

# BIM4Infra

22/11/2022



Vlaamse  
overheid



BELEIDSDOMEIN  
MOBILITEIT &  
OPENBARE  
WERKEN

DEPARTEMENT  
MOBILITEIT &  
OPENBARE  
WERKEN

AGENTSCHAP  
MARITIEME  
DIENSTVERLENING en  
KUST

AGENTSCHAP  
WEGEN &  
VERKEER

 De Vlaamse  
Waterweg nv

**lantis** bouwen  
aan  
verbinding

 **DE WERKENNOOTSCHAP**



# 5.1 OTLMOW

OTL-conforme data beheren met Python.

David Vlamincx (AWV)

# Databeheerder van AWV Infra db

Beheren van data: hoe?

Op welke manier dan ook

Liefst rechtstreeks via databank query's  
maar dat mag niet 😞

Meest voorkomende scenario:

Data inladen (DAVIE bestand of iets anders) ⇒ bewerken ⇒ OTL-conform bestand aanmaken



# Data beheren

## Meest voorkomende scenario

Data inladen ⇒ **bewerken** ⇒ OTL-conform bestand aanmaken



## Laagdrempelige taal

Populair voor scripts, in applicaties of plugins

Ondersteunt overerving van meerdere classes (multiple inheritance probleem)





# Project OTLMOW

## Meerdere Python libraries

Groter project uitgesplitst in meerdere kleine libraries

Dit is geen afgewerkte applicatie

Wel een **basis**, voor een applicatie, op maat van de use cases van de gebruikers

Erg geschikt voor kleine scripts en meer...

**Open source**



# Python code

## Praktisch

Uitvoerbare code! (Icoon + naam)



Gedeeld via AWW GitHub:

<https://github.com/WegenenVerkeer/OTLMOW/tree/main/BIM4Infra22>

Voorzien van een readme bestand voor Python “dummies”

Uitvoerbare code start steeds met:

```
if __name__ == "__main__":
```

Code erna wordt alleen uitgevoerd als dat specifieke bestand wordt uitgevoerd

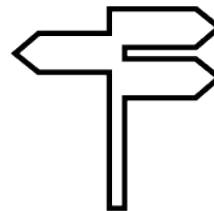




Vlaamse  
overheid

# otlmow\_model

Klassenmodel



# otlmow\_model

## Klassenmodel

Basis library, met een Python klasse voor elke klasse in de OTL

Zo “juist” mogelijke implementatie van de OTL

- inclusief overerving
- attributen met types die overeenkomen met de datatypes
  - validatie/conversie van ingevulde waardes
- validatie bij leggen van relaties mogelijk
- CLI voor informatie over objecten en attributen



# Wegkantkast

## Klassenmodel in otlmow\_model

Overerving zoals in de OTL

Definitie toegevoegd als docstring

TypeURI is een klasse attribuut

De attributen zijn properties

```
# coding=utf-8
from typing import List
from otlmow_model.BaseClasses.OTLAttribuut import OTLAttribuut
from otlmow_model.Classes.Abstracten.Buitenkast import Buitenkast
from otlmow_model.BaseClasses.BooleanField import BooleanField
from otlmow_model.Datatypes.DtcDocument import DtcDocument, DtcDocumentWaarden
from otlmow_model.Datatypes.KlWegkantkastType import KlWegkantkastType

# Generated with OTLClassCreator. To modify: extend, do not edit
class Wegkantkast(Buitenkast):
    """Behuizing in de vorm van een kast typisch gebruikt buiten, langs de kant van de weg."""

    typeURI = 'https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Wegkantkast'
    """De URI van het object volgens https://www.w3.org/2001/XMLSchema#anyURI."""

    def __init__(self):...

    @property
    def elektrischSchema(self) -> DtcDocumentWaarden:
        """Elektrisch aansluitschema van de kast."""
        return self._elektrischSchema.get_waarde()

    @elektrischSchema.setter
    def elektrischSchema(self, value):...

    @property
    def heeftMaaibescherming(self) -> bool:
        """Geeft aan of de kast voorzien is van bescherming tegen schade
        bij het maaien van de omgeving rond de kast."""
        return self._heeftMaaibescherming.get_waarde()

    @heeftMaaibescherming.setter
    def heeftMaaibescherming(self, value):...
```



# Wegkantkast

## Klassenmodel in otlmow\_model

Relaties zijn toegevoegd

```
self.add_valid_relation(relation='https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging',
                        target='https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Verkeersregelaar')
self.add_valid_relation(relation='https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging',
                        target='https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#WIMDataLogger')
```

De properties zijn achterliggend een OTLAttribuut  
waarin het datatype gespecificeerd wordt

```
self._elektrischSchema = OTLAttribuut(field=DtcDocument,
                                       naam='elektrischSchema',
                                       label='elektrisch schema',
                                       objectUri='https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Wegkantkast.elektrischSchema',
                                       definition='Elektrisch aansluitschema van de kast.',
                                       owner=self)

self._heeftMaaibeschermining = OTLAttribuut(field=BooleanField,
                                              naam='heeftMaaibeschermining',
                                              label='heeft maaibeschermining',
                                              objectUri='https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Wegkantkast.heeftMaaibeschermining',
                                              definition='Geeft aan of de kast voorzien is van bescherming tegen schade bij het maaien van de omgeving rond de kast.',
                                              owner=self)
```





# Gebruik klassenmodel in de praktijk

## Aanmaken van een wegkantkast



# Gebruik klassenmodel in de praktijk

## Aanmaken van een wegkantkast

IDE geeft aan welke attributen je nog kan invullen

```
if __name__ == "__main__":  
    kast = Wegkantkast()  
    kast.isActief = True  
    kast.toestand = 'in-gebruik'  
    kast.naam = 'A0013.K'  
    kast.heeftMaaibescherming = True  
    kast.
```



adres	Behuizing
afmeting	Kast
datumOprichtingObject	AIMObject
elektrischSchema	Wegkantkast
heeftVerlichting	Kast
indelingsplan	Kast
ipKlasse	Buitenkast







# Validatie in klassenmodel

## Meteen validatie

Wat gebeurt er bij het foutief invullen?

Niet interpreteerbare waarde geeft foutmelding

```
if __name__ == "__main__":  
    kast = Wegkantkast()  
    kast.isActief = True  
    kast.toestand = 'in-gebruik'  
    kast.naam = 'A0013.K'  
    kast.heeftMaaibescherming = 20
```



Traceback (most recent call last):

File `"/home/davidlinux/PycharmProjects/OTLMOW-Model/SampleCode/printing_instances.py"`, line 9, in `<module>`

`kast.heeftMaaibescherming = 20`

File `"/home/davidlinux/PycharmProjects/OTLMOW-Model/otlmow_model/Classes/Onderdeel/Wegkantkast.py"`, line 78, in `heeftMaaibescherming`

`self._heeftMaaibescherming.set_waarde(value, owner=self)`

File `"/home/davidlinux/PycharmProjects/OTLMOW-Model/otlmow_model/BaseClasses/OTLAttribuut.py"`, line 167, in `set_waarde`

`converted_value = self.field.convert_to_correct_type(value)`

File `"/home/davidlinux/PycharmProjects/OTLMOW-Model/otlmow_model/BaseClasses/BooleanField.py"`, line 33, in `convert_to_correct_type`

`raise CouldNotConvertToCorrectTypeError(f'{value} could not be converted to correct type (implied by {cls.__name__})')`

`otlmow_model.Exceptions.CouldNotConvertToCorrectTypeError.CouldNotConvertToCorrectTypeError: 20 could not be converted to correct type (implied by BooleanField)`





# Validatie in klassenmodel

## Meteen conversie

Wat gebeurt er bij het foutief invullen?

Indien wel interpreteerbaar: enkel waarschuwing

```
if __name__ == "__main__":  
    kast = Wegkantkast()  
    kast.isActiveief = True  
    kast.toestand = 'in-gebruik'  
    kast.naam = 'A0013.K'  
    kast. heeftMaaibescher ming = 'True'
```



1.1

WARNING:root:Assigned a string to a boolean datatype. Automatically converted to the correct type  
Please change the type



## Toon een object

<Wegkantkast> object

```
heeftMaaibescherming : True
```

naam : A0013.K

toestand : in-gebruik

 1.1

# Gebruik klassenmodel

## Toon de meta info van een object

Met `print(meta_info(...))` toon je attribuut info van de asset

```
if __name__ == "__main__":
    kast = Wegkantkast()
    kast.isActief = True
    kast.toestand = 'in-gebruik'
    kast.naam = 'A0013.K'
    kast.heeftMaaibescherming = True

    print(meta_info(kast))
```



Showing metadata of Wegkantkast:

typeURI: <https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Wegkantkast>

definition: Behuizing in de vorm van een kast typisch gebruikt buiten, langs de kant van de weg.

attributes:

- adres (type: DtcAdres)
- afmeting (type: DtcAfmetingBxLxhInMm)
- assetId (type: DtcIdentifier)
- bestekPostNummer (type: String)
- datumOprichtingObject (type: Date)
- elektrischSchema (type: DtcDocument)
- geometry (type: WKT)
- heeftMaaibescherming (type: Boolean)
- heeftVerlichting (type: Boolean)
- indelingsplan (type: DtcDocument)
- ipKlasse (type: KLAIngIngressProtectionCode)
- isActief (type: Boolean)
- kastmateriaal (type: KLAIngMateriaal)
- keuringsfrequentie (type: KwantWrdInJaar)
- mplan (type: DtcDocument)
- naam (type: String)
- notitie (type: String)
- risicoanalyse (type: DtcDocument)
- standaardBestekPostNummer (type: String)
- tabelUitwendigeInvloeden (type: DtcDocument)
- theoretischeLevensduur (type: KwantWrdInMaand)
- toestand (type: KLAIntToestand)
- type (type: KLVegkantkastType)
- verfraaid (type: KLBuitenkastVerfraaid)



# Gebruik klassenmodel

## Toon de meta info van een object

Met `print(meta_info(..., attribute='...'))` toon je info van attributen

```
if __name__ == "__main__":  
    kast = Wegkantkast()  
    print(meta_info(kast, attribute='assetId'))
```



Showing metadata of assetId:

typeURI: <https://wegenverkeer.data.vlaanderen.be/ns/implementatieelement#AIMObject.assetId>

definition: Unieke identificatie van de asset zoals toegekend door de assetbeheerder of n.a.v.  
eerste aanlevering door de leverancier.

attributes:

identificator (type: String)

toegekendDoor (type: String)





# Validatie in klassenmodel

## Deprecation warning

Wat gebeurt er bij het invullen van een deprecated attribuut?

```
if __name__ == "__main__":  
    camera = Camera()  
    camera.heeftAid = True
```



```
if __name__ == "__main__":  
    camera = Camera()  
    print(meta_info(camera))
```

**WARNING:root:**<https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Camera.heeftAid>  
is deprecated since version 2.3.0

Showing metadata of Camera:

typeURI: <https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Camera>

definition: Een CCTV-camera, closed-circuit television camera, kortweg camera, produceert beelden of opnames voor bewaking van een regio vanop afstand. Het is een element dat beelden neemt van een locatie en deze doorgeeft naar verschillende partijen om zo de werkelijke situatie te kunnen inschatten vanop afstand. Deze camera kan van het analoge type zijn of digitaal.

attributes:

- assetId (type: DtcIdentifier)
- beeldverwerkingsinstelling (type: DtcCameraBeeldverwerking)
- bestekPostNummer (type: String)
- configBestandAid (type: DtcDocument) <deprecated since 2.3.0>
- datumOprichtingObject (type: Date)
- dnsNaam (type: String)
- geometry (type: WKT)
- heeftAid (type: Boolean) <deprecated since 2.3.0>
- heeftSpitsstrook (type: Boolean) <deprecated since 2.3.0>



# Validatie in klassenmodel

## Keuzelijsten

Wat gebeurt er bij het foutief invullen van een keuzelijst?

```
if __name__ == "__main__":  
    camera = Camera()  
    camera.toestand = 'in gebruik'
```

ValueError: in gebruik is not a valid option for toestand, find the valid options using `print(meta_info(<object>, attribute="toestand"))`



```
if __name__ == "__main__":  
    camera = Camera()  
    print(meta_info(camera,  
                    attribute='toestand'))
```

Showing metadata of toestand:

typeURI: <https://wegenverkeer.data.vlaanderen.be/ns/implementatieelement#AIMToestand.toestand>

definition: Geeft de actuele stand in de levenscyclus van het object.

valid values:

geannuleerd

gepland

in-gebruik

in-ontwerp

in-opbouw

overgedragen

uit-gebruik

verwijderd





# Validatie in klassenmodel

## Relaties

Gebruik het klassenmodel om te bevragen of een relatie mogelijk is

```
if __name__ == "__main__":  
    camera = Camera()  
    kast = Wegkantkast()  
    steun = RechteSteun()  
  
    camera_kast_bevestiging = RelationValidator.is_valid_relation(source=camera, relation=Bevestiging, target=kast)  
    print(f'Bevestiging tussen Camera en Wegkantkast mogelijk? {camera_kast_bevestiging}')  
    camera_steun_bevestiging = RelationValidator.is_valid_relation(source=camera, relation=Bevestiging, target=steun)  
    print(f'Bevestiging tussen Camera en RechteSteun mogelijk? {camera_steun_bevestiging}')  
    steun_camera_bevestiging = RelationValidator.is_valid_relation(source=steun, relation=Bevestiging, target=camera)  
    print(f'Bevestiging tussen RechteSteun en Camera mogelijk? {steun_camera_bevestiging}')
```



1.6

Bevestiging tussen Camera en Wegkantkast mogelijk? False

Bevestiging tussen Camera en RechteSteun mogelijk? True

Bevestiging tussen RechteSteun en Camera mogelijk? True







Vlaamse  
overheid

# otlmow\_modelbuilder

Bouwt het klassenmodel



# otlmow\_modelbuilder

## Bouwt het klassenmodel

Ondersteunende library die het klassenmodel aanmaakt

Op basis van SQLite (OTL of subset) het OTL model opbouwen

- eventueel inclusief GeometrieArtefact

Nuttig wanneer je

- enkel een beperkt aantal klassen wilt gebruiken (= subset)
- een OTL versie wilt gebruiken verschillend van de meest recent gepubliceerde

De huidige versie wordt steeds gebruikt in de meest recent versie van otlmow\_model





# otlmow\_modelbuilder

## Gebruik

Enkel locaties van bestanden en map invullen

```
if __name__ == '__main__':  
    logging.basicConfig(level=logging.INFO, format='%(message)s')  
  
    current_dir = Path(__file__).parent  
    otl_subset_path = Path(current_dir / 'InputFiles' / 'OTL 2.5.db')  
    GA_file_path = Path(current_dir / 'InputFiles' / 'Geometrie_Artefact_2.5.db')  
    model_directory = Path(current_dir.parent.parent / 'OTLMOW-Model/otlmow_model')  
  
    ModelBuilder.build_otl_datamodel(otl_subset_location=otl_subset_path,  
                                     geometry_artefact_location=GA_file_path,  
                                     directory=model_directory)
```



2.1





# otlmow\_converter

Omvormen van assets uit en naar bestanden



# otlmow\_converter

## Omvormen van assets uit en naar bestanden

“Core” library, die het klassenmodel gebruikt

Assets aanmaken door

- helpers te gebruiken
- OTL-conforme bestanden\* in te lezen

Exporteren naar een OTL-conform bestand\*

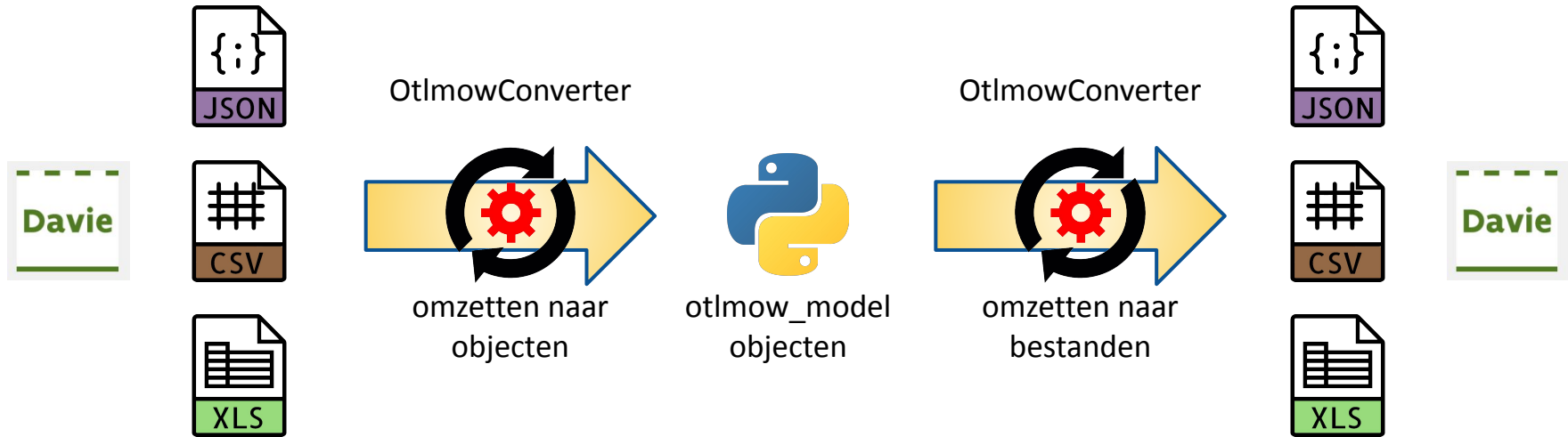
- zoveel mogelijk validatie doen voor de import via DAVIE

\* bestand met OTL-conforme data, dat door DAVIE aangemaakt/verwerkt wordt



# Omzetten van en naar bestanden

## OtlmowConverter



Zeer relevante use case: best om steeds te vertrekken van de huidige data uit DAVIE:  
inlezen  $\Rightarrow$  aanpassen/aanvullen  $\Rightarrow$  exporteren



# OtlmowConverter gebruik

## Camera dataset

CSV bestand met alle OTL conforme Camera assets

typeURI	assetId.identificator	datumOprichtingObject	geometry	isActief	isPtz	merk	naam	opstelhoogte	toestand
#Camera	14cc3c81-0e13-49c4-805f-7a89ecf70038-	2010-01-01	POINT Z (122548.78 179658.28 0)	True	True			19	in-gebruik
#Camera	15956fad-95f0-4635-8d52-53270fc54a0e-b	2010-01-01	POINT Z (169097.2 171897 0)	True	True		CAMV6127		in-gebruik
#Camera	15b552aa-5f7e-47ae-a982-12c608695eb9-		POINT Z (161227.3 173223.1 0)	True	True		CAMV6163		in-gebruik
#Camera	164d5e7f-da2c-4c80-b739-a9ed1beae032-	1970-01-01	POINT Z (151544.4 209812.7 0)	True	True	bosch	CAMA0354		in-gebruik
#Camera	16625e34-6558-43bc-8928-46839f0d3db1-	2014-12-02	POINT Z (154342.8 177373.1 0)	True	True		CAMV6185		in-gebruik
#Camera	166cd082-90d0-4f1e-8be2-f66c68b90996-	2020-09-29	POINT Z (41664.3 203954.1 0)	True	True		CAMW8227		in-gebruik
#Camera	1695be04-dad8-4e58-bfbc-3aee7b7d94c9-	2018-02-07	POINT Z (104658.7 188524.1 0)	True	True	bosch	CAMO4767		in-gebruik
#Camera	265eddfc-5d46-425b-b6da-fc3c648b8047-	1970-01-01	POINT Z (139439.2 174817.4 0)	True	False		CAMV6075		in-gebruik
#Camera	16aec9c5-bec7-4fff-8e4d-43b9598037c5-b	2016-10-10	POINT Z (66734.3 162810.6 0)	True	True		CAMW8423		in-gebruik
#Camera	171ada59-ed38-42c0-a395-883102aecb5f-	1970-01-01	POINT Z (152113.9 213014 0)	True	False		CAMA1007		in-gebruik
#Camera	172b9b71-a2d5-434e-89ba-a296d8557227-	2014-01-13	POINT Z (70816.7 204533.7 0)	True	True		CAMW8073		in-gebruik
#Camera	17309759-fd32-46fd-a191-fa8b8f26c392-b	2010-01-01	POINT Z (131879.7 213457 0)	False	False		CAMO5007.0UD		verwijderd

Open data





# OtlmowConverter gebruik

## Create\_assets\_from\_file()

```
if __name__ == '__main__':  
    logging.basicConfig(level=logging.INFO, format='%(message)s')  
  
    converter = OtlmowConverter()  
  
    cameras = converter.create_assets_from_file(filepath=Path('./cameras_open_data.csv'))  
    print_overview_assets(cameras)
```



3.1

counting 1547 assets of type <https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Camera>





# OtlmowConverter gebruik

## Create\_file\_from\_assets()

```
if __name__ == '__main__':  
    logging.basicConfig(level=logging.INFO, format='%(message)s')  
    converter = OtlmowConverter()  
    cameras = converter.create_assets_from_file(filepath=Path('./cameras_open_data.csv'))  
  
    ptz = list(filter(lambda c: c.isActief and c.isPtz, cameras))  
  
    converter.create_file_from_assets(filepath=Path('./ptz_open_data.json'),  
                                     list_of_objects=ptz)
```



```
{  
    "assetId": {  
        "identifier": "008d697f-236b-469c-8893-7202545d1230-b25kZXJkZWVsI0NhbWVvYQ",  
        "toegekendDoor": "AWV"  
    },  
    "datumOprichtingObject": "2002-01-01",  
    "geometry": "POINT Z (112142.8 193631.4 0)",  
    "isActief": true,  
    "isPtz": true,  
    "merk": "bosch",  
    "naam": "CAM04625",  
    "toestand": "in-gebruik",  
    "typeURI": "https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Camera"  
},
```

# Helpers

## Helpers/functies

**AssetFactory:** instantieert assets

Gebruikt ofwel

- URI
- namespace + naam

```
if __name__ == '__main__':  
    asset = AssetFactory.dynamic_create_instance_from_ns_and_name(namespace='onderdeel',  
                                                                    class_name='Wegkantkast')  
    print(asset)
```



<Wegkantkast> object

typeURI : <https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Wegkantkast>



# Helpers

## Helpers/functies

### RelationCreator: instantieert en valideert relaties

- gebruikt assets en relatietype

```
if __name__ == '__main__':  
    camera = Camera()  
    camera.assetId.identificator = 'camera0001'  
    paal = RechteSteun()  
    paal.assetId.identificator = 'paal12345'  
    bevestiging = create_relation(source=camera, target=paal,  
                                relation=Bevestiging)  
    print(bevestiging)
```

<Bevestiging> object

typeURI : <https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging>

assetId :

identificator : camera0001\_-paal12345

toegekendDoor : OTLMOW

bron :

typeURI : <https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Camera>

bronAssetId :

identificator : camera0001

doel :

typeURI : <https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#RechteSteun>

doelAssetId :

identificator : paal12345





Vlaamse  
overheid

# otlmow\_template

Templates maken o.b.v. subset

# otlmow\_template

## Templates

Maakt templates van gewenst formaat op basis van een subset (zie starterspakket)

Bijna alle bouwblokken waren al gemaakt

- leest een subset in (modelbuilder)
- maakt assets aan voor elk aanwezig assettype (converter)
- vult elk attribuut op met dummy data
- exporteert de assets naar een gewenst formaat





# otlmow\_template

## Gebruik

```
if __name__ == '__main__':  
    logging.basicConfig(level=logging.INFO, format='%(message)s')  
  
    template_creator = SubsetTemplateCreator()  
    subset_location = Path('./camera_steun.db')  
    csv_location = Path('./template_file.csv')  
  
    template_creator.generate_template_from_subset(path_to_subset=subset_location,  
                                                  path_to_template_file_and_extension=csv_location,  
                                                  split_per_type=True)
```



# otlmow\_template

## Resultaat

### Camera

J	K	L	M	N	O	P	Q	R	S
datumOprichtingObject	dnsNaam	ipAdres	isActief	isPtz	merk	modelnaam	naam	notitie	opstelhoogte
2004-01-13	hkNPSHxykLcUwf	wEGjhEKQX	True	False	bosch	ulisse-hd	WStGpetQzwwM	qhtsmosEfs	56.64

### Bevestiging

A	B	C	D	E	F	G	H
typeURI	assetId.identificator	assetId.toegekendDoor	bronAssetId.identificator	bronAssetId.toegekendDoor	doelAssetId.identificator	doelAssetId.toegekendDoor	isActief
<a href="https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging">https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging</a>	ZscNEaYcsOkOk	SujkUTJ	ytOEmS	AUAQFpPzIvyOPnu	rhgnSgy	fjXvRyioVZs	False

### RechteSteun

B	C	D	E	F	G	H	I	J
assetId.identificator	assetId.toegekendDoor	beschermlaag	bestekPostNummer[]	bijzonderTransport	datumOprichtingObject	elektrischeBeveiliging	fabrikant	hoogteBovenkant
wZRikXGbVvMqe	tFskDdyA	gecoat	ZnYgSHL	kantelbaar	2016-09-27	differentieelautomaat	TXVCnr	46.24



# Voorbeelden



# Normalisatie Exoten naar InvasieveExoten

## Deprecatie van Exoten

Exoten is een deprecated class

⇒ “vervangen” door InvasieveExoten

Identiek qua attributen

## Normalisatie

- inladen van Exoten
- aanmaken van InvasieveExoten met identieke waardes voor attributen
- exoten inactief maken
- export bestand maken en inladen in DAVIE





# Normalisatie Exoten naar InvasieveExoten

```
if __name__ == '__main__':  
    converter = OtlmowConverter()  
    list_exoten = converter.create_assets_from_file(Path('DA-2022-00004_export.json'))    Inladen van Exoten  
    list_objects = []  
  
    list_of_attributes_to_copy = AssetFactory.get_attribute_list_from_object(list_exoten[0])  
  
    for exoten in list_exoten:    Aanmaken van InvasieveExoten met identieke waarden voor attributen  
        new_invasieve_exoten = AssetFactory.create_aimObject_using_other_aimObject_as_template(  
            orig_aim_object=exoten, typeURI=InvasieveExoten.typeURI, fields_to_copy=list_of_attributes_to_copy)  
  
        new_invasieve_exoten.assetId.identificator = f'normalized_{exoten.assetId.identificator}'  
        list_objects.append(new_invasieve_exoten)    InvasieveExoten toevoegen aan exportlijst  
  
        exoten.isActief = False    Exoten inactief maken en toevoegen aan exportlijst  
        list_objects.append(exoten)  
  
    converter.create_file_from_assets(list_of_objects=list_objects,    Export bestand maken voor DAVIE  
                                    filepath=Path('DA-2022-00004_exoten_normalisation_prd_import.json'))
```



# Normalisatie Exoten naar Invasieve Exoten

```
"assetId": {
  "identificator": "078740ec-2f5b-40e2-a7cb-508ba1bb5e9f-b25kZXJkZWVsI0V4b3RlbG",
  "toegekendDoor": "AWV"
},
"typeURI": "https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Exoten",
"datumOprichtingObject": "2020-07-03",
"drassigheid": "niet-drassig",
"geometry": "POINT Z (191097.553 214375.984 20.494)",
"heeftObstakels": true,
"isActief": false,
"soort": [
  {
    "soortnaamNederlands": "Japanse duizendknoop",
    "soortnaamWetenschappelijk": "Fallopia japonica"
  }
],
"taludwaarde": "tot-1-3",
"toestand": "in-gebruik"
```

```
"assetId": {
  "identificator": "normalized_078740ec-2f5b-40e2-a7cb-508ba1bb5e9f-b25kZXJkZWVsI0V4b3RlbG",
},
"typeURI": "https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#InvasieveExoten",
"datumOprichtingObject": "2020-07-03",
"drassigheid": "niet-drassig",
"geometry": "POINT Z (191097.553 214375.984 20.494)",
"heeftObstakels": true,
"isActief": true,
"soort": [
  {
    "soortnaamNederlands": "Japanse duizendknoop",
    "soortnaamWetenschappelijk": "Fallopia japonica"
  }
],
"taludwaarde": "tot-1-3",
"toestand": "in-gebruik"
```



# Lichtmasten langs een lijn

## Elke X aantal meter een asset langs een lijn

Use case:

Assets creëren, langs een opgegeven lijngeometrie, elke X aantal meter

Nuttig voor ontwerpen of geplande toestand

Voorbeeld op GitHub



# Lichtmasten langs een lijn

## 2 mogelijke helper functies

shapely (en numpy)

```
def divide_line_in_points_by_distance(linestring: LineString, distance: float) -> MultiPoint:
    distances = numpy.arange(0, linestring.length, distance)
    return MultiPoint([linestring.interpolate(distance) for distance in distances])

def divide_line_in_equally_distanced_points(linestring: LineString, number_of_points: int) -> MultiPoint:
    distances = numpy.arange(0, linestring.length, linestring.length / (number_of_points - 1))
    return MultiPoint([linestring.interpolate(distance) for distance in distances] + [linestring.boundary[1]])
```





# Lichtmasten langs een lijn

## Script

```
if __name__ == '__main__':  
    line = wkt.loads("LINESTRING (141478.84234000742435455 206252.95497500523924828 0, ...)")  
    distanced_points = divide_line_in_points_by_distance(line, 50.0)  
  
    list_of_assets = []  
    for index, point in enumerate(distanced_points.geoms):  
        mast = AssetFactory.dynamic_create_instance_from_ns_and_name('onderdeel', 'WVLichtmast')  
        mast.geometry = Point(point.x, point.y, 0).wkt  
        mast.assetId.identificator = f'mast_{index}'  
        mast.naam = f'mast_{index}'  
        mast.toestand = 'in-ontwerp'  
        list_of_assets.append(mast)  
  
    converter = OtlmowConverter()  
    converter.create_file_from_assets(filepath=Path('./masten_langs_lijn.csv'),  
                                     list_of_objects=list_of_assets)
```



5.1



# Lichtmasten langs een lijn

## Resultaat

Export naar csv bestand

typeURI	assetId.identificator	as	geometry	naam	toestand
#WVLichtmast	mast_0		POINT Z (141478.84234000742 206252.95497500524 0)	mast_0	in-ontwerp
#WVLichtmast	mast_1		POINT Z (141517.5332718902 206284.62533638685 0)	mast_1	in-ontwerp
#WVLichtmast	mast_2		POINT Z (141556.6621640567 206315.7527807326 0)	mast_2	in-ontwerp
#WVLichtmast	mast_3		POINT Z (141595.41411304675 206347.34831286265 0)	mast_3	in-ontwerp
#WVLichtmast	mast_4		POINT Z (141634.2929189576 206378.78705719596 0)	mast_4	in-ontwerp
#WVLichtmast	mast_5		POINT Z (141673.2993157092 206410.06805801843 0)	mast_5	in-ontwerp
#WVLichtmast	mast_6		POINT Z (141712.06759648785 206441.6414279432 0)	mast_6	in-ontwerp
#WVLichtmast	mast_7		POINT Z (141750.89615846777 206473.1422179894 0)	mast_7	in-ontwerp
#WVLichtmast	mast_8		POINT Z (141790.00775119307 206504.28917085307 0)	mast_8	in-ontwerp
#WVLichtmast	mast_9		POINT Z (141828.73655254766 206535.91321148103 0)	mast_9	in-ontwerp
#WVLichtmast	mast_10		POINT Z (141867.53744046896 206567.44840958083 0)	mast_10	in-ontwerp





# Lichtmasten langs een lijn

## Op kaart

CSV bestand ingelezen in QGIS  
met Google satellietbeeld als  
achtergrondkaart





# OTLMOW in QGIS

# QGIS plugin

## OTLMOW als basis voor QGIS plugin

QGIS gebruikt o.a. Python voor plugins

Basic plugin die

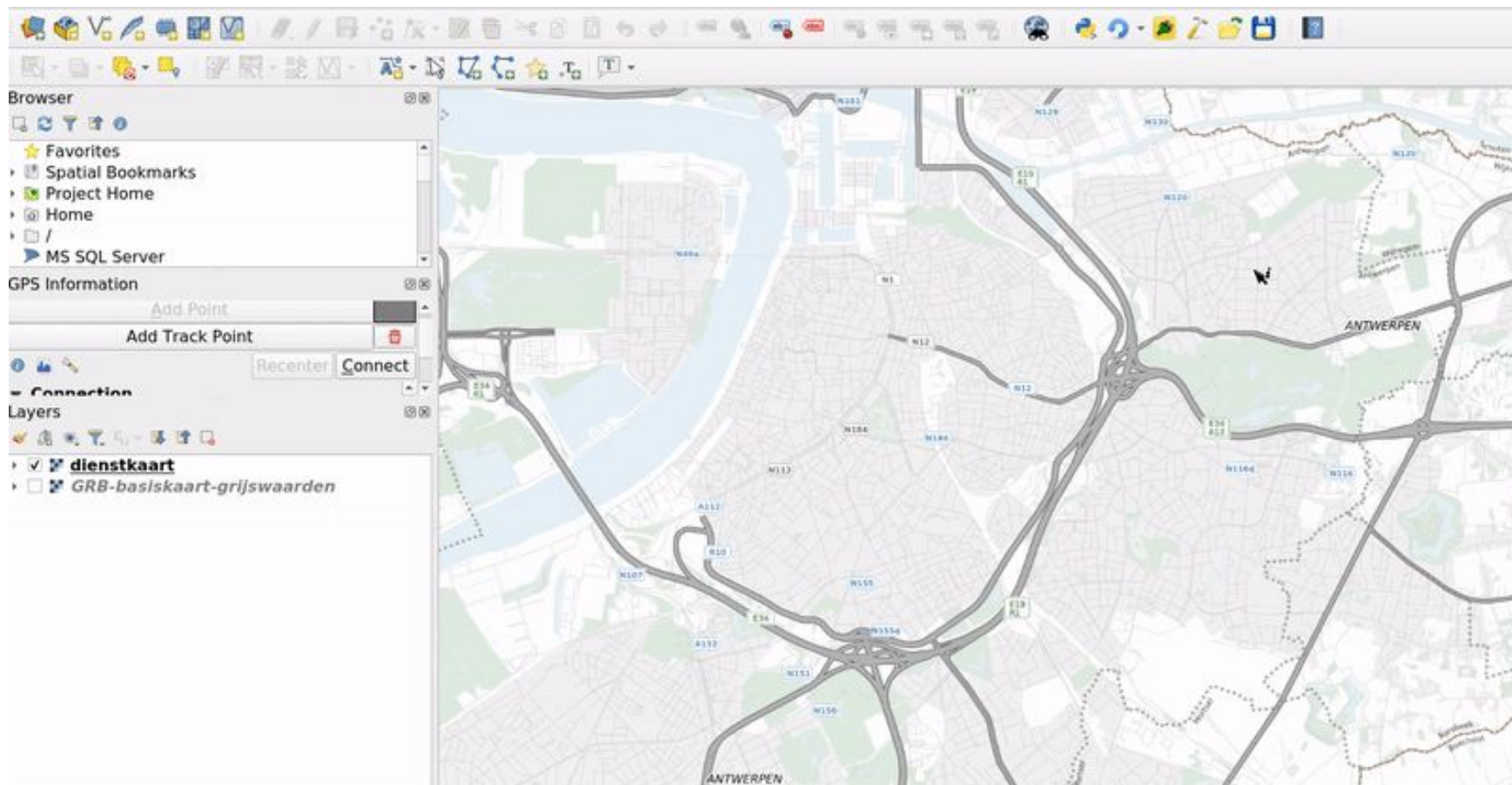
- bestanden kan inladen (csv / json)
- assets als features aanmaakt in een vectorlaag
- features zijn aanpasbaar
- features kan opslaan naar een bestand (csv / json)

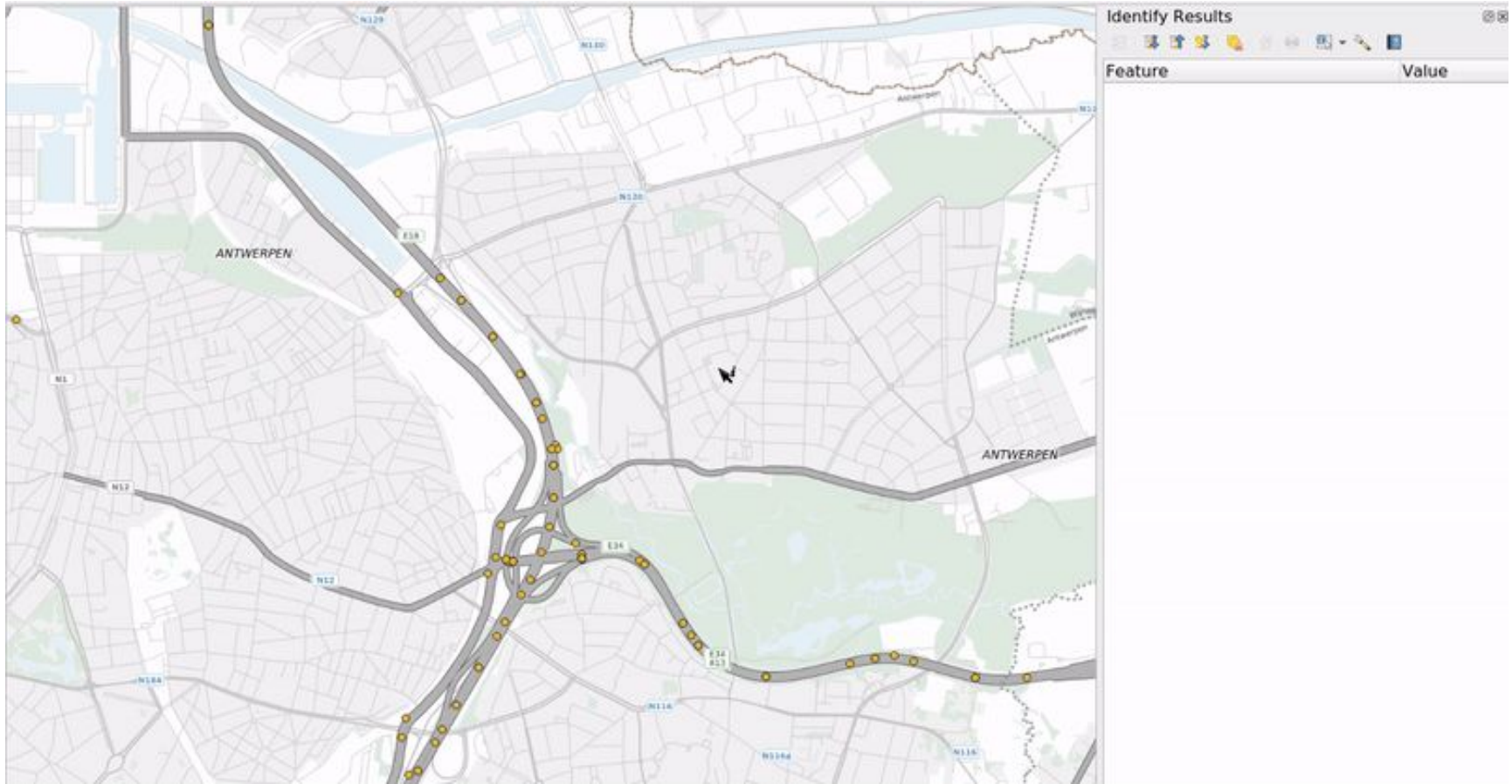


Downloadbaar van de GitHub als zip bestand (otlmow\_ie.zip)



# QGIS plugin





# QGIS plugins

## Resultaat na aanpassing

```
{  
  "assetId": {  
    "identificator": "06e3a7f5-23f8-484f-b13e-499c8b7d745d-b25kZXJkZWVsI0NhbWVyYQ",  
    "toegekendDoor": "AWV"  
  },  
  "datumOprichtingObject": "2018-03-14",  
  "geometry": "POINT Z (4.45116226009433458 51.22117498972576755 0)",  
  "isActief": true,  
  "isPtz": true,  
  "naam": "CAMA0660",  
  "toestand": "in-gebruik",  
  "typeURI": "https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#Camera"  
},
```







Vlaamse  
overheid

# Vragen?

Zoek mij tijdens de lunch!



# 5.2 OTL in SDF praktisch gebruik

Ruben Dewitte (AWV)

# Topics

## Overzicht

### IMKL vs OTL

- OTL formaten
- OTL subset

### OTL in SDF

- Schema
- IDs
- Geometrie
- Datatypes
- Keuzelijsten
- Bijlages

### OTL relaties

- Concept
- Formaat
- Tools







IMKL vs OTL

# Schema

## IMKL-model

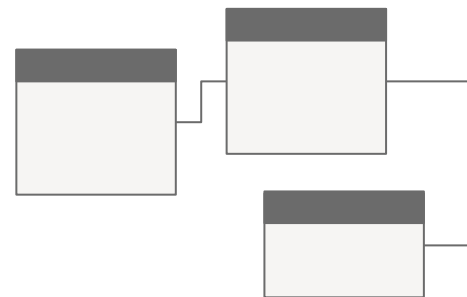
Cabinet  
Manhole  
Appurtenance  
Cable  
Pipe  
ActivityComplex  
Extraplan  
Werkplan

**Mapping**

*niet 1 op 1*

## OTL model

Gedetailleerde  
objecten met attributen en  
relaties



# Vereenvoudigde mapping

## Voorbeeld

IMKL	OTL	SUBSET
<b>Cabinet</b>	Cabine	
	Container	
	HSCabine	
	Hulppostkast	
	<b>Montagekast</b>	Montagekast
	<b>Wegkantkast</b>	Wegkantkast

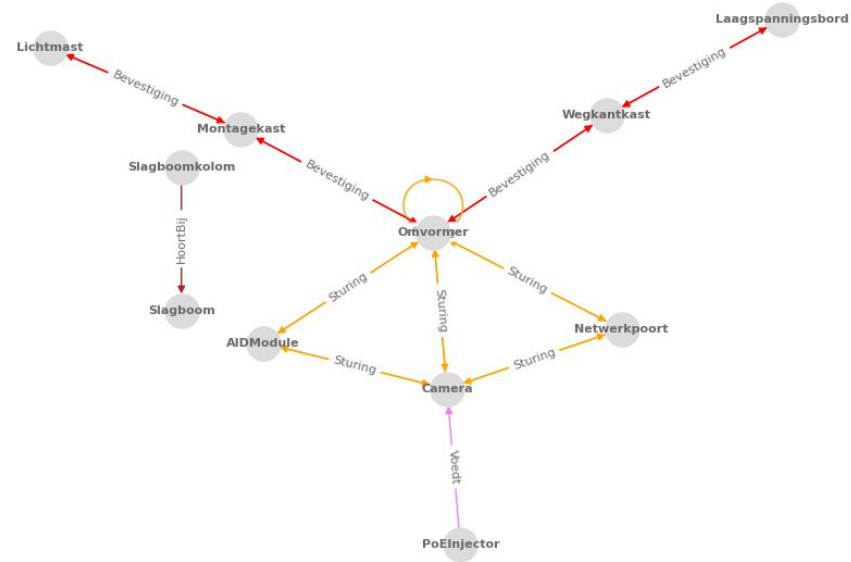
# OTL subset

**Projectspecifieke** selectie van  
OTL onderdelen en relaties

Kan per opdracht in samenspraak en  
ondersteund door team BIM









- geeft overzicht van alle klassen en attributen die je nodig hebt
- geeft overzicht van alle relaties die je nodig hebt
- onnodige zaken uitfilteren

SQLite database, te bekijken via  
subset-tool



visualisatie subset camerabestek

# DAVIE Formaten

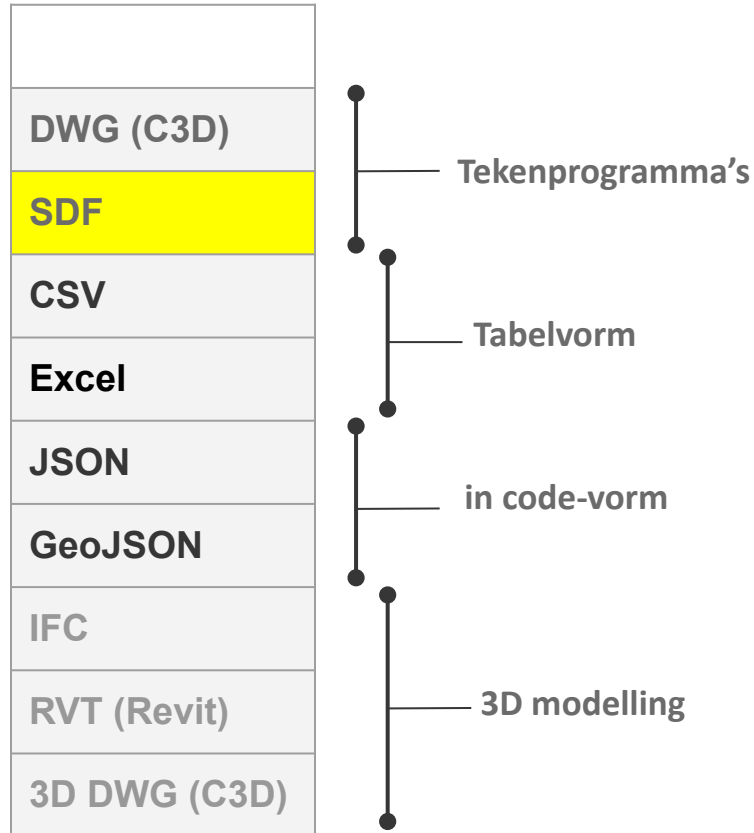
	Download	Aanlevering
 <b>DWG (C3D)</b>	X	✓
 <b>SDF</b>	✓	✓
 <b>CSV</b>	✓	✓
 <b>Excel</b>	✓	✓
 <b>JSON</b>	✓	✓
 <b>GeoJSON</b>	✓	✓
 <b>RVT (Revit)</b>	X	✓
 <b>IFC</b>	X	✓



# DAVIE Formaten

	(LOG 0) Geometrie	Data	Relaties
DWG (C3D)	✓	✓ in propertysets	✗
SDF	✓	✓	✗
CSV	✓ <sub>wkt</sub>	✓	✓
Excel	✓ <sub>wkt</sub>	✓	✓
JSON	✓ <sub>wkt</sub>	✓	✓
GeoJSON	✓	✓	✓
IFC	✗	✓ in ifcpropertyset	✗
RVT (Revit)	✗	✓ in shared parameters	✗

# DAVIE Formaten



## Welk formaat te kiezen?

Projectspecifiek afgesproken in

BIM Uitvoeringsplan op basis van

BIM opstartvergadering

(kan ook combinatie zijn).



OTL in SDF



# Stappenplan

Beschikbaar als webinar

Stap voor stap uitleg

[wegenenverkeer.be/zakelijk/bim/webinars-presentaties](https://wegenenverkeer.be/zakelijk/bim/webinars-presentaties)



# De verschillen

## Overzicht

IMKL conforme SDF	OTL conforme SDF/DWG
relatie tool (IMKL-tools)	<b>Relaties buiten SDF / DWG</b> <ul style="list-style-type: none"> <li>- via ander formaat (CSV/Excel/Json)</li> </ul>
voorgedefinieerd, vast schema	<b>Dynamisch schema</b> <ul style="list-style-type: none"> <li>- nodige klassen en attributen kunnen verschillen per aanlevering.</li> <li>- update mee met OTL updates</li> </ul>
vooringevulde keuzelijstopties, via dropdown	<b>Keuzelijsten opvragen</b> via OTL <ul style="list-style-type: none"> <li>- geen vaste template</li> <li>- keuzelijstopties kunnen uitgebreid worden</li> </ul>

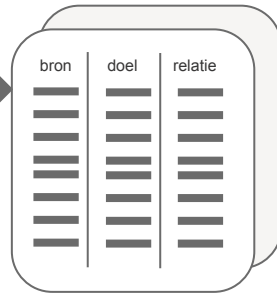
# Overzicht



samen in ZIP file  
bij downloaden



**SDF file**  
met OTL onderdelen en hun  
geometrie in juiste  
featureclasses.



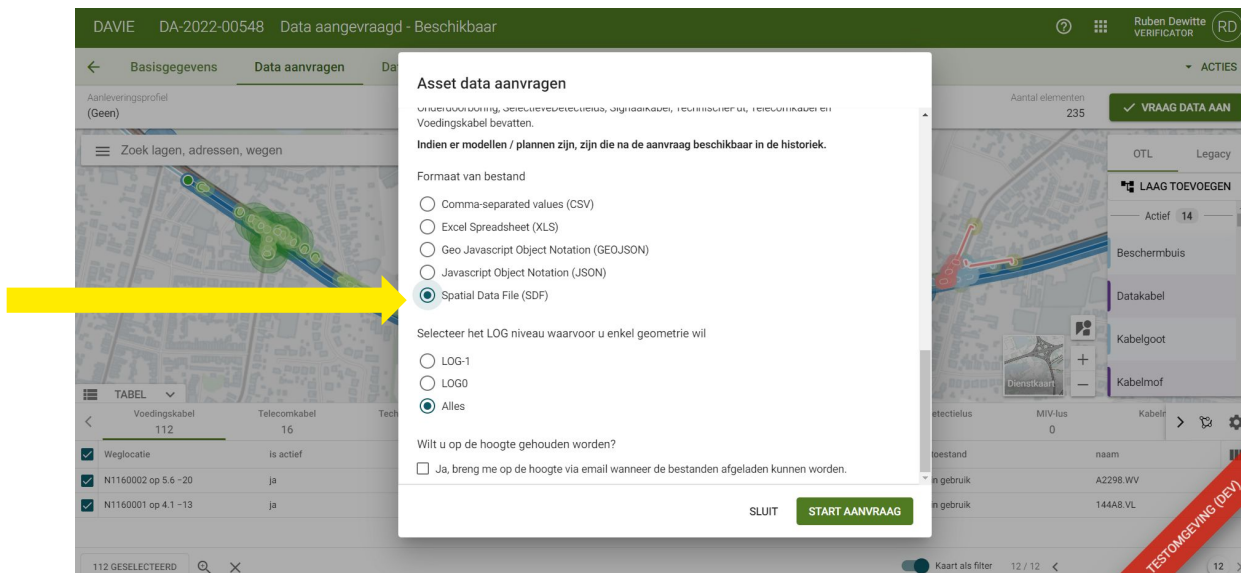
**CSV file**  
met OTL relaties tussen deze onderdelen.

# SDF download via DAVIE

## kaartweergave in DAVIE applicatie



# SDF download via DAVIE



DAVIE DA-2022-00548 Data aangevraagd - Beschikbaar

Ruben Dewitte  
VERIFICATOR

← Basisgegevens Data aanvragen

Aanleveringsprofiel  
(Geen)

Zoek lagen, adressen, wegen

Formaat van bestand

- ☐ Comma-separated values (CSV)
- ☐ Excel Spreadsheet (XLS)
- ☐ Geo Javascript Object Notation (GEOJSON)
- ☐ Javascript Object Notation (JSON)
- ☒ Spatial Data File (SDF)

Selecteer het LOG niveau waarvoor u enkel geometrie wilt

- ☐ LOG-1
- ☐ LOG0
- ☒ Alles

Wilt u op de hoogte gehouden worden?

☐ Ja, breng me op de hoogte via email wanneer de bestanden afgeladen kunnen worden.

SLUIT START AANVRAAG

Aantal elementen  
235

✓ VRAAG DATA AAN

OTL Legacy

LAAG TOEVOEGEN

Actief 14

Beschermhuis

Datakabel

Kabelgoot

Kabelmof

Dienstkant

etctielus MIV-lus 0 Kabel

bestand naam

gebruik A2298.VV

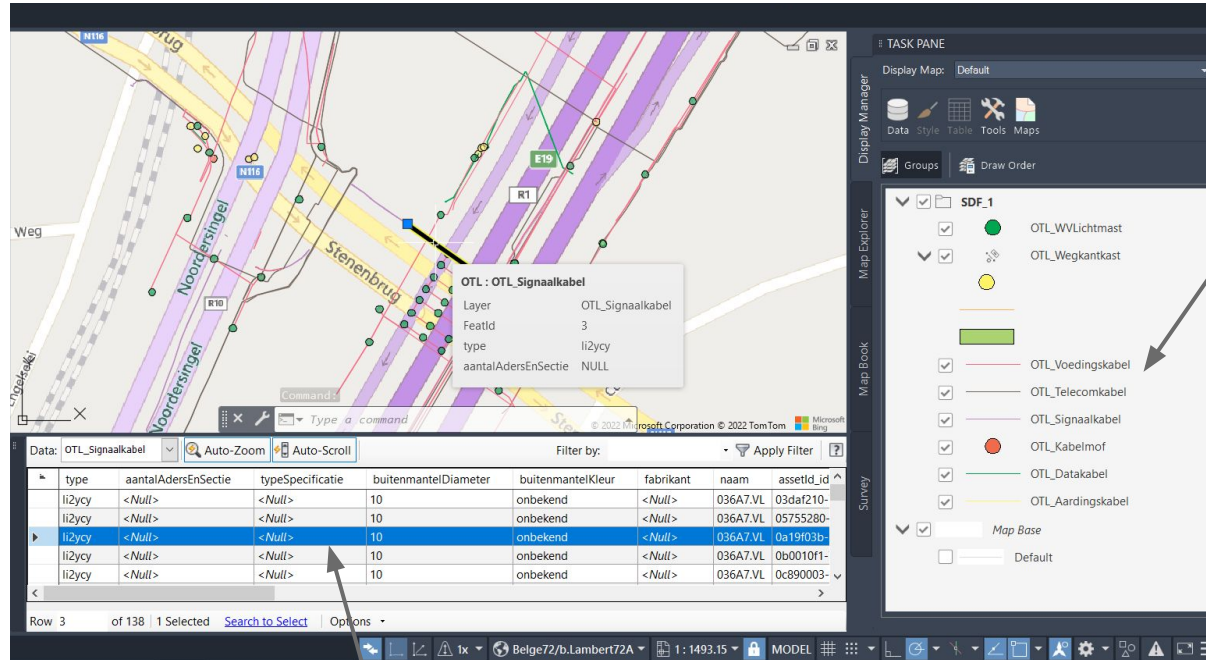
gebruik 144A8.VL

112 GESELECTEERD

Kaart als filter 12 / 12

TESTOMGEVING (DEV)

# Voorbeeld OTL in SDF



The screenshot shows the Autodesk MAP interface. The main map area displays a network of roads and utility lines. A tooltip for 'OTL : OTL\_Signaalkabel' is visible, showing attributes: Layer: OTL\_Signaalkabel, FeatId: 3, type: li2ycy, and aantalAdersEnSectie: NULL. Below the map is a data table with columns: type, aantalAdersEnSectie, typeSpecificatie, buitenmantelDiameter, buitenmantelKleur, fabrikant, naam, and assetId. The table contains 6 rows of data. To the right of the map is a 'TASK PANE' with a 'Feature Class' list for 'SDF\_1'. The list includes: OTL\_WVLichtmast, OTL\_Wegkantkast, OTL\_Voedingskabel, OTL\_Telecomkabel, OTL\_Signaalkabel, OTL\_Kabelmof, OTL\_Datakabel, and OTL\_Aardingskabel. An arrow points from the 'Feature Class' list to the text 'Feature Class per OTL onderdeel'. Another arrow points from the data table to the text 'OTL attributen in tabelvorm met dotnotatie ook attributen zonder info in getoond'.

type	aantalAdersEnSectie	typeSpecificatie	buitenmantelDiameter	buitenmantelKleur	fabrikant	naam	assetId
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	03daf210-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	05755280-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	0a19f03b-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	0b0010f1-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	0c890003-

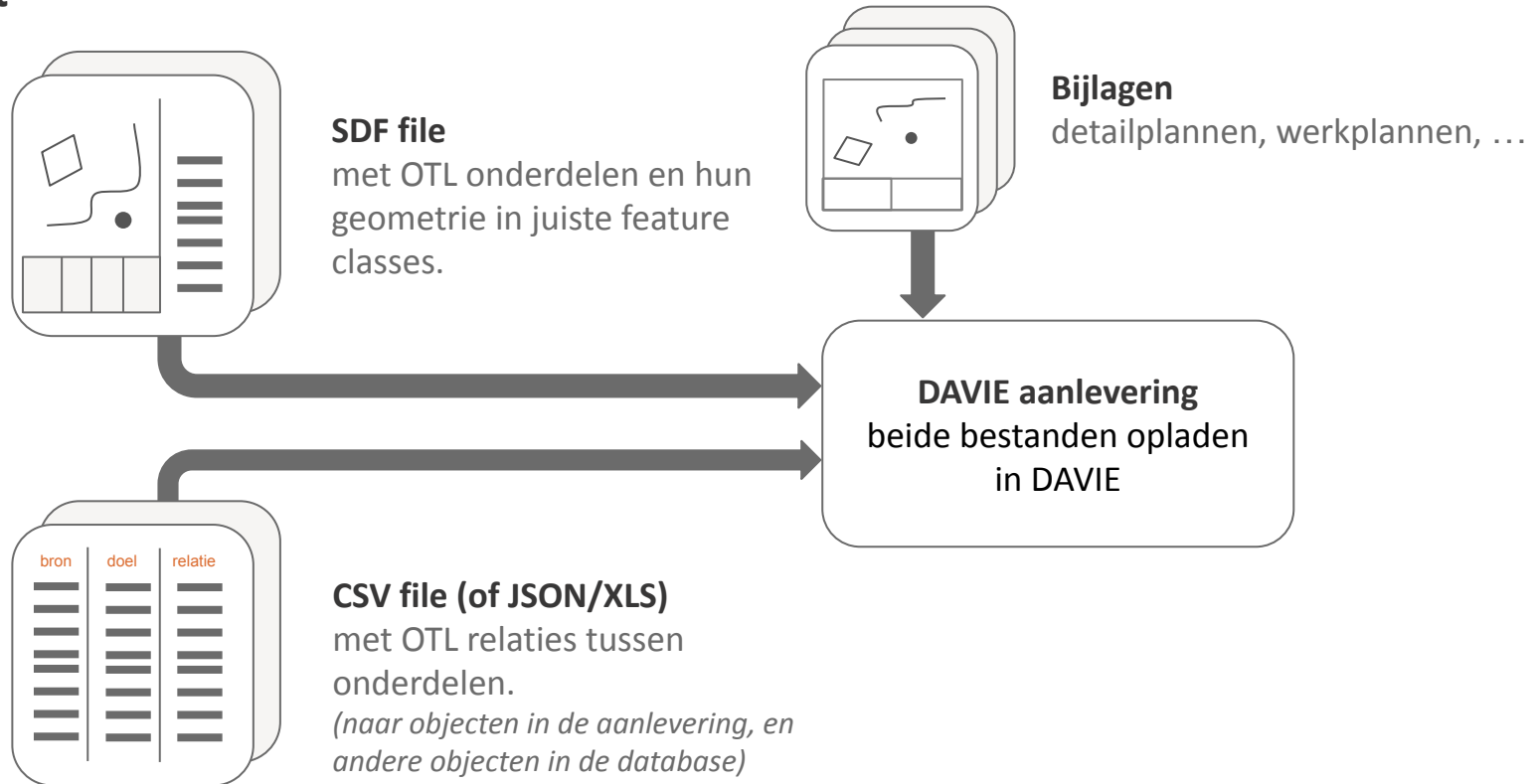
**Feature Class**  
per OTL onderdeel

**OTL attributen**

in tabelvorm met dotnotatie  
ook attributen zonder info in getoond

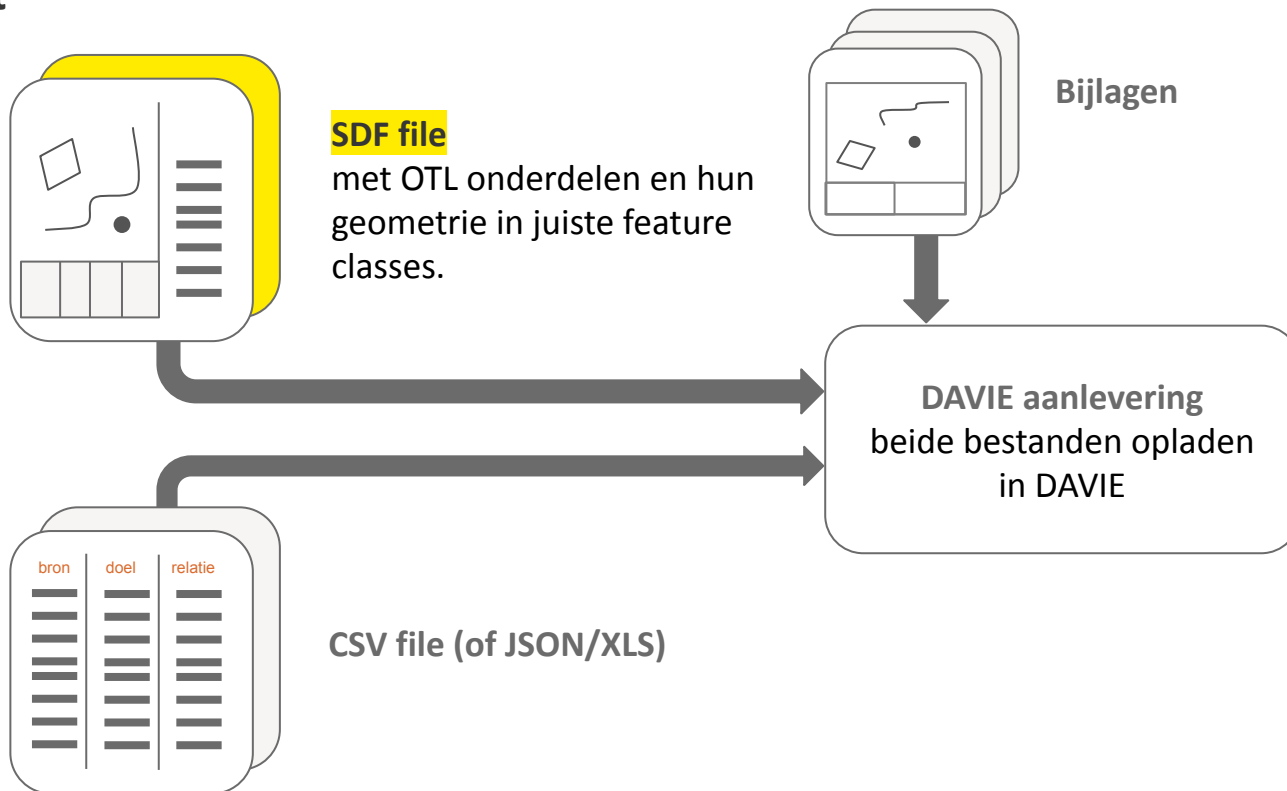
# SDF aanmaken

## Overzicht



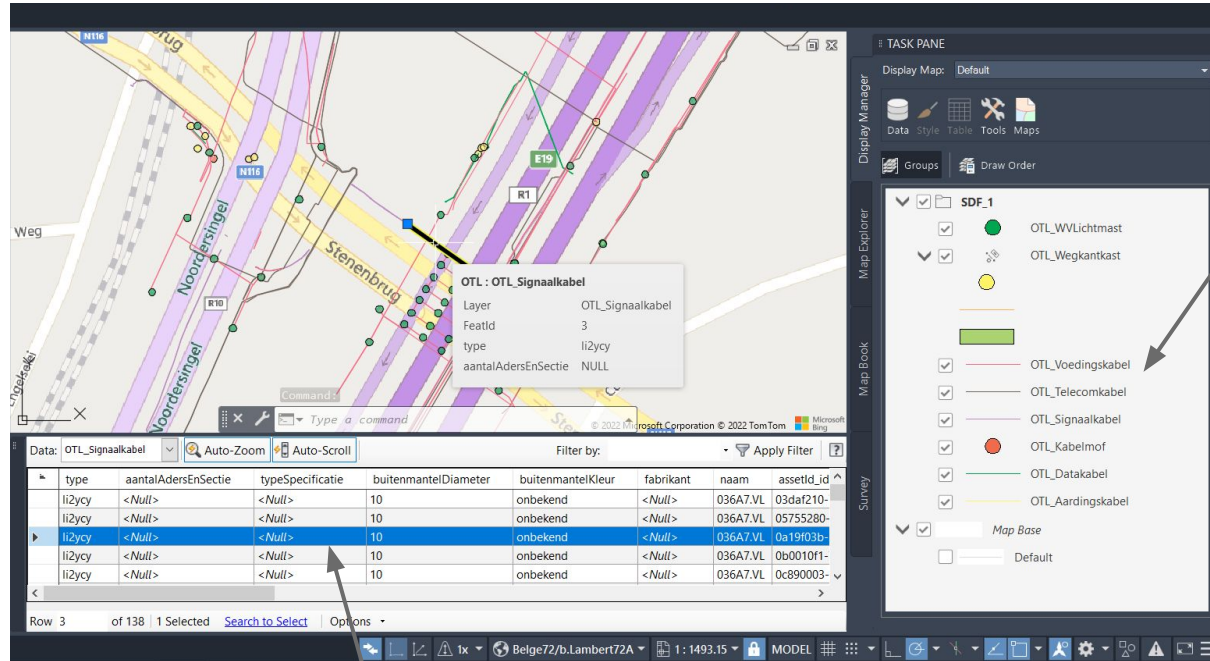
# SDF aanmaken

## Overzicht





# Voorbeeld OTL in SDF



The screenshot shows the Autodesk MAP interface. The main map area displays a network of roads and utility lines. A tooltip for 'OTL : OTL\_Signaalkabel' is visible, showing attributes: Layer: OTL\_Signaalkabel, FeatId: 3, type: li2ycy, and aantalAdersEnSectie: NULL. Below the map is a data table with columns: type, aantalAdersEnSectie, typeSpecificatie, buitenmantelDiameter, buitenmantelKleur, fabrikant, naam, and assetId. The table contains 6 rows of data. To the right of the map is a 'TASK PANE' with a 'Feature Class' list for 'SDF\_1'. The list includes: OTL\_WVLichtmast, OTL\_Wegkantkast, OTL\_Voedingskabel, OTL\_Telecomkabel, OTL\_Signaalkabel, OTL\_Kabelmof, OTL\_Datakabel, and OTL\_Aardingskabel. An arrow points from the 'Feature Class' list to the text 'Feature Class per OTL onderdeel'. Another arrow points from the data table to the text 'OTL attributen in tabelvorm met dotnotatie ook attributen zonder info in getoond'.

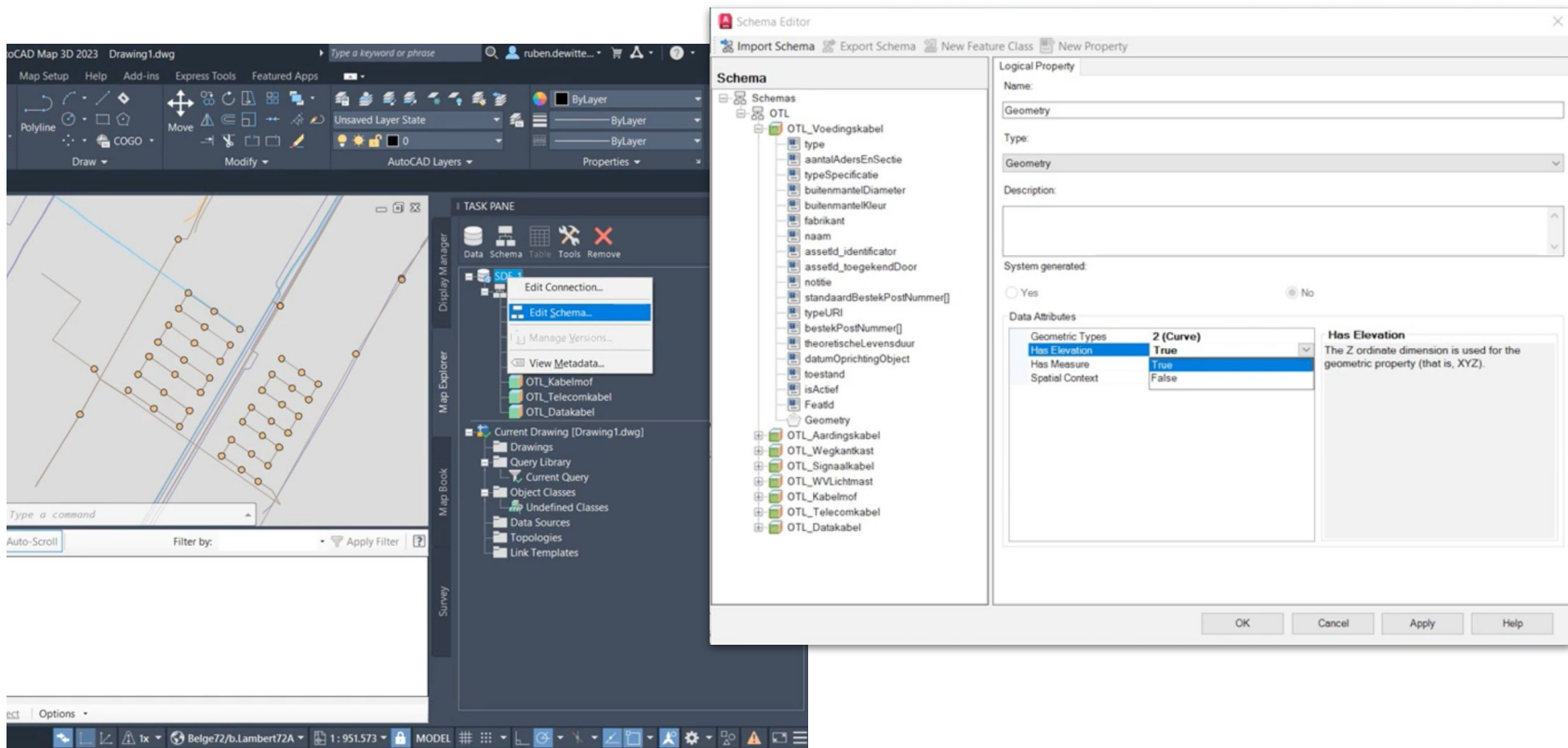
type	aantalAdersEnSectie	typeSpecificatie	buitenmantelDiameter	buitenmantelKleur	fabrikant	naam	assetId
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	03daf210-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	05755280-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	0a19f03b-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	0b0010f1-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	0c890003-

**Feature Class**  
per OTL onderdeel

**OTL attributen**

in tabelvorm met dotnotatie  
ook attributen zonder info in getoond

# OTL in SDF schema

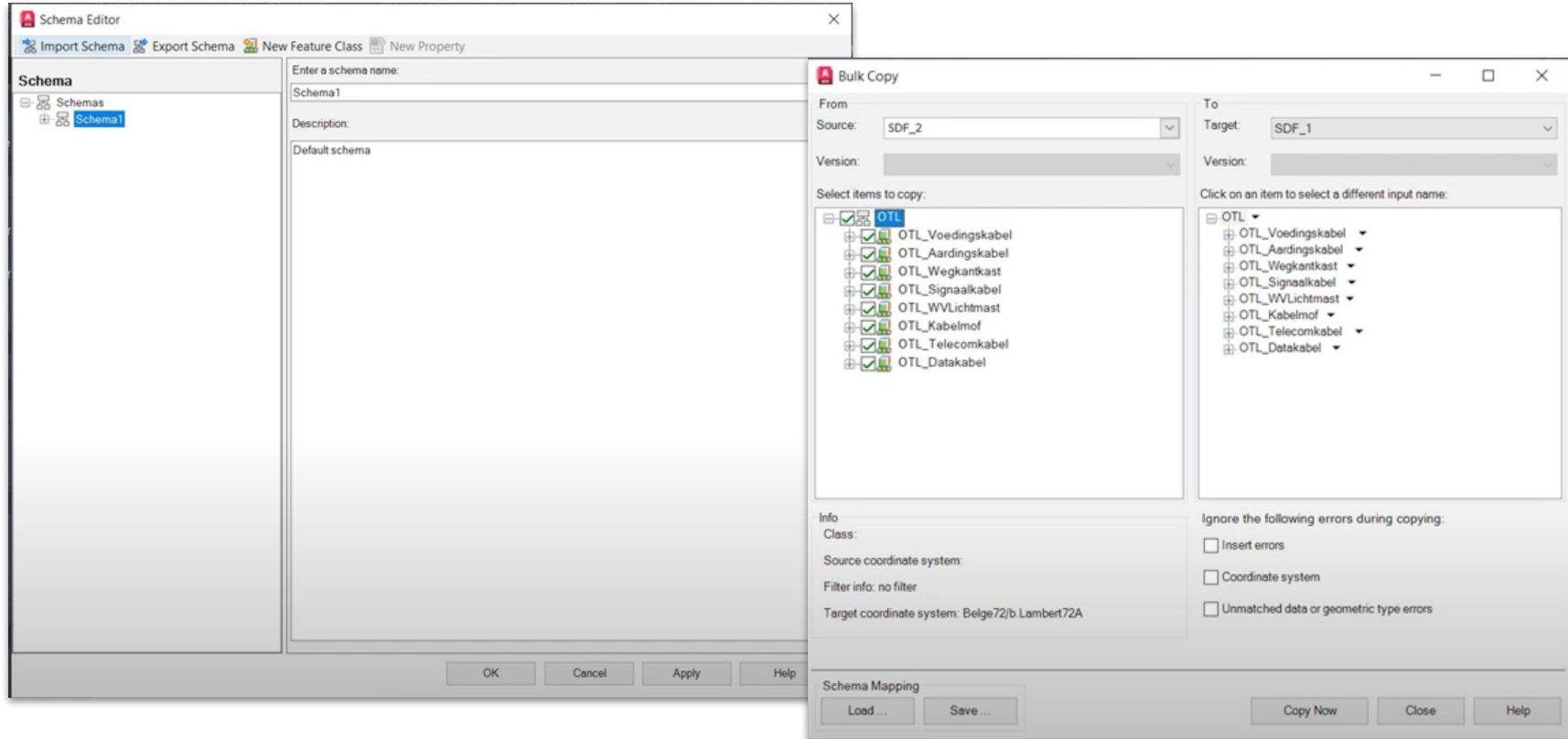


The screenshot displays the AutoCAD Map 3D 2023 interface with the SDF schema editor open. The main drawing area shows a map with various features. The Task Pane on the right contains the SDF schema editor, which is currently editing the OTL schema. The schema tree on the left lists various properties for the OTL schema, including type, aantalAdersEnSectie, typeSpecificatie, buitenmantelDiameter, buitenmantelKleur, fabrikant, naam, asseid\_identificator, asseid\_toegekendDoor, notie, standaardBestekPostNummer[], typeURI, bestekPostNummer[], theoretischLevensduur, datumOpriepingObject, toestand, isActief, FeatId, and Geometry. The right pane shows the Logical Property settings for the selected property, including Name, Geometry, Type, Description, System generated, and Data Attributes. The Data Attributes table shows the following values:

Geometric Types	2 (Curve)
Has Elevation	True
Has Measure	True
Spatial Context	False

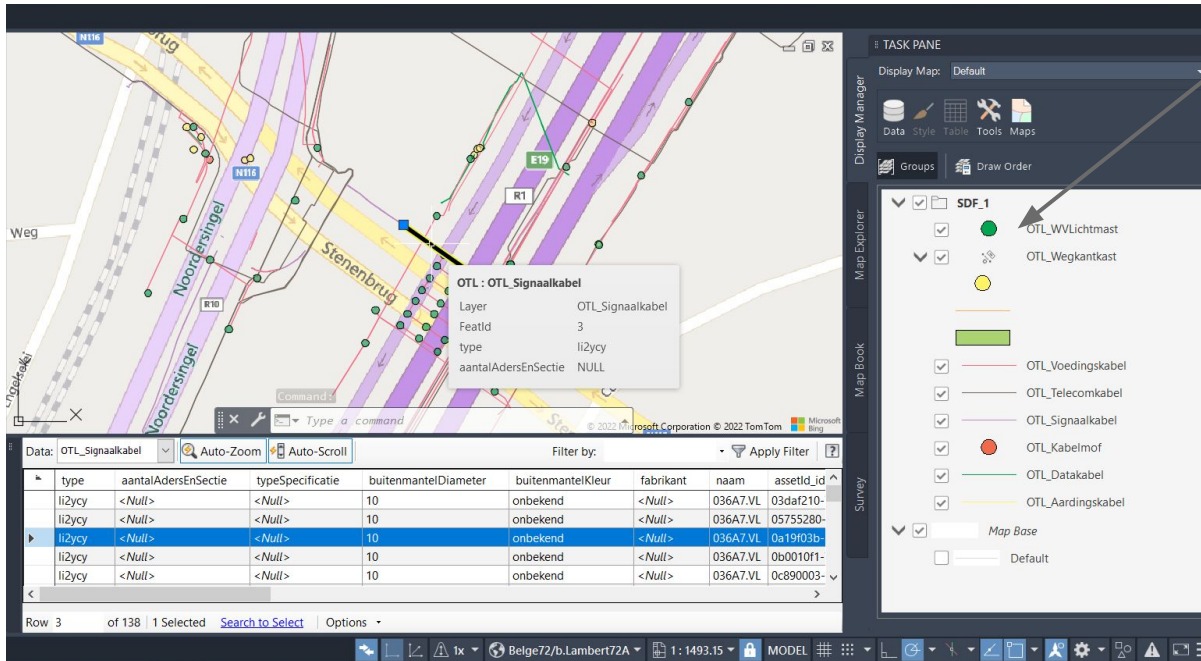
The 'Has Elevation' property is highlighted, and a note indicates: "The Z ordinate dimension is used for the geometric property (that is, XYZ)." The bottom status bar shows the current drawing is 'Drawing1.dwg' and the coordinate system is 'Belge72/b.Lambert72A'.

# OTL in SDF schema kopiëren en importeren



# Geometrie types in SDF

## adhv Geometrie Artefact



The screenshot shows the Autocad MAP interface. The main map area displays a road network with various colored lines representing different types of infrastructure. A tooltip for 'OTL : OTL\_Signaalkabel' is visible, showing properties like Layer, FeatId, type, and aantalAdersEnSectie. Below the map is a data table with columns for type, aantalAdersEnSectie, typeSpecificatie, buitenmantelDiameter, buitenmantelKleur, fabrikant, naam, and assetId. The table contains 138 rows, with the first row selected. To the right of the map is a 'TASK PANE' with a 'Display Map' dropdown and a 'Map Explorer' panel. The 'Map Explorer' panel shows a tree view of the map data, with 'SDF\_1' expanded. Under 'SDF\_1', there are several items, including 'OTL\_WVLichtmast' and 'OTL\_Wegkantkast'. An arrow points from the text 'Geometrie Artefact' to the 'OTL\_WVLichtmast' item in the 'Map Explorer' panel.

type	aantalAdersEnSectie	typeSpecificatie	buitenmantelDiameter	buitenmantelKleur	fabrikant	naam	assetId
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	03daf210-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	05755280-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	0a1903b-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	0b0010f1-
li2ycy	<Null>	<Null>	10	onbekend	<Null>	036A7.VL	0c890003-

### Geometrie Artefact

welk geometrietype  
per OTL onderdeel

# Geometrie types in SDF

## adhv Geometrie Artefact

Topografisch Legendeboek	Geometrie Artefact	SDF
Symbol	3D punt	<ul style="list-style-type: none"><li>- point</li><li>- multipoint</li></ul>
Veelhoekslijn	3D lijn	<ul style="list-style-type: none"><li>- line</li><li>- multiline</li></ul>
Gesloten veelhoekslijn	3D polygoon	<ul style="list-style-type: none"><li>- polygon</li><li>- multipolygon</li></ul>


elke vertex **3D-coördinaten**

(X,Y in Lambert72, en een TAW waarde als Z)




# Attributen invullen

## Overzicht



id	heeftStopcontact	kleur	specialeUitvoeringswijze	naam	assetId_identificator	assetId_oudergeometrie	notitie	standaardBestekPostNummer
<Null>	<Null>	<Null>	<Null>	<Null>	e5e6bc55-9852-4dd4-a4f8-b47b612b9f4-b25k2XkZwVv1dWTGijahRrYXNO	AWV	<Null>	<Null>
<Null>	<Null>	<Null>	<Null>	<Null>	ea665fa2-de52-4bfa-b6e5-ad067a456a07-b25k2XkZwVv1dWTGijahRrYXNO	AWV	<Null>	<Null>
<Null>	<Null>	<Null>	<Null>	<Null>	9f0a51b-08b7-4be0-8bae-2bfb14631b1-b25k2XkZwVv1dWTGijahRrYXNO	AWV	<Null>	<Null>
<Null>	<Null>	<Null>	<Null>	<Null>	1a9993b-08b2-4335-8b34-a2b354388b3-b25k2XkZwVv1dWTGijahRrYXNO	AWV	<Null>	<Null>
<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>



typeURI	bestekPostNummer	theoretischeLevensduur	datumOprichtingObject	toestand	isActief	elektrischeBeveiliging	FeatId
https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#WWLichtmast	<Null>	<Null>	2010-01-01	in-gebruik	True	<Null>	59
https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#WWLichtmast	<Null>	<Null>	2010-01-01	in-gebruik	True	<Null>	60
https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#WWLichtmast	<Null>	<Null>	2010-01-01	in-gebruik	True	<Null>	61
https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#WWLichtmast	<Null>	<Null>	2010-01-01	in-gebruik	True	<Null>	62
https://wegenverkeer.data.vlaanderen.be/ns/onderdeel#WWLichtmast	<Null>	<Null>	<Null>	<Null>	<Null>	<Null>	63



# Attributen invullen

## Overzicht datatypes

boolean	datum	keuzelijst	lege waarden (te negeren)	getallen	tekstvelden
adhv dropdown	niet via kalender maar via tekstveld als YYYY-MM-DD	als tekstveld mogelijke opties te bekijken via github pagina of via OTL publicatie website	Als een er geen waarde moet worden doorgegeven, neem je de OSLOAttribute niet op, laat je de waarde leeg, of kan je <Null> gebruiken.	integers en decimalen	string

heeftStopcontact

<Null>

True

False

<Null>

datumOprichtingObject

12:15:01

april 2022

ma	di	wo	do	vr	za	zo
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

Vandaag: 21/04/2022

kleur

<Null>



# Keuzelijstopties

## Via OTL web publicatie

<https://wegenverkeer.data.vlaanderen.be/>

**WV lichtmast**

**Beschrijving**  
Paal voorzien van armen voor het bevestigen van wegverlichtingstoestellen. Omvat het deurtje, klemmenblok, montagekastje, de bevestigingsmaterialen (bv. voetplaten) en fundering of verankeringsmassief. Gebruik Lichtmast voor de bevestiging van andere toestellen dan wegverlichting.

**Subklasse van**  
[Lichtmast](#)

**Eigenschappen**  
Voor deze entiteit zijn de volgende eigenschappen gedefinieerd: [aantal armen](#), [armenlengte](#), [bevestiging](#), [bevestiging toestellen](#).

Eigenschap	Verwacht Type	Kardinaliteit	Beschrijving	Gebruik	Keuzelijst
<a href="#">aantal armen</a>	<a href="#">WV-mast aantal armen</a>	1	Aantal armen van de lichtmast.		<a href="#">Link</a>
<a href="#">armenlengte</a>	<a href="#">WV-mast armenlengte</a>	1	Lengte van de arm van de lichtmast in meter.		<a href="#">Link</a>

## Conceptscheme: KIWV LichtmastAantArmen

< <https://wegenverkeer.data.vlaanderen.be/id/conceptscheme/KIWV LichtmastAantArmen>

**Gegevens**

label  
definitie  
heeft topconcepten

WV-mast aantal armen  
Aantal armen van de lichtmast.

> 0  
> 1  
> 2  
> 3

[Meer](#)

dataset  
<https://data.vlaanderen.be/id/dataset/codelist>

OF

## Github pagina: keuzelijsten als ttl-files

<https://github.com/Informatievlaanderen/OSLOthema-wegenverkeer/tree/master/codelisten>

Informatievlaanderen / OSLOthema-wegenverkeer Public

< Code Issues 41 Pull requests Actions Projects Wiki Security Insights

master OSLOthema-wegenverkeer / codelisten /

OAT-beheer Oat Tool

- AntiParkeerpaalType.ttl Pub ttls with status
- KL LuidsprekerVormgeving.ttl Pub ttls with status
- KI AIDModuleType.ttl Pub ttls with status
- KI AIDToestand.ttl Pub ttls with status
- KIANPRMerk.ttl Pub ttls with status
- KIANPRModelnaam.ttl Pub ttls with status
- KI AanplantingswijzeSierbeplanting.ttl Pub ttls with status

Info: Pub ttls with status ✓

Latest commit: vub018: v1.0.0 History

All 2 contributors

49 Lines 143 Slots 3.43 KB

```

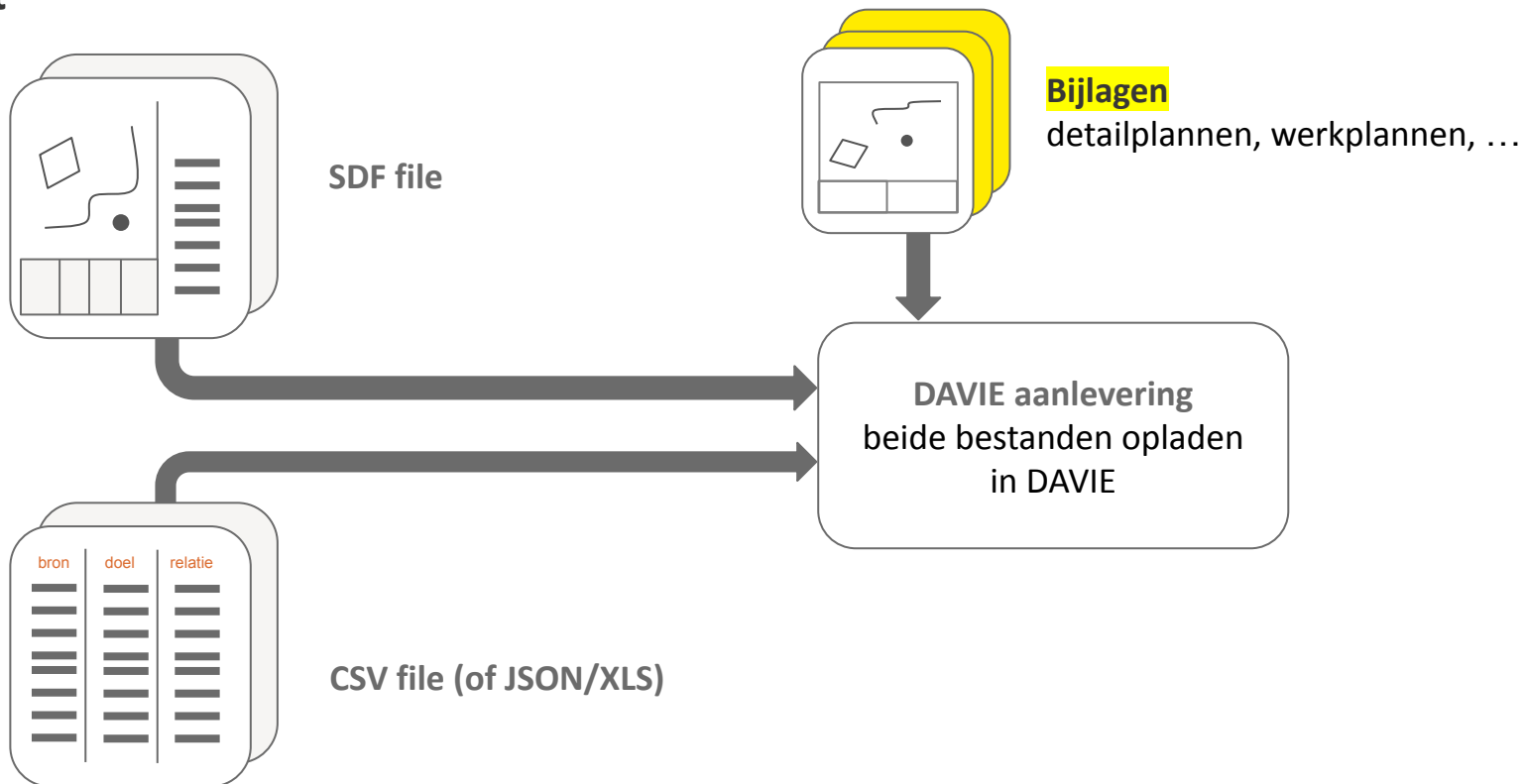
1  @prefix owl: <http://www.w3.org/2002/07/owl#> .
2  @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
3  @prefix skos: <http://www.w3.org/2004/02/skos/core#> .
4  @prefix owl: <http://www.w3.org/2002/07/owl#> .
5
6  <https://wegenverkeer.data.vlaanderen.be/id/conceptscheme/KIWV LichtmastAantArmen> a skos:ConceptScheme ;
7  <https://wegenverkeer.data.vlaanderen.be/id/conceptscheme/KIWV LichtmastAantArmen> a skos:Concept ;
8  skos:inScheme <https://wegenverkeer.data.vlaanderen.be/id/conceptscheme/KIWV LichtmastAantArmen> ;
9  skos:prefLabel "WV-mast aantal armen" ;
10
11 <https://wegenverkeer.data.vlaanderen.be/id/conceptscheme/KIWV LichtmastAantArmen> a skos:Concept ;
12 <https://wegenverkeer.data.vlaanderen.be/id/conceptscheme/KIWV LichtmastAantArmen> a skos:Concept ;
13 skos:prefLabel "WV-mast aantal armen van de Lichtmast" ;
14 skos:inScheme <https://wegenverkeer.data.vlaanderen.be/id/conceptscheme/KIWV LichtmastAantArmen> ;
15 skos:prefLabel "WV" ;
16 skos:prefLabel "WV" ;
17 skos:prefLabel "WV" ;
18

```



# SDF aanmaken

## Overzicht



# Aanvullende geometrie

## werkplan, detailplan, ..

wegenverkeer.data.vlaanderen.be/doc/implementatiemodel/master/#Aanvullende%20geometrie

### Aanvullende geometrie

**Beschrijving**  
Beschrijft een geometrie die aanvullend is bij de de werking van een asset maar beschrijft niet de asset zelf, bv. een detailplan of een werkingsgebied met een specifieke locatie, enz..De aanvullende geometrie kan al dan niet een bijlage bevatten.

**Subklasse van**  
[Abstracte aanvullende geometrie](#)

**Eigenschappen**  
Voor deze entiteit zijn de volgende eigenschappen gedefinieerd: [type](#).

Eigenschap	Verwacht Type	Kardinaliteit	Beschrijving	Gebruik	Keuzelijst
<a href="#">type</a>	<a href="#">Bijlage met geometrie type</a>	1	Het type van aanvullende geometrie.		<a href="#">Link</a>



Vlaanderen DATA VLAANDEREN

## Conceptscheme: KlBijlageMetGeometrieType

< <https://wegenverkeer.data.vlaanderen.be/id/conceptscheme/KlBijlageMetGeometrieType>

**Gegevens**

label  
Bijlage met geometrie type

definitie  
De mogelijke opties als type van een bijlage met geometrie.

heeft topconcepten

- > [andere](#)
- > [werkplan](#)
- > [detailplan](#)
- > [field of view](#)
- > [veiligheidsvoorschriften](#)
- > [lengteprofiel](#)

[Meer](#)

dataset  
<https://data.vlaanderen.be/id/dataset/codelist>

# Aanvullende geometrie

## Heeft aanvullende geometrie relatie



**geometrie** die aanvullend is bij de werking van een asset maar beschrijft niet de asset zelf

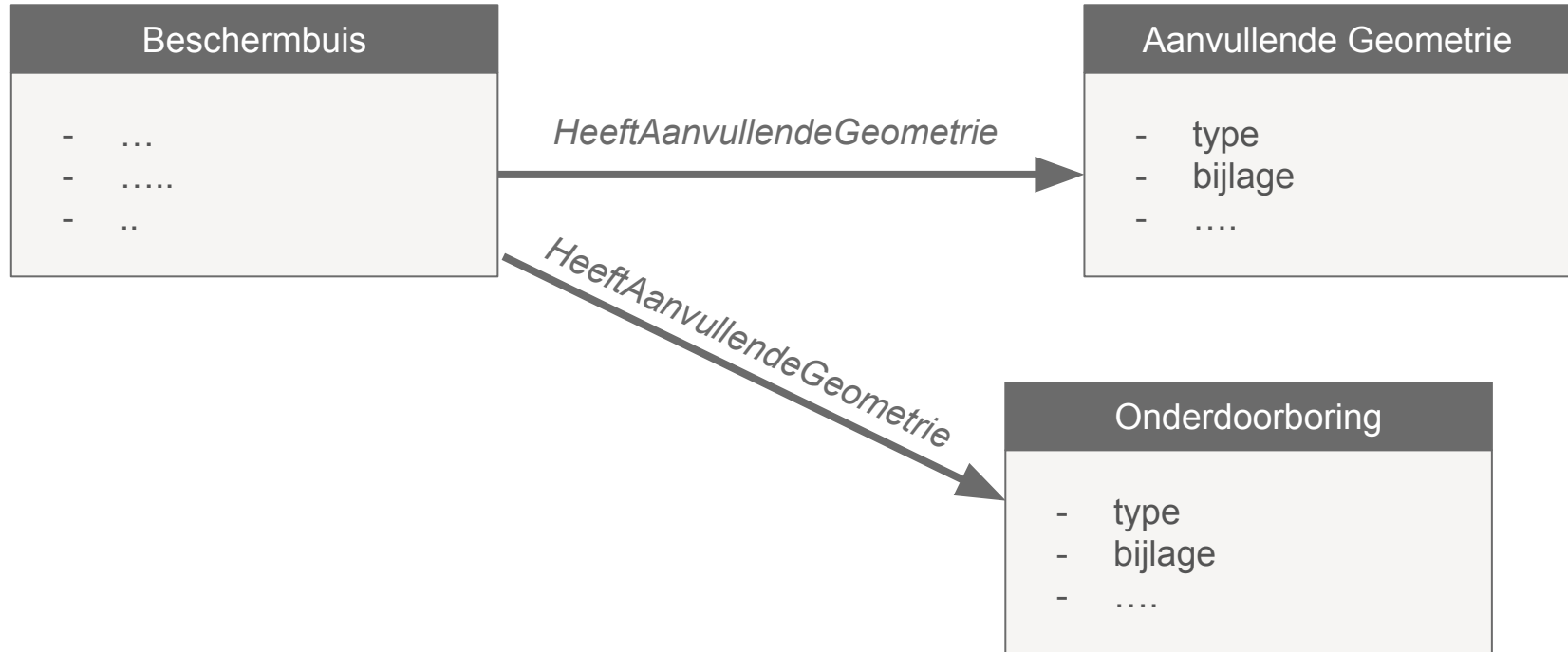
### voorbeeld

- een detailplan
- een werkingsgebied met een specifieke locatie

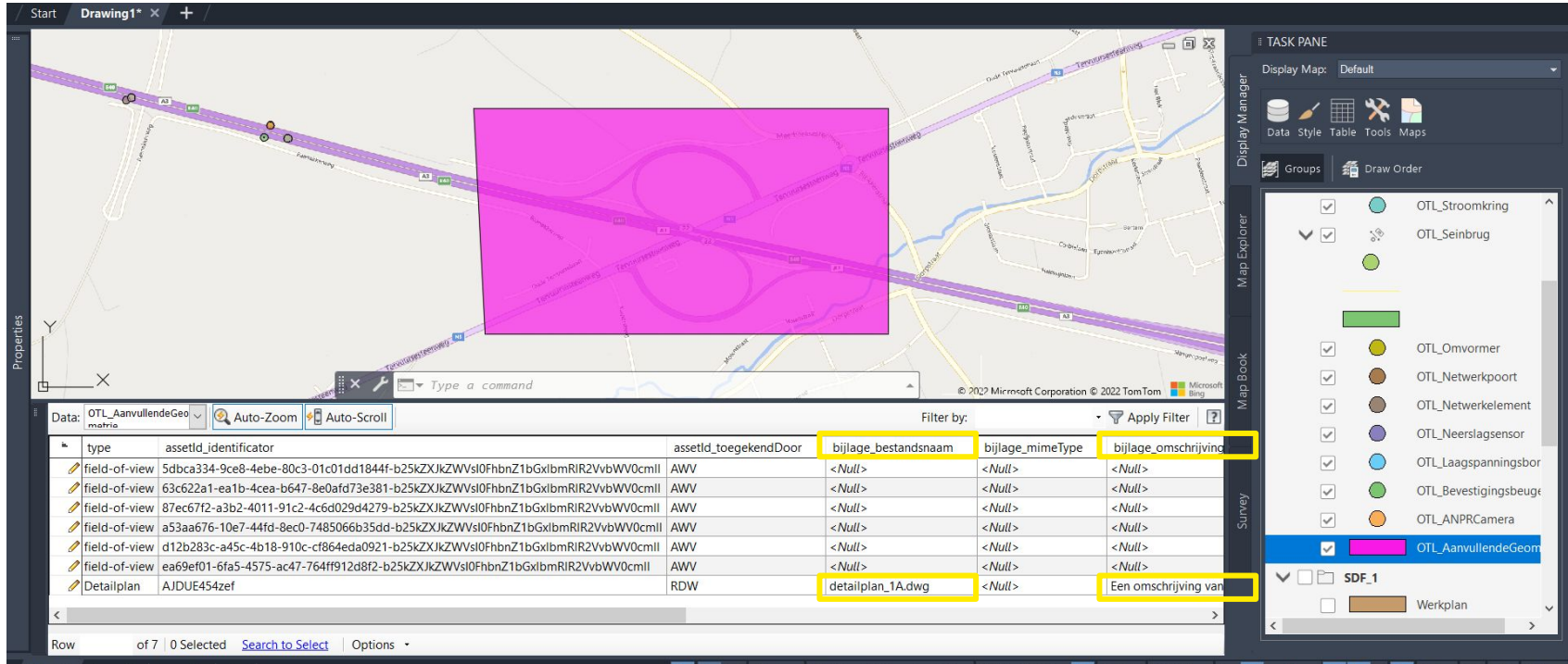
In het geval de aanvullende geometrie van het type detailplan is, zal deze via de attributen een link vormen naar het desbetreffende plan in bijlage.

# Aanvullende geometrie

## Heeft aanvullende geometrie relatie



# Voorbeeld werkplan / extraplan



The screenshot shows a GIS application interface. The main map area displays a pink polygon overlaid on a street map. Below the map is a data table with the following columns: type, assetId, assetId\_identificator, assetId\_toegekendDoor, bijlage\_bestandsnaam, bijlage\_mimeType, and bijlage\_omschrijving. The table contains 7 rows of data, with the first 6 rows representing 'field-of-view' assets and the 7th row representing a 'Detailplan' asset. The 'OTL\_AanvullendeGeometrie' asset is highlighted in blue in the table and in the map. The 'Detailplan' asset is also highlighted in yellow in the table and in the map.

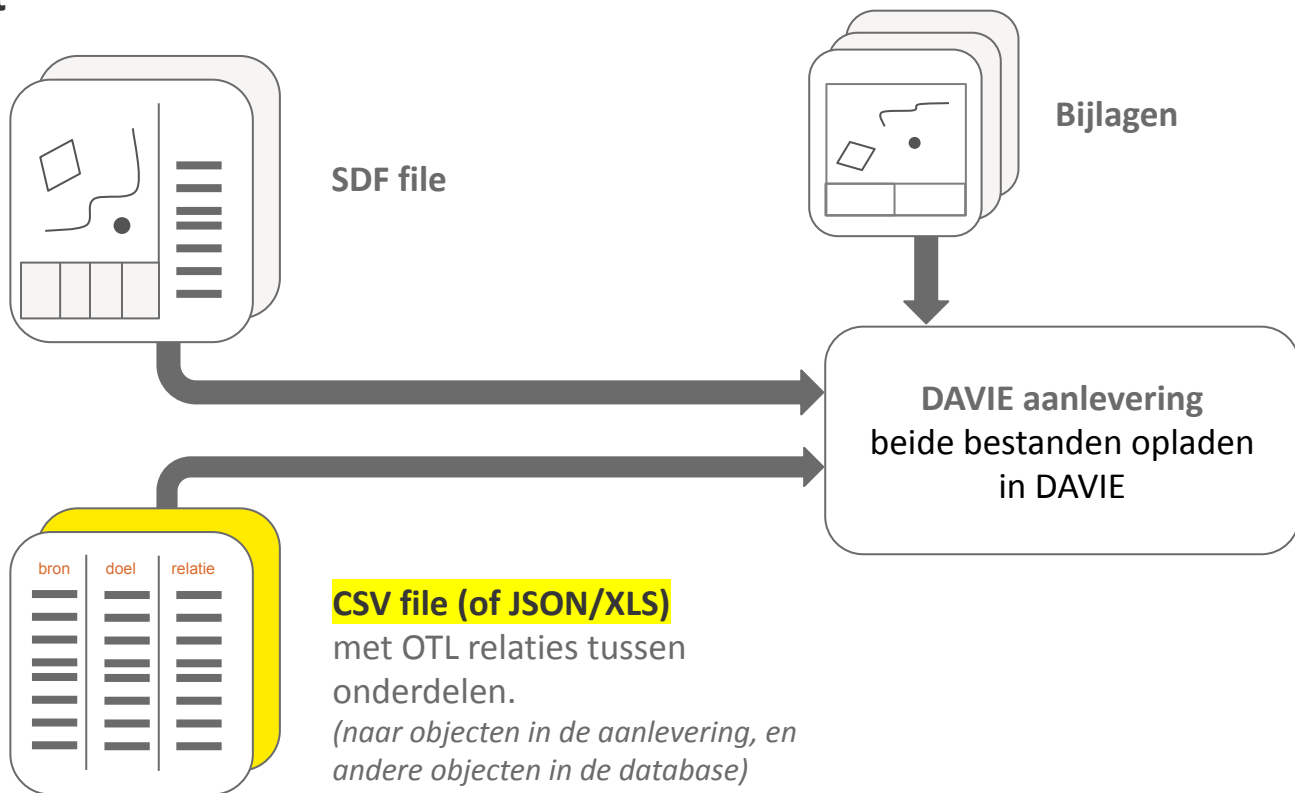
type	assetId	assetId_identificator	assetId_toegekendDoor	bijlage_bestandsnaam	bijlage_mimeType	bijlage_omschrijving
field-of-view	5dbc334-9ce8-4ebe-80c3-01c1dd1844f-b25kZKJkZWVsl0FhbnZ1bGxlbmRIR2VvbWV0cmll	AWV	<Null>	<Null>	<Null>	<Null>
field-of-view	63c622a1-ea1b-4cea-b647-8e0afd73e381-b25kZKJkZWVsl0FhbnZ1bGxlbmRIR2VvbWV0cmll	AWV	<Null>	<Null>	<Null>	<Null>
field-of-view	87ec67f2-a3b2-4011-91c2-4c6d029d4279-b25kZKJkZWVsl0FhbnZ1bGxlbmRIR2VvbWV0cmll	AWV	<Null>	<Null>	<Null>	<Null>
field-of-view	a53aa676-10e7-44fd-8ec0-7485066b35dd-b25kZKJkZWVsl0FhbnZ1bGxlbmRIR2VvbWV0cmll	AWV	<Null>	<Null>	<Null>	<Null>
field-of-view	d12b283c-a45c-4b18-910c-cf864eda0921-b25kZKJkZWVsl0FhbnZ1bGxlbmRIR2VvbWV0cmll	AWV	<Null>	<Null>	<Null>	<Null>
field-of-view	ea69ef01-6fa5-4575-ac47-764ff912d8f2-b25kZKJkZWVsl0FhbnZ1bGxlbmRIR2VvbWV0cmll	AWV	<Null>	<Null>	<Null>	<Null>
Detailplan	AJDUE454zef	RDW	<Null>	detailplan_1A.dwg	<Null>	Een omschrijving van

Alle assets getoond op dit plan worden gelinkt via **HeeftAanvullendeGeometrie Relatie**

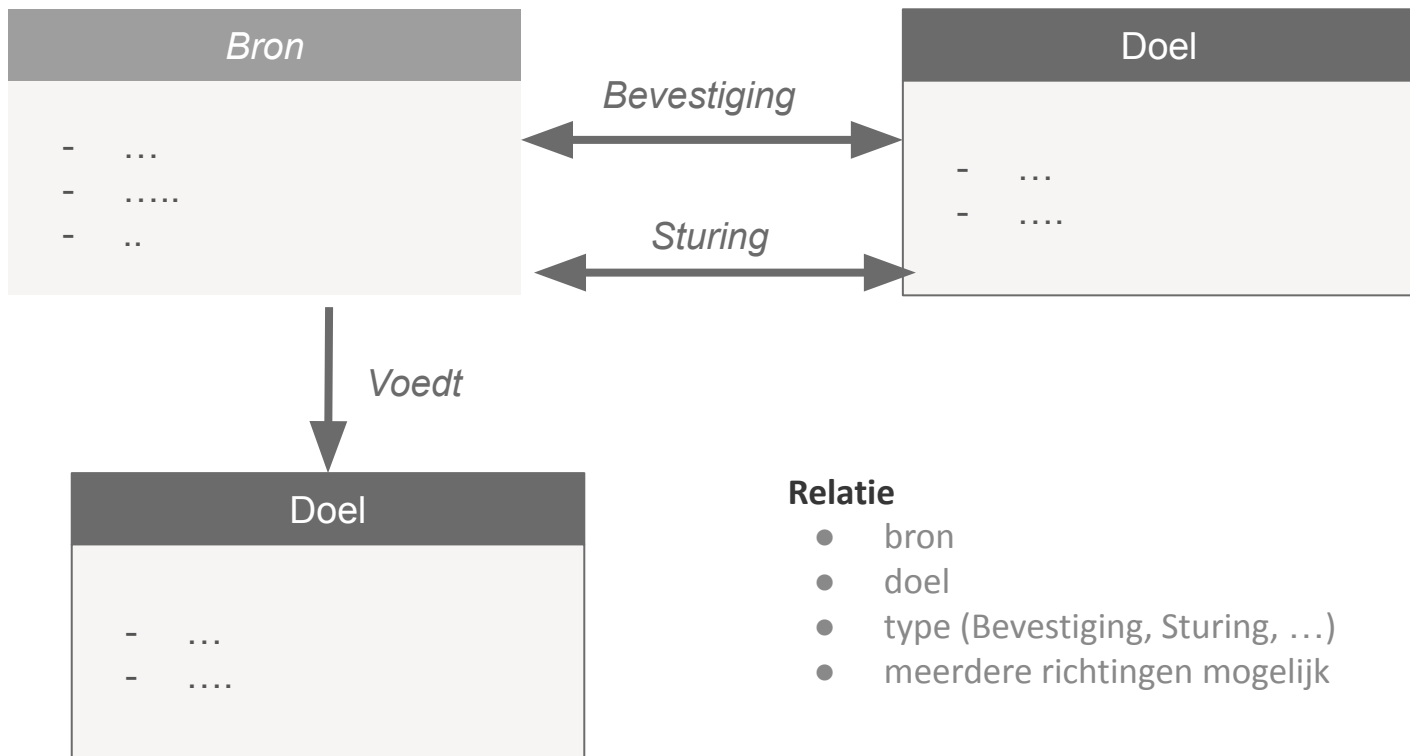


OTL relaties

## Overzicht



# OTL relaties





# OTL relaties

## Voorbeeld van aangemaakte relaties tussen OTL onderdelen

- CSV, Excel, JSON
- Op OTL AssetId's (niet feature IDs)
- isActief gebruiken om bestaande relatie te verwijderen

OTL\_Bevestiging

assetId.identificator	bronAssetId.identificator	doelAssetId.identificator	typeURI	isActief	assetId.toegekendDoor
c1598ce4-d317-4232-af40-3864c5745d6d	e5bcbe44-8c7a-49b7-8bbd-ee703dc222cb	bf8a6c11-e250-46f3-a5aa-370a91179274	<a href="https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging">https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging</a>	true	AWV
12826eb7-6f4e-41b3-86fe-70a5c73a0c48	5c30742d-ce77-423c-a827-88330987f4ca	9cc75e6c-186b-441e-8dc9-57beab73f838	<a href="https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging">https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging</a>	true	AWV
cc974a0d-eb6e-4ae3-b25a-2610fc76a615	26127a65-c2ae-4f9a-8863-176089ed287f	bf8dedbd-c2cc-44b7-964f-91a173b16ccc	<a href="https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging">https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging</a>	true	AWV
ed390cee-e6a3-4469-9875-a31ae647c5fe	26127a65-c2ae-4f9a-8863-176089ed287f	bf8dedbd-c2cc-44b7-964f-91a173b16ccc	<a href="https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging">https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging</a>	false	AWV
0a684268-0a89-4eba-8726-adf298d1cf63	5c30742d-ce77-423c-a827-88330987f4ca	9cc75e6c-186b-441e-8dc9-57beab73f838	<a href="https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging">https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging</a>	true	AWV

	A	B	C	D	E	F	G	H	I	J	K
1	assetId.id	bronAssetId.identificator	doelAssetId.identificator	typeURI							
2	reltype1	AIDModule	Agent	<a href="https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#HeeftBetrokkene">https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#HeeftBetrokkene</a>							
3	reltype3	ANPRCamera	Verkeersbordsteun	<a href="https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging">https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Bevestiging</a>							
4	reltype4	ANPRCamera	Netwerkpoot	<a href="https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Sturing">https://wegenenverkeer.data.vlaanderen.be/ns/onderdeel#Sturing</a>							

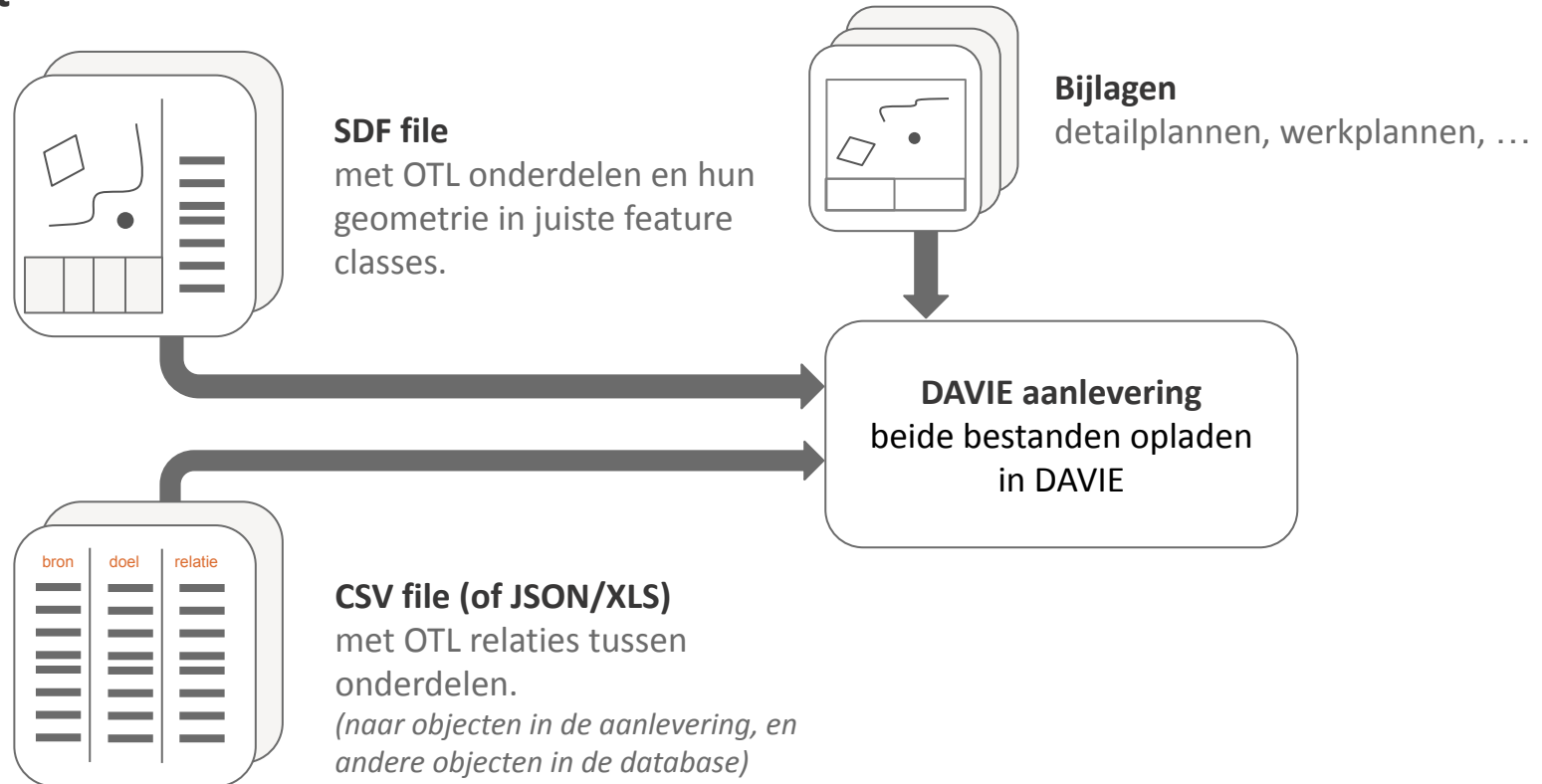
# OTL relaties

## Beschikbare hulpmiddelen

Vroeger	Nieuwe tools
<p data-bbox="200 390 602 426"><b>Autocad map IMKL plugin</b></p> <p data-bbox="200 481 755 517"><b><u>Niet</u></b> bruikbaar voor OTL aanlevering</p> <ul data-bbox="239 528 958 694" style="list-style-type: none"><li>- ander schema</li><li>- relaties anders opgebouwd</li><li>- verwijderen van assets en relaties werkt op andere manier</li></ul>	<p data-bbox="1010 390 1180 426"><b>OTL wizard</b></p> <ul data-bbox="1049 437 1676 603" style="list-style-type: none"><li>- tool met gebruikersinterface</li><li>- voorbeeldbestanden genereren</li><li>- tool om relaties aan te maken</li><li>- meer info in starterspakket en sessies</li></ul> <p data-bbox="1010 661 1348 697"><b>Software leveranciers</b></p> <ul data-bbox="1049 708 1740 831" style="list-style-type: none"><li>- ontwikkelingen lopende</li><li>- contacteer uw reseller of spreek ze aan in de netwerk-tuin</li></ul> <p data-bbox="1010 888 1437 924"><b>Open source Python library</b></p> <ul data-bbox="1049 954 1692 1027" style="list-style-type: none"><li>- <a href="https://github.com/davidvlaminck/OTLMOW">https://github.com/davidvlaminck/OTLMOW</a></li><li>- readme op github</li></ul>



# Overzicht





# Technische specificaties + Ondersteuning voor OTL via SDF

Aanvragen via: [TeamBim@verzendlijst.wegenenverkeer.be](mailto:TeamBim@verzendlijst.wegenenverkeer.be)

## Stap voor stap uitleg

[wegenenverkeer.be/zakelijk/bim/webinars-presentaties](https://wegenenverkeer.be/zakelijk/bim/webinars-presentaties)





Vlaamse  
overheid

Vragen?

