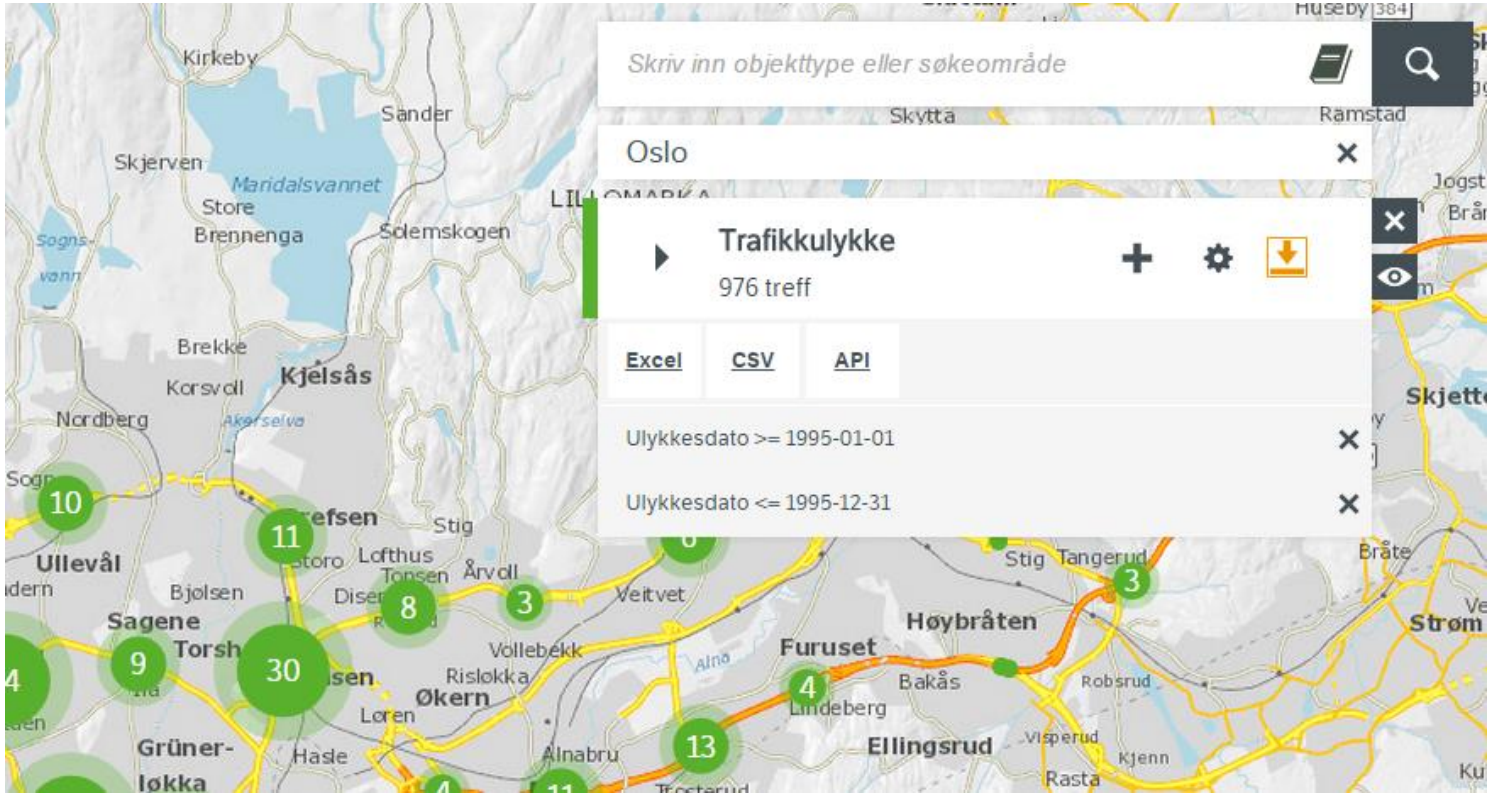


traffic accidents within tunnels

More filter!


traffic accidents
where someone died
https://www.vegvesen.no/nvdb/api/v2/vegobjekter/570?egenskap="5074=6427"
&overlapp=105(2021=2738)
and speed limit = 80 km/h
Data catalogue definitions
https://www.vegvesen.no/nvdb/apidokumentasjon/#/get/vegobjekttyper

Vegkart http://vegkart.no is your friend!

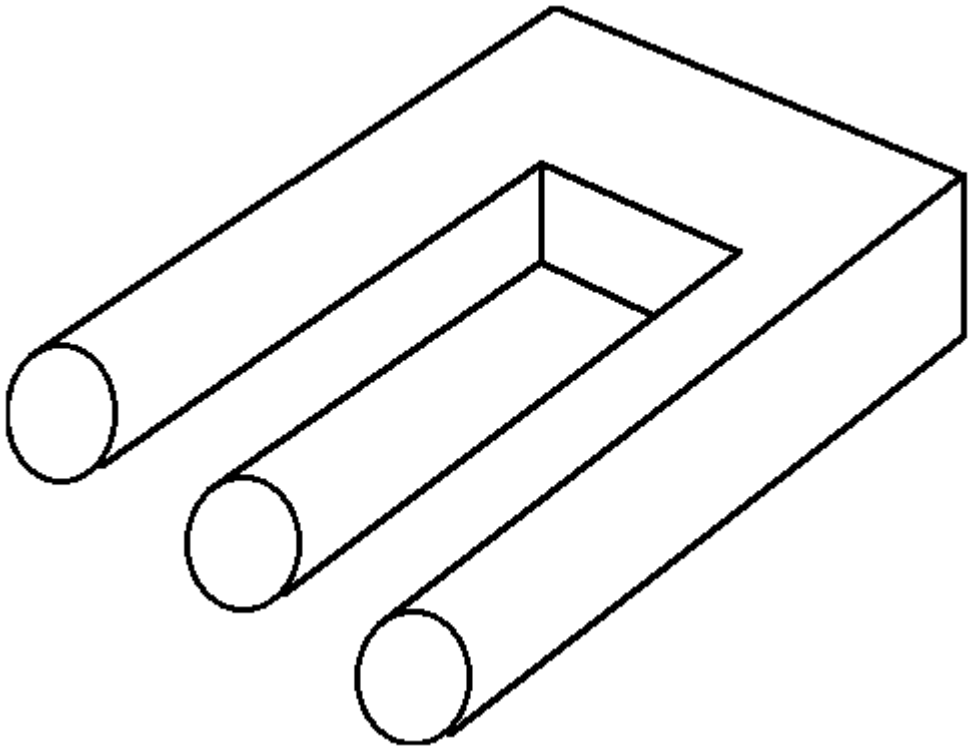


Caveat: Vegkart reads V1 of NVDB api (per aug-2016)

http://www.vegdata.no/2015/10/16/na-er-det-enda-enklere-a-lage-sok-mot-nvdb-api/

Why is NVDB so insanely difficult?

Pitfalls & headaches



Why is NVDB so insanely difficult?

Gis-unfriendly data model

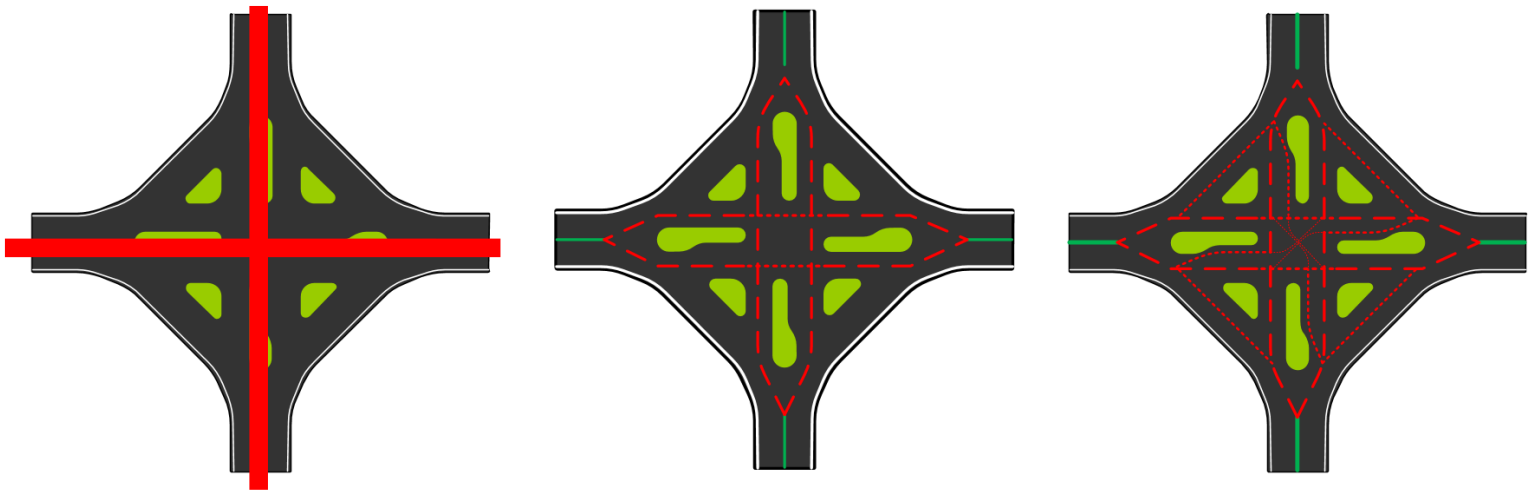

```
{
    id: 1078,
    navn: "Navn bomstasjon",
    datatype: 1,
    datatype_tekst: "Tekst",
    verdi: "RV409 KVERNEVIKVEIEN"
},
{
    id: 7992,
    navn: "Eier",
    datatype: 30,
    datatype_tekst: "FlerverdiAttributt, Tekst",
    verdi: "Stat",
    enum_id: 10258
},
```


Property	Value
Navn bomstasjon	RV409 KVERNEVIKVEIEN
Eier	Stat

Why is NVDB so insanely difficult?

Topology levels

Road network = 1, 2 or 3 topology levels **where needed**



Vegtrasenivå

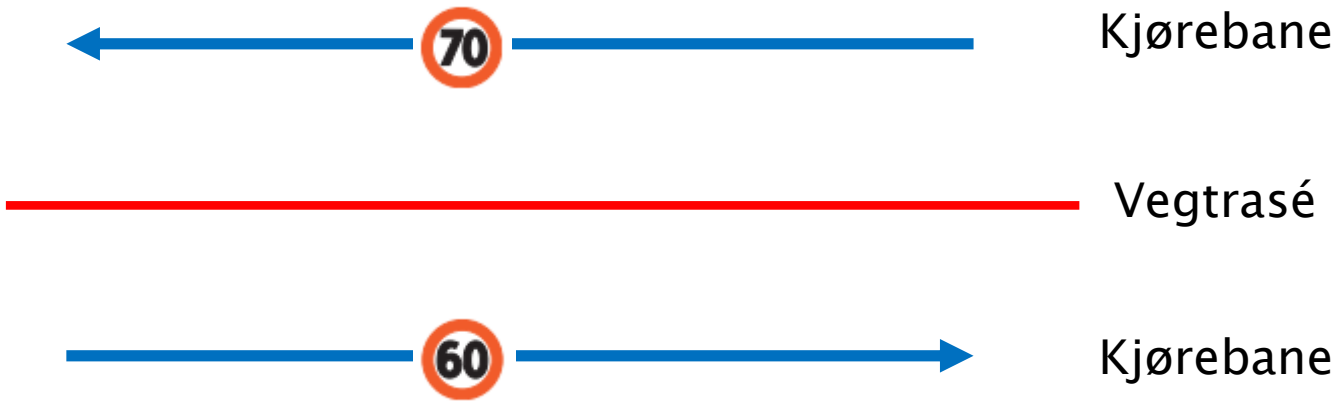
Kjørebanenivå

Kjørefeltnivå

Why is NVDB so insanely difficult?

Topology levels

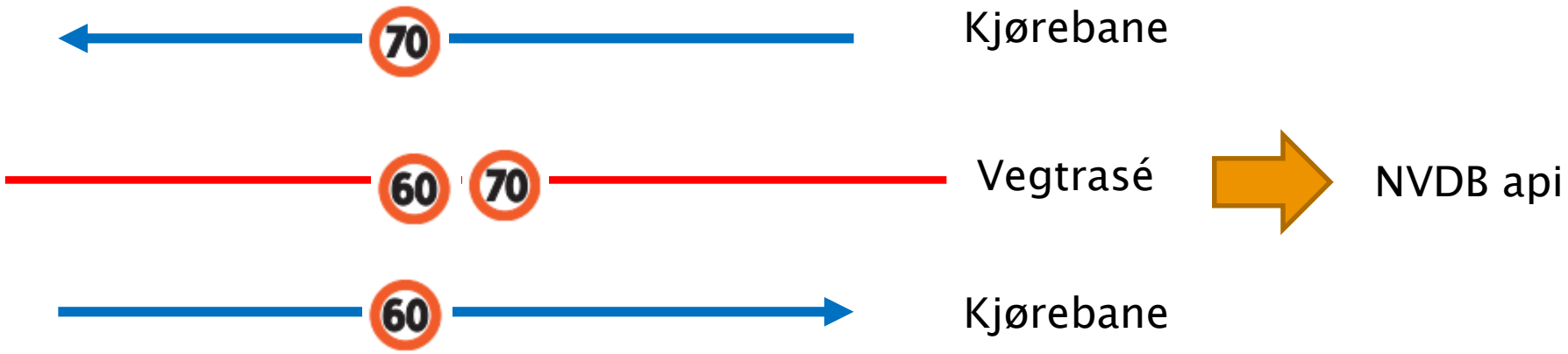
Data are almost always at uppermost level
Mapping to other levels as needed…



Why is NVDB so insanely difficult?

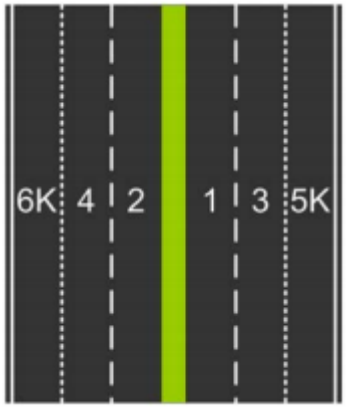
Topology levels

Data mapped between topology levels
=> all sort of confusion

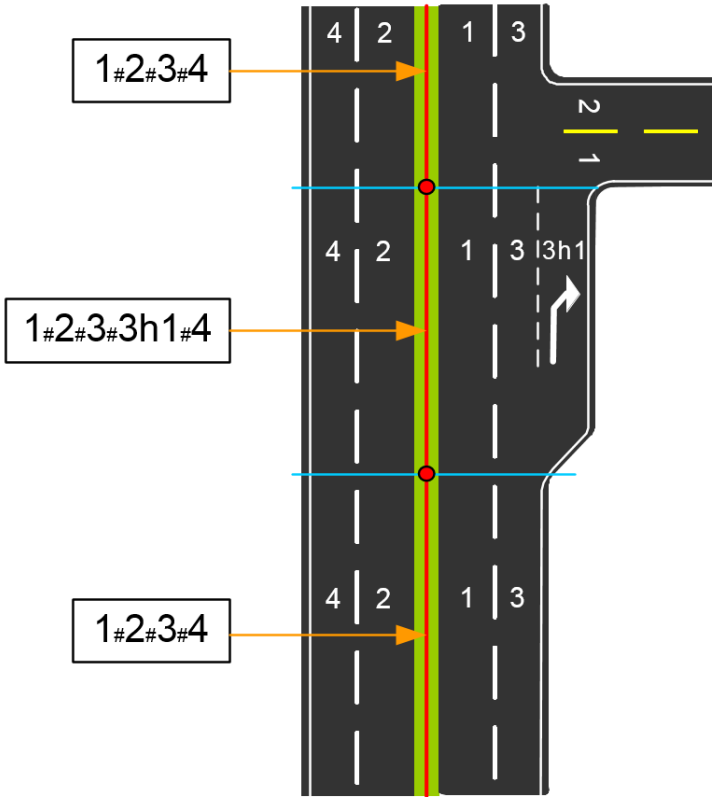


Why is NVDB so insanely difficult?

Lane definition, oh my



1#2#3#4#5K#6K

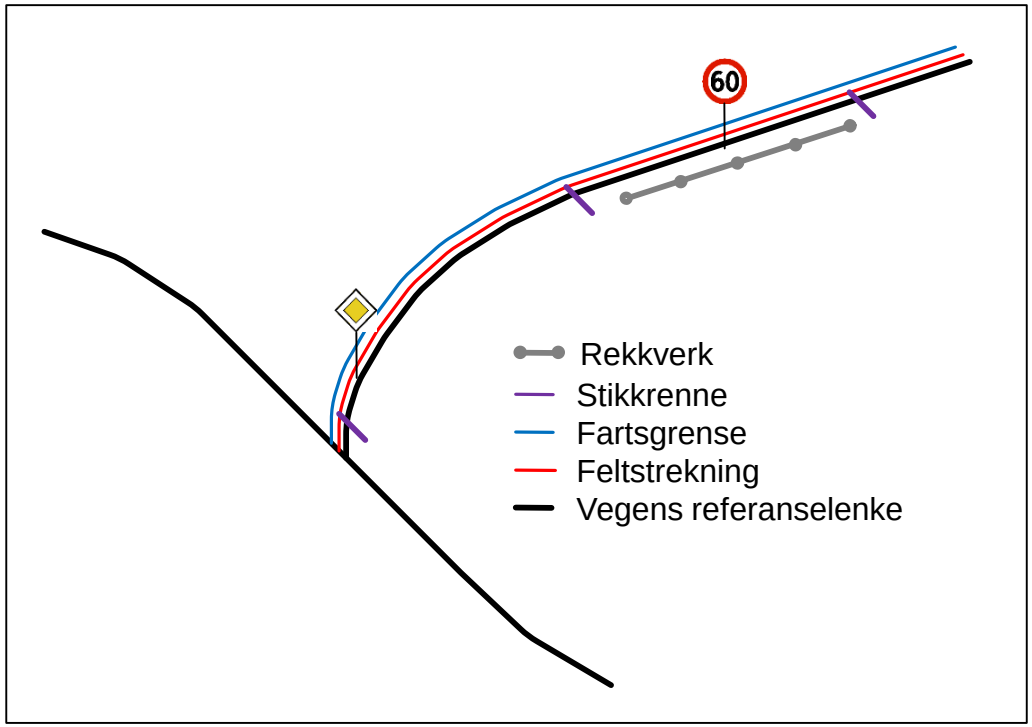


Object type 616 Feltstrekning
https://www.vegvesen.no/nvdb/api/datakatalog/objekttyper/616

Why is NVDB so insanely difficult?

Linear references? Drives me nuts…

- Lineare references
  - Non-dimensional [0..1]
  - Metering system
- Coordinates




```
- {
    - stedfesting: {
          veglenkeid: 384037,
          posisjon: 0.832626690665394,
          kortform: "0.832626690665394@384037",
          retning: "MED"
      },
    - geometri: {
          wkt: "POINT (82550.19999999997 6809184.100000001 140.1000000000516)",
          srid: 32633
      },
```

Why is NVDB so insanely difficult?

GIS-friendly = homogenous segments

Speed limit

Road class

70km/h
BK1

Repositories

Magnus Haugsand
haugsand

https://github.com/haugsand/

- Not maintained, will become stale (V1 NVDB api)
- Javascript, python

https://github.com/nvdb-vegdata

- Java: reading (NVDB api V2) and writing to NVDB
- Javascript & python: Mostly V1
- «Official» repos

nvdb-vegdata

https://github.com/ltglahn

- My own playground
- Python NVDB api V2 (first version)
- FME (proprietary tool, $$)

Jan Kristian Jensen
LtGlahn

Writing to NVDB? Try the docker mock-up

- Mock-up NVDB api server under your control
  - Same login
  - Same strict data validation (format, consistency ++)
  - *Same everything, exept you don't actually write to NVDB*

- Good documentation
- https://www.vegdata.no/2016/03/09/utviklerutgave-av-skrive-apiet-tilgjengelig-pa-docker-hub/

Come to our developer conference!

- 23. sept 2016, Trondheim
- https://www.vegdata.no/2016/08/24/nvdb-apne-vegdata-utviklerkonferanse-2016/

Statens vegvesen

Thank you!