

# ONBOARDING VSIDS

Implementation Guideline

Versie /// 0.9

Publicatiedatum /// 05 December 2023

Auteur: Koenraad Verduyn

Datum aanmaak: 5 December 2023

Datum afdruk: 8 december 2023

Interne bestandsnaam: Implementation guideline V0.91.docx

Documenthistoriek:

<b>Versie</b>	<b>Opmerking</b>	<b>Datum</b>	<b>Auteur</b>	<b>Status</b>
[versie]	[opmerking]	[datum]	[auteur]	[status]

**Digitaal Vlaanderen**

Havenlaan 88, 1000 Brussel  
+32 (0)2 553 72 02

Koningin Maria Hendrikaplein 70, 9000 Gent  
+32 (0)9 276 15 00

digitaal.vlaanderen@vlaanderen.be



# INHOUD

Inhoud..... 3

1 Context..... 5

1.1 VSDS ..... 5

1.2 OSLO..... 5

1.3 LDES ..... 6

2 Use case Verkeersmetingen – Doorsnede meting ..... 7

2.1 Wat is een doorsnede meting? ..... 7

2.2 Implementatiemodel ..... 8

2.3 Data vereisten ..... 8

2.3.1 Telgegevens ..... 8

2.3.2 Locatie ..... 9

2.3.3 Tijdstip ..... 10

2.3.4 Voertuigtype..... 10

2.3.5 Snelheidsmeting ..... 10

2.3.6 Telling ..... 10

2.3.7 Type Sensor ..... 11

2.3.8 Observatieprocedure..... 11

2.3.9 Extra klassen ..... 11

2.4 data Voorbeeld ..... 12

2.5 Architectuur ..... 18

2.6 Implementatie Proces ..... 20

2.6.1 Voorbereiding..... 21

2.6.2 OSLO mapping ..... 21

2.6.3 Configuratie LDIO ..... 21

2.6.4 Deployment..... 21

2.6.5 Metadata deployment..... 22

2.7 referenties..... 22





# 1 CONTEXT

## 1.1 VSIDS

Met de Vlaamse Smart Data Space zetten we in op het vindbaar, toegankelijk en makkelijk (her)bruikbaar maken van data, kortom duurzaam data delen. Dit doen we door technische standaarden en bouwblokken te ontwerpen, en daar een afsprakenkader rond te voorzien dat het vertrouwen tussen de partijen bevordert.

We nemen vanuit Vlaanderen een verbindende rol op tussen de publieke actoren en private actoren die data (met elkaar) uitwisselen. Dit moet leiden tot een open ecosysteem waarin partijen op een betrouwbare en gestandaardiseerde manier data publiceren en consumeren. Via standaarden en samenwerkingsprotocollen stimuleren we de uitbouw van smart-data en personal data space-toepassingen.

De Vlaamse Smart Data Space zorgt zo voor een standaardisering van de data-integratie. Dat wil zeggen dat we ernaar streven om het maatwerk dat bij data-uitwisseling komt kijken, zo veel mogelijk te beperken. Zo kan elke partij in het ecosysteem zich op zijn kerntaken (data-kwaliteit, inzichten halen uit de data, nieuwe toepassingen, ...) focussen bij het uitwisselen van data. Op basis van goede afspraken en de principes van Linked Data kan eenzelfde databron ingezet worden voor meerdere use cases of meerdere doeleinden. Zo krijgt elke partij de best mogelijke inzichten uit de data.

## 1.2 OSLO

De Vlaamse overheid wil haar dienstverlening optimaliseren en de uitwisseling van informatie vlotter laten verlopen. Maar overheidsdiensten aan burgers en ondernemers worden ondersteund door gespecialiseerde toepassingen van verschillende softwareleveranciers. Daarom is er nood aan een eenduidige standaard: OSLO (Open Standaarden voor Linkende Organisaties). Het is de bedoeling om te zorgen voor meer samenhang, een betere begrijpbaarheid en een betere vindbaarheid van informatie en dienstverlening. Op die manier kan iedereen de gegevens makkelijker gebruiken.

Overheden op lokaal, Vlaams, interfederaal en Europees niveau moeten in het kader van hun dienstverlening vaak samenwerken. In praktijk moeten bijgevolg heel wat gegevens uitgewisseld worden tussen de verschillende administraties. Deze gegevens zijn afkomstig uit verschillende systemen, zijn mogelijk niet in hetzelfde technische formaat beschikbaar, en volgen niet noodzakelijk dezelfde semantiek. Zonder het maken van afspraken, wordt kwaliteitsvolle gegevensuitwisseling onmogelijk. Dit onderstreept het belang van de ontwikkeling van datastandaarden. De verschillende standaarden kunnen onderverdeeld worden op basis van waar ze zich bevinden in de levenscyclus, namelijk:

**Erkende standaarden:** werden na het doorlopen van een erkenningsprocedure goedgekeurd door de Werkgroep Datastandaarden en het Stuurorgaan Vlaams Informatie- en ICT-beleid als standaard binnen de Vlaamse overheid.



**Kandidaat standaarden:** een stabiele specificatie voor de standaard werd ontwikkeld, aan de hand van een publieke reviewperiode wordt implementatie-ervaring opgedaan en feedback van buiten de thematische werkgroep verzameld.

**Standaarden in ontwikkeling:** werden reeds aangemeld bij de Werkgroep Datastandaarden en worden uitgewerkt door een thematische werkgroep aan de hand van publieke werksessies.

**Technische standaarden:** de technische specificaties voor gegevensuitwisseling, ze bieden manieren aan van gestandaardiseerde gegevensuitwisseling aan.

**Vocabularium:** de basis voor open semantische informatiestandaarden, ze bieden een gedeeld begrippenkader voor bepaalde concepten met een focus op gegevensuitwisseling.

**Applicatieprofielen:** een specificatie voor gegevensuitwisseling voor applicaties die een bepaalde use case vervullen. Het laat toe om naast een gedeelde semantiek ook bijkomende restricties op te leggen, zoals het vastleggen van kardinaliteiten of het gebruik van bepaalde codelijsten.

**Implementatiemodel:** een specificatie voor gegevensuitwisseling voor applicaties die een bepaalde use case vervullen, deze modellen kunnen een deel van de interne gegevensverwerking van specifieke applicaties bevatten. Het laat toe om naast een gedeelde semantiek ook bijkomende restricties op te leggen, zoals het vastleggen van kardinaliteiten of het gebruik van bepaalde codelijsten. De ontwikkeling hiervan volgt dezelfde ideologie als applicatieprofielen en vocabularia, maar heeft nog geen officiële erkenning als proces en methode.

Op <https://data.vlaanderen.be/standaarden/> vind je het Standaarden register, waar je kunt zoeken naar het gepaste applicatieprofiel. Voor deze use case gebruiken we de OSLO standaard Verkeersmetingen. Het applicatieprofiel vind je hier : <https://data.vlaanderen.be/standaarden/standaard-in-ontwikkeling/applicatieprofiel-verkeersmetingen.html> en het vocabularium hier : <https://data.vlaanderen.be/standaarden/standaard-in-ontwikkeling/vocabularium-verkeersmetingen.html> . Daarnaast vind je alle presentaties en verslagen van de thematische werkgroep die de ontwikkeling van deze standaard op zich genomen heeft.

In de context van VSIDS is OSLO de functionele standaard aan dewelke de data inhoudelijk voldoet, zonder uitspraak te doen over de technische drager. Het OSLO model legt de vorm van de data vast, zodat alle data die aan dit model conformeert dezelfde structuur heeft, tot en met de properties van alle velden en eventuele mogelijke beperkingen in waarden (codelijsten). Er wordt hier verder ingegaan op de specifieke structuur van data voor doorsnedetellingen.

## 1.3 LDES

LDES staat voor Linked Data Event Stream. De LDES Server is een configureerbaar component die gebruikt kan worden om updates van data via het LDES-protocol te ontsluiten. Binnen de context van het Vlaamse Smart Data Space project zijn er open-source LDES componenten gebouwd die de uitwisseling van open data makkelijker maken. De technische documentatie over deze componenten is hier te vinden :

[https://informatievlaanderen.github.io/VSIDS-Tech-Docs/introduction/LDES\\_server](https://informatievlaanderen.github.io/VSIDS-Tech-Docs/introduction/LDES_server) . De servercomponenten zelf zijn hier te vinden : <https://hub.docker.com/r/ldes/ldes-server/tags> , en de code is publiek beschikbaar op GitHub : <https://github.com/informatievlaanderen/VSIDS-LDESServer4j>.

////////////////////////////////////

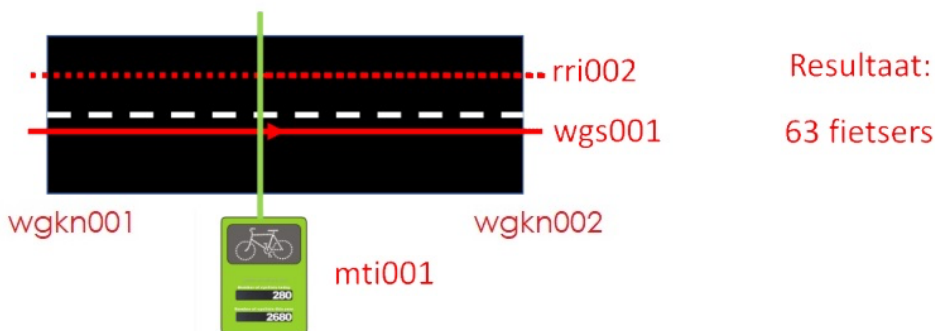
Daarnaast zijn er ook open-source componenten uitgewerkt voor de LDES client, die deze LDES stromen kan binnenhalen en converteren. High-level technische documentatie over de LDES client staat hier : [https://informatievlaanderen.github.io/VSDS-Tech-Docs/introduction/LDES\\_client](https://informatievlaanderen.github.io/VSDS-Tech-Docs/introduction/LDES_client) , en de componenten zelf staan hier : <https://github.com/Informatievlaanderen/VSDS-Linked-Data-Interactions>. LDES Client maakt deel uit van een grotere suite om met Linked Data aan de slag te gaan, namelijk Linked Data Interactions (LDI): <https://informatievlaanderen.github.io/VSDS-Linked-Data-Interactions/>. Er worden vanuit LDI twee frameworks voor data pipelining ondersteunt: Apache Nifi & Linked Data Interactions Orchestrator. LDES is de technische drager waarlangs de data opengesteld wordt. LDES bestaat uit componenten die specifiek ontworpen zijn om linked data op een efficiënte en duidelijke manier te ontsluiten, maar is generiek wat betreft de inhoud van de data, deze wordt in het OSLO model vastgelegd.

## 2 USE CASE VERKEERSMETINGEN – DOORSNEDE METING

### 2.1 WAT IS EEN DOORSNEDE METING?

Het domein van verkeersmetingen is een zeer breed domein. De use case doorsnede metingen definieert dit verder als een meting die aan de volgende voorwaarden voldoet :

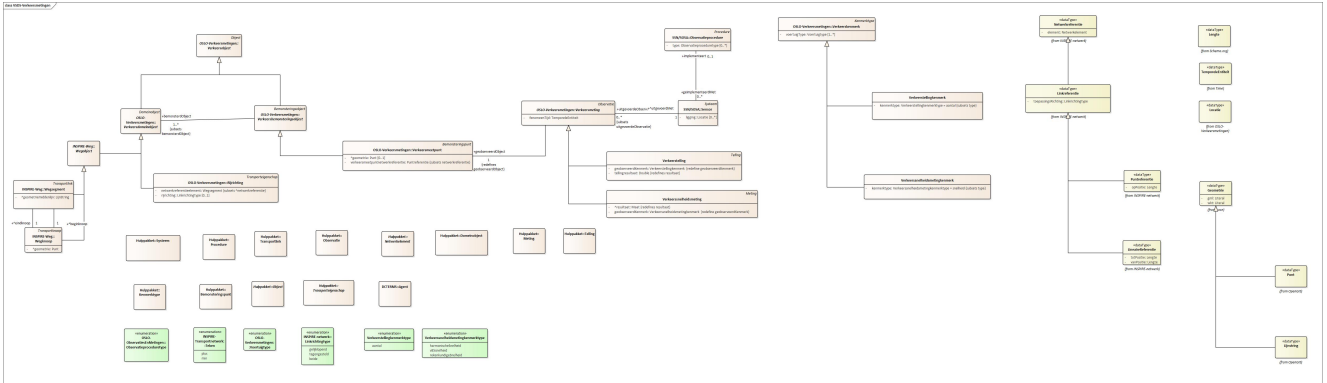
- Het betreft een meting op één punt in de ruimte, namelijk op een doorsnede van een wegvak. Herkomst-bestemmings metingen of kruispunt tellingen horen niet in deze usecase thuis.
- De meting slaat op een doorsnede van de weg, m.a.w. een optelsom van alle rijstroken (inclusief voetpaden en/of fietspaden). Rijstrook tellingen horen hier niet thuis.
- De meetwaarden zijn geaggregeerd per 15 minuten per vervoersmodus (zie verder voor codelijsten)
- De metingen slaan ofwel op het aantal verkeersdeelnemers (telling) ofwel op de gemeten snelheid (snelheidsmeting)



Dit kunnen zowel historische meetcampagnes zijn waarbij er data voor een bepaalde periode (in 15 minuten intervallen) aanwezig is, als live telcampagnes waarbij de data telkens aangevuld wordt.



## 2.2 IMPLEMENTATIEMODEL



Het implementatiemodel Doorsnedetellingen (<https://implementatie.data.test-vlaanderen.be/doc/implementatiemodel/verkeersmetingen/>) is een verdere specificatie van het OSLO applicatieprofiel verkeersmetingen (<https://data.vlaanderen.be/doc/applicatieprofiel/verkeersmetingen/>).

In het implementatiemodel wordt er niet gewerkt met een generieke klasse verkeersmeting (zoals in het applicatieprofiel verkeersmetingen), maar zijn twee klassen gedefinieerd die de klasse Verkeersmeting verder specificeren, namelijk Verkeersstelling en Verkeerssneldmeting. Het kenmerk van een Verkeersstelling is altijd van het type "Aantal", en het resultaat is een getal van het type double (decimaal getal).

Voor een verkeerssneldmeting kan er gekozen worden tussen 4 types (V85, mediaan, tijds gemiddelde snelheid en plaats gemiddelde snelheid) als kenmerk, en dient bij het resultaat de eenheid meegegeven te worden.

De overige onderdelen van het model zijn overgenomen van het applicatieprofiel verkeersmetingen.

De volgende hoofdstukken bevatten zowel de vereisten aan de inputdata om dit te kunnen modelleren, en een voorbeeld.

## 2.3 DATA VEREISTEN

### 2.3.1 Telgegevens

Er dient minimaal de volgende gegevens aanwezig te zijn :

- Locatie van de telling (zie verder voor specificaties)
- Tijdstip/datum van de telling
- Voertuigtype
- Getelde waarde : dit kan zowel snelheid als aantal zijn
- Type van de sensor

Optioneel kunnen de volgende zaken meegegeven worden :

- observatieprocedure





Elke meting die opgenomen wordt in de LDES stroom bevat slechts 1 waarde voor de meting. M.a.w., als op 1 locatie zowel aantallen als snelheid gemeten is, dan worden dat twee metingen, één voor het aantal voertuigen en 1 voor de snelheid.

Verder in het proces wordt besproken hoe de data OSLO compliant gemaakt dient te worden (met referentie naar het implementatiemodel).

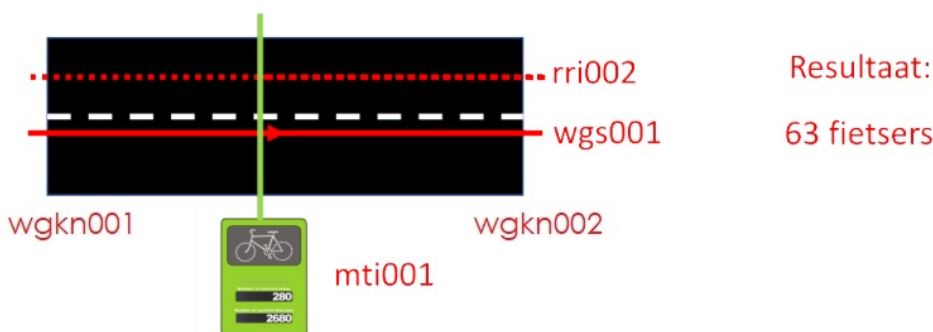
### 2.3.2 Locatie

Er is gekozen om locatiedata onafhankelijk van een digitale kaart op te slaan in het model. Daarom volgen we de OpenLR specificatie voor het specificeren van locatie (<http://www.openlr.org/>). OpenLR is door TomTom ontwikkeld om netwerkgerelateerde gegevens te kunnen mappen op een kaart zonder voorafgaandelijke kennis van de kaart waarop gemapt zal worden. OpenLR is als open source ontwikkeld (<https://github.com/tomtom-international/openlr>) onder de Apache license 2.0 (compatibel met GPL v3), wat toelaat om de software te gebruiken, te verspreiden en te modificeren zonder enige royalties. Op de Tomtom github pagina staan 9 componenten waaronder een decoder (die map agnostische data terug linkt aan een kaart).

Er worden geen OpenLR software componenten in dit proces gebruikt, alleen de specificaties voor locatie data worden gevolgd (hier te downloaden : [http://www.openlr.org/fileadmin/user\\_upload/openlr-whitepaper\\_v1.5.pdf](http://www.openlr.org/fileadmin/user_upload/openlr-whitepaper_v1.5.pdf)).

Dat wil zeggen dat twee zaken meegegeven worden : een wegsegment (minstens bestaande uit 1 beginpunt en 1 eindpunt met XY coördinaten in WGS84), en een offset in meter. Deze offset bepaalt waar het meetpunt zich precies op het wegsegment bevindt ten opzichte van de beginknoop.

We illustreren op basis van een figuur :



In dit voorbeeld is er een wegsegment gedefinieerd WGS001, dat een A knoop en een B knoop heeft, namelijk wgkn001 en wgkn002. Voor elke wegknoop wordt een X en Y coördinaat in WGS84 meegegeven. De volgorde van deze knoop bepaalt de digitaliseringsrichting. Dit wil zeggen dat elke andere richting hiernaar verwijst. Zo is een rijrichting (gelijklopend of tegengesteld) altijd ten opzichte van deze digitaliseringsrichting.

De locatie van de meting mti001 wordt dan bepaald door een offset mee te geven ten opzichte van de wegknoop A, in het voorbeeld dus ten opzichte van wgkn001. Deze offset wordt in meter gegeven.



## 2.3.3 Tijdstip

Voor doorsnedetellingen verwerken we data in 15 minuten intervallen. In het OSLO model wordt dit gecodeerd als een interval met een begin- en eindtijd (of begintijd en duratie), van het formaat XML schema DateTime. In de input data dient minimaal een datum en een begin uur meegegeven te worden (in 15 minuten intervallen). Indien een einduur van het interval meegegeven is kan dit overgenomen worden (dan zijn andere intervallen per meting technisch mogelijk, maar niet wenselijk), anders moet in de mapping van de data meegegeven worden wat het interval is voor alle tellingen.

## 2.3.4 Voertuigtype

Voor elke meting dient meegegeven te worden welk voertuigtype geteld is. Het OSLO model voorziet hiervoor een codelijst die de volgende mogelijkheden bevat :

- Auto : deze klasse omvat naast personenwagens ook lichte vracht (>3,5 ton) en motorfietsen
- Vrachtwagen : naast vrachtwagens wordt hier ook bussen meegenomen.
- Fiets : deze klasse omvat zowel klassieke fietsen als elektrische fietsen (inclusief speed pedelecs) en bromfietsen.
- Voetganger : naast voetgangers zijn hier ook rolstoelen en mobility scooters inbegrepen.

Indien deze informatie niet aanwezig is in de inputdata moet er aangenomen worden dat de gemeten voertuigen allemaal tot dezelfde klasse behoren, en wordt dit bij het mappen van de data naar het OSLO model als constante aangenomen.

## 2.3.5 Snelheidsmeting

Voor een snelheidsmeting kan gekozen worden tussen vier types gemeten snelheden. Het OSLO model voorziet hiervoor een codelijst die de volgende mogelijkheden bevat :

- V85 : de snelheid die 85% van het verkeer niet overschrijdt
- Mediaan snelheid : de snelheid die 50% van het verkeer niet overschrijdt
- Plaatsgemiddelde snelheid : harmonisch gemiddelde van de snelheid van alle voertuigen
- Tijdsgemiddelde snelheid : rekenkundig gemiddelde van de snelheid van alle voertuigen

Een meegegeven waarde voor een snelheidsmeting moet één van deze vier categoriën zijn. Ofwel wordt het type meegegeven in de data, ofwel wordt aangenomen dat alle data van dezelfde categorie is en wordt dit meegegeven bij het opmaken van de mapping van data naar het OSLO model. De voorkeur gaat naar de eerste mogelijkheid, omdat dit uiteraard toelaat om meerdere types snelheid mee te geven in de dataset.

In het OSLO model dient ook de eenheid (meestal km/u, soms m/s) meegegeven worden. Dit kan ofwel in de input meegegeven worden per meting, ofwel bij het opmaken van de mapping naar het OSLO model opgegeven worden. Let wel, dit kan alleen als alle snelheidsmetingen dezelfde eenheid gebruiken.

## 2.3.6 Telling

Het resultaat van een telling bevat het aantal voertuigen. Het resultaat mag als een decimaal getal opgegeven worden (het OSLO model verwacht het type "double"), wat bijvoorbeeld bij geïnterpoleerde waarden van toepassing kan zijn. Voor de use case doorsnedetellingen is dit de optelsom van alle voertuigen van het aangegeven voertuigtype in het betreffende kwartier.

////////////////////////////////////

### 2.3.7 Type Sensor

Het type sensor dient opgegeven te worden in het OSLO model. Ook hier voorziet het OSLO model een codelijst met de mogelijke waarden :

- ANPR Camera : Dit is een camera die kentekens herkent en registreert, meestal voor handhavingsdoeleinden of toegangscontrole.
- Glasvezel : sensor die glasvezel gebruikt
- Inductielus : sensor die voertuigen detecteert via magnetische veldveranderingen in ingegraven lussen
- Manuele telling : een manueel uitgevoerde telling
- Piezzo sensor : sensor die de verkeersdruk meet via piëzoelektrische sensoren bij overrijden door voertuigen
- Radar : sensor die radiogolven uitzendt en vervolgens de gereflecteerde golven detecteert om de positie, snelheid en richting van objecten te bepalen
- Rubberslang : sensor met één of meerdere rubberen slangen voor meten van verkeersvolume door luchtdrukverandering bij het overrijden door voertuigen
- Camera : een camera die beeldherkenning gebruikt om voertuigen te tellen zonder kentekenregistratie
- Telraam : telling uitgevoerd door een telraam toestel (<https://telraam.net/>), een crowd sourced oplossing

Indien het type sensor niet aanwezig is in de input data kan ook in de mapping naar het OSLO model het type sensor als constante opgegeven worden. Dit vereist echter dat alle tellingen met dezelfde type sensor uitgevoerd zijn.

### 2.3.8 Observatieprocedure

Optioneel kan de geïmplementeerde observatieprocedure meegegeven worden. Bij de diverse onboardings van doorsnedetellingen is dit niet relevant gebleken, er wordt dan ook niet verder op ingegaan. In het OSLO model vindt u de specificaties hier : <https://implementatie.data.test-vlaanderen.be/doc/implementatiemodel/verkeersmetingen/#Observatieprocedure>

### 2.3.9 Extra klassen

Naast de beschreven codelijsten is het mogelijk om meerdere waarden op te geven voor voertuigtype, type sensor en snelheidsmeting. Dit wordt gedaan door gescheiden door een array van waarden op te nemen (het zogenaamde “double-typing”) waarin verwezen wordt naar een tweede codelijst en een tweede waarde uit die codelijst. We illustreren met een voorbeeld :

Firma A heeft de mogelijkheid om met hun meetapparatuur het onderscheid te maken tussen rode en blauwe voertuigen, en wil dit detailniveau meegeven in de gepubliceerde open data. Zo hebben ze voor 1 interval 100 blauwe auto’s geteld en 50 rode auto’s.

In de resulterende dataset zijn dit twee aparte metingen in hetzelfde tijdssegment, namelijk één meting met als voertuigtype “[codelijst VSDS :auto codelijst Firma A:rode auto]” en als resultaat 50, en één meting met als voertuigtype “[codelijst VSDS:auto codelijst Firma A: blauwe auto]” en als resultaat 100.



De belangrijkste voorwaarde hiervoor is dat beide codelijsten toegankelijk zijn. De URL naar de codelijst van VSIDS is opgenomen in de standaard context die beschikbaar is bij het implementatiemodel (<https://implementatie.data.test-vlaanderen.be/doc/implementatiemodel/verkeersmetingen/ontwerpstandaard/2023-09-27/context/Verkeersmetingen-im.jsonld>), de URL van de gedetailleerde codelijst dient toegevoegd te worden aan de context van de data en gehost te worden bij de data owner.

Er is één nadeel aan deze methode, namelijk dat bij data met zeer veel detail het aantal metingen (en dus het aantal records) zeer snel stijgt. Het is dan ook aangeraden om de impact op storage en bandbreedte te monitoren.

## 2.4 DATA VOORBEELD

In dit hoofdstuk illustreren we de transformatie van input data in JSON formaat naar een OSLO compatibel JSON-LD formaat. We gebruiken data van de stad Brugge (beschikbaar als open data) om dit te illustreren, maar de geïllustreerde principes slaan op 90% van alle verkeersdata die beschikbaar is. Noot : JSON-LD is hier puur gekozen om illustratieve redenen, de LDES componenten schrijven de resulterende OSLO data direct weg naar een database. JSON-LD is de meest leesbare notatievorm van linked data, vandaar de keuze.

De input data valt uiteen in twee delen, namelijk een beschrijving van de locatie, en een beschrijving van de meting. In JSON formaat ziet de locatie er als volgt uit :

```
1. {
2.   "id": "05d50e16-2e4e-414a-921a-91a1ea11cb02",
3.   "name": "Site 38",
4.   "address": "Noorweegse Kaai",
5.   "geometry": {
6.     "type": "Point",
7.     "coordinates": [
8.       3.2421013712883,
9.       51.2257868862067
10.    ]
11.  },
12.  "country": "BE",
13.  "subdivision": "BE-BRU",
14.  "origins": [
15.    {
16.      "name": "A",
17.      "geometry": {
18.        "type": "Point",
19.        "coordinates": [
20.          3.24239104986191,
21.          51.2259355430829
22.        ]
23.      }
24.    },
25.    {
26.      "name": "B",
27.      "geometry": {
28.        "type": "Point",
29.        "coordinates": [
30.          3.24177950620651,
31.          51.2256331896095
32.        ]

```



```

33.     }
34.   }
35. ]
36. }

```

De eerste drie regels bevatten een ID, de naam van de site en het adres. Daar is er een blok die de XY coördinaten bevat (regel 5 t.e.m. 11).

Daarna komen er twee regels met administratieve locaties, en vervolgens het blok dat het wegsegment beschrijft waarop de telling gebeurt is (regels 14 t.e.m. 35). In dat blok is er een beginpunt beschreven ("A"), met XY coördinaat, en een eindpunt ("B"), ook met een XY coördinaat.

Na de omzetting naar een OSLO compatibel JSON-LD (linked data) ziet het resultaat (1 gecombineerd bestand met zowel locatie als telling) er als volgt uit :

```

1. {
2.   "@context": [
3.     "https://implementatie.data.vlaanderen.be/doc/implementatiemodel/verkeersmetingen/ontwerpstandaard/2023-09-27/context/Verkeersmetingen-im.jsonld",
4.     "https://data.vlaanderen.be/doc/applicatieprofiel/sensoren-en-bemonstering/kandidaatstandaard/2022-04-28/context/ap-sensoren-en-bemonstering.jsonld",
5.     "https://data.vlaanderen.be/doc/applicatieprofiel/observaties-en-metingen/kandidaatstandaard/2022-04-28/context/ap-observaties-en-metingen.jsonld",
6.     "https://raw.githubusercontent.com/samuvack/context/main/wegenregister.jsonld",
7.     "https://data.vlaanderen.be/doc/applicatieprofiel/GEODCAT-AP-VL/erkendstandaard/2022-04-21/context/geodcatap-vlaanderen.jsonld",
8.     "https://data.vlaanderen.be/doc/applicatieprofiel/generiek-basis/zonderstatus/2019-07-01/context/generiek-basis.jsonld",
9.     {
10.      "schema": "http://schema.org/",
11.      "dct": "http://purl.org/dc/terms/",
12.      "xsd": "http://www.w3.org/2001/XMLSchema#",
13.      "geosparql": "http://www.opengis.net/ont/geosparql#",
14.      "qudt-unit": "http://qudt.org/vocab/unit/",
15.      "qudt-schema": "https://qudt.org/schema/qudt/",
16.      "dcterms": "http://purl.org/dc/terms/",
17.      "time": "http://www.w3.org/2006/time#",
18.      "adms": "http://www.w3.org/ns/adms#",
19.      "ucum": "https://w3id.org/cdt/",
20.      "Verkeersmeting.resultaat": {
21.        "@type": "http://www.w3.org/2001/XMLSchema#double",
22.        "@id": "http://def.isotc211.org/iso19156/2011/Observation#OM_Observation.result"
23.      },
24.      "cl-vrt": "https://data.vlaanderen.be/doc/concept/VkmVoertuigType/",
25.      "cl-vkt": "https://data.vlaanderen.be/doc/concept/VkmVerkeersKenmerkType/",
26.      "cl-trt": "https://inspire.ec.europa.eu/codelist/LinkDirectionValue/",
27.      "cl-mit": "https://data.vlaanderen.be/doc/concept/VkmMeetInstrumentType/",
28.      "cl-op": "https://data.vlaanderen.be/doc/concept/VkmObservatieProcedure/",
29.      "cl-access": "http://publications.europa.eu/resource/authority/access-right/"
30.    }
31.   ],
32.   "@graph": [
33.     {
34.       "@id": "_:GM001",
35.       "@type": "Dataset",
36.       "Dataset.titel": {
37.         "@language": "nl",
38.         "@value": "GeoMobility."
39.       },

```



```

40.     "Dataset.beschrijving": [
41.         {
42.             "@language": "nl",
43.             "@value": "Floating Car Data geeft een zeer goed beeld van de verkeerssituatie in een stad,
maar soms is het essentieel om exact te weten hoeveel verkeer er in bothDirections richtingen door een
straat rijdt en wat de modal split is."
44.         }
45.     ],
46.     "Dataset.toegankelijkheid": "cl-access:PUBLIC",
47.     "Dataset.trefwoord": [
48.         {
49.             "@language": "nl",
50.             "@value": "GeoMobility"
51.         }
52.     ]
53. },
54. {
55.     "@id": "_:rri001",
56.     "@type": "Rijrichting",
57.     "Rijrichting.netwerkreferentieelement": {
58.         "@type": "Linkreferentie",
59.         "Linkreferentie.toepassingsRichting": "cl-trt:inDirection",
60.         "Netwerkreferentie.element": "_:wgs001"
61.     },
62.     "Rijrichting.rijrichting": "cl-trt:inDirection"
63. },
64. {
65.     "@id": "_:wgs001",
66.     "@type": "Wegsegment",
67.     "Wegsegment.beginknoop": "_:wgkn001",
68.     "Wegsegment.eindknoop": "_:wgkn002",
69.     "Wegsegment.geometriemiddenlijn": {
70.         "@type": "LineString",
71.         "Geometrie.wkt": {
72.             "@value": "<http://www.opengis.net/def/crs/EPSSG/0/4326> LINESTRING (3.24239104986191
51.2259355430829, 3.24177950620651, 51.2256331896095, 3.24177950620651 51.2256331896095)",
73.             "@type": "geosparql:wktLiteral"
74.         }
75.     }
76. },
77. {
78.     "@id": "_:wgkn001",
79.     "@type": "Wegknoop",
80.     "Wegknoop.geometrie": {
81.         "@type": "Punt",
82.         "Geometrie.gml": {
83.             "@value": "<gml:Point
srsName=\"http://www.opengis.net/def/crs/EPSSG/0/4326\"><gml:coordinates> 3.24239104986191,
51.2259355430829</gml:coordinates><gml:Point>",
84.             "@type": "geosparql:gmlLiteral"
85.         }
86.     }
87. },
88. {
89.     "@id": "_:wgkn002",
90.     "@type": "Wegknoop",
91.     "Wegknoop.geometrie": {
92.         "@type": "Punt",
93.         "Geometrie.gml": {

```



```

94.         "@value": "<gml:Point
srsName=\"http://www.opengis.net/def/crs/EPSG/0/4326\"><gml:coordinates>3.24177950620651,
51.2256331896095</gml:coordinates><gml:Point>",
95.         "@type": "geosparql:gmlLiteral"
96.     }
97. }
98. },
99. {
100.     "@id": "_:vkmauto001",
101.     "@type": "Verkeerstelling",
102.     "Verkeerstelling.geobserveerdKenmerk": {
103.         "@type": "Verkeerstellingkenmerk",
104.         "Verkeerstellingkenmerk.kenmerktype": "cl-vkt:aantal",
105.         "Verkeerskenmerk.voertuigType": "cl-vrt:fiets"
106.     },
107.     "Verkeersmeting.geobserveerdObject": "_:mpt001",
108.     "Verkeersmeting.fenomeenTijd": "_:fenomtime001",
109.     "Verkeerstelling.tellingresultaat": 5,
110.     "Verkeersmeting.uitgevoerdMet": "_:mti001",
111.     "dct:memberOf": "_:GM001"
112. },
113. {
114.     "@id": "_:fenomtime001",
115.     "Verkeersmeting.fenomeenTijd": {
116.         "@type": "time:ProperInterval",
117.         "time:hasBeginning": {
118.             "@type": "time:Instant",
119.             "time:inXSDDateTime": {
120.                 "@type": "xml-schema:dateTime",
121.                 "@value": "20161122T09:00:00.000Z"
122.             }
123.         },
124.         "time:hasEnd": {
125.             "@type": "time:Instant",
126.             "time:inXSDDateTime": {
127.                 "@type": "xml-schema:dateTime",
128.                 "@value": "20161122T10:00:00.000Z"
129.             }
130.         }
131.     }
132. },
133. {
134.     "@id": "_:mpt001",
135.     "@type": "Verkeersmeetpunt",
136.     "Verkeersmeetpunt.geometrie": {
137.         "@type": "Punt",
138.         "Geometrie.gml": {
139.             "@value": "<gml:Point
srsName=\"http://www.opengis.net/def/crs/EPSG/0/4326\"><gml:coordinates>3.24177950620651,
51.2256331896095</gml:coordinates><gml:Point>",
140.             "@type": "geosparql:gmlLiteral"
141.         }
142.     },
143.     "Verkeersmeetpunt.verkeersmeetpuntnetwerkreferentie": "_:pr001",
144.     "Verkeersbemonsteringsobject.bemonsterdObject": "_:rri001"
145. },
146. {
147.     "@id": "_:pr001",
148.     "@type": "Puntreferentie",
149.     "Puntreferentie.opPositie": {

```



```

150.         "@type": "Lengte",
151.         "KwantitatieveWaarde.waarde": "300",
152.         "KwantitatieveWaarde.standaardEenheid": {
153.             "@value": "m",
154.             "@type": "ucum:ucumunit"
155.         }
156.     },
157.     "Linkreferentie.toepassingsRichting": "cl-trt:bothDirections"
158. },
159. {
160.     "@id": "._:mti001",
161.     "@type": "Sensor",
162.     "Systeem.type": "cl-mit:rubberslang"
163. }
164. ]
165. }
166.

```

Regel 1 t.e.m. 31 bevat de context, m.a.w. de labels die in de text vervangen worden door persistente URI's op machineniveau. Regel 34 t.e.m. 54 bevat de algemene beschrijving van de dataset.

Van regel 66 t.e.m. regel 74 wordt het wegsegment gedefinieerd. Er wordt een object met de naam "wgs001" aangemaakt, dat van het type "wegsegment" is en een beginknoop ("wgkn001") en eindknoop ("wgkn002") heeft. Dan wordt de geometrie beschreven als een linestring, met drie vormpunten, namelijk het beginpunt van het wegsegment, het meetpunt en het eindpunt van het wegsegment. Op deze manier bevat de data steeds een wegsegment waar het meetpunt op ligt. Indien in de inputdata meer vormpunten meegegeven worden voor het wegsegment kunnen deze in de linestring meegegeven worden voor een nauwkeuriger vorm.

Daarna worden de twee wegknoopen beschreven, met als type "wegknoop" en een geometrie als punt volgens de GML syntax. Tijdens de transformatie van JSON naar JSON-LD wordt de GeoJSON notatie omgezet naar GML om OSLO compatibel te zijn.

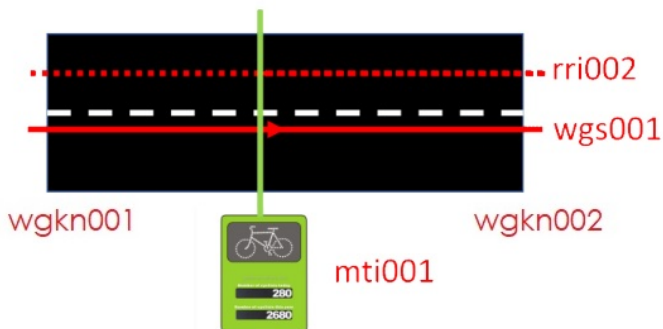
Er staan nog twee objecten in die voor de locatie van belang zijn, namelijk de rijrichting en een puntreferentie. De puntreferentie bevat als waarde de offset van het meetpunt t.o.v. het startpunt van het wegsegment (uitgedrukt in meter, hier 300 meter), met specificatie van de gebruikte eenheid ("m").

De rijrichting is een object ("rri001") dat verwijst naar het wegsegment wgs001, en dat als toepassingsrichting "inDirection" heeft, ofwel van A knoop naar B knoop. Dit object wordt gebruikt als referentie bij het definiëren van de meting op deze plek, dit wordt zodadelijk toegelicht.

Deze figuur (zie ook eerder) illustreert dit :







De input data voor de meting ziet er als volgt uit :

```

1. {
2.   "id": "5ff87dcd9dc02be13c259e28",
3.   "type": "atc",
4.   "origin": "A",
5.   "destination": "B",
6.   "classification": "fiets",
7.   "timestamp": "2016-11-22T09:00:00.000Z",
8.   "count": 5,
9.   "poiId": "05d50e16-2e4e-414a-921a-91a1ea11cb02",
10.  "surveyId": "b18222b9-7cd9-4c3c-adbe-41c0c597ee7d"
11. }
12.

```

Ook hier begint de data met een ID, een veld “type” wat slaat op gebruikte sensor, een referentie naar beginpunt en eindpunt van het gemeten wegvak (“origin” en “destination”), een voertuigtype (fiets), een tijdsaanduiding (datum en tijd), het meetresultaat, en een ID voor de locatie en een ID voor de telcampagne. Als we kijken naar het resulterend OSLO-compliant JSON-LD formaat (hierboven) zien we dit. Op regel 100 wordt een object met de ID “vkmauto001” gedefinieerd. Dit krijgt als type “verkeerstelling”, van het type “aantal” (m.a.w. een telling) van voertuigen van het type “fiets” (regel 105).

In regel 107 wordt gesteld dat de meting op het hierboven gedefinieerde meetpunt gebeurt (in semantische termen : het geobserveerde object van de meting is het meetpunt). Regel 108 stelt dat het object “fenomtime001” het tijdsvak bevat (zie verder), regel 109 bevat het telresultaat, en regel 110 stelt dat de meting uitgevoerd is met object mti001, wat verder gedefinieerd wordt. Tenslotte wordt de telling beschouwd als lid van dataset GM001.

Vanaf regel 115 wordt het tijdsvak gedefinieerd (object “fenomtime001”), als een interval met een begin (vanaf regel 117) en een einde (vanaf regel 124).

Vanaf regel 160 wordt het meetinstrument gedefinieerd (object “mti001”) als een sensor van het type “rubberslang”.

Dit voorbeeld toont aan hoe data transformeert van een niet-linked data bestand in JSON naar linked data volgens de OSLO standaard in JSON-LD formaat. Dit is ter illustratie, de verder beschreven IT componenten vertalen deze data onmiddellijk door naar database objecten die aan de LDES server aangeboden worden.



## 2.5 ARCHITECTUUR

Digitaal Vlaanderen voorziet verschillende bouwblokken om verkeersdata via VSDS als open data te ontsluiten. Deze bouwblokken doen kort samengevat twee zaken : data converteren naar het betreffende OSLO model (in dit geval het implementatiemodel Doorsnedetellingen), en data ontsluiten via een LDES server naar de buitenwereld.

De technische documentatie over deze componenten is hier te vinden :

[https://informatievlaanderen.github.io/VSDS-Tech-Docs/introduction/LDES\\_server](https://informatievlaanderen.github.io/VSDS-Tech-Docs/introduction/LDES_server) . De servercomponenten zelf zijn hier te vinden : <https://hub.docker.com/r/ldes/ldes-server/tags> , en de code is publiek beschikbaar op GitHub : <https://github.com/informatievlaanderen/VSDS-LDESServer4j>.

Er zijn twee mogelijke architecturen die gebruikt kunnen worden, namelijk een data stroom die via APACHE NIFI componenten verwerkt wordt en aan de server aangeboden wordt, of een datastroom die de LDIO (Linked Data Interaction Orchestrator) componenten gebruikt hiervoor. Apache NiFi (<https://nifi.apache.org>) is een open source software pakket dat een zeer uitgebreide feature set heeft voor allerhande data verwerking en visualisatie doeleinden. LDIO (<https://informatievlaanderen.github.io/VSDS-Linked-Data-Interactions/>) zijn Java componenten die specifiek voor de Vlaamse Smart Data Space (VSDS) ontwikkeld zijn.

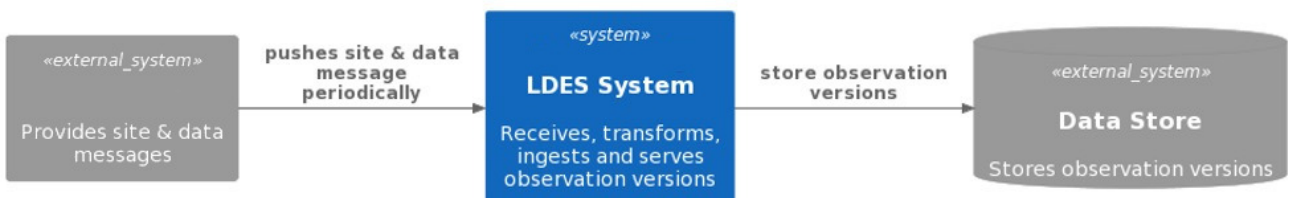
Omwille van de complexiteit van Apache NiFi wordt hier gekozen om met LDIO componenten te werken. Bij zeer grote en complexe datasets is Apache NiFi eventueel een betere keuze omwille van robuustheid en features, maar voor 90% van de toepassingen is LDIO de betere keuze, vandaar dat de beschrijving hier verder op gebaseerd is.

De componenten bestaan uit 5 categorieën :

- Input
- Core
- Output
- Transformatoren
- Adapters

De input componenten lezen data ofwel rechtstreeks uit een database, ofwel via een API (push or pull). De adapter componenten bevatten de functionaliteit om van reguliere data (vb JSON of NGSI) naar linked data gaan. De transformatoren bevatten functionaliteit om de data inhoudelijk te bewerken, zoals bijvoorbeeld een functie om GeoJSON naar WKT om te zetten, of een query om linked data naar het OSLO model te mappen.

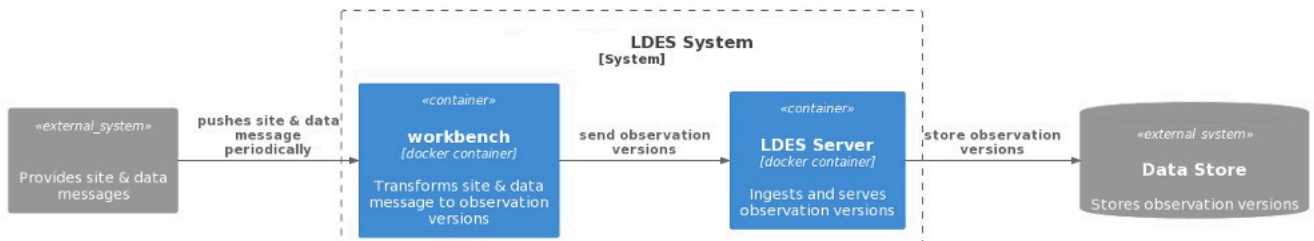
Als we even kijken naar een praktijkvoorbeeld, dan is dit een samenvatting hoe de implementatie voor open data van de stad Brugge er uitziet



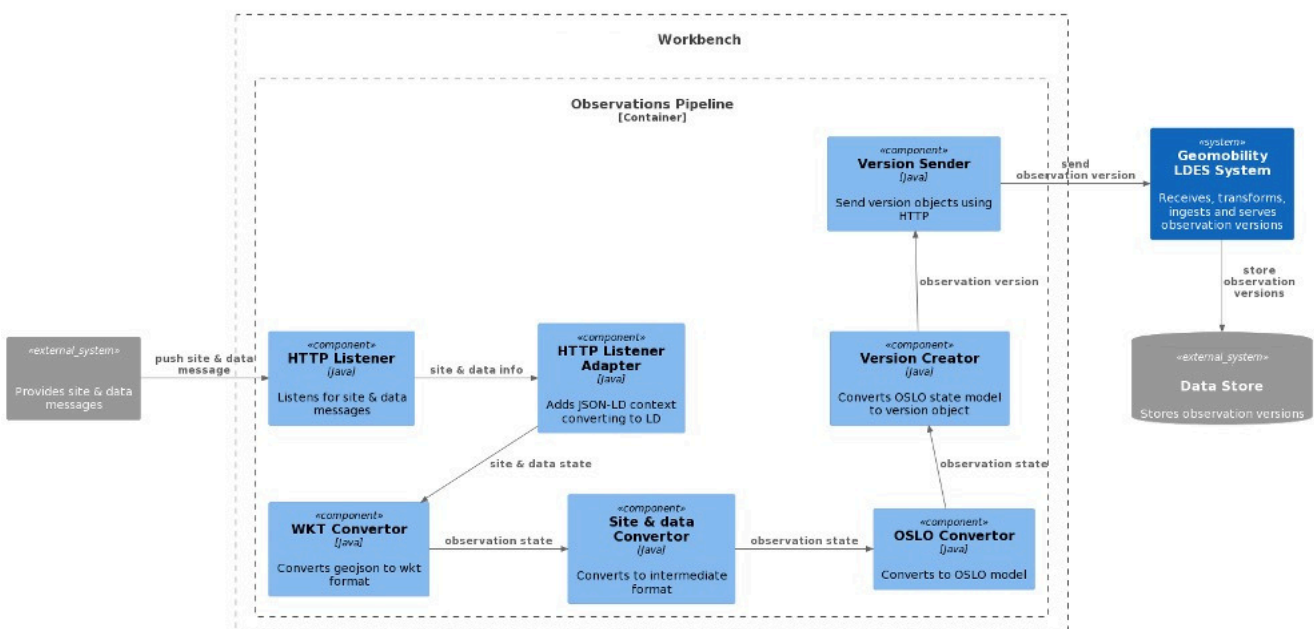
////////////////////////////////////

Vanuit het Backend systeem worden files gepusht die (in dit geval in JSON formaat) de locaties en de tellingen bevatten. Het LDES systeem neemt deze data op, verwerkt ze, en biedt ze aan aan de buitenwereld.

Als we dit functioneel verder opsplitsen komen we tot de volgende figuur, waarbij het LDES systeem in 2 grote delen uiteenvalt : de workbench, waar de data input en conversie gebeurt, en de LDES server die de data aan de buitenwereld aanbiedt.



Als we ook de workbench verder uitsplitsen voor geomobility komen we tot de volgende componenten :



- HTTP listener : In het voorbeeld wordt er data gepusht vanuit het backend systeem, de HTTP listener pikt deze data op en geeft ze door aan de volgende component.  
<https://informatievlaanderen.github.io/VSDS-Linked-Data-Interactions/Idio/Idio-inputs/Idio-http-in>
- HTTP listener adapter : converteert JSON naar JSON-LD door een context toe te voegen.  
<https://informatievlaanderen.github.io/VSDS-Linked-Data-Interactions/Idio/Idio-adapters/Idio-json-to-json-ld>
- WKT convertor : deze zet de coördinaten in geojson formaat over naar Well known text (WKT)  
<https://informatievlaanderen.github.io/VSDS-Linked-Data-Interactions/Idio/Idio-transformers/Idio-geojson-to-wkt>





### 2.6.1 Voorbereiding

In de voorbereidingsfase wordt nagekeken of de aanwezige data alle informatie bevat die noodzakelijk is (zoals hierboven beschreven). Indien deze data niet aanwezig is moet nagegaan worden of de data verrijkt kan worden (eenmalig in het geval van een historische dataset, permanent in het geval van een lopende telling). Een typisch voorbeeld hiervan is de beschrijving van een locatie, waarbij het model een wegsegment met offset verwacht. Als die data ontbreekt (bijvoorbeeld doordat enkel een XY coördinaat in de data aanwezig is) moet nagegaan worden of de data kan toegevoegd worden binnen het bestaande backend systeem (bijvoorbeeld door data permanent te importeren in de bron database), ofwel kan er een preprocessing script uitgewerkt worden die data ophaalt en combineert voor ze aan te bieden aan de LDIO componenten (bijvoorbeeld met een python script die wegsegmenten uit het GRB haalt).

Ten tweede moet er beslist worden over de hosting van de LDIO componenten. Deze draaien in een docker container die op een server gehost kan worden. Afhankelijk van de backend architectuur kan gekozen worden om deze docker containers op een server te draaien naast de bron database (makkelijker voor datatoegang), op een externe server (met API toegang naar de data) of in de cloud (AWS, Azure...). Afhankelijk van de grootte van de dataset (aantal records) dient storage grootte bepaald te worden. Voor doorsnedetellingen kan momenteel gerekend worden op 2,5KB per record als vuistregel.

### 2.6.2 OSLO mapping

Ten tweede wordt een mapping uitgewerkt van de huidige data naar het OSLO model. Waar het OSLO model een codelijst gebruikt (vb bij voertuigcategorie, type snelheidsmeting en type sensor) dient een tabel gemaakt te worden die de waarden in de brondata mapt op de gewenste waarde in het OSLO model. Een aantal verwachte waarden in het OSLO model zullen simpelweg niet aanwezig zijn in de brondata omdat ze een constante zijn. Zo komt het regelmatig voor dat alle metingen in een dataset met hetzelfde type sensor uitgevoerd is, en dat dit type niet expliciet in de data staat. Eenzelfde redenering gaat dikwijls op voor rijrichting : meestal zijn alle tellingen van een bepaald type in dezelfde richting uitgevoerd. Ook dit dient gedocumenteerd te worden.

### 2.6.3 Configuratie LDIO

Vervolgens kunnen de LDIO componenten geconfigureerd worden. Dit houdt de volgende zaken in :

- Initialisatie van docker containers via YML. In totaal worden 4 containers geïnitieerd : LDIO workbench, LDES Server, MongoDB, NGINX (reverse proxy). De laatste twee componenten kunnen uiteraard ingewisseld worden voor een database en reverse proxy naar keuze.
- Configuratie van de LDIO componenten. Dit gebeurt door het aanpassen van een YML configuratie file. Voor sommige componenten (transformatie naar linked data en transformatie naar OSLO) dient er een SPARQL query (.RQ formaat) aangemaakt worden die in de configuratie wordt opgenomen. De sparql construct query bouwt de data op volgens de uitgeschreven query en schrijft ze weg als linked data.

### 2.6.4 Deployment

Na configuratie (en testing op lokale docker containers) kan overgegaan worden tot deployment. Dit houdt in dat de nodige docker containers op de correcte locaties opgespind worden met correcte configuratiefiles. Vervolgens kan de data gepusht worden naar de configuratie indien van toepassing.



## 2.6.5 Metadata deployment

Om data vindbaar te maken is het noodzakelijk dat de metadata (dcat) van de dataset aan de hand van linked data naar de server gestuurd wordt, in stukjes relevant voor de LDES stroom in kwestie en de gedefinieerde views. Dit wordt gedaan door een TTL file aan te passen en door te sturen (seeding). De server lijmt deze stukjes aan mekaar en maakt er een DCAT catalog van, die opgepikt kan worden door een metadata crawler (zoals bijvoorbeeld datavindplaats).

## 2.7 REFERENTIES

<https://data.vlaanderen.be/standaarden/> : standaarden register

<https://data.vlaanderen.be/standaarden/standaard-in-ontwikkeling/applicatieprofiel-verkeersmetingen.html> : Applicatieprofiel Verkeersmetingen

<https://data.vlaanderen.be/standaarden/standaard-in-ontwikkeling/vocabularium-verkeersmetingen.html> : vocabularium verkeersmetingen

[https://informatievlaanderen.github.io/VSDS-Tech-Docs/introduction/LDES\\_server](https://informatievlaanderen.github.io/VSDS-Tech-Docs/introduction/LDES_server) : LDES Server documentatie

<https://hub.docker.com/r/ldes/ldes-server/tags> : LDES Server docker images

[https://informatievlaanderen.github.io/VSDS-Tech-Docs/introduction/LDES\\_client](https://informatievlaanderen.github.io/VSDS-Tech-Docs/introduction/LDES_client) : LDES client technische documentatie

<https://github.com/Informatievlaanderen/VSDS-Linked-Data-Interactions> : LDIO componenten repository

<https://implementatie.data.test-vlaanderen.be/doc/implementatiemodel/verkeersmetingen/> : implementatiemodel doorsnedetellingen

<http://www.openlr.org> : openLR organisation

<https://github.com/tomtom-international/openlr> : Open LR Repository

[http://www.openlr.org/fileadmin/user\\_upload/openlr-whitepaper\\_v1.5.pdf](http://www.openlr.org/fileadmin/user_upload/openlr-whitepaper_v1.5.pdf) : openLR specifications

<https://implementatie.data.test-vlaanderen.be/doc/implementatiemodel/verkeersmetingen/ontwerpstandaard/2023-09-27/context/Verkeersmetingen-im.jsonld> : Doorsnedetellingen JSONLD context

<https://nifi.apache.org> : apache NiFi

<https://informatievlaanderen.github.io/VSDS-Linked-Data-Interactions/> : LDIO project

[https://informatievlaanderen.github.io/VSDS-Tech-Docs/quickstart/Publish\\_LDES](https://informatievlaanderen.github.io/VSDS-Tech-Docs/quickstart/Publish_LDES) : quick guide LDES client

