Usecase Verblijfsobjecten

Doel van de usecase

Met deze usecase wordt onderzocht of het mogelijk is om met datavirtualisatie, tabellen vanuit verschillende geografische bronnen in te lezen en het resultaat vervolgen met elkaar te correleren.

Huidige situatie

Momenteel worden niet alle beschikbare databronnen bevraagd. Daarnaast worden de ontvangen gegevens niet met elkaar gecorreleerd.

Situatie van de usecase

Vanuit Apache Drill wordt rechtstreeks een koppeling gelegd met drie databronnen:

ArcGIS PR database

> Alle gegevens in bezit van de gemeente Sittard-Geleen om collega's te voorzien van GEO informatie. Deze databron is geen landelijke databron maar bevat data welk is afgestemd op de behoefte van de gemeente Sittard-Geleen. Denk hierbij aan informatie uit de basis registratie, de kadastrale registratie en objecten registratie.

Webservice PDOK (Publieke Dienstverlening op de Kaart)

> PDOK toont geografische data uit basisregistraties en publieke data, deze worden geografisch getoond in een landelijke voorziening op basis van een GEO-service Web Feature Service (WFS)

DSO

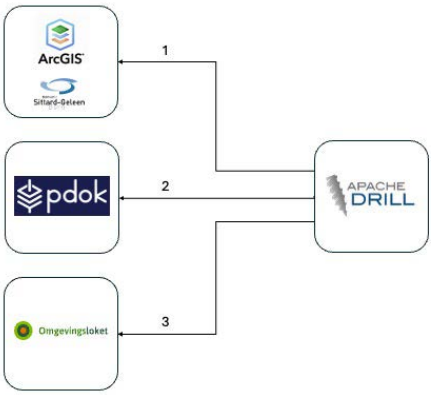
> Digitaal Stelsel Omgevingswet (DSO) is een landelijke voorziening voor de leefomgeving en bevat o.a. het bestemmingsplan van de gemeente Sittard-Geleen. De verbinding met het DSO wordt via het omgevingsloket gemaakt middels een API-call.

Na deze technische realisatie wordt de koppeling gemaakt met het Gemeentelijk GegevensModel (GGM). In een eerdere fase van het project Datavirtualisatie is door een externe partij (Inin) een inventarisatie en een werkwijze gemaakt waarbij dataelementen gekoppeld kunnen worden aan het Gemeentelijk Gegevens Model (GGM).

Praktisch

In onderstaande figuren is de usecase weergegeven.

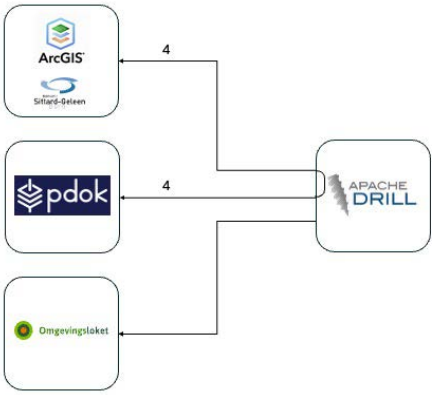
Eerste fase



Figuur 1

In de eerste fase van de usecase zijn de databronnen individueel bevraagd vanuit ApacheDrill (zie figuur 1). Bij query 1 en 2 wordt BAG-data opgehaald van panden en verblijfsobjecten. Bij query 3 worden de gegevens uit het bestemmingsplan van de gemeente Sittard-Geleen opgehaald. Het resultaat van deze drie queries is in bijlage 1 weergegeven.

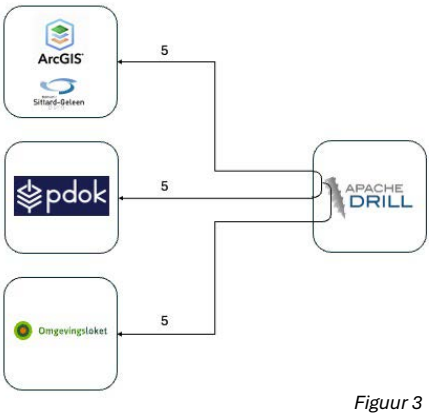
Tweede fase



Figuur 2

In de tweede fase van de usecase (zie figuur 2) worden uit de ArcGIS database en uit de PDOK webservice de BAG-data opgehaald m.b.t. de eerder geselecteerde panden uit de eerste fase en wordt er een join uitgevoerd (query 4) op de BAG-data.

Derde fase



Figuur 3

In de derde fase van deze usecase (zie figuur 3) wordt het resultaat van de tweede fase gecorreleerd aan de gegevens in het DSO. Deze correlatie (query 5) vindt plaats via een zgn. geografische join.

Vierde fase

In de vierde fase van deze usecase worden de verkregen resultaten gekoppeld aan het Gemeentelijk Gegevens Model (GGM). Met het GGM worden gemeentes in staat gesteld om op een uniforme wijze data te categoriseren, waardoor er zonder translatie 1-op-1 dat uitgewisseld of vergeleken kan worden. Vanuit de VNG is er voor een groot aantal onderwerpen een datastructuur beschikbaar die als gemeenschappelijk model gebruikt wordt. Deze gemeenschappelijke datastructuur is ook voor GEO beschikbaar.

Geslaagde usecase

De usecase verblijfsobjecten is geslaagd wanneer:

- Aangetoond wordt dat het mogelijk is om vanuit Apache Drill data uit verschillende
  openbare registers te halen;
- Deze data in een lijst gepresenteerd kan worden en geanalyseerd kan worden;
- De opgehaalde dataelementen gekoppeld zijn aan het Gemeentelijk GegevensModel
  (GGM);
- Het met een één query mogelijk is om uit verschillende databronnen data op te halen en
  te correleren;
- Er een acceptabele performance is.
  (Hierbij is nog niet gedefinieerd welke waarde acceptabel is).

Belangrijk om hierbij op te merken is dat de usecase een afgebakende scope heeft waarbij het mogelijk is dat de getoonde informatie in de verschillende overzichten en queries nog niet aansluit bij de behoefte vanuit het werkproces.

Resultaat

In bijlage 2 is de complete technische aanpak van de engineer weergegeven, evenals zijn technische conclusie.

Resultaat eerste fase

Het resultaat van de eerst fase van deze usecase is dat het 1-op-1 ophalen van gegevens uit externe- en interne bronnen goed gaat. In een beginstadium is samen met een engineer van het Kadaster de koppeling met het Omgevingsloket (DSO) gerealiseerd en is er gekeken naar mogelijke performance issues. Mogelijke performance belasting is in de 1-op-1 koppeling aan de bron bijna niet waar te nemen.

Resultaat tweede fase

Technisch gezien werkte deze bevraging, maar door de logische opbouw van de query werd er een zeer grote hoeveelheid data ontvangen waardoor het moeilijk werd deze te verwerken. Daarnaast gaven performance issues aan de presentatie kant dat deze tweede fase niet kon worden voortgezet.

Resultaat derde fase

Omdat de uitvoering van de derde fase voortbouwt op het resultaat van de tweede fase, waarbij de uitvoering van de tweede fases afgebroken is, is er geen resultaat in de derde fase te melden.

Resultaat vierde fase

De data die opgehaald is in de eerste fase van deze usecase is 1-op-1 te matchen aan de datastructuur van het GGM. Het is hier wel belangrijk om bij het definiëren van de queries in Apache Drill de veldnamen van de databronnen in de queries al te labelen naar de uniforme namen uit het GGM. Met deze werkwijze wordt verkregen dat in het gehele proces direct met de GGM namen gewerkt wordt. Dit verhoogt direct de herkennaarheid en voorkomt re-work.

Algehele conclusie

In algemene zin kan worden geconcludeerd dat het gebruik van datavirtualisatie bij 1-op-1 koppeling goed werkt. Wanneer er vervolgens meerdere verschillende bronnen worden gecombineerd en hier vanuit geografische vlakken naar de samenhang gekeken wordt, de hoeveelheid ontvangen data en de daaruit volgende performance issue maken dat datavirtualisatie in deze casus geen werkbare situatie opleverde. Omdat hierbij op technisch vlak op voorhand geen issues zijn opgetreden is het advies om nog te onderzoeken of o.a. query optimalisatie en een andere opbouw van de infrastructuur een beter resultaat geeft.

Bijlage 1

BAG-data (pand informatie)

PDOK-data (verblijfsobjecten informatie)

DSO-data (bestemmingsvlakken)

Bijlage 2 – Technische aanpak

Begin van de query
Om erachter te komen of de query voor ST_Intersects goed werkt heb ik wat testen uitgevoerd om te achterhalen of de synthax van de query goed is. Gauw bleek al dat de synthax in orde was, maar dat Apache Drill wat anders verwachtte dan er als data in ging. Gedurende het onderzoek, met de goede hulp van ChatGPT, dat het mogelijk zat in het wegschrijven van de data in onze Oracle-database.

WKT geometry
Onze data voor geo leggen wij, vooralsnog, vast in een Oracle Spatial database. Vanuit deze centrale database kunnen wij de data via services of rechtstreeks aanbieden bij onze collega's. Oracle legt geometrieën vast als SDO_GEOMETRY. Voor alles wat wij gebruiken is dit meer dan prima en lopen we nergens tegen problemen aan. Behalve dan in dit geval in Apache Drill. Drill verwacht de data namelijk in een WKT (Well-Known Text) format en is dus heel anders dan hoe Oracle dit opslaat. Hierdoor zal Drill dus nooit een geometrie kunnen verwerken, mits deze natuurlijk voldoet aan de WKT norm die gevraagd wordt. Standaard gaan wij hier natuurlijk tegen het probleem aanlopen dat de data uit onze database niet toereikend genoeg is voor Drill om GIS functionaliteiten uit te voeren.

Omzetten naar WKT geometry
Om de data dus goed voor Drill te krijgen dient de geometry van de lagen in de use case omgezet te worden naar WKT formaat. Om hier te komen heb ik de volgende stappen uitgevoerd:

1. Zet een FME flow op waarin je de data van BAG panden en verblijfsobjecten wilt wegschrijven naar een test tabel.
2. Schrijf de gegevens weg voor 1 object uit elke tabel, waarvan bekend is dat zij een geografische overlap hebben. Zodoende gaan we de functie ST_Intersects niet testen op meer dan 50000 objecten, maar op 2.
3. Voeg de transformer GeometryExtractor toe en laat de geometrie wegschrijven als een OGC Well-Known Text in een nieuwe kolom.

4. Open SQL developer en bekijk de tabellen in Oracle.
5. Voeg een nieuwe kolom toe aan de tabellen onder met als type VARCHAR. Dit doen we omdat we in FME de data netjes hebben weggeschreven als CHAR, maar Oracle ziet dit alsnog als een NCLOB. Apache Drill heeft de geometrieën nodig in een VARCHAR, dus vandaar dat we wederom een nieuwe kolom moeten aanmaken in de tabellen.
6. Cast de gegevens van de al bestaande geometrie kolom met datatype NCLOB naar de nieuwe geometrie kolom met datatype VARCHAR.
7. Doe dit uitvoeren voor beide tabellen en je bent gereed.

Query Apache Drill
Vervolgens gaan we naar Apache Drill om de query uit te laten voeren (beide storages voor BAG pand en verblijfsobject staan al ingericht. De volgende query voeren we dan uit:

```
SELECT
  p.PAND_ID AS pand_id,
  v.VERBLIJFSOBJECT_ID AS vbo_id,
  ST_Intersects(
    ST_GeomFromText(p.GEOM_WKT),
    ST_GeomFromText(v.GEOM_WKT)
  ) AS heeft_overlap
FROM arcgispr.BAG_PAND_TEST_DV p
CROSS JOIN arcgispr.BAG_VBO_TEST_DV v;
```

Vervolgens komt hier het volgende resultaat uit:

pand_id	vbo_id	heeft_overlap
1883100000014735	1883010000046203	true

Conclusie

De GIS functionaliteit ST_Intersects werkt! Daar moet natuurlijk wel de kanttekening bij geplaatst worden dat dit enkel is gelukt voor een 1 op 1 koppeling. Van uitgaande hoe de functie werkt betekent dit dat voor elk object in beide tabellen een true or false resultaat uit moet komen. In de BAG panden tabel zitten 68059 panden. In de tabel BAG verblijfsobjecten zitten 55952 verblijfsobjecten in. Als je die wilt nalopen of er geografisch raakvlak is tussen pand en verblijfsobject gaat Drill dit doen voor 68059 * 55952. Dat geeft 3808037168 resultaten terug waarbij via true or false wordt aangegeven of er overlap is. Ik wilde dit testen, maar ik ga er eigenlijk al vanuit dat Drill dit niet gaat trekken. Helaas kreeg ik dit niet getest omdat ik de geometry niet omgezet kreeg voor alle BAG panden.

Het werkt, maar ik weet dus niet of Apache Drill geschikt is om GIS analyses op uit te voeren met grote datasets.

Ik denk dat het hierom verstandiger is om, voor nu, GIS-analyses gewoon in GIS of in FME uit te laten voeren i.p.v. Apache Drill. Het omzetten van de data was erg omslachtig, maar hier kan je in de toekomst eventueel rekening mee houden als je de data gaat opzetten.