



# Temamøde om entitetsbaserede WFS-tjenester og hændelser

Modernisering af Datafordeleren

Temamøde den 6. maj 2025





OBS: Mødet bliver optaget, så det kan genses i 30 dage





# Velkomst

---

## Gabi Omdahl

- Projektleder for KDS
- [x164540@kds.dk](mailto:x164540@kds.dk)



Klimadastyrrelsen

## Jørgen Petersen

- Product Owner for KDS
- [jopet@kds.dk](mailto:jopet@kds.dk)



Klimadastyrrelsen

## August Clement Leve

- Domæne lead for Netcompany
- [aucl@netcompany.com](mailto:aucl@netcompany.com)

Netcompany

## Thomas Krohn

- Product Owner for KDS
- [thkro@kds.dk](mailto:thkro@kds.dk)



Klimadastyrrelsen

## Enis Hajzeri

- Domæne lead for Netcompany
- [enh@netcompany.com](mailto:enh@netcompany.com)

Netcompany



# Agenda

---

- Velkomst
- Overblik over ny funktionalitet i de tre udviklingsleverancer
- Entitetsbaserede WFS-tjenester
- Entitetsbaserede hændelser
- Det videre forløb
- Spørgsmål

## **Praktisk information:**

- Stil gerne spørgsmål undervejs  
(virtuelle deltagere: skriv spørgsmål i chatten)
- Slides udsendes efterfølgende
- Video gøres tilgængelig i en begrænset periode

# Overblik over ny funktionalitet i de tre udviklingsleverancer



# De 3 udviklingsleverancer



## Udviklingsleverance 1

- Funktionelle fokusdomæner:
  - Fildownloads (ikke beskyttede data)
  - Geodatatjenester (Eksisterende WFS tjenester på Geoserver)

## Udviklingsleverance 2

- Funktionelle fokusdomæner:
  - Fildownloads (Inklusiv beskyttede data, CSV, Rasterdata downloads)
  - Entitetsbaserede GraphQL-tjenester
  - Sikkerhed og beskyttede data
  - Ny portal til administration og brugerstyring

## Udviklingsleverance 3

- Funktionelle fokusdomæner:
  - GraphQL-tjenester (Specialbyggede for EIJ og CPR)
  - Entitetsbaserede WFS tjenester
  - Entitetsbaserede Hændelser
  - CPR tjenester
  - Fleksibel opslagslogik
  - Geografisk filtrering





# Overblik

---

- Et uddybende overblik over indholdet i de tre udviklingsleverancer er under udvikling og vil findes på [www.datafordeler.dk](http://www.datafordeler.dk) fra fredag den 9. maj 2025.
- Her har vi indsat links til detaljeret dokumentation for samtlige funktionelle leverancer.
- Finder du ikke svar på dine spørgsmål i den tilgængelige dokumentation – eller har du som anvender særlige behov eller udfordringer – er du meget velkommen til at kontakte os. Skriv gerne til [daf2@kds.dk](mailto:daf2@kds.dk), så vender vi hurtigt tilbage med et svar.



# De 3 udviklingsleverancer



## Udviklingsleverance 1

- Funktionelle fokusdomæner:
  - Fildownloads (ikke beskyttede data)
  - Geodatatjenester (Eksisterende WFS tjenester på Geoserver)

## Udviklingsleverance 2

- Funktionelle fokusdomæner:
  - Fildownloads (Inklusiv beskyttede data, CSV, Rasterdata downloads)
  - Entitetsbaserede GraphQL-tjenester
  - Sikkerhed og beskyttede data
  - Ny portal til administration og brugerstyring

## Udviklingsleverance 3

- Funktionelle fokusdomæner:
  - GraphQL-tjenester (Specialbyggede for EJF og CPR)
  - **Entitetsbaserede WFS tjenester**
  - **Entitetsbaserede hændelser**
  - CPR tjenester
  - Fleksibel opslagslogik
  - Geografisk filtrering

Paralleldrif

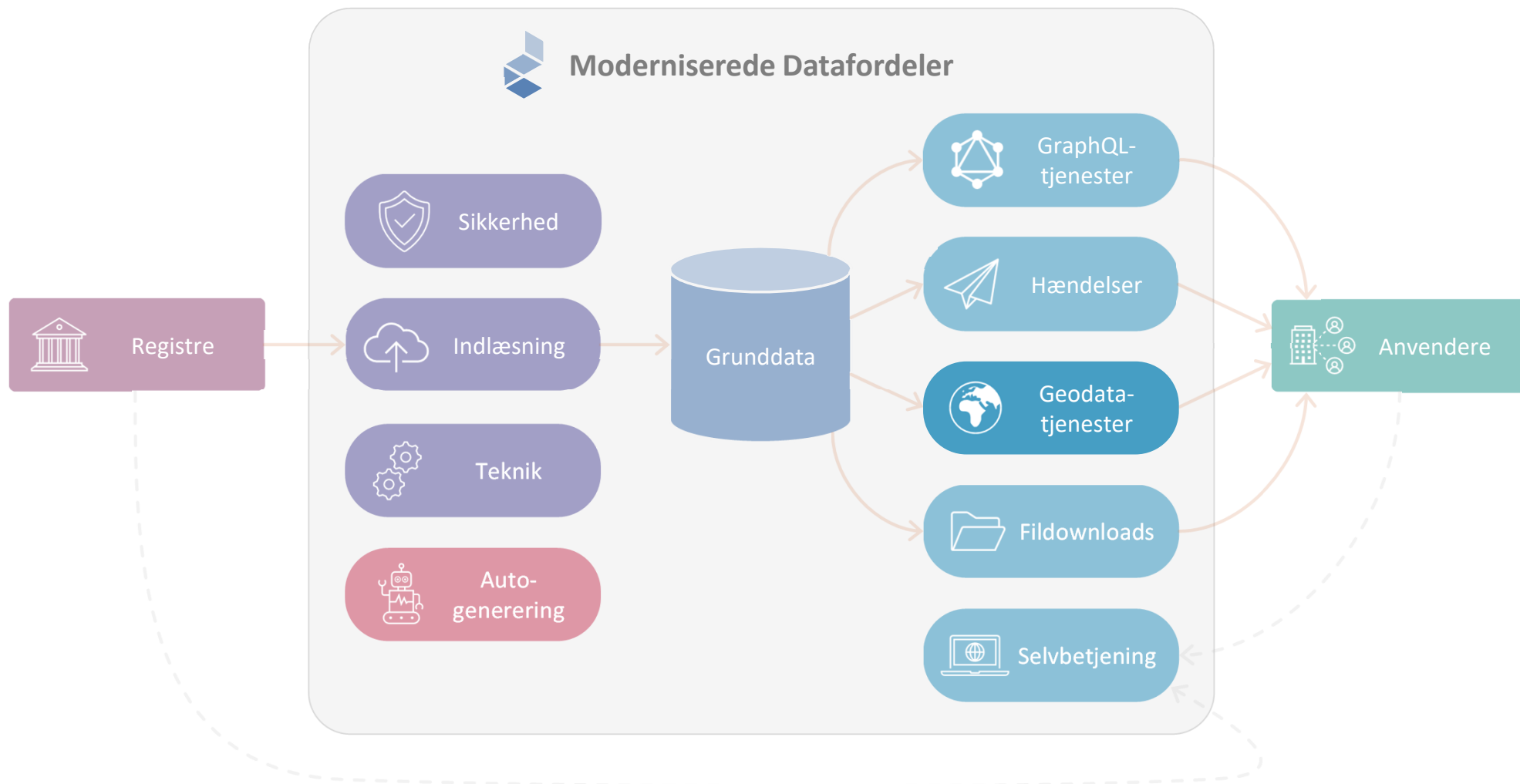




# Entitetsbaserte WFS-tjenester



## Den Moderniserede Datafordeler funktionelle og tekniske domæner





## Gevinster ved geodatatjenester



### Nuværende Datafordeler

- ÷ Komplekst for registre at specificere udstilling af WFSs via DLS – høj time-to-market for implementering heraf
- ÷ Meget manuelt arbejde ifm. Implementering af nye og ændrede WFSs
- ÷ Inkonsistent udformning af DLS'er på tværs af registre
- ÷ Muligheder for fejl ved manuel implementering
- ÷ Tid forbrugt på vedligehold



### Moderniserede Datafordeler

- + Kortere time-to-market
- + Ingen register-specifikation (DLS) af udstillingskemaer, tjenesteparametre eller tjenestelogik
- + Autogenerering af nye og ændrede WFS', med reduceret manuelt arbejde
- + Samme logik og struktur på tværs af registre
- + Reduceret mængde fejl fra manuel implementering da det nu autogenereres
- + Reduceret mængde tid brugt på vedligehold som følge af autogenerering



## Geodata-tjenester

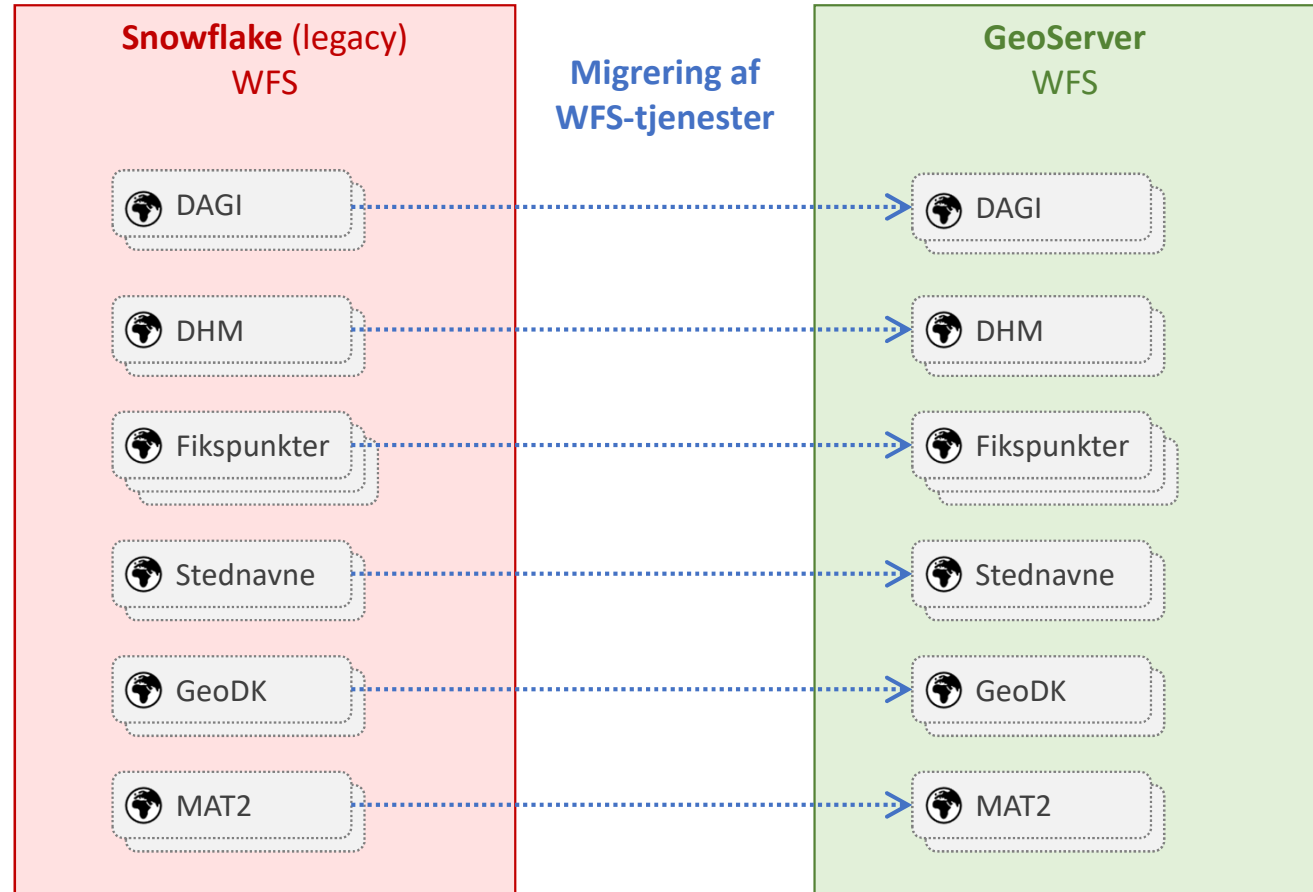
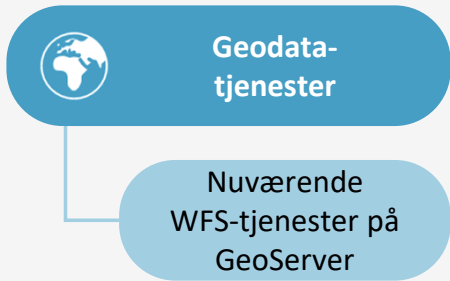
Nuværende  
WFS-tjenester på  
GeoServer

Entitetsbaserede  
WFS-tjenester

## Recap: WFS-tjenester

- WFS-tjenester på Snowflake GO Publisher (Legacy)
- WFS-tjenester på GeoServer (UL1)
- **Entitetsbaserede WFS-tjenester (UL3)**
- Fremtidig videreudvikling efter UL3: POC på OGC API Features

# Nuværende WFS-tjenester er flyttet til GeoServer



*Efterfølgende paralleldrift i en periode*



## Geodata-tjenester

Nuværende  
WFS-tjenester på  
GeoServer

## Plan for udvikling og udfasning af WFS tjenester på Datafordeleren

Register navn	WFS Specialtjenester (UL 1)	Fortsætte WFS Specialtjenester (UL 1)	Udvikling af WFS Entitetsbaseret (UL 3)
Matrikel	Gældende og Foreløbig	Ja	Ja
	Med historik	Udfases*	Ja
Geodanmark	Vektor	Udfases*	Ja
	Vektor med Historik	Udfases*	
DHM	Højdekurver	Udfases*	Ja
	Oprindelse	Udfases*	
DAGI	Område	Udfases*	Ja
	Område Historisk	Udfases*	
Danske Stednavne	Bearbejdet	Ja	Ja
	Ubearbejdet	Ja	
Fikspunkter	Simpel struktur	Ja	Ja
	Fuld struktur	Ja	
	Fuld struktur med Historik	Ja	
DAR	<i>Pilotversion ikke i drift</i>	-	Ja
BBR	<i>Ingen WFS tjeneste i dag</i>	-	Ja

\* Tidsplan for udfasning aftales mellem DAF-Operatøren og registret.



## Geodata-tjenester

Nuværende  
WFS-tjenester på  
GeoServer

Entitetsbaserede  
WFS-tjenester

## Registre som udstiller entitetsbaserede WFS-tjenester



BBR



GeoDanmark  
Vektor



DAGI



Matriklen



DHM



DAR



Danske  
Stednavne



Fikspunkter

Detaljer kan  
ændre sig

## Autogenerering

Revideret  
DLS-format

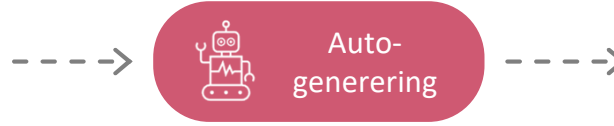
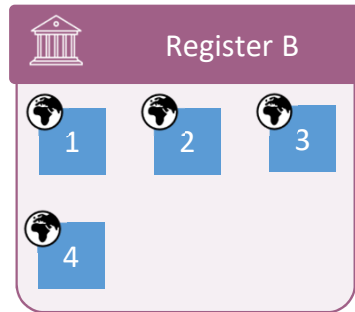
Autogenerering af  
fildownloads

Autogenerering af  
GraphQL-tjenester

Autogenerering af  
hændelser

Autogenerering af  
geodatatjenester

## Register datamodeller

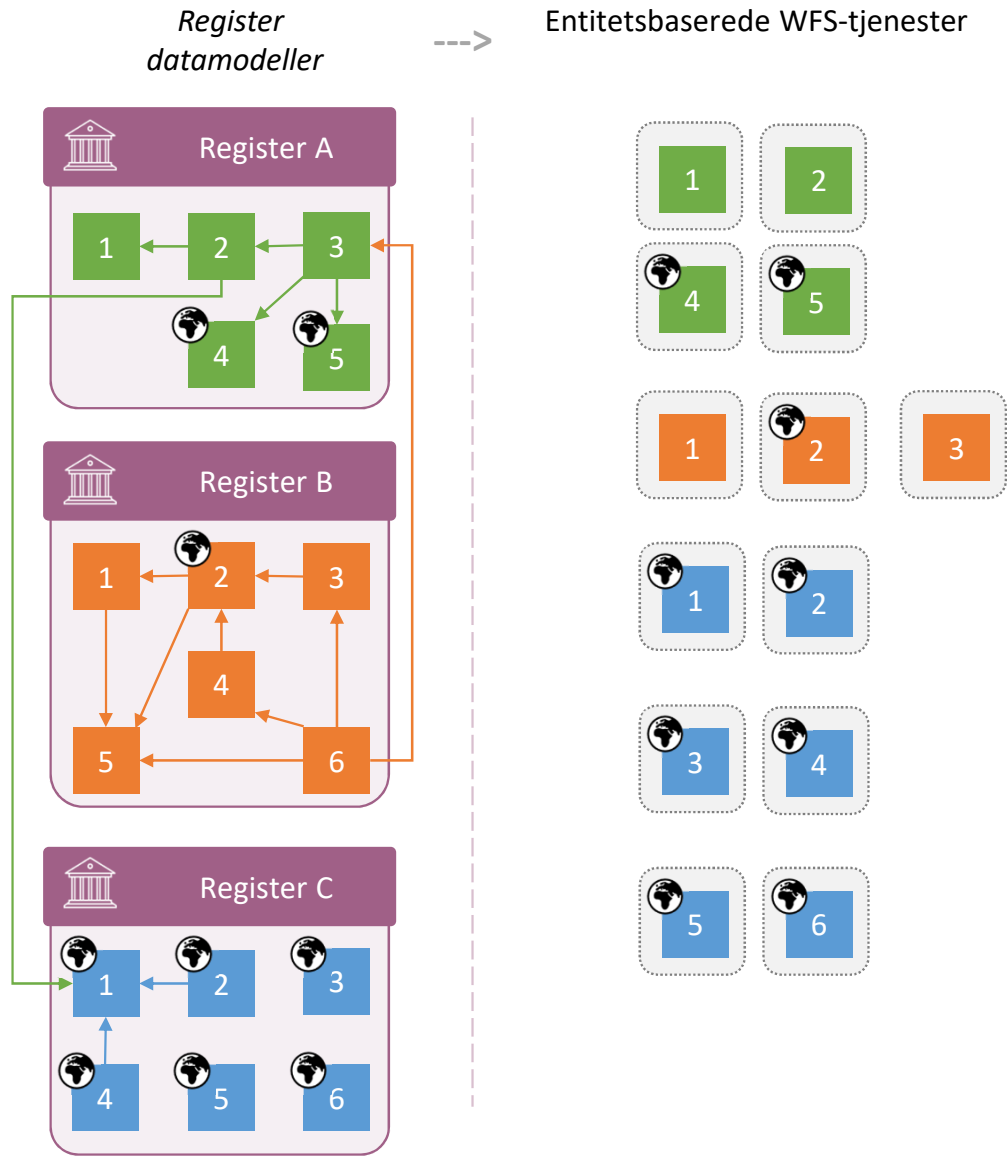
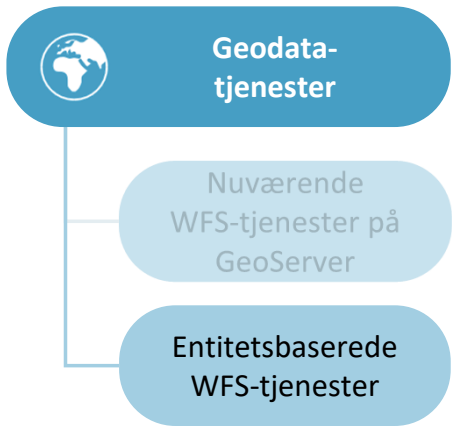


## Entitetsbaserede WFS-lag



Detaljer kan  
ændre sig





**Bemærk: Sammenstillede tjenester udstilles via eksisterende specialtjenester**

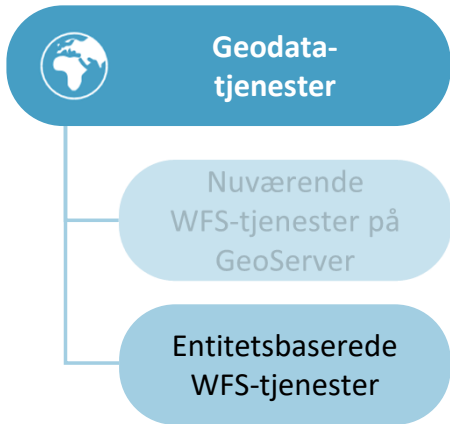
- Udstilles i 2 temporaliteter:

- Aktuelle data
- Bitemporale data



- Tilgås via standard OGC-compliant WFS forespørgsel
- Register kan filtrere lag fra, der ikke skal udstilles.
- Bemærk at det ikke nødvendigvis er alle lag som indeholder geometrier

Detaljer kan ændre sig



## Forespørgsel på WFS-tjenester

- WFS version 2.0.0 benyttes
- URL'en kan angives direkte i GIS-klienter (testes i QGIS, ArcMap, MapInfo)
- Data returneres som GeoJSON eller GML ved at angive det i outputFormat-parameteret
- Anvendt projektion er EPSG:25832
- Parametre som kan angives følger OGC-standard

<https://wfs.datafordeler.dk/>

**BBR/WFS?Service=WFS&Version=2.0.0&Request=GetFeature&TypeNames=bbr:Bygning\_Current&APIKey=ABC**

Angivelse af Register, fx  
BBR eller DHM

OGC-parametre

Navn på **registeret** som  
entiteten tilhører, fx  
BBR eller DHM

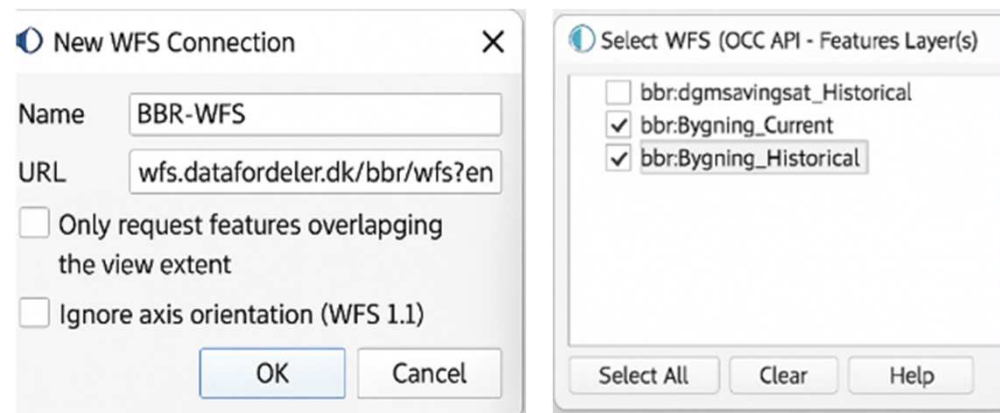
Navn på entitet og  
historik, fx  
**Bygning\_Current** eller  
**Bygning\_Historical**

Autentificering af den  
kaldende anvender,  
API key for  
ubeskyttede data

Detaljer kan  
ændre sig

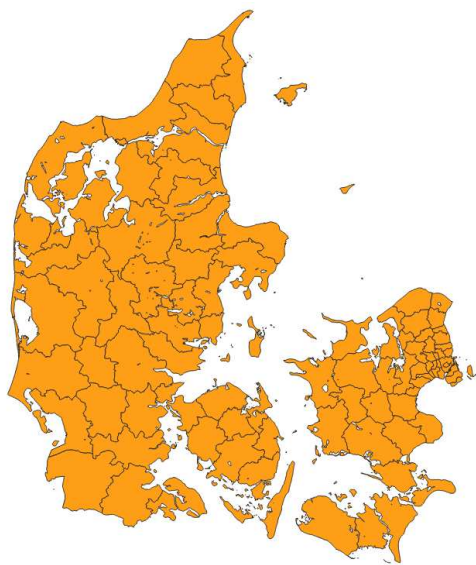
# Eksempel på anvendelse

1. Opret en ny WFS-forbindelse i fx QGIS
  1. a) Indstil basis-URL:  
<https://wfs.datafordeler.dk/bbr/wfs?service=WFS>
  2. B) Klik **Connect** – QGIS læser GetCapabilities og viser alle entitetslag
2. Vælg det ønskede lag, fx
  - bbr:Bygning\_Current (aktuelle data) eller
  - bbr:Bygning\_Historical (historiske data)
3. Tryk **Add** – laget tilføjes til *Layers*-panelet og tegnes i kortet
  - Nu kan du zoome, style, filtrere og analysere direkte fra WFS-kilden



Detaljer kan ændre sig

# Filtrering



SQL Query Composer

```
SELECT * FROM MatrikelKommune_Current where kommunekode = '0201'
```

SQL Statement

Columns \*

Table(s) MatrikelKommune\_Current

Joint layer	ON condition
-------------	--------------

Where kommunekode = '0201'

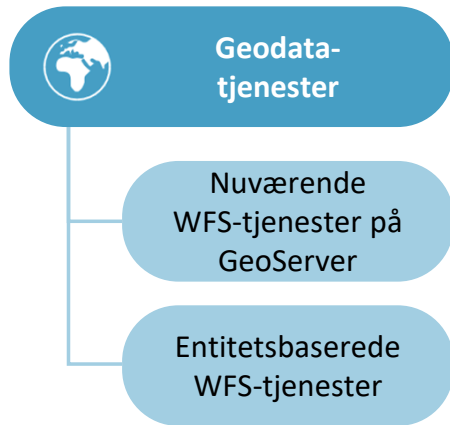
Order by

Reset OK Cancel Help

Data  
Tables  
Columns  
Functions  
Spatial predicates  
Operators



Detaljer kan ændre sig



## Forskel mellem UL3 og UL1 WFS-tjenester

Entitetsbaserede UL3 WFS-tjenester leveres uden:

- Sammenstilling
- Præfiltrering
- Dybe forespørgsler (f.eks. ved brug af "til-mange" relationer)

UL1 WFS-tjenester udstilles som specificeret i nuværende specialtjenester.

Alternativt kan filtrering foretages i eget system.

# Entitetsbaserede hændelser



# Introduktion til hændelser

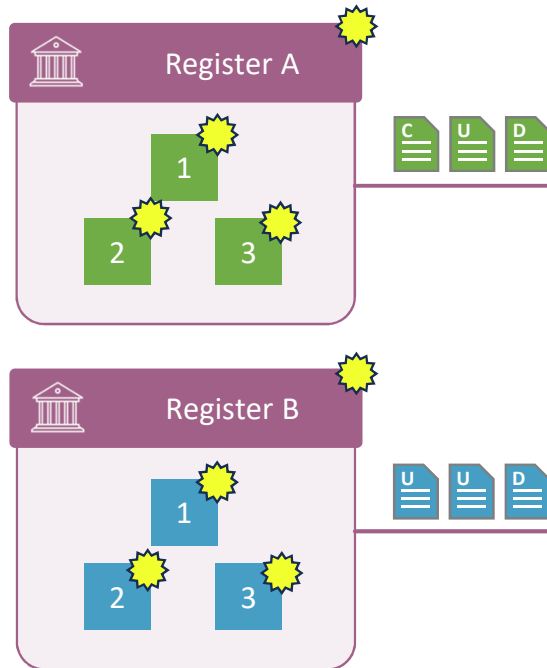
---

- Dataopdateringer fra registre afføder hændelser
  - Anvendere kan modtage/hente information, når registerdata bliver opdateret
    - Eks. Virksomhed skifter navn -> Anvendere får det at vide
  - Hændelser kan fx. blive brugt i følgende scenarier
    - Opdatere sin egen register kopi
    - Notificere sine egne anvendere om ændringer
  - Nuværende hændelser: Aktivt brugt men teknisk forældet
- Moderniserede hændelser: Mere gennemsigtigt for anvendere, registre og Datafordeleren
  - Der hentes blot de hændelser der ønskes, når det passer for anvenderen – også tidligere hændelser

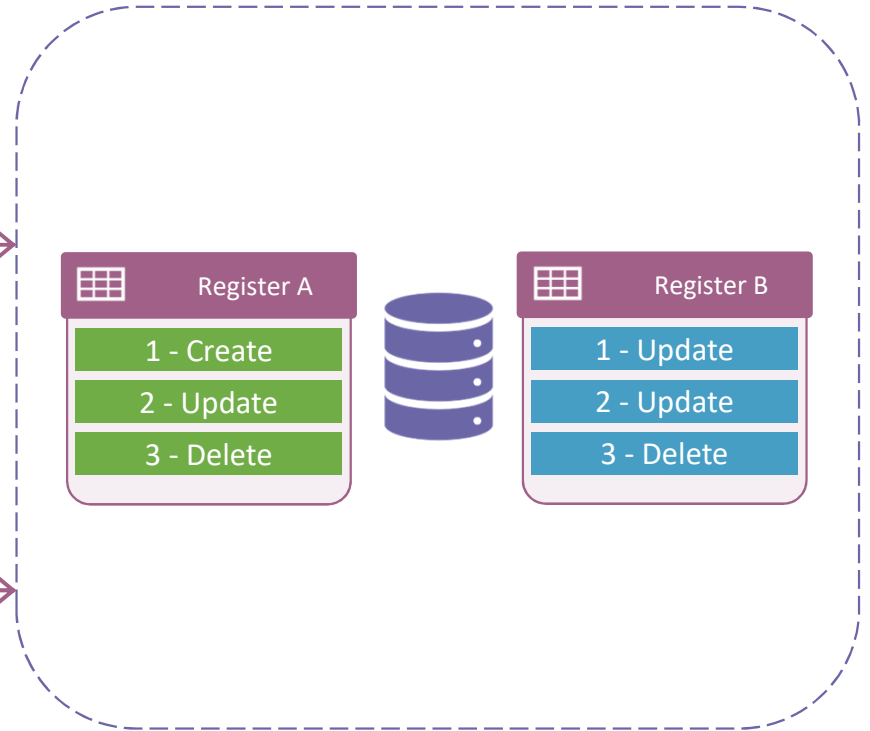
Indlæsning

Indlæsnings-  
understøttelse af  
hændelser

Register dataopdatering



Opdatering i  
hændelsestabeller





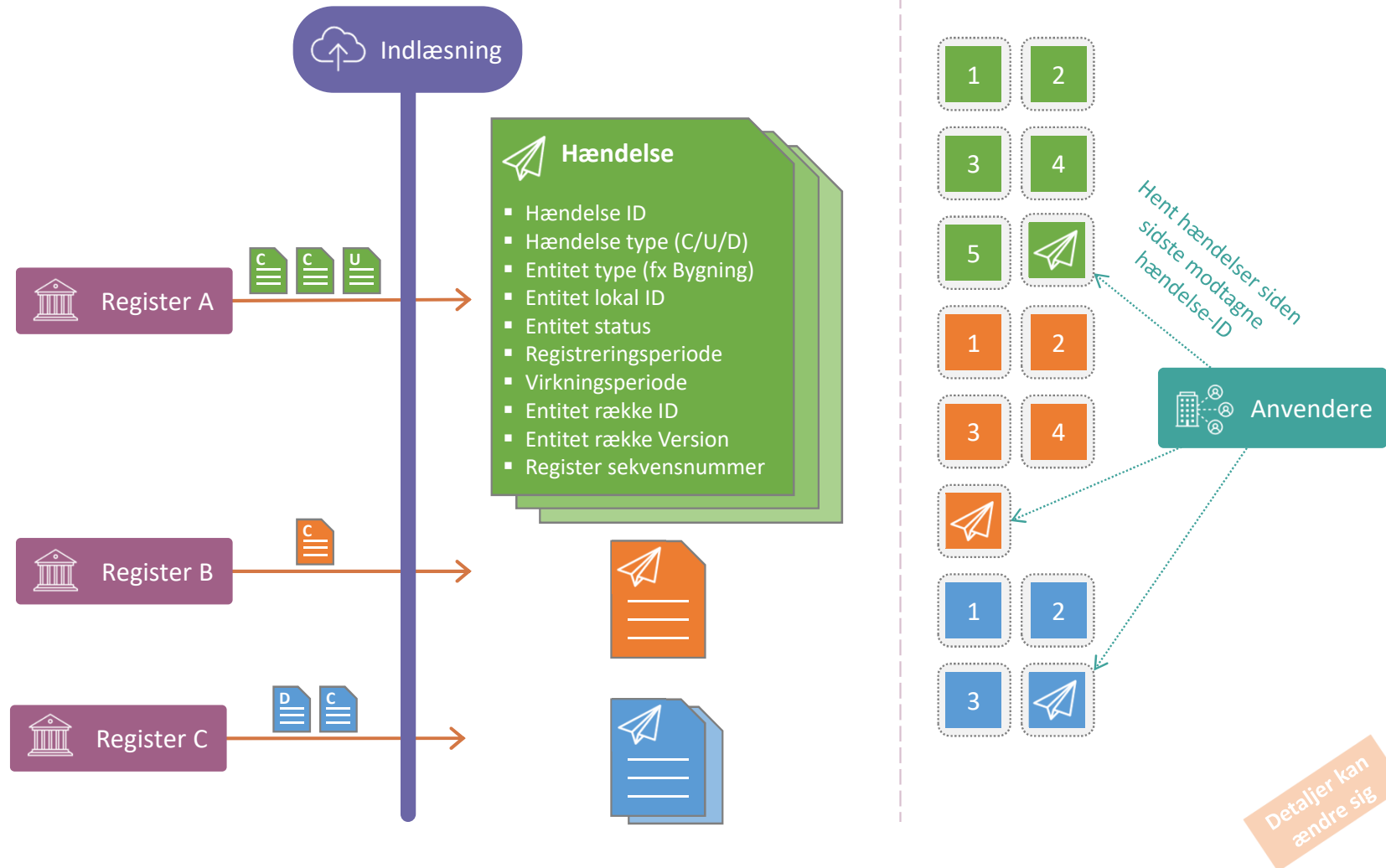
# Hændelser

## Entitetsbaserede hændelser

Dataopdateringer fra registre

Data opdateres i registers entitetstabel og datahandling tilføjes til registers hændelsestabel

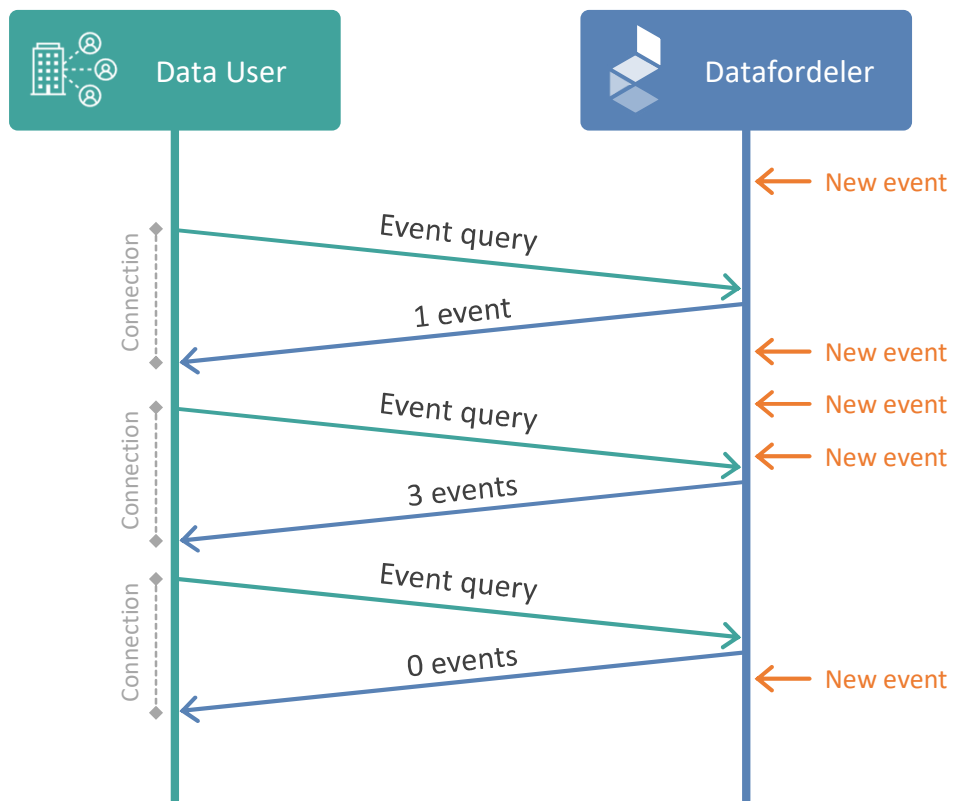
Hændelsestabel udstilles via GraphQL-tjenester, tilsvarende til registerentiteter



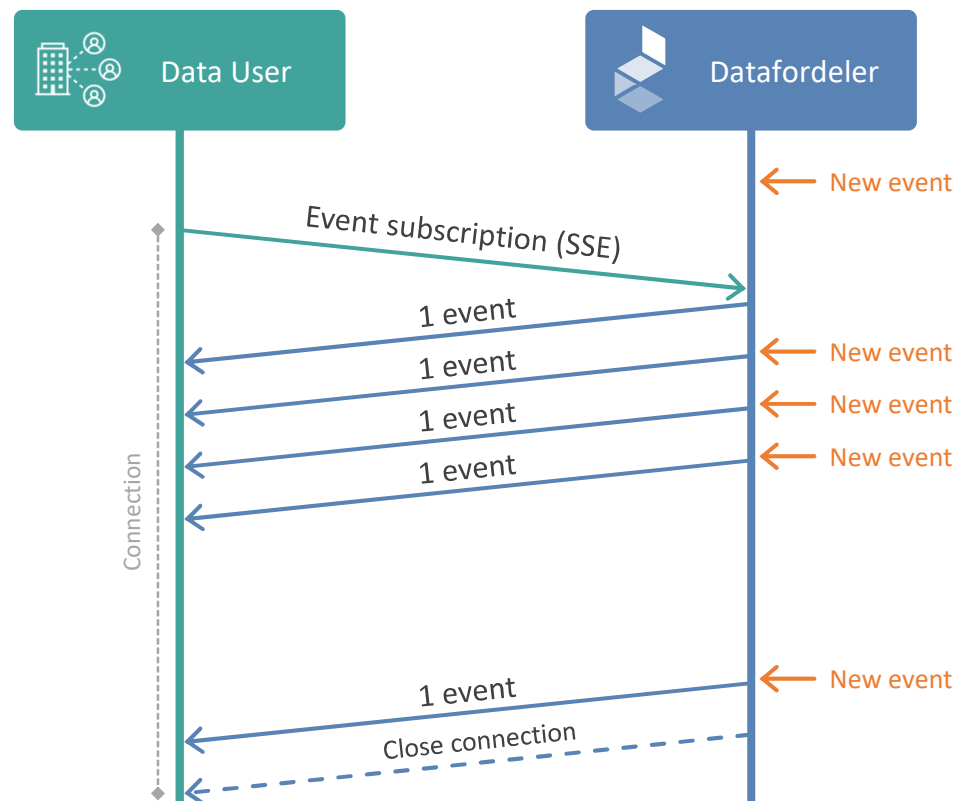


## Get events: GraphQL query vs. subscription

**GraphQL query**  
*Without Server-Sent Event (SSE)*

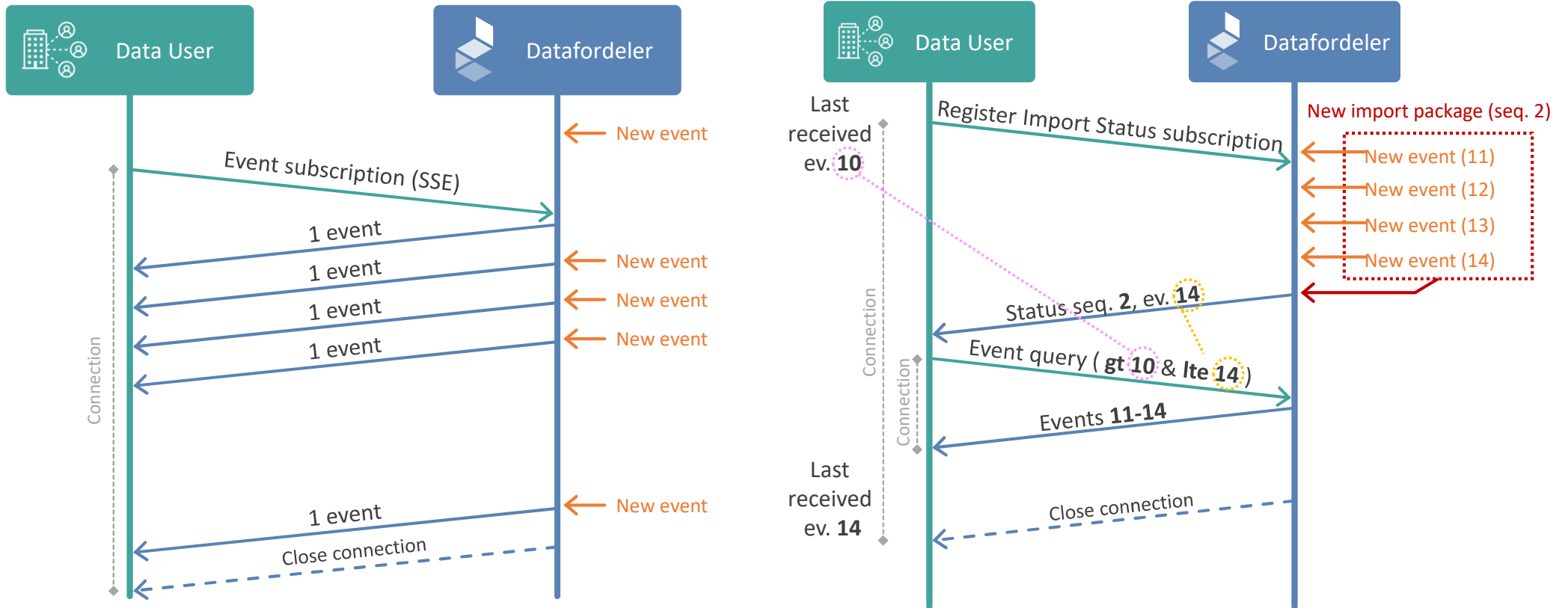


**GraphQL subscription**  
*With Server-Sent Event (SSE)*





## Register data consistency via events and register import status (trigger: events)





# GraphQL Query Registerimportstatus

**POST** https://test07-graphql.datafordeler.dk/BBR/v1/

Query Pre-request Post-request Auth •

(Send query)

```
1 query {
2   DAF_RegisterImportStatus(
3   ) {
4     lastEventId
5     lastSequenceNumber
6     lastUpdated
7   }
8 }
```

**Result** Response headers

200 OK ⌚ 565ms

```
1 {
2   "data": {
3     "DAF_RegisterImportStatus": {
4       "lastEventId": 3,
5       "lastSequenceNumber": 2,
6       "lastUpdated": "2025-01-23T12:27:03.000000Z"
7     }
8   }
9 }
```

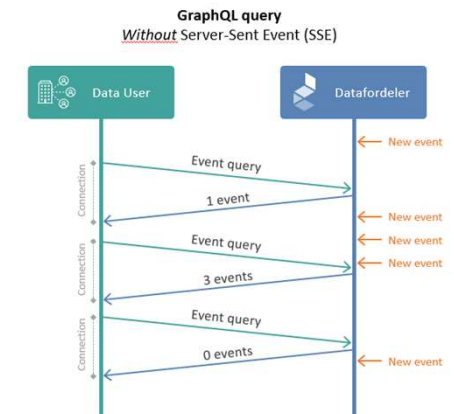


## Brugerscenarie 1 – Brugere anvender Query for at modtage hændelser

1. Brugerne ønsker at modtage hændelser fra BBR\_Bygning for at få information omkring data ændringer til bygning

Brugerne laver en GraphQL Query for at modtage eventuelle hændelser

2. Brugerne henter opdateret registerdata, via GraphQL for BBR\_Bygning og kan efterfølgende opdatere lokale data
3. Brugerne notificerer sine egne anvendere, at opdateringer er sket





## Brugerscenarie 1 – GraphQL Query til hændelseskald

---

### Tidligere metode: PULL

1. Brugeren opsætter et abonnement, med ønskede filtre
2. Derefter foretages REST-kald, f.eks.
  1. [https://services.datafordeler.dk/system/EventMessages/1.0.0/Custom?datefrom=2025-01-08&dateto=2025-01-09&username=\\*\\*\\*&password=\\*\\*\\*](https://services.datafordeler.dk/system/EventMessages/1.0.0/Custom?datefrom=2025-01-08&dateto=2025-01-09&username=***&password=***)
3. Autentificeringsmetode: Brugernavn og adgangskode
4. Filtrering og svar varierer fra register til register



# Brugerscenarie 1 – GraphQL Query til hændelseskald

POST https://test07-graphql.datafordeler.dk/BBR/v1/

Query Pre-request Post-request Auth -

```
1 query {
2   BBR_events(
3     where: {
4       eventId: { gt: 20 },
5       entityname: { eq: "Bygning" }
6     }
7   ) {
8     nodes {
9       eventId,
10      datafordelerOpdateringstid,
11      entityname,
12      eventaction,
13      eventId,
14      object_id,
15      object_registreringfra,
16      object_registreringtil,
17      object_datafordelerRowId,
18      object_datafordelerRowVersion,
19      object_status,
20      object_virkningfra,
21      object_virkningtil,
22      datafordelerRegisterImportSequenceNumber
23    }
24    pageInfo {
25      hasNextPage
26      hasPreviousPage
27      startCursor
28      endCursor
29    }
30  }
31 }
```

VARIABLES

```
1 {}
```

Result Response headers

200 OK 446ms Clear 9:24:02 AM

```
1 {
2   "data": {
3     "BBR_events": {
4       "nodes": [
5         {
6           "eventId": 26,
7           "datafordelerOpdateringstid": "2025-04-02T12:38:18.462707Z",
8           "entityname": "Bygning",
9           "eventaction": "u",
10          "object_id": "3b29635f-7104-4500-94f6-007fa6847bc3",
11          "object_registreringfra": "2023-03-20T06:19:31.681063Z",
12          "object_registreringtil": "2023-10-23T04:44:06.555776Z",
13          "object_datafordelerRowId": "c2deaac8-a32c-4699-a13c-
14          f64ba07f3f9d",
15          "object_datafordelerRowVersion": 1,
16          "object_status": "3",
17          "object_virkningfra": "2023-03-20T06:19:31.681063Z",
18          "object_virkningtil": null,
19          "datafordelerRegisterImportSequenceNumber": 3453506
20        },
21        {
22          "eventId": 27,
23          "datafordelerOpdateringstid": "2025-04-02T12:38:18.464021Z",
24          "entityname": "Bygning",
25          "eventaction": "i",
26          "object_id": "3b29635f-7104-4500-94f6-007fa6847bc3",
27          "object_registreringfra": "2023-10-23T04:44:06.555776Z",
28          "object_registreringtil": null,
29          "object_datafordelerRowId": "3d1e0bf8-168d-4db3-ab34-
30          90e2e4283568",
31          "object_datafordelerRowVersion": 1,
32          "object_status": "9",
33          "object_virkningfra": "2023-10-23T04:44:06.555776Z",
34          "object_virkningtil": null,
35          "datafordelerRegisterImportSequenceNumber": 3453506
36        }
37      ]
38    }
39  }
40 }
```



# Brugerscenarie 1 – GraphQL Query til registerkald

POST https://test07-graphql.datafordeler.dk/BBR/v1/

Query Pre-request Post-request Auth

```
1 query {
2   BBR_Bygning(
3     where: {
4       datafordelerRowId: { eq: "c2deaac0-a32c-4699-a13c-f64ba07f3f9d" }
5       datafordelerRowVersion: { eq: 1 }
6     }
7   ) {
8     nodes {
9       datafordelerRowId
10      datafordelerRowVersion
11      forretningshaendelse
12      forretningsomraade
13      forretningsproces
14      kommunekode
15      jordstykke
16      husnummer
17      grund
18    }
19  }
20 }
21 }
```

Result Response headers

200 OK 25ms

```
1 {
2   "data": {
3     "BBR_Bygning": {
4       "nodes": [
5         {
6           "datafordelerRowId": "c2deaac0-a32c-4699-a13c-f64ba07f3f9d",
7           "datafordelerRowVersion": 1,
8           "forretningshaendelse": "Bygning",
9           "forretningsomraade": "54.15.05.05",
10          "forretningsproces": "11",
11          "kommunekode": "8386",
12          "jordstykke": "2531661",
13          "husnummer": "0a3f5884-c9a6-32b8-e044-0083ba298818",
14          "grund": "c42d1356-a75d-4cb4-a762-1e7a05e76558"
15        }
16      ]
17    }
18  }
19 }
```



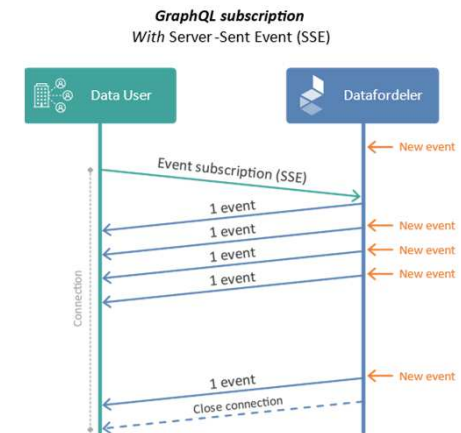


## Brugerscenarie 2 – Brugere anvender Subscription for at modtage hændelser

1. Brugere ønsker at modtage hændelser fra BBR\_Bygning for at få information omkring data ændringer til bygning

Brugere opretter en SSE forbindelse via GraphQL Subscription, for at blive notificeret når der er opdateringer for BBR\_Bygning

2. Brugere henter opdateret registerdata, via GraphQL for BBR\_Bygning og kan efterfølgende opdatere lokale data
3. Brugere notificerer deres egne anvendere, at opdateringer er sket





## Brugerscenarie 2 – GraphQL Subscription til hændelseskald

---

### **Tidligere metode: PUSH**

1. Anvenderen opsætter et endpoint, der kan modtage data
2. Anvenderen opretter et abonnement hos Datafordeleren:
  1. Endpoint og konfiguration
  2. Entiteter
  3. Filtrering
3. Datafordeleren overvåger løbende nye genererede hændelser for alle abonnementer
4. Datafordeleren forsøger at sende hændelserne ud



## Brugerscenarie 2 – GraphQL Subscription til hændelseskald

---

```
Request Run > Cancel Response success status 1 events N/A duration 230 B size {} Copy More  
1 subscription Subscription {  
2   BBR_events {  
3     eventId  
4     object_datafordelerRowId  
5     object_datafordelerRowVersion  
6     datafordelerRegisterImportSequenceNumber  
7     entityname  
8   }  
9 }  
10  
11  
1 {  
2   "data": {  
3     "BBR_events": {  
4       "eventId": 24,  
5       "object_datafordelerRowId": "26382856-fa1d-4eb3-9272-a636dfc68ba4",  
6       "object_datafordelerRowVersion": 1,  
7       "datafordelerRegisterImportSequenceNumber": 9,  
8       "entityname": "Bygning"  
9     }  
10  }  
11 }
```



## Brugerscenarie 2 – GraphQL Query til registerkald

POST https://test07-graphql.datafordeler.dk/BBR/v1/

Query Pre-request Post-request Auth

Result Response headers

200 OK 25ms

```
query {
  BBR_Bygning(
    where: {
      datafordelerRowId: { eq: "c2deaac8-a32c-4699-a13c-f64ba87f3f9d" }
      datafordelerRowVersion: { eq: 1 }
    }
  ) {
    nodes {
      datafordelerRowId
      datafordelerRowVersion
      forretningshaendelse
      forretningsomraade
      forretningsproces
      kommunekode
      jordstykke
      husnummer
      grund
    }
  }
}
```

```
{
  "data": {
    "BBR_Bygning": {
      "nodes": [
        {
          "datafordelerRowId": "c2deaac8-a32c-4699-a13c-f64ba87f3f9d",
          "datafordelerRowVersion": 1,
          "forretningshaendelse": "Bygning",
          "forretningsomraade": "54.15.05.05",
          "forretningsproces": "11",
          "kommunekode": "8306",
          "jordstykke": "2531661",
          "husnummer": "0a3f5084-c9a6-32b8-e044-0003ba298018",
          "grund": "c42d1356-a75d-4cb4-a762-1e7a05e76558"
        }
      ]
    }
  }
}
```



## Brugerscenarie 3 – GraphQL hændelser med Fildownloads

---

1. Brugerne henter en lokal kopi af BBR\_Bygning gennem Fildownloads
2. Brugerne spørger, via GraphQL Query eller Subscription, hvis der er opdateringer for BBR\_Bygning
3. Ved nye hændelser, henter bruger opdateret registerdata
4. Brugerne opdaterer deres egen kopi af BBR\_Bygning



## Brugerscenarie 3 – Fildownloads med hændelsesflow

---

1. Nye systemkolonner er tilgængelige i registerdata via GraphQL, totaldownloads og deltdownloads
2. Anvenderen henter kun opdaterede registerdata, når der er nye hændelser:
  1. GraphQL Query for hændelser
  2. GraphQL Subscription for hændelser
  3. RegisterImportStatus
3. Anvenderen henter herefter de opdaterede data via GraphQL for at modtage ændringerne med det samme
4. Anvenderen opdateres deres eget kopi med opdateringerne



## Gevinster ved hændelser



### Nuværende Datafordeler

- ÷ Uens udstilling af hændelser fra register til register
- ÷ Hændelser kun tilgængelig for en brøkdel af entiteter
- ÷ Komplex og håndholdt for anvendere at opsætte hændelsesabonnementer
- ÷ Anvendere skal afvente at der kommer nye hændelser og det er ikke muligt at hente hændelser fra før abonnement er oprettet
- ÷ Komplekst for registre at specificere hændelser via DLS – og høj time-to-market for implementering heraf
- ÷ En del manuelt arbejde ifm. implementering af nye og ændrede hændelser
- ÷ Komplex og forældet teknisk løsning



### Moderniserede Datafordeler

- + Ensartet udstilling af hændelser, uanset register
- + Hændelser for alle entiteter (medmindre fravalgt af register) og alle typer af bitemporale dataopdateringer
- + Ingen abonnementer – der hentes blot de hændelser der ønskes, når det passer for anvenderen – også tidligere hændelser
- + Ingen PUSH-hændelser, men i stedet GraphQL subscription som kan bruges til samme formål
- + Ingen register-specifikation (DLS) af hændelsesstrukturer eller felt-mapninger
- + Autogenerering af implementering af nye og ændrede hændelser, med minimalt manuelt arbejde

Detaljer kan ændre sig

# Videre forløb





# Videre forløb

---

## I hører fra os!

- Temamøde 2 om UL3, torsdag den 22. maj kl. 9:30 – 13:00
  - Fleksibel opslagslogik
  - CPR-tjenester + EJF
- Løbende nyhedsbreve.
- Overblik over leverancer på Datafordeler.dk.





# Nyhedsbrev og mere information

---

- Læs mere om projektet:  
<https://datafordeler.dk/modernisering>
- Tilmeld jer til vores [nyhedsbrev](#) og få besked, når der er nyt.
  - Via ovenstående link findes tilmelding:

