



Informationsmøde om endeligt scope i UL3 og ekstern test

Modernisering af Datafordeleren

Informationsmøde den 17. september 2025



Klimadatastyrelsen



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



Slide tilføjet efter informationsmødet

Bemærk vedrørende paralleldrift

Paralleldrift udfasninger:

1.april 2026: (30. april blev fejlagtigt nævnt på mødet)

- WFS på Snowflake - erstattet af GeoServer
- Rasterdata filudtræk - erstattet af fildownload

30.juni 2026: ikke-moderniserede tjenester

Info via datafordeler.dk og nyhedsbrev



Agenda

- Velkomst - v. *Nanna Barndorff, Kontorchef KDS*
- Status på Moderniseringsprojektet - v. *Natalia Hansen, Projektleder KDS*
- Præsentation af funktionalitet implementeret i UL3 - v. *Kristian Kjølstad Jørgensen PO'er KDS, Thomas Krohn PO'er KDS, Jørgen Petersen, PO'er KDS, og Marcus Norton Quistgaard, Team Lead Netcompany*
- Ekstern test - v. *Kaja Hansen, Testmanager KDS*
- Transition til den Moderniserede Datafordeler - v. *Linda Strøm Lundum, Implementeringsansvarlig KDS*
- Spørgsmål - *alle*
- Afrunding - v. *Natalia Hansen, Projektleder KDS*

Praktisk information:

- Stil gerne spørgsmål undervejs (virtuelle deltagere: skriv i chatten)
- Specifikke spørgsmål besvares efterfølgende
- Slides udsendes efterfølgende
- Video gøres tilgængelig i min. 30 dage

Status på Moderniseringsprojektet

v. Natalia Hansen, ProjeKtleder KDS



Projektstatus

- Projektet er på spor
- UL2 – driftsprøven er afsluttet
- UL3 – sidste Sprint 18 er afsluttet
- Ekstern Test 22. september – 24.oktober
- UL3 idriftsættes 27. november 2025
- UL3 driftsprøven afholdes 12. december – 12. januar
- Videreudvikling fortsætter
- Drift og vedligehold

Paralleldrift udfasninger:

1.april 2026:

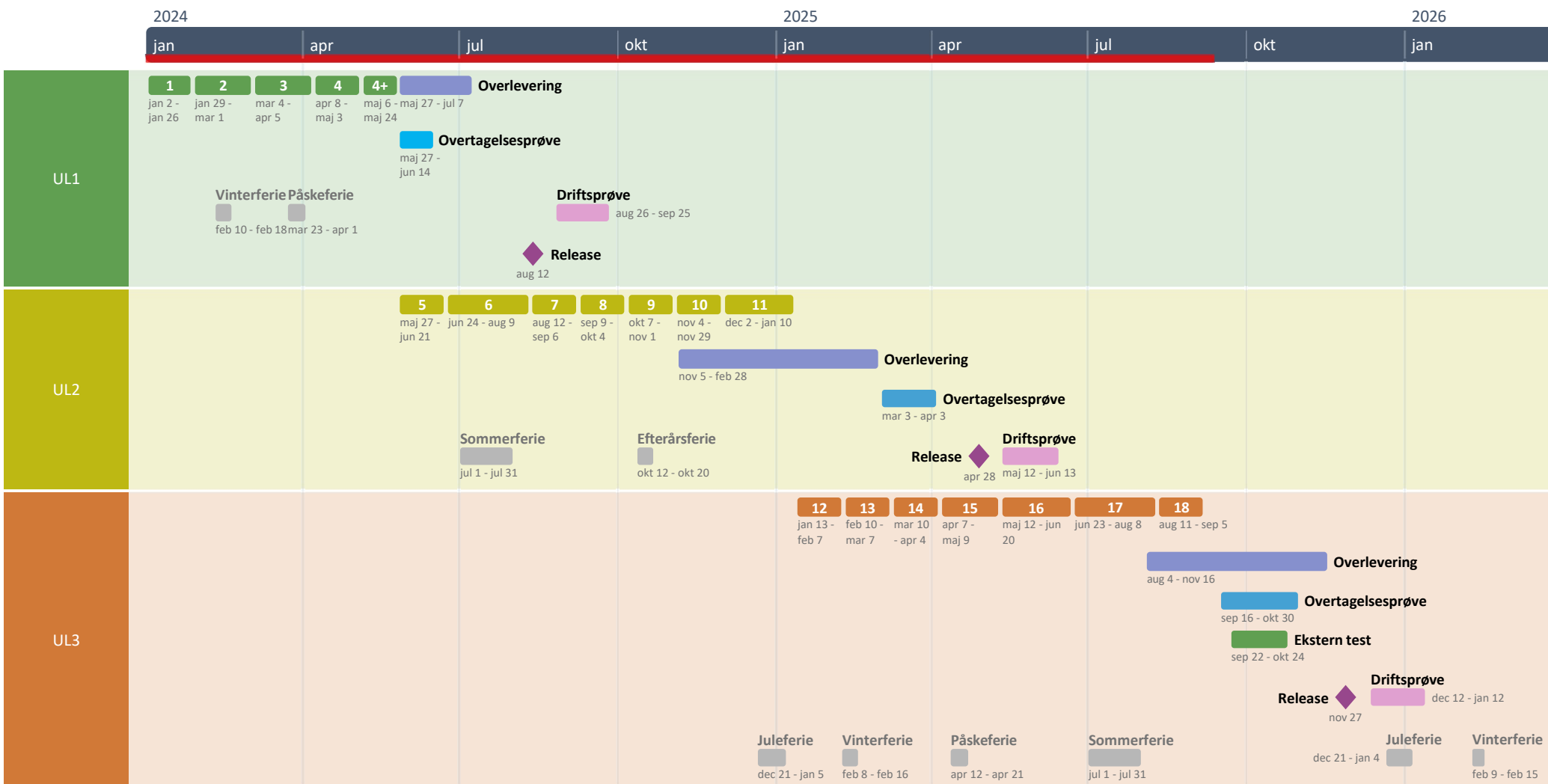
- WFS på Snowflake - erstattet af GeoServer
- Rasterdata filudtræk - erstattet af fildownload

30.juni 2026: ikke-moderniserede tjenester

Info via datafordeler.dk og nyhedsbrev

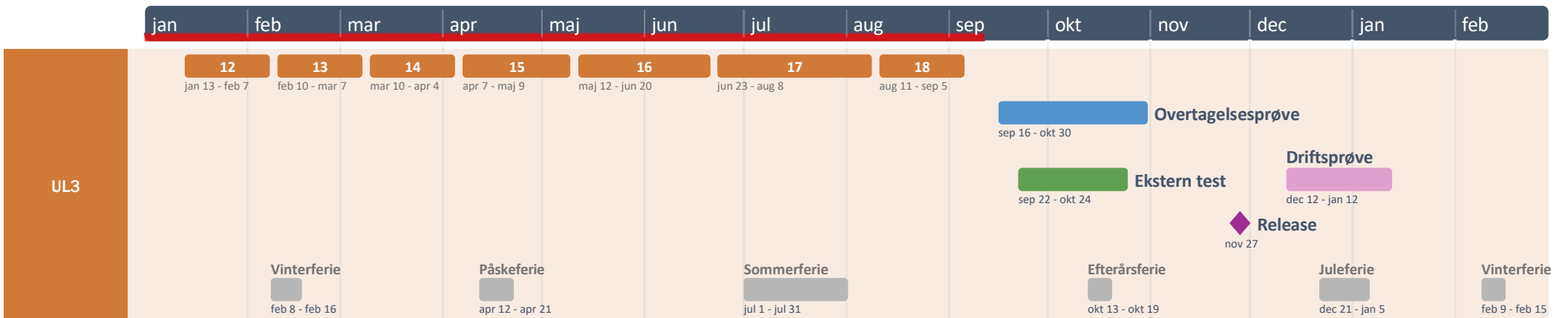


Tidslinje for Leveranceplan





Tidslinje for UL3



Moderniseringsbehov

- Indfriet via ny funktionalitet

MB1: Etablering af modulær it-arkitektur

MB2: udvidet understøttelse af versionering

MB3: Etablering af funktionalitet ved brug af Standardprogrammel

MB4: Øget brug af dynamiske skalering af kapacitet

MB5: Øget automatisering af testmiljøer

MB6: Øget brug af automatiseret test

MB7: Optimering af Hændelser samt hændelsesgenerering

MB8: Optimering af logning og overvågning via logning

MB9: Optimering af distribution af geodata

MB10: Optimering af indlæsning af data

MB11: Optimering af forespørgsels- og udstillingsværktøjer

MB12: Optimering af understøttelse af udstilling af data i nær realtid

MB13: Autogenerering af REST-tjenester

MB14: Optimering af filudtræk

MB15: Optimering af funktionalitet til selvbetjening

MB16: Optimering af understøttelse af kopiregistre

MB17: Optimering af håndtering af metadata



De 3 udviklingsleverancer



August
2024

28. April
2025

27. Nov
2025

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

Paralleldrift

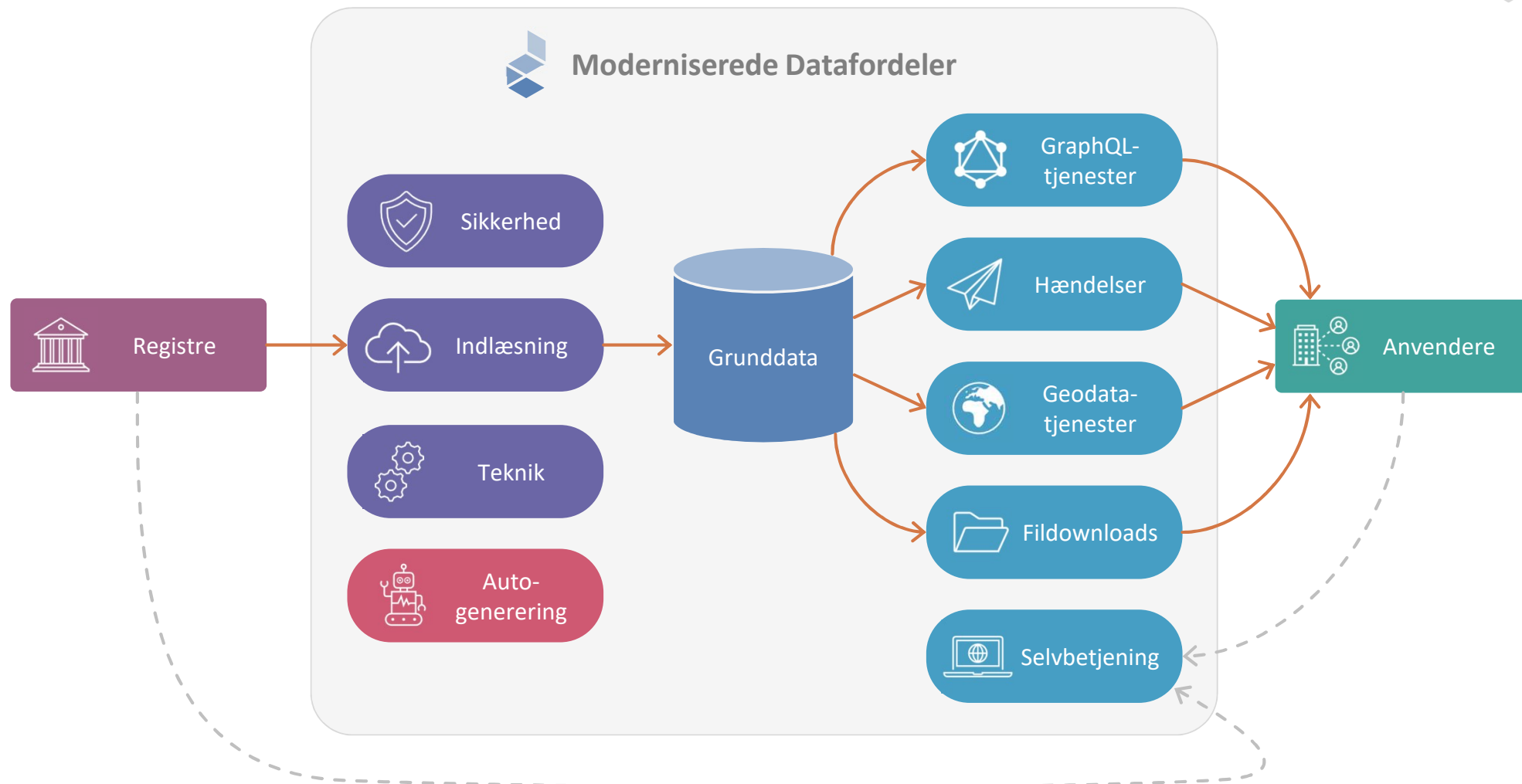


Funktionalitet implementeret i UL3

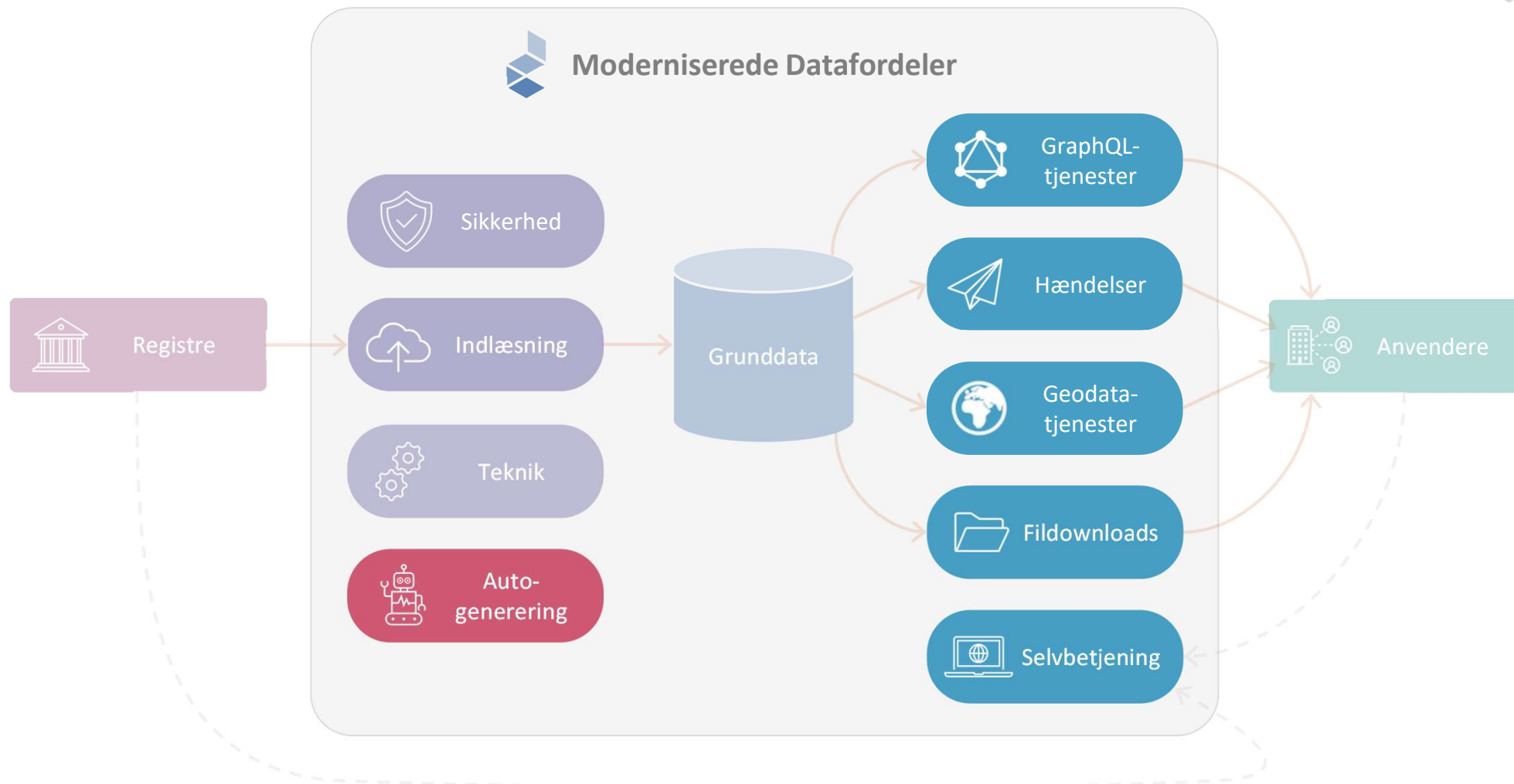
v. Kristian Kjølstad Jørgensen, Product Owner KDS



Den Moderniserede Datafordeler funktionelle og tekniske domæner



Funktionelle og tekniske domæner for UL3



Modernisering i UL3



- Anvenderrettet funktionalitet:
 - Fleksibel opslagslogik til GraphQL-tjenester
 - CPR tjenester i den Moderniserede Datafordeler
 - EJV tjenester i den Moderniserede Datafordeler
 - Entitetsbaserede Hændelser
 - Entitetsbaserede WFS-tjenester
 - Entitetsbaserede Fildownloads med geografisk kommuneopdelt indhold
 - Flere rasterdata Fildownload
 - MD5 checksum på totaldownload af tabulæredata
- Autogenerering ud fra registerdatamodel/DLS af:
 - GraphQL-tjenester for fleksibel opslagslogik
 - Geodatatjenester
 - Hændelser
- Transitions- og kopiregisterguides

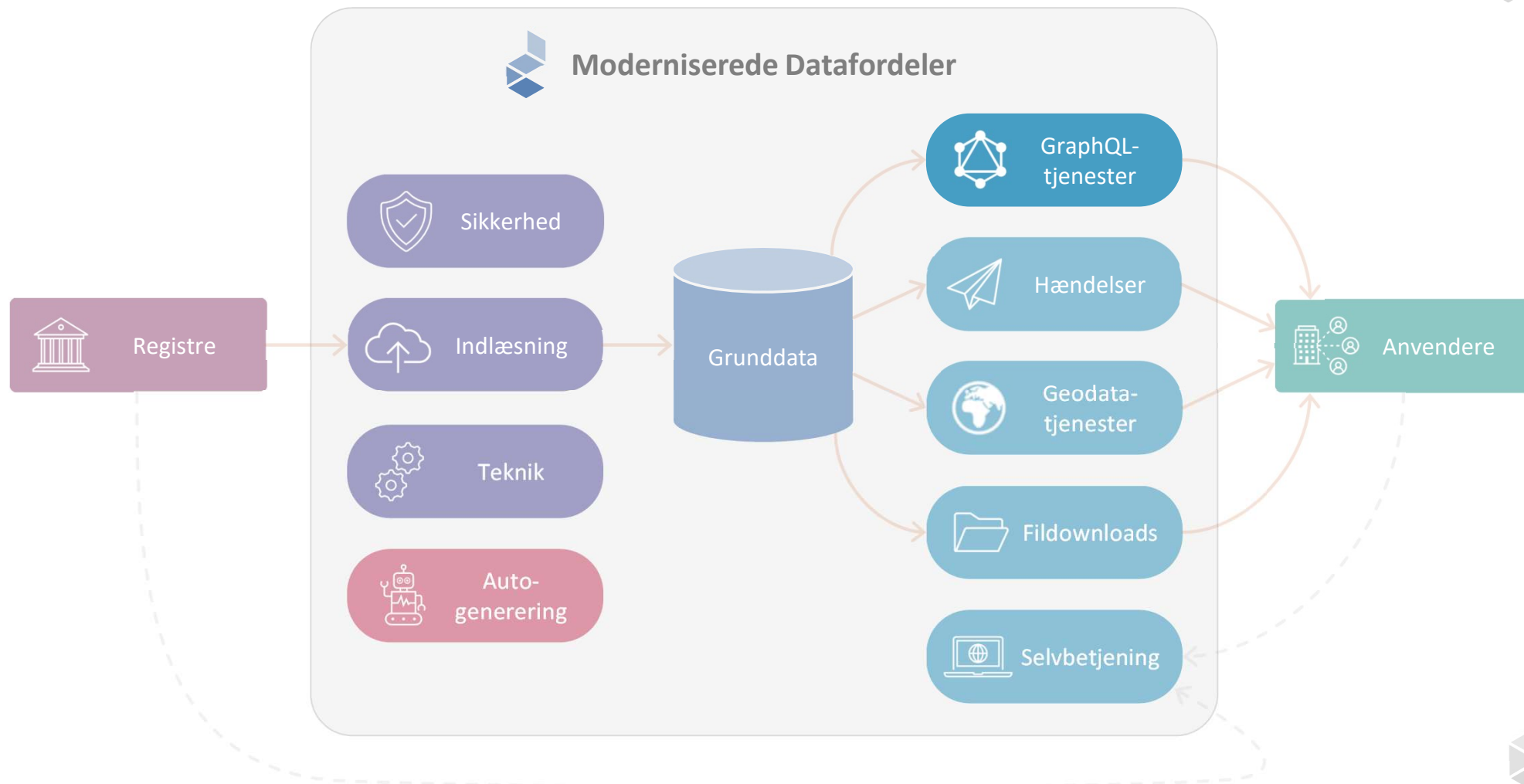
Fleksibel opslagslogik til GraphQL-tjenester

v. Thomas Krohn, Product Owner KDS

v. Marcus Norton Quistgaard, Team Lead Netcompany



Funktionelle og tekniske domæner





Gevinster ved GraphQL-tjenester



Nuværende Datafordeler

- ÷ Nuværende REST-tjenester med forskellig logik og struktur fra register til register
- ÷ Komplekst for registre at specificere REST-tjenester via DLS – og høj time-to-market for implementering heraf
- ÷ Meget manuelt arbejde ifm. implementering af nye og ændrede REST-tjenester
- ÷ Mange data, der ellers er på Datafordeleren, er ikke udstillet via REST-tjenester
- ÷ Udstillingens output skemaer følger ikke altid grunddatamodellen



Moderniserede Datafordeler

- + GraphQL-udstilling for alle entiteter, som anvendere har adgang til
- + Samme logik og struktur for opslag af data, uanset register
- + Ingen register-specifikation (DLS) af udstillings-skemaer, tjenesteparametre eller tjenestelogik
- + Autogenerering af implementering af nye og ændrede GraphQL-tjenester, med minimalt manuelt arbejde
- + Anvendere kan nøjes med at hente de data der er brug for
- ! Anvendere skal (som hidtil) forholde til sammenhæng mellem data ud fra grunddatamodellen



Sammenstilling af data inden for og mellem registre

- Fleksibel opslagslogik er en viderebygning af de entitetsbaserede GraphQL-tjenester, der muliggør joins inden for og mellem registre, hvilket giver mulighed for mere komplekse datarelationer og forespørgsler
 - I fleksibel opslagslogik indarbejdes alle relationer, der er defineret i [\[Grunddatamodellen\]](#)



Fleksibel opslagslogik

- Med entitetsbaseret GraphQL har det hidtil været vanskeligt at hente større mængder sammenhængende data ad gangen. Her har man skulle lave flere forskellige kald, for at få samme mængde data som i REST-tjenesterne. Med indførelsen af fleksibel opslagslogik bliver det nu muligt at oversætte REST-tjenester direkte til GraphQL



Fleksibel opslagslogik

- Sammenstillinger er specificeret af registrene i DLS'en i relationsbilaget
- Forespørgsler som inkluderer beskyttede entiteter kræver adgang til samtlige entiteter i forespørgslen
 - Entiteten med de strengeste adgangskrav sætter samlet adgangskrav for hele forespørgslen
 - Adgang søges i portalen



Fleksibel opslagslogik - fortsat

- Tilgås på 2 endepunkter
 - Flexible (<https://graphql.datafordeler.dk/Flexible/{version}/>)
 - Alle queries har krav om filtrering på registreringstid og virkningstid.
 - FlexibleCurrent (<https://graphql.datafordeler.dk/FlexibleCurrent/{version}/>)
 - Alle queries har krav om filtrering på virkningstid. Registreringstid er altid sat til forespørgselstidspunktet.
- Sammensætninger af data mellem entiteter foregår via **left joins**
 - Bitemporal filtrering mellem register-entiteter håndteres, men ikke anden forretningslogik
- Paginering og filtrering fungerer på same måde som i entitetsbaserede GraphQL-tjenester. Det er yderligere muligt at filtrere og paginere på underentiteter.



Fleksibel opslagslogik – Overblik over registre

Register	Flexible	FlexibleCurrent	Adgangsbegrænset
BBR	Ja	Ja	Nej
CPR	Ja (uden joins)	Ja (uden joins)	Ja
CVR	Nej	Ja	Nej (På nær CVRPerson)
DAGI	Ja	Ja	Nej
DAR	Ja	Ja	Nej
DHM Højdekurver	Ja	Ja	Nej
DHM Oprindelse	Ja	Ja	Nej
Danske Stednavne	Ja	Ja	Nej
EBR	Ja	Ja	Nej
EJF	Ja (uden PersonSimpel)	Ja	Ja
Fikspunkter	Ja	Ja	Nej
GeoDanmark Vektor	Ja	Ja	Nej
MAT	Ja	Ja	Nej
SVR	Ja	Ja	Ja
VUR	Ja	Ja	Nej



Fleksibel opslagslogik

Adgangsbegrænsning

- VUR
 - 10 beskyttede felter udeladt fra snitfladen
 - Data udstilles uden adgangsbegrænsning
- CVR
 - CVRPerson er adgangsbegrænset
 - CVRPersonBegraenset er ikke inkluderet i Fleksibel opslagslogik
- CPR, SVR & EJF
 - Ved manglende rettigheder følger tjenesterne GraphQL-over-HTTP specifikationen, og returnerer passende HTTP statuskode og fejlbesked.



Fleksibel opslagslogik - Versionering

- Alle registre versioneres samlet I Fleksibel Opslagslogik
 - Dvs. hvis BBR laver en breaking change I deres datamodel i Fleksibel Opslagslogik, rykkes DAGI også op I version, trods ingen direkte relation mellem registrene.

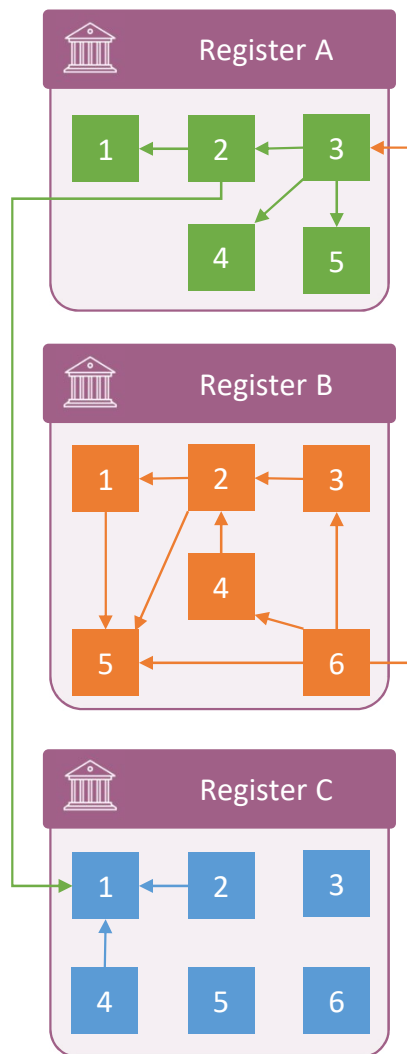
<u>BBR Version</u>	<u>DAGI Version</u>	<u>CVR Version</u>	<u>Flexible Query Logic Services Version</u>
<u>v1</u>	<u>v1</u>	<u>v1</u>	<u>flexible/v1</u> <u>flexibleCurrent/v1</u>
<u>v2</u>	<u>v1</u>	<u>v1</u>	<u>flexible/v2</u> <u>flexibleCurrent/v2</u>
<u>v2</u>	<u>v2</u>	<u>v1</u>	<u>flexible/v3</u> <u>flexibleCurrent/v3</u>
<u>v2</u>	<u>v3</u>	<u>v2</u>	<u>flexible/v4</u> <u>flexibleCurrent/v4</u>



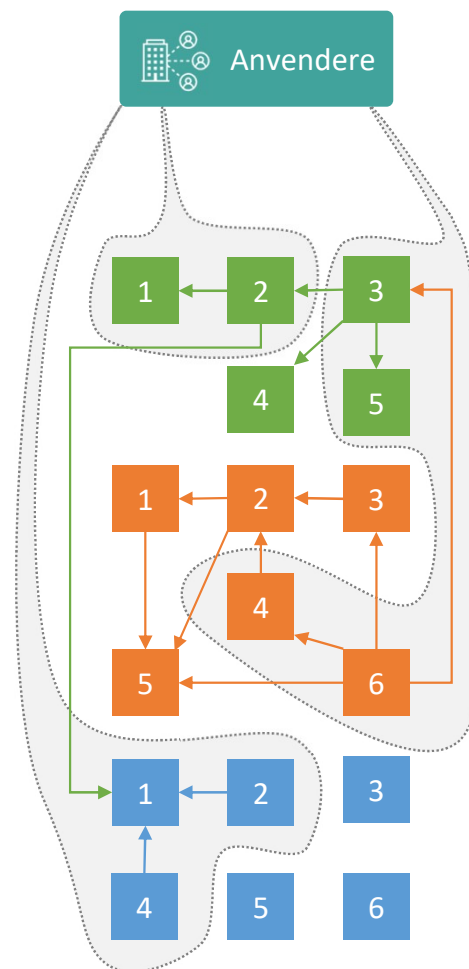
GraphQL-tjenester

Fleksibel
opslagslogik

Register datamodeller



Fleksibel opslagslogik



Funktionalitet i GraphQL- tjenesterne

- Mulighed for at kombinere attributter på tværs af entiteter i ét kald

Et eksempel kunne være:

```
https://graphql.datafordeler.dk/flexible/v1
{
  BBR_Bygning(
    first: 1
    virkningstid: "2024-09-29T16:49:20.251798Z"
    registreringstid: "2024-07-01T22:11:18.251798Z"
  ) {
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    id_lokalid
    status
    bygningHarHusnummer {
      id_lokalid
      status
    }
  }
}
```





GraphQL-tjenester

Fleksibel
opslagslogik

Eksempel på GraphQL query



<https://graphql.datafordeler.dk/flexible/v1>



Query

Register

Hovedentitet

```
{
  DAR_Husnummer
  {
    id_lokalId
    husnummertekst
    postnummer
    kommuneinddeling
    husnummerHarAdgangTilBygning
    {
      byg007Bygningsnummer
    }
  }
}
```

Felter på hovedentitet

Join til BBR_Bygning

Felter på underentitet



Resultat

```
{
  "data": {
    "DAR_Husnummer": [
      {
        "id_lokalId": "2b190e33-7f85-3634-e044-0003ba298018",
        "husnummertekst": "25",
        "postnummer": "d92cd14b-c832-45d7-a465-c93dfe1a3a85",
        "kommuneinddeling": "389142",
        "husnummerHarAdgangTilBygning":
          {
            "byg007Bygningsnummer": 3
          }
      },
      ...
    ]
  }
}
```

NOTE: Viste query er simplificeret

Visualisering af forskel mellem GraphQL- og REST-tjenester



Nuværende Datafordeler

`https://services.datafordeler.dk/BBR/BBRPublic/1/REST/bygning?
nord=6147800 & syd=6147880 & oest=530500 & vest=530510 & status=6`

```
[  
  {  
    "datafordelerOpdateringstid": "2021-04-02T04:25:59.562061+02:00",  
    "byg007Bygningsnummer": 21,  
    "byg021BygningensAnvendelse": "421",  
    "byg026Opførelsesår": 1960,  
    "byg032YdervæggensMateriale": "1",  
    "byg033Tagdækningsmateriale": "3",  
    "byg037KildeTilBygningensMaterialer": "1",  
    "byg038SamletBygningsareal": 65,  
    "byg040BygningensSamledeErhvervsAreal": 65,  
    "byg041BebyggetAreal": 65,  
    "byg053BygningsarealerKilde": "1",  
    "byg054AntalEtagér": 1,  
    "byg404Koordinat": "POINT(530502.29 6147876.88)",  
    "husnummer": "0a3f5090-1981-32b8-e044-0003ba298018",  
    "id_lokalId": "0ea64718-1182-4a1a-be1a-4fbe293ca5ce",  
    "kommunekode": "0621",  
    ... }  
]
```



Moderniserede Datafordeler

`https://graphql.datafordeler.dk/Flexible/v1?query={BBR_Bygning(whe
re: status: { eq: "6" }
byg404koordinat: { StWithin: "POLYGON((530500 6147880, 530510
6147880, 530510 6147800, 530500 6147800, 530500 6147880))" }
}) {
 byg007Bygningsnummer, byg026Opførelsesår, byg404Koordinat,
 husnummer, id_lokalId, kommunekode
}}`

```
{  
  "data": {  
    "BBR_Bygning": {  
      {  
        "byg007Bygningsnummer": 21,  
        "byg026Opførelsesår": 1960,  
        "byg404Koordinat": "POINT(530502.29 6147876.88)",  
        "husnummer": "0a3f5090-1981-32b8-e044-0003ba298018",  
        "id_lokalId": "0ea64718-1182-4a1a-be1a-4fbe293ca5ce",  
        "kommunekode": "0621"  
      }  
    }  
  }  
}
```

NOTE: Viste query er simplificeret



GraphQL-tjenester

Fleksible GraphQL-tjenester

Udstilling af registermetadata - Skema



<https://graphql.datafordeler.dk/Flexible/v1/schema?>

<https://graphql.datafordeler.dk/FlexibleCurrent/v1/schema?>

1
Navn
Definition
URI
Kilde



Navn:	byg007Bygningsnummer
URI:	https://data.gov.dk/model/profile/bbr/byg007Bygningsnummer
Foretrukken term:	Bygningsnummer
Definition:	angiver bygningens nummer indenfor ejendommen
Kommentar:	Løbenummer under ejendommen. Værdierne 1-9999 er tilladte
Juridisk kilde:	https://www.retsinformation.dk/eli/ta/2019/797
Kilde:	https://instruks.bbr.dk/instruks/0/30
Multiplicitet:	1
Type:	Integer
Navn:	byg021BygningensAnvendelse
URI:	https://data.gov.dk/model/profile/bbr/byg021BygningensAnvendelse
Foretrukken term:	Bygningens anvendelse
Definition:	angiver bygningens hovedanvendelse
Kommentar:	Bygningens hovedanvendelse. Har en bygning flere anvendelser registreres den anvendelse som hovedparten af bygningens areal har.



Fleksibel Opslagslogik – Eksempel på anvendelse

- Anvender har behov for at vide hvilke kommuner som vejen "Østjyske Motorvej" går igennem





Fleksibel Opslagslogik - Eksempel



Query

```
{
  DAR_NavngivenVej(
    first: 1
    where: { vejnavn: { eq: "Østjyske Motorvej" }, status: { eq: "3" } }
    virkningstid: "2025-01-01T01:01:01Z"
    registreringstid: "2025-01-01T01:01:01Z"
  ){
    nodes {
      vejnavn
      status
      navngivenVejBestaarAfNavngivenVejKommunedel{
        nodes {
          kommune
          status
        }
      }
    }
  }
}
```



Resultat

```
{
  "data": {
    "DAR_NavngivenVej": {
      "nodes": [
        {
          "vejnavn": "Østjyske Motorvej",
          "status": "3",
          "navngivenVejBestaarAfNavngivenVejKommunedel": {
            "nodes": [
              {
                "kommune": "0751",
                "status": "3"
              },
              {
                "kommune": "0766",
                "status": "3"
              },
              {
                "kommune": "0630",
                "status": "3"
              },
              {
                "kommune": "0710",
                "status": "3"
              },
              {
                "kommune": "0607",
                "status": "3"
              }
            ]
          }
        }
      ]
    }
  }
}
```



Fleksibel Opslagslogik – Paginering / filtrering



Query

```
{
  BBR_Grund(
    first: 1
    where: { id_lokalId: { eq: "347ffdde-de10-427d-bbd7-02f8afd7a0ef" } }
    virkningstid: "2000-02-05T20:31:05Z"
    registreringstid: "2025-01-01T01:01:01Z"
  ){
    pageInfo {
      hasNextPage
      hasPreviousPage
      startCursor
      endCursor
    }
    nodes {
      gru009Vandforsyning
      gru010Aflobsforhold
      grundSamletFastEjendom {
        first: 1
        where: { id_lokalId: { eq: "291a660b-e2d7-4e46-b0b9-d723af787444" } }
        pageInfo {
          hasNextPage
          hasPreviousPage
          startCursor
          endCursor
        }
        nodes {
          bfeNummer
          ejendomsnummer
          ejendomstype
          id_lokalId
          kommunekode
          samletFastEjendom
          status
        }
      }
    }
  }
}
```



Resultat

```
{
  "data": {
    "BBR_Grund": {
      "pageInfo": {
        "hasNextPage": true,
        "hasPreviousPage": false,
        "startCursor": "TkRBd09EazVOeIF3Tmc9PTt",
        "endCursor": "TkRBd09EazVOeIF3Tmc9PTt"
      },
      "nodes": [
        {
          "gru009Vandforsyning": "1",
          "gru010Aflobsforhold": "10",
          "grundSamletFastEjendom": {
            "pageInfo": {
              "hasNextPage": true,
              "hasPreviousPage": false,
              "startCursor": "TkRBd09EazVOeIF3Tmc9PQ==",
              "endCursor": "TkRBd09EazVOeIF3Tmc9PQ=="
            },
            "nodes": [
              {
                "bfeNummer": 6022778,
                "ejendomsnummer": 455904,
                "ejendomstype": "1",
                "id_lokalId": "291a660b-e2d7-4e46-b0b9-d723af787444",
                "kommunekode": "0101",
                "samletFastEjendom": "6022778",
                "status": "7",
              }
            ]
          }
        }
      ]
    }
  }
}
```

CPR tjenester i den Moderniserede Datafordeler

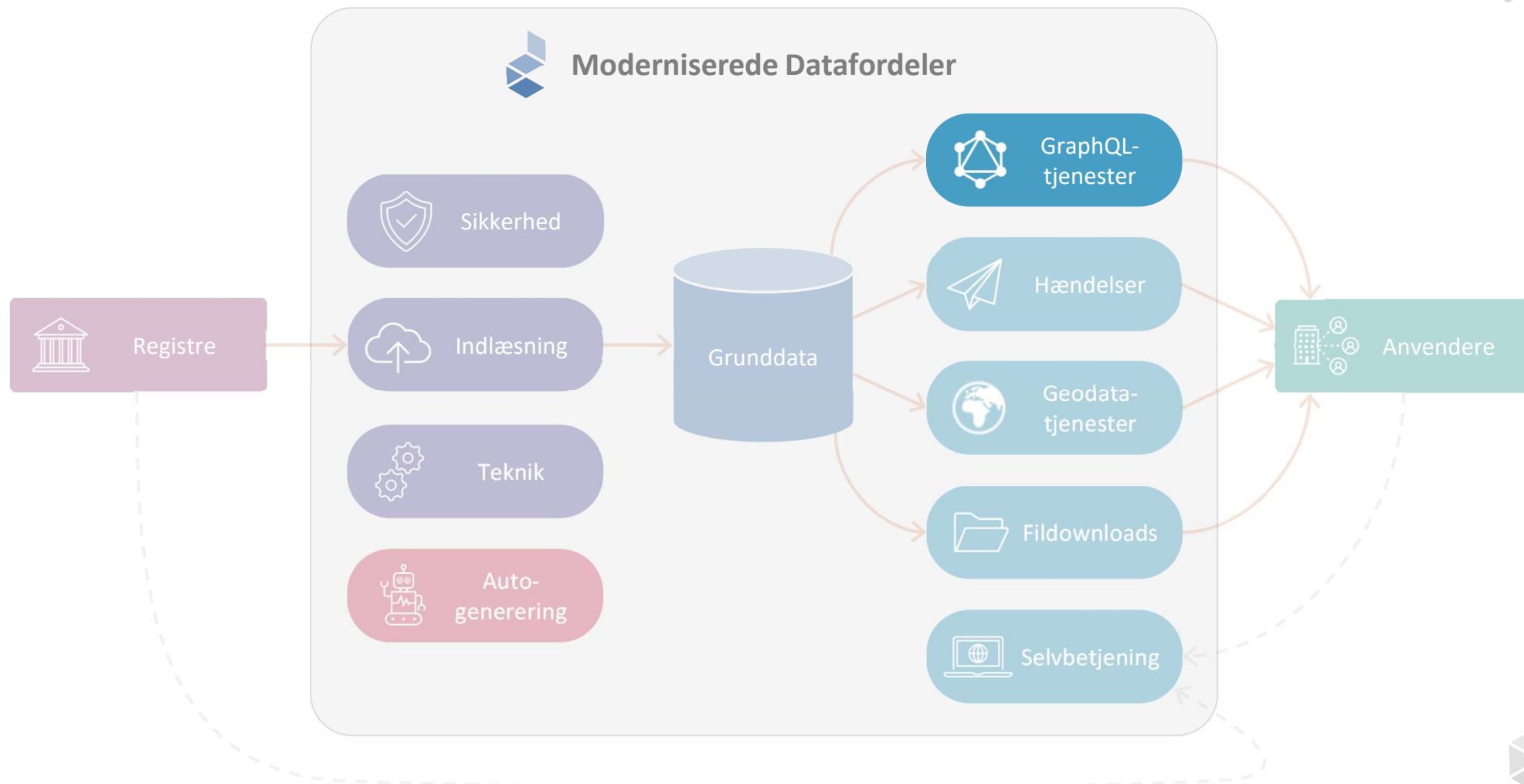
Specielle GraphQL-tjenester til CPR2

v. Kristian Kjølstad Jørgensen, Product Owner KDS

v. Marcus Norton Quistgaard, Team Lead Netcompany



Funktionelle og tekniske domæner





CPR tjenester



Nuværende Datafordeler

- ÷ En række forskellige REST-tjenester med forskellige mængder af data
- ÷ Statisk mængde af data ved hvert kald
- ÷ Understøtter CPR datamodel



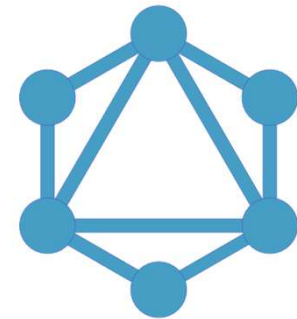
Moderniserede Datafordeler

- + Kun to GraphQL-tjenester:
 - Én til den **offentlige sektor**: CPRPublicSector
 - Én til den **private sektor**: CPRPrivateSector
- + Anvender angiver selv hvilke data der returneres, hvilket gør dataminimering nemmere
- + Understøtter CPR2 datamodel



Specielle GraphQL-tjenester til CPR

- Nuværende Datafordeler: En række **REST-tjenester**, der returnerer forskellige mængder statiske CPR-data.
- Moderniseret Datafordeler (UL3): To **GraphQL-tjenester**:
 - ⇒ én til den offentlige sektor (CPRPublicSector), fremsøge flere personer på én gang.
 - ⇒ én til den private sektor (CPRPrivateSector), som returnerer de CPR-data, anvenderen selv angiver, hvilket gør det nemmere for anvenderen at dataminimere





To specielle GraphQL-tjenester til CPR: PublicSector og PrivateSector



CPR PublicSector

- Tilgås på:
<https://graphql.datafordeler.dk/CPR/custom/PublicSector/<version>>
- Schema'et for tjenesten hentes på ovenstående URL med **/schema** til sidst
- Man kan fremsøge flere personer samtidig
- Kan tage en række søgeparametre (inkl. lister):
 - Id, status, fødselsdato, statusdato
 - Adresseoplysninger
 - Navn
 - Personnummer
 - Statsborgerskab
 - CPR-hændelser og CPR-hændelseskoder
- Data er beskyttet og kræver dataadgang til CustomPublicSectorPerson
- Autentificering kun ved OAuth flow



CPR PrivateSector

- Tilgås på:
<https://graphql.datafordeler.dk/CPR/custom/PrivateSector/<version>>
- Schema'et for tjenesten hentes på ovenstående URL med **/schema** til sidst
- Man kan kun fremsøge én person ad gangen
- Personer skal fremsøges på en af følgende måder:
 - Personnummer
 - Navn + fødselsdato
 - Navn + adresse
- Returnerer kun aktuel data
- Data er beskyttet og kræver dataadgang til CustomPrivateSectorPerson
- Autentificering kun ved OAuth flow



GraphQL-tjenester

Sammenstillede
GraphQL-tjenester
fundament og PoC

GraphQL query – CPR PublicSector

<https://graphql.datafordeler.dk/CPR/custom/PublicSector/v1?query=>



Query

```
query {  
  CPRCustom_PublicSectorPerson(  
    first: 3  
    input: {  
      navn: { fornavne: { eq: "John" }, efternavn: { eq: "Doe" } }  
    }  
  ) {  
    nodes {  
      id  
      navne {  
        fornavne  
        mellemnavn  
        efternavn  
      }  
      adresseoplysninger {  
        cprAdresse {  
          daradresse  
          ...  
        }  
      }  
    }  
  }  
}
```



Resultat

```
{  
  "data": {  
    "CPRCustom_PublicSectorPerson": {  
      "nodes": [  
        {  
          "id": "xxxxxxx-0000-000x-0x00-xx000000x00x"  
          "navne": [  
            {  
              "fornavne": "John"  
              "mellemnavn": null  
              "efternavn": "Doe"  
            }  
          ],  
          "adresseoplysninger": [  
            {  
              "cprAdresse": {  
                "daradresse": "xxxxxxx-0000-000x-0x00-xx000000x00x"  
                ...  
              }  
            }  
          ]  
        }  
      ]  
    }  
  }  
}
```



GraphQL-tjenester

Sammenstillede
GraphQL-tjenester
fundament og PoC

GraphQL query – CPR PrivateSector

<https://graphql.datafordeler.dk/CPR/custom/PrivateSector/v1?query=>



Query

```
query {  
  CPRCustom_PrivateSectorPerson(  
    input: {  
      navn: { fornavne: { eq: "John" }, efternavn: { eq: "Doe" } }  
      adresse: {  
        postnr: 1550,  
        vejnavn: "Rådhuspladsen",  
        husnr: "1"  
      }  
    }  
  ) {  
    navn {  
      fornavne  
      mellemnavn  
      efternavn  
    }  
    adresseoplysninger {  
      cprAdresse {  
        daradresse  
      }  
    }  
  }  
}
```



Resultat

```
{  
  "data": {  
    "CPRCustom_PrivateSectorPerson": [  
      {  
        "navn": {  
          "fornavne": "John"  
          "mellemnavn": null  
          "efternavn": "Doe"  
        },  
        "adresseoplysninger": {  
          "cprAdresse": {  
            "daradresse": "xxxxxxx-0000-000x-0x00-xx000000x00x"  
          }  
        }  
      }  
    ]  
  }  
}
```

EJF tjenester i den Moderniserede Datafordeler

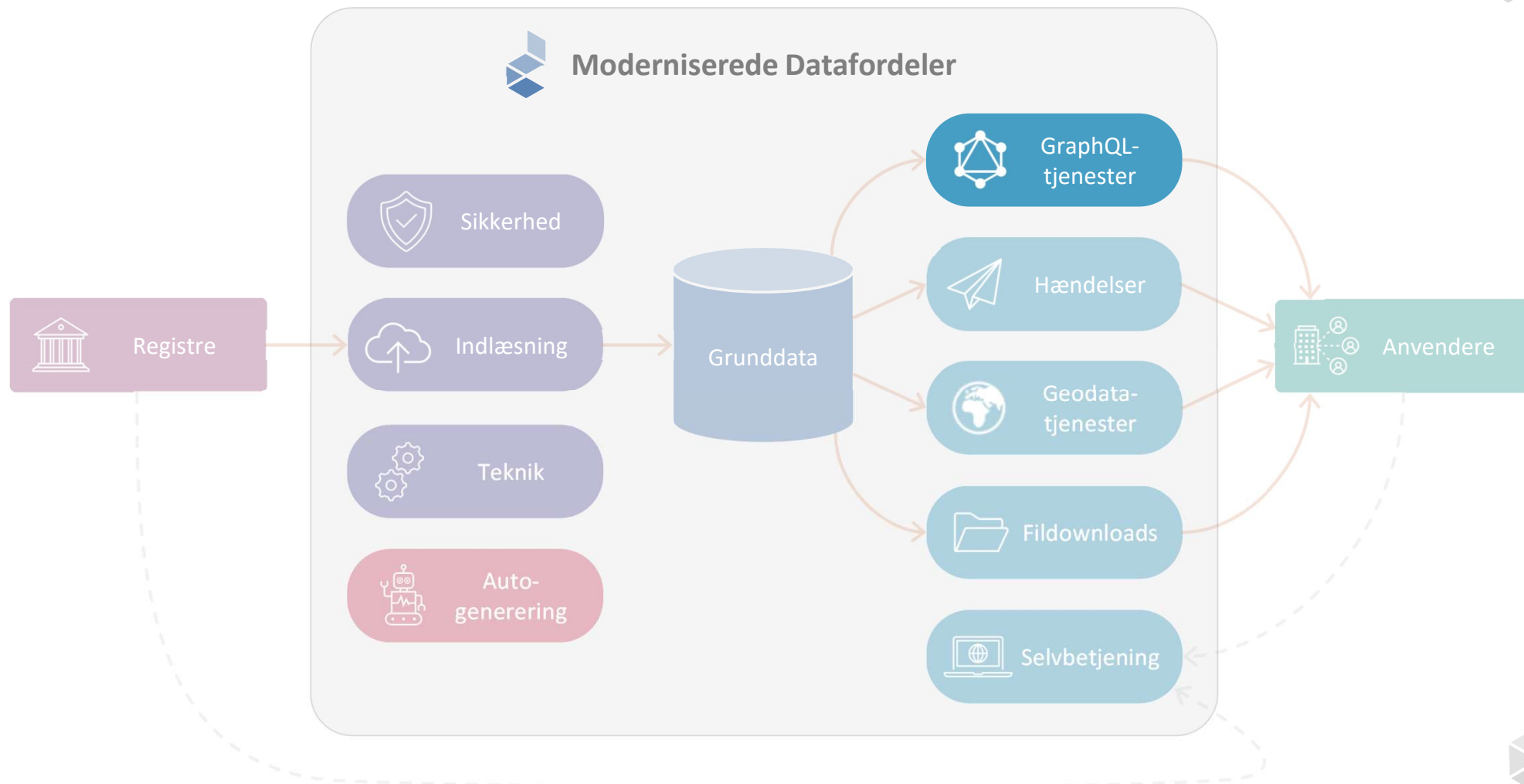
Specielle GraphQL-tjenester til EJF

v. Thomas Krohn, Product Owner KDS

v. Marcus Norton Quistgaard, Team Lead Netcompany



Funktionelle og tekniske domæner





EJF-tjenester

- Der udvikles speciel-tjenester til EJF som kan bruges til at hente relevante CPR-data (Stamoplysninger)
- Tiltænkt anvendere som har hjemmel til at se CPR data – men ikke se CPR-nummer
- Tager højde for CPR's regler vedr. navne- og adressebeskyttelse
- Tjenesterne indgår i Fleksibel Opslagslogik og kan dermed sammenstilles med fx CVR/MAT/DAR



EJF-tjenester

- Data for speciel-tjenesterne til EJF er kun tilgængelige for brugere, der har fået eksplicit adgang til dem
- Der søges rettigheder inde i portalen og det er EJF der giver adgang



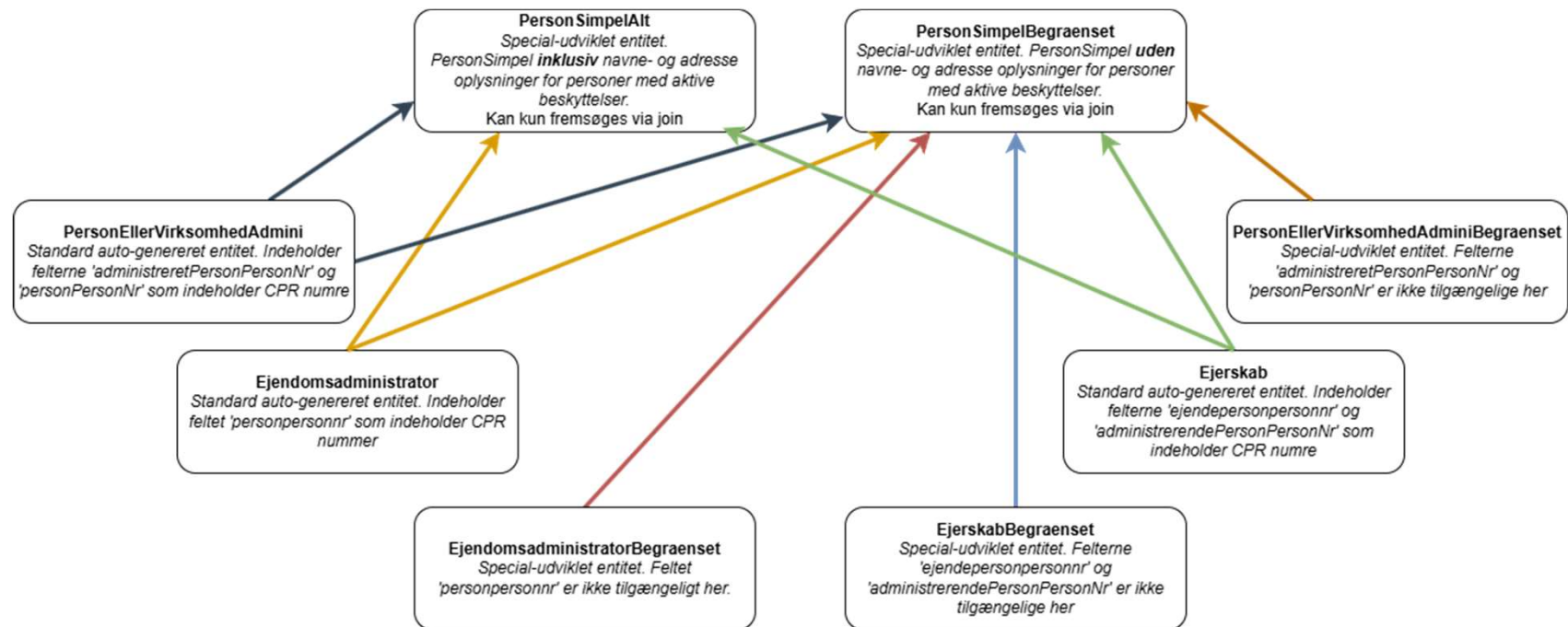
EJF-tjenester – Sammenligning mellem REST og GraphQL

Ejerfortegnelsen	GraphQL-entiteter
EjendomsadministratorMedStamoplysninger	EjendomsadministratorBegraenset + PersonSimpelBegraenset
PersonVirksomhedsAdminiMedStamoplysninger	PersonVirksomhedsAdminiBegraenset + PersonSimpelBegraenset
EjerskabMedStamoplysninger	EjerskabBegraenset + PersonSimpelBegraenset

EjerfortegnelsenFortrolig	GraphQL-entiteter
EjendomsadministratorMedStamoplysninger	Ejendomsadministrator + PersonSimpelBegraenset
PersonVirksomhedsAdminiMedStamoplysninger	PersonVirksomhedsAdmini + PersonSimpelBegraenset
EjerskabMedStamoplysninger	Ejerskab + PersonSimpelBegraenset

EjerfortegnelsenFortroligBeskyttet	GraphQL-entiteter
EjendomsadministratorMedStamoplysninger	Ejendomsadministrator + PersonSimpelAlt
PersonVirksomhedsAdminiMedStamoplysninger	PersonVirksomhedsAdmini + PersonSimpelAlt
EjerskabMedStamoplysninger	Ejerskab + PersonSimpelAlt

EJF-tjenester - Overblik



Hver pil repræsenterer et eller flere tilladte joins i Fleksibel Opslagslogik snitfladen som anvender kan vælge at hente data fra, såfremt de har fået tildelt de nødvendige rettigheder.

Join udføres på baggrund af EJF entitetens CPR-nummer felt(er) samt de af-EJF specificerede forretningsregler for PersonSimpel



EJF-tjenester - Rettigheder

GraphQL entitet	Indeholder CPR-nummer	Indeholder beskyttede navne og adresser	Kan fremsøges direkte
Ejendomsadministrator	Ja	Nej	Ja
PersonVirksomhedsAdmini	Ja	Nej	Ja
Ejerskab	Ja	Nej	Ja
PersonSimpelAlt	Nej	Ja	Nej
EjerskabBegraenset	Nej	Nej	Ja
PersonVirksomhedsAdminiBegraenset	Nej	Nej	Ja
EjendomsadministratorBegraenset	Nej	Nej	Ja
PersonSimpelBegrasenset	Nej	Nej	Nej

KORT PAUSE

-5 minutter-



Entitetsbaserede Hændelser

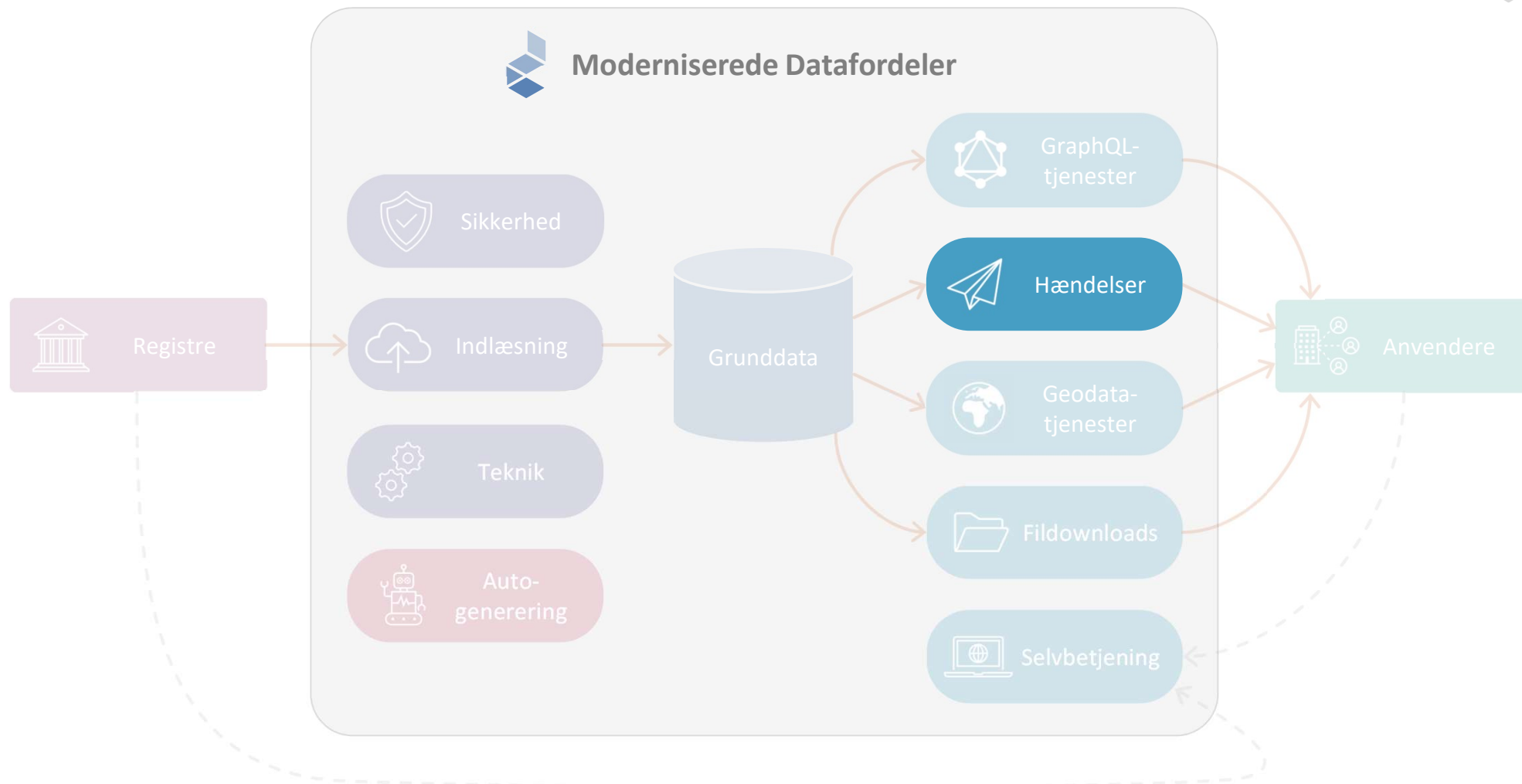
v. Thomas Krohn, Product Owner KDS

v. Marcus Norton Quistgaard, Team Lead Netcompany



DATAFORDELER

Funktionelle og tekniske domæner





Introduktion til hændelser

- Dataopdateringer fra registre -> hændelser
- Ved at bruge hændelser, kan anvendere hente opdateringer, og holde egne data synkroniseret med de nyeste ændringer
 - For eksempel hvis en virksomhed skifter navn, vil anvendere få en hændelsesbesked, hvorefter de kan opdatere deres kopiregister
- Der kan være forskellige scenarier for, hvordan en anvender ønsker at modtage hændelser. Har man ikke behov for at modtage opdateringer med det samme, kan man hente hændelser med GraphQL queries
- Har man derimod behov for at modtage hændelser i nær realtid, kan man i stedet abonnere på hændelser med GraphQL subscriptions



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 register-specifikation (DLS) af hændelsesstrukturer eller felt-mapninger
- + Autogenerering af implementering af nye og ændrede hændelser, med minimalt manuelt arbejde
- + Ingen PUSH-hændelser, men i stedet GraphQL subscription som kan bruges til samme formål



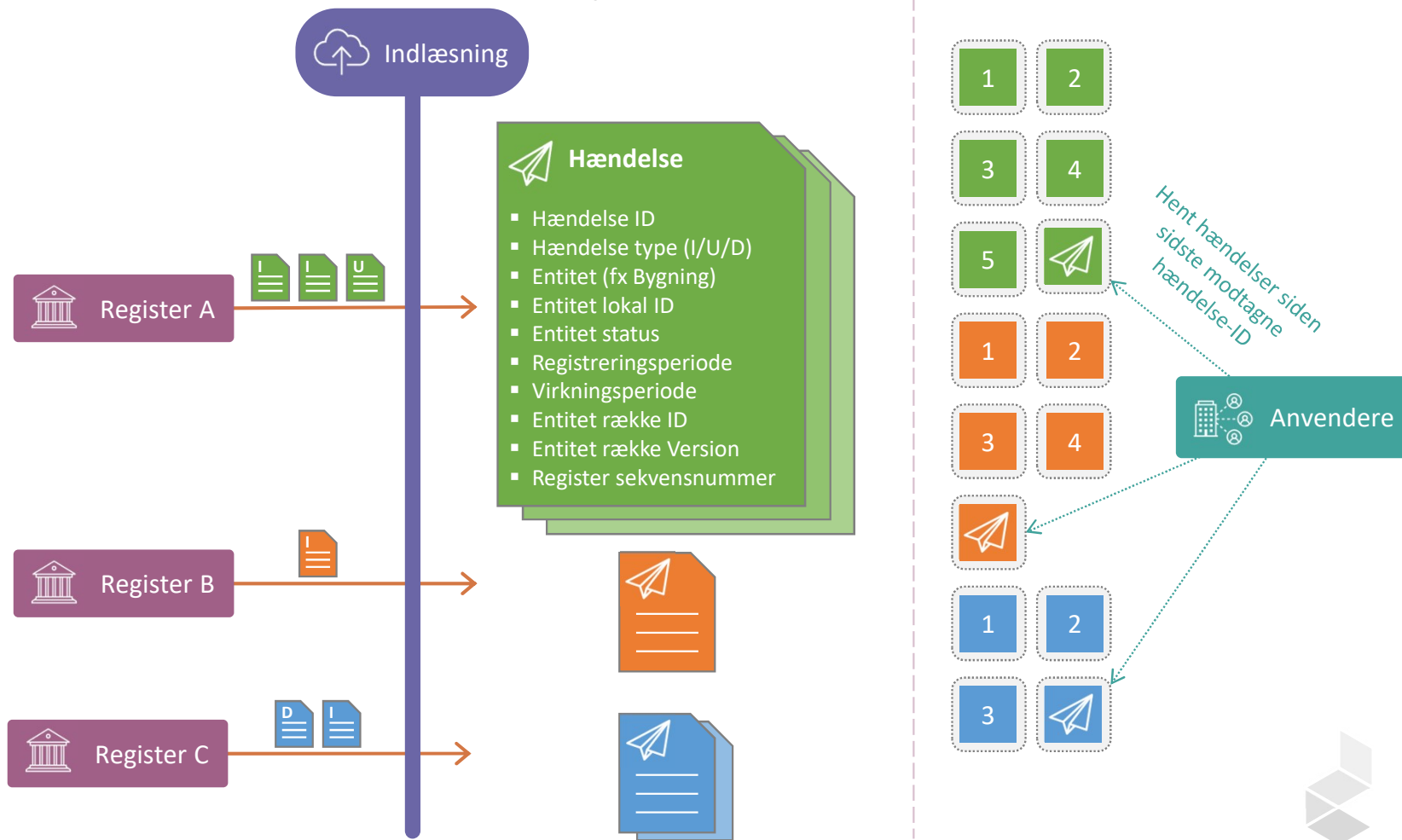
Hændelser

Entitetsbaserede
hændelser

Dataopdateringer
fra registre

Data opdateres i
registers entitetstabel og
datahandling tilføjes til
registers hændelsestabel

Hændelsestabel udstilles via
GraphQL-tjenester, tilsvarende
til registerentiteter





Nye kolonner for alle register entiteter



- 3 nye kolonner er tilføjet til alle register-entiteter
 - F.eks: BBR_Bygning, CVR_Navn
- Disse bruges for at kunne identificere rækker og opdateringer der er sket
- Udstilles i GraphQL og nye versioner af fildownloads

Attribut	Type	Beskrivelse
datafordelerRowId	Guid	Unik identifikator for en række.
datafordelerRowVersion	Int	Rækkeversion. Begynder på 1 og øger hver gang rækken bliver opdateret.
datafordelerRegisterImportSequenceNumber	Int	Sekvensnummer for det dataindlæsningspakke der senest opdateret rækken.



Ny entitet: Events

- Hændelser udstilles med en ny entitet der udstilles under hvert registers endpoint
 - F.eks.: BBR_Events
- Indeholder data for hændelsen og data fra den relaterede række
 - Data fra den relaterede række har et "object_" præfiks



Attribut	Type
eventId	Int
eventAction	String
entityName	String
datafordelerRegisterImportSequenceNumber	Int
fromFailedImport	Bool
object_datafordelerRowId	Guid
object_datafordelerRowVersion	Int
object_id	String
object_status	String
object_registreringfra	DateTime
object_registreringtil	DateTime
object_virkningfra	DateTime
object_virkningtil	DateTime

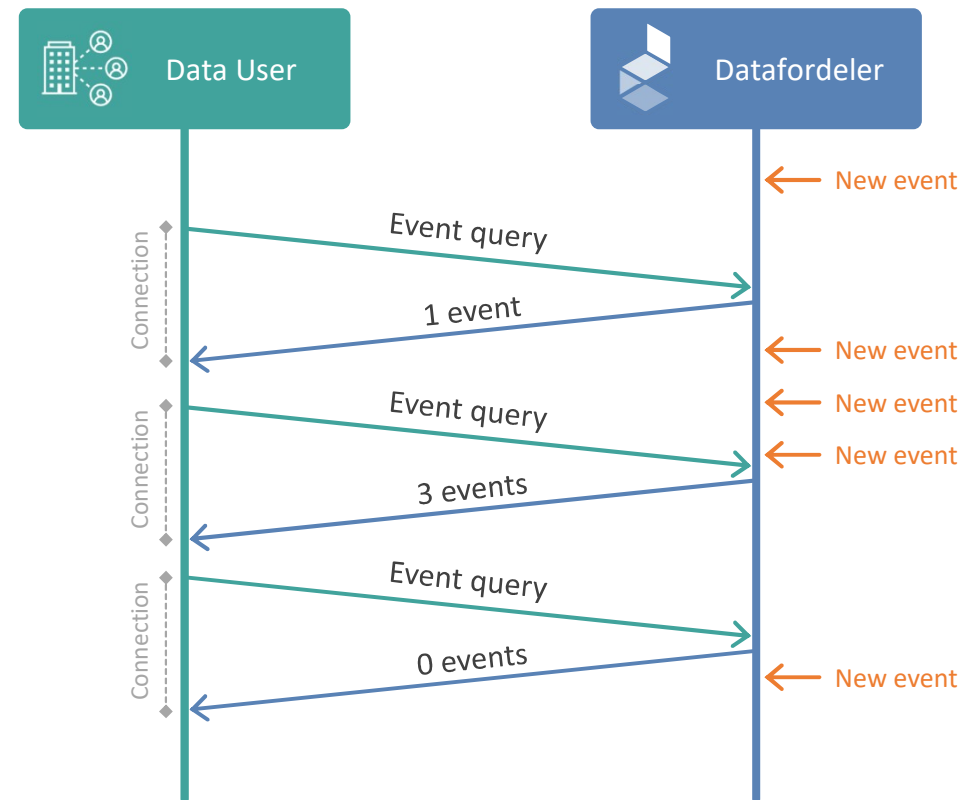


Get events: GraphQL query

- For at hente nye hændelser kan man enten filtrere på:
 - Det sidste kendte "eventid" eller
 - Hændelser oprettet efter en sat dato med "datafordelerOpdateringstid"
- Hvilket giver det flow der vises på billedet



GraphQL query flow





Brugerscenarie – Brugere anvender Query for at modtage hændelser

Brugerne ønsker at modtage hændelser fra BBR_Bygning for at få information omkring dataændringer til bygning:

1. Brugerne laver en GraphQL Query for at modtage eventuelle hændelser
2. Brugerne henter opdateret registerdata, via GraphQL for BBR_Bygning og kan efterfølgende opdatere lokal data
3. Brugerne notificerer sine egne anvendere, at opdateringer er sket

Brugerscenarie, trin 1: GraphQL query til hændelsetabel

```
query Demo {  
  BBR_Events(where: { eventid: { gt: 33 }, entityname: { eq: "Bygning" } }) {  
    nodes {  
      entityname  
      eventaction  
      eventid  
      object_id  
      object_datafordelerRowId  
      object_registreringfra  
      object_registreringtil  
      object_virkningfra  
      object_virkningtil  
    }  
    pageInfo {  
      hasNextPage  
      endCursor  
    }  
  }  
}
```

```
2  
3  "data": {  
4    "BBR_Events": {  
5      "nodes": [  
6        {  
7          "entityname": "Bygning",  
8          "eventaction": "u",  
9          "eventid": 1177313,  
10         "object_id": "85209566-b4a4-4769-a016-13f4b7d4ce1d",  
11         "object_datafordelerRowId": "4c91b626-e30a-467b-a54a-62600b707e9d",  
12         "object_registreringfra": "2023-02-28T07:54:30.700555Z",  
13         "object_registreringtil": "2023-10-23T04:53:19.487369Z",  
14         "object_virkningfra": "2023-02-28T07:54:30.700555Z",  
15         "object_virkningtil": null  
16       },  
17       {  
18         "entityname": "Bygning",  
19         "eventaction": "i",  
20         "eventid": 1177314,  
          "object_id": "85209566-b4a4-4769-a016-13f4b7d4ce1d",
```

Eksempel på GraphQL kald for at hente hændelser for BBR, filtreret til kun at returnere hændelser for Bygning, med event id højere end 33



Brugerscenarie, trin 2: GraphQL query til register



```
query {  
  BBR_Bygning(  
    where: { datafordelerRowId: { eq: "4c91b626-e30a-467b-a54a-62600b707e9d" } }  
  ) {  
    nodes {  
      datafordelerRowVersion  
      datafordelerRowId  
      datafordelerRegisterImportSequenceNumber  
      husnummer  
      id_lokalId  
      kommunekode  
    }  
  }  
}
```

```
2  
3  "data": {  
4    "BBR_Bygning": {  
5      "nodes": [  
6        {  
7          "datafordelerRowVersion": 2,  
8          "datafordelerRowId": "4c91b626-e30a-467b-a54a-62600b707e9d",  
9          "datafordelerRegisterImportSequenceNumber": 3453517,  
10         "husnummer": "0a3f5084-bc3b-32b8-e044-0003ba298018",  
11         "id_lokalId": "85209566-b4a4-4769-a016-13f4b7d4ce1d",  
12         "kommunekode": "0306"  
13       }  
14     ]  
15   }  
}
```

Eksempel på GraphQL kald for at hente det opdaterede data fra BBR_Bygning, med datafordelerRowId fra hændelsequery i tidligere slide



Get events: GraphQL subscription



- Det er også muligt at abonnere på nye hændelser
- Sender automatisk nye hændelser i nær realtid
- Også i GraphQL, med samme syntax
 - Dog uden paginering da én hændelse bliver returneret ad gangen

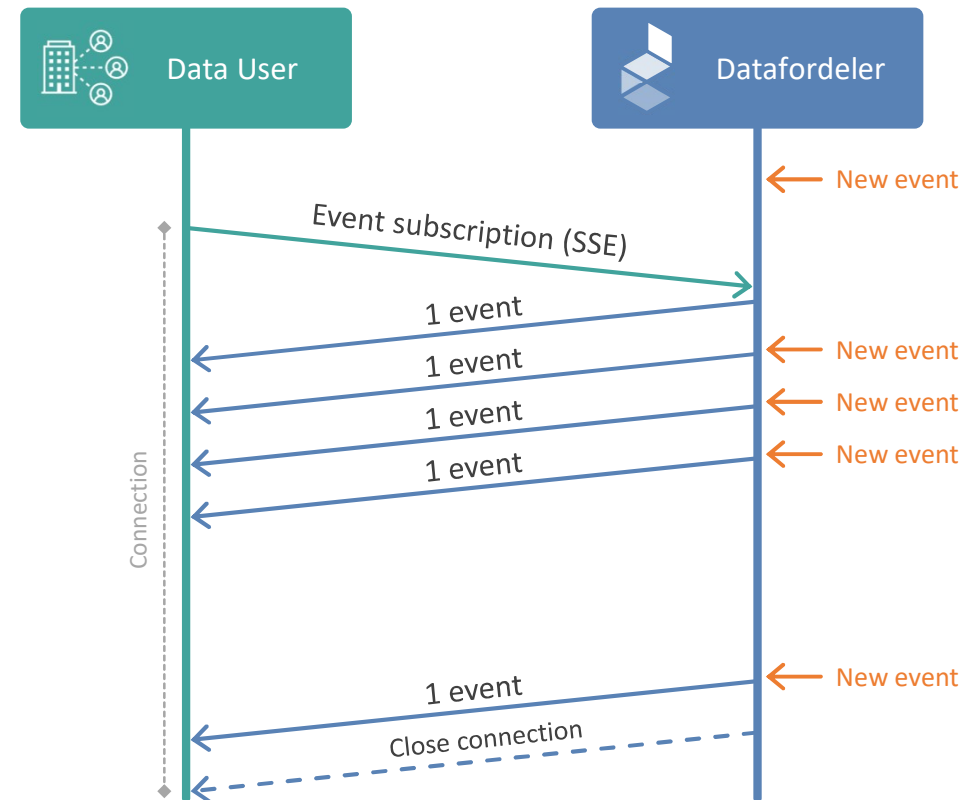
```
subscription Test {  
  BBR_Events(where: { entityname: { eq: "Bygning" } }) {  
    entityname  
    eventaction  
    object_datafordelerRowId  
    object_id  
  }  
}
```



Get events: GraphQL subscription

1. Anvenderen opretter en forbindelse ved at sende sin GraphQL request
2. En stream oprettes så anvenderen kan modtage data (via Server Sent Events)
3. Nye hændelser bliver sendt ud i nær realtid mens forbindelsen er åben
4. Forbindelsen lukkes efter 10 min
5. Anvenderen kan oprette en nye forbindelse efter den er lukket og heraf i praksis have en konstant åben forbindelse

GraphQL subscription With Server-Sent Event (SSE)





Ny entitet: Register Import Status



- En anden tabel der udstilles er Register Import Status
- Indeholder information om det seneste indlæsningspakke der er indlæst for et register
- Opdateres efter en hel pakke er indlæst og kan således bruges for at sikre dataintegritet
 - Eller for at vide hvis ny data er blevet indlæst

Attribut	Type
lastSequenceNumber	Int
lastEventId	Long
lastUpdated	DateTime

Attribut for den nye Register Import Status-tabel der udstilles i GraphQL



Hente data: Register Import Status

- Register Import Status udstilles under register endpoints
- Status for BBR tilgås altså af at query "DAF_RegisterImportStatus" under BBR endpoint
- Tillader både queries og subscriptions

```
query Test {  
  DAF_RegisterImportStatus {  
    lastEventId  
    lastSequenceNumber  
    lastUpdated  
  }  
}
```

Query

```
subscription Test {  
  DAF_RegisterImportStatus {  
    lastEventId  
    lastSequenceNumber  
    lastUpdated  
  }  
}
```

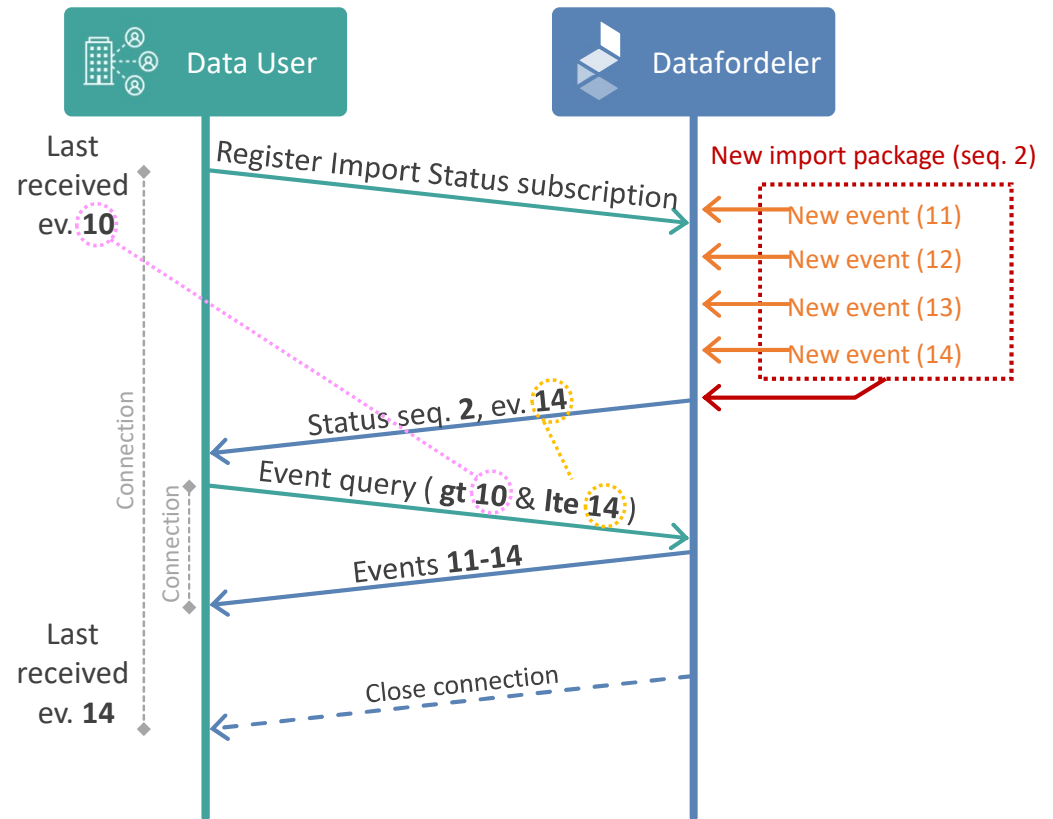
Subscription





Eksempel på anvendelse

- Register Import Status og hændelser kan godt kombineres for at hente nye hændelser når en indlæsningspakke er helt indlæst:
1. Abonner på Register Import Status med en GraphQL subscription
 2. Når ny data er modtaget for Register Import Status er en datapakke indlæst
 3. Hent nye hændelser fra det seneste eventid "lasteventid" fra Register Import Status med en GraphQL query
 4. Kan nu sikre at all data findes fra den sidste indlæsningspakke





Kombinere med fildownloads



- Hændelser kan også bruges med fildownloads
- For at vedligeholde et kopiregister i nær realtid kan man:
 1. Hente hændelser som beskrevet med GraphQL
 - Enten med Queries eller Subscriptions, baseret på hvor ofte man har brug for at opdatere sit kopiregister
 2. Hente opdateret data via GraphQL ved at query den entitet hændelsen er relateret til
 3. Opdatere sit kopiregister med den opdaterede data
- De nye systemkolonner, f.eks. datafordelerRowId, udstilles også i fildownloads og kan bruges for at mappe hændelser til data
 - Kun i nye fildownload versioner, så nuværende anvendere ikke påvirkes



Adgang til hændelser



- Automatisk adgang til at se hændelser, hvis man har adgang til den entitet hændelsen er genereret for
- Undtaget CPR, på grund af deres Custom tjenester, hvor man skal have adgang til CustomPublicSectorPerson for at kunne modtage hændelser for beskyttede CPR-entiteter



Hændelser for CPR



- Hændelser for CPR har 2 ekstra kolonner for kommunekode:
 - object_kommunekodetidligere
 - object_kommunekodenuværende
- Disse kolonner har kun værdier for hændelser for CPRHændelse entiteten
 - Indeholder nuværende **og** tidligere kommunekode
 - object_id mapper til CPR Person ID hvilket kan bruges for CprCustomPublic tjenesten som "ID"-parametre

```
query Test {  
  CPR_Events(  
    where: {  
      object_kommunekodenuvaerende: { eq: "0101" }  
      entityname: { eq: "Cprhaendelse" }  
    }  
  ) {  
    nodes {  
      object_id  
      object_kommunekodetidligere  
      object_kommunekodenuvaerende  
      eventaction  
    }  
  }  
}
```

Eksempel-query med kommunekode

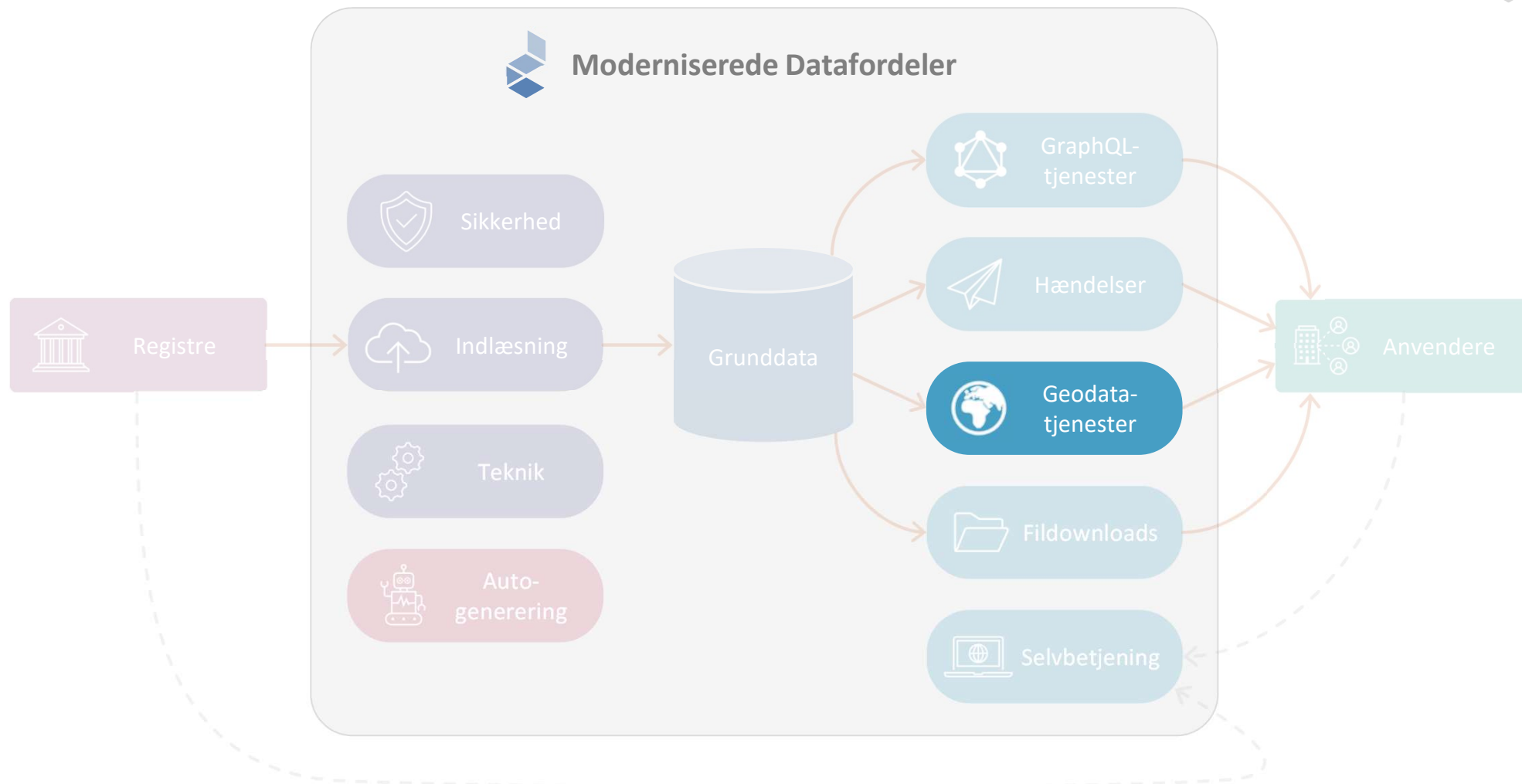
Entitetsbaserede WFS-tjenester

v. Jørgen Petersen, Product Owner KDS

v. Marcus Norton Quistgaard, Team Lead Netcompany



Funktionelle og tekniske domæner





Gevinster ved Entitetsbaserede WFS-tjenester



Nuværende Datafordeler

- ÷ WFS tjenester indeholdt nyttige men få og meget specifikke lag
- ÷ Komplex manuel process for nye tjenester krævede omfattende koordinering mellem registre og udviklere for at specificere og skabe hver tjeneste
- ÷ Kun WFS 1.0.0 og 1.1.0 understøttelse af eksisterende platform - begrænset funktionalitet og ældre protokol
- ÷ Begrænset bitemporal understøttelse for få tjenester



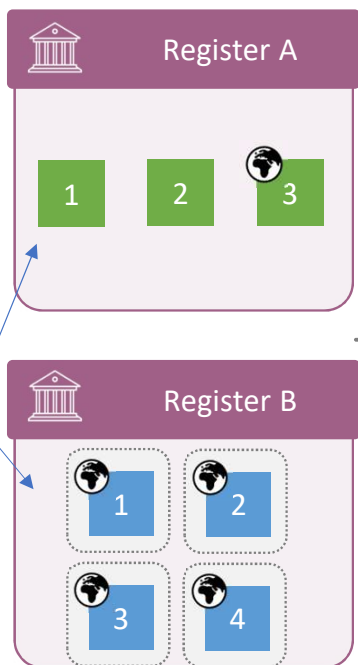
Moderniserede Datafordeler

- + WFS tjenester og lag udstilles for alle registre med alle entiteter som de vælger at udstille
- + Automatiseret og standardiseret generering fra registres ønske om at udstille entiteter til at de bliver tilgængelige for anvendere
- + Standard OGC WFS 2.0.0 som alle GIS-værktøjer understøtter
- + Bitemporal understøttelse indbygget – både current og historiske lag genereres automatisk for alle entiteter
- + Versionering som sikrer bagudkompatibilitet – eksisterende integrationer fortsætter uændret ved ændringer af datamodellen

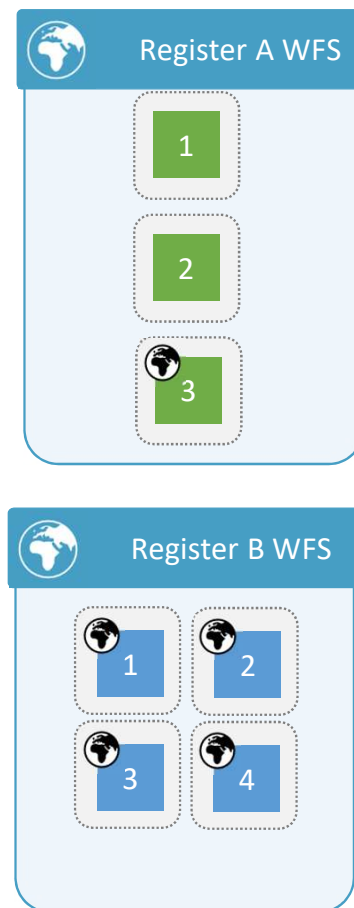
Autogenerering af Entitetsbaserede WFS-tjenester

WFS services

Register
data modeller



Generering af
WFS-tjenester



- Udstilles i 2 temporaliteter:
 - Aktuelle data (current)
 - Bitemporale data (historic)



- 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

 = Entiteter med geometri register vil eksponere som WFS lag





Liste af register der udstiller entitetsbaserede WFS-tjenester



Register	Replikeringskanal	Udstillingstype	Antal entiteter	Specifikke entiteter
DAGI	02620	Alle entiteter	Alle	-
DHMhoejdekurver	00030	Alle entiteter	Alle	-
DHMOprindelse	00038	Alle entiteter	Alle	-
FIKSPUNKT	00055	Alle entiteter	Alle	-
GeoDKV	00279	Alle entiteter	Alle	-
BBR	00018	Selektiv	2	Bygning, TekniskAnlæg
MAT	00336	Selektiv	22	
DAR	00022	Selektiv	1	NavngivenVej



To typer af WFS-tjenester i MoDAF



1. Migrerede fra NuDAF (UL1)

- Ikke standardiseret navngivning. Servicenavn beskriver indhold
- Eksempel: /FIKSPUNKT/FIKSPUNKTER_HIST_GML3SFP/1.0.0/WFS
 - *Indeholder specifikke lag-repræsentationer i hist representation i GML3 format*

2. Entity-baserede (UL3)

- Standardiseret navngivning
- Eksempel: /FIKSPUNKT/FIKSPUNKT_WFS/1.0.0/WFS
 - *Indeholder alle entiteter FIKSPUNKT vil udstille i både current og hist, muligt at tilgå i både GML3 og GeoJSON*

Forskellen:

- **UL1:** Forskellige navne per indhold – lag er håndplukket og service-specifikt
- **UL3:** Ensartet mønster /{REGISTER}/{REGISTER}_WFS/{VERSION}/WFS – alle entiteter tilgængelige gennem samme endpoint



Endpoint struktur for entitesbaserede WFS-tjenester, eksempel



I eksemplet nedenfor bruges:

- Register: BBR
- Service version: 1.0.0
- OGC parametre
- Entitet_bitemporalitet: Bygning_current
- Authentication: API key

[https://wfs.datafordeler.dk/
BBR/BBR_WFS/1.0.0/WFS?Service=WFS&Version=2.0.0&Request=GetFeature&TypeNames=bygning_current&APIKey=ABC](https://wfs.datafordeler.dk/BBR/BBR_WFS/1.0.0/WFS?Service=WFS&Version=2.0.0&Request=GetFeature&TypeNames=bygning_current&APIKey=ABC)

Angivelse af Register, fx
BBR eller DHM

Service version & service
identifier

OGC-parametre

Navn på lag og
bitemporalitet, fx
bygning_Current eller
Bygning_Hist

Autentificering af den
kaldende anvender,
API key eller
OAuthToken



Brugerscenarie, Bruno – Hent alle lag for BBR



GetCapabilities request:

BBR/BBR_WFS/1.0.0/WFS?service=WFS&request=GetCapabilities&apikey=

Query

Name	Path	
service	WFS	☒
request	GetCapabilities	☒
apikey	YOUR_API_KEY	☒

Se tilgængelige lag (entiteter)



```
<FeatureTypeList>
  <FeatureType xmlns:bbr_v001="http://datafordeler.dk/schemas/bbr/v001/gml3">
    <Name>bbr_v001:bygning_current</Name>
    <Title>bygning_current</Title>
    <Abstract>bbr_bygning_v001_current</Abstract>
    <DefaultCRS>urn:ogc:def:crs:EPSG::25832</DefaultCRS>
    <ows:WGS84BoundingBox>
      <ows:LowerCorner>-60.0 54.5</ows:LowerCorner>
      <ows:UpperCorner>15.2 84.0</ows:UpperCorner>
    </ows:WGS84BoundingBox>
  </FeatureType>
  <FeatureType xmlns:bbr_v001="http://datafordeler.dk/schemas/bbr/v001/gml3">
    <Name>bbr_v001:bygning_hist</Name>
    <Title>bygning_hist</Title>
    <Abstract>bbr_bygning_v001_hist</Abstract>
    <DefaultCRS>urn:ogc:def:crs:EPSG::25832</DefaultCRS>
    <ows:WGS84BoundingBox>
      <ows:LowerCorner>-60.0 54.5</ows:LowerCorner>
      <ows:UpperCorner>15.2 84.0</ows:UpperCorner>
    </ows:WGS84BoundingBox>
  </FeatureType>
  <FeatureType xmlns:bbr_v001="http://datafordeler.dk/schemas/bbr/v001/gml3">
    <Name>bbr_v001:tekniskanlaeg_current</Name>
    <Title>tekniskanlaeg_current</Title>
    <Abstract>bbr_tekniskanlaeg_v001_current</Abstract>
    <DefaultCRS>urn:ogc:def:crs:EPSG::25832</DefaultCRS>
    <ows:WGS84BoundingBox>
      <ows:LowerCorner>-60.0 54.5</ows:LowerCorner>
      <ows:UpperCorner>15.2 84.0</ows:UpperCorner>
    </ows:WGS84BoundingBox>
  </FeatureType>
```

Brugerscenarie, Bruno – Se alle attributer for BBR bygning



DescribeFeature request:

BBR/BBR_WFS/1.0.0/WFS?service=WFS&request=DescribeFeatureType&typeName=
=bygning_current&count=1

query

Name	Path	
service	WFS	☒ 
request	DescribeFeatureType	☒ 
typeName	bygning_current	☒ 
count	1	☒ 

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:bbr_v001="http://datafordeler.dk/schemas/bbr/v001/gml3" xmlns:gml="http://www.openg
www.w3.org/2001/XMLSchema" elementFormDefault="qualified" targetNamespace="http://datafordeler.dk/sche
<xsd:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="https://test07-wfs.datafor
<xsd:complexType name="bygning_currentType">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="0" name="forretningshaendelse" nillable="tru
        <xsd:element maxOccurs="1" minOccurs="0" name="forretningsomraade" nillable="true"
        <xsd:element maxOccurs="1" minOccurs="0" name="forretningsproces" nillable="true"
        <xsd:element maxOccurs="1" minOccurs="0" name="id_namespace" nillable="true" type=
        <xsd:element maxOccurs="1" minOccurs="0" name="id_lokalId" nillable="true" type="
        <xsd:element maxOccurs="1" minOccurs="0" name="kommunekode" nillable="true" type=
        <xsd:element maxOccurs="1" minOccurs="0" name="registreringFra" nillable="true" t
        <xsd:element maxOccurs="1" minOccurs="0" name="registreringsaktør" nillable="true
        <xsd:element maxOccurs="1" minOccurs="0" name="registreringTil" nillable="true" t
        <xsd:element maxOccurs="1" minOccurs="0" name="virkningFra" nillable="true" type=
        <xsd:element maxOccurs="1" minOccurs="0" name="virkningsaktør" nillable="true" ty
        <xsd:element maxOccurs="1" minOccurs="0" name="virkningTil" nillable="true" type=
        <xsd:element maxOccurs="1" minOccurs="0" name="status" nillable="true" type="xsd:
        <xsd:element maxOccurs="1" minOccurs="0" name="byg007Bygningsnummer" nillable="tr
        <xsd:element maxOccurs="1" minOccurs="0" name="byg021BygningensAnvendelse" nillab
        <xsd:element maxOccurs="1" minOccurs="0" name="byg024AntalLejlighederMedKøkken" n
        <xsd:element maxOccurs="1" minOccurs="0" name="byg025AntalLejlighederUdenKøkken"
```

Se tilgængelige attributer for specifikt lag (entitet) →

Brugerscenarie, Bruno – Hent konkret Bygningsdata

GetFeature request:

/BBR/BBR_WFS/1.0.0/WFS?service=WFS&request=GetFeature&typeName=bygnin
g_current&count=10&apikey=

Query

Name	Path		
service	WFS	☒	☐
request	GetFeature	☒	☐
typeName	bygning_current	☒	☐
count	10	☒	☐

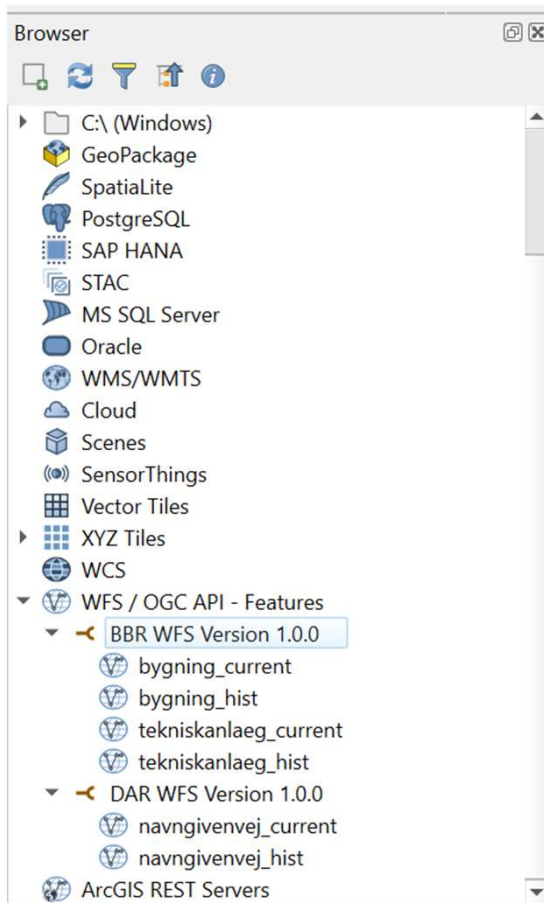
Hent data fra 10 bygninger



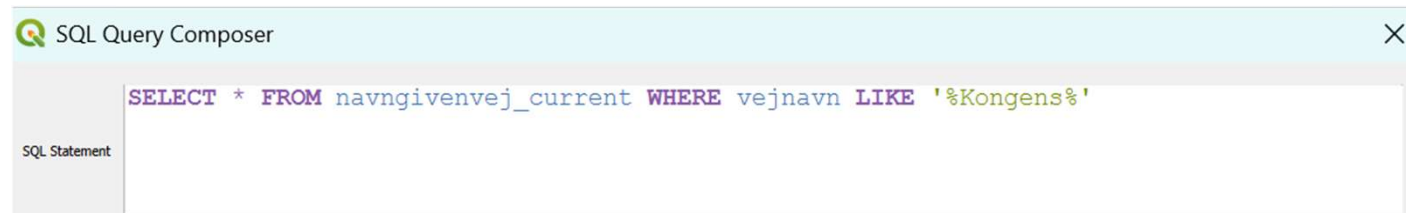
```
<wfs:member>
  <bbr_v001:bygning_current gml:id="bygning_current.fid--6aa98df3_19938f344ea_15e5">
    <bbr_v001:forretningshaendelse>Bygning</bbr_v001:forretningshaendelse>
    <bbr_v001:forretningsomraade>54.15.05.05</bbr_v001:forretningsomraade>
    <bbr_v001:forretningsproces>11</bbr_v001:forretningsproces>
    <bbr_v001:id_namespace>http://data.gov.dk/bbr/bygning</bbr_v001:id_namespace>
    <bbr_v001:id_lokalId>21931cff-af7f-4368-9725-a69b80f2fc2f</bbr_v001:id_lokalId>
    <bbr_v001:kommunekode>0575</bbr_v001:kommunekode>
    <bbr_v001:registreringFra>2022-08-17T06:54:01.096Z</bbr_v001:registreringFra>
    <bbr_v001:registreringsaktør>BBR</bbr_v001:registreringsaktør>
    <bbr_v001:virkningFra>2022-08-17T06:54:01.096Z</bbr_v001:virkningFra>
    <bbr_v001:virkningsaktør>Registerfører</bbr_v001:virkningsaktør>
    <bbr_v001:status>14</bbr_v001:status>
    <bbr_v001:byg021BygningensAnvendelse>910</bbr_v001:byg021BygningensAnvendelse>
    <bbr_v001:byg033Tagdekningmateriale>2</bbr_v001:byg033Tagdekningmateriale>
    <bbr_v001:byg041BebyggetAreal>18</bbr_v001:byg041BebyggetAreal>
    <bbr_v001:byg136PlaceringPåSøterritorie>0</bbr_v001:byg136PlaceringPåSøterritorie>
    <bbr_v001:jordstykke>2375877</bbr_v001:jordstykke>
    <bbr_v001:husnummer>0a3f508e-8ca3-32b8-e044-0003ba298018</bbr_v001:husnummer>
    <bbr_v001:grund>99f3817c-665c-4adb-acef-55b9bb0d79bf</bbr_v001:grund>
  </bbr_v001:bygning_current>
</wfs:member>
<wfs:member>
  <bbr_v001:bygning_current gml:id="bygning_current.fid--6aa98df3_19938f344ea_15e7">
    <bbr_v001:forretningshaendelse>Bygning</bbr_v001:forretningshaendelse>
    <bbr_v001:forretningsomraade>54.15.05.05</bbr_v001:forretningsomraade>
    <bbr_v001:forretningsproces>2</bbr_v001:forretningsproces>
    <bbr_v001:id_namespace>http://data.gov.dk/bbr/bygning</bbr_v001:id_namespace>
    <bbr_v001:id_lokalId>8bc5436a-b208-4498-bdba-0062a021d897</bbr_v001:id_lokalId>
    <bbr_v001:kommunekode>0813</bbr_v001:kommunekode>
    <bbr_v001:registreringFra>2017-06-02T12:29:37Z</bbr_v001:registreringFra>
    <bbr_v001:registreringsaktør>BBR</bbr_v001:registreringsaktør>
    <bbr_v001:virkningFra>2014-04-23T09:54:59.010Z</bbr_v001:virkningFra>
    <bbr_v001:virkningsaktør>Konvertering2017</bbr_v001:virkningsaktør>
    <bbr_v001:status>2</bbr_v001:status>
    <bbr_v001:byg007Bygningsnummer>3</bbr_v001:byg007Bygningsnummer>
    <bbr_v001:byg021BygningensAnvendelse>920</bbr_v001:byg021BygningensAnvendelse>
    <bbr_v001:byg136PlaceringPåSøterritorie>0</bbr_v001:byg136PlaceringPåSøterritorie>
    <bbr_v001:jordstykke>1894932</bbr_v001:jordstykke>
    <bbr_v001:husnummer>0a3f509b-8f4f-32b8-e044-0003ba298018</bbr_v001:husnummer>
    <bbr_v001:grund>e227efa9-ecdb-430d-b691-f20777588084</bbr_v001:grund>
  </bbr_v001:bygning_current>
</wfs:member>
```



Brugerscenarie, QGIS - Hent bygnings- og vejdata

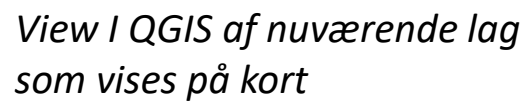
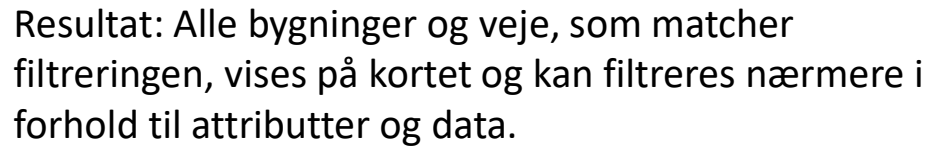


Bruger CQL-filtrering på bygning-attributter for at finde bygninger i København, bygget mellem 1960 og 1970, med et areal mindre end 150 kvadratmeter



Bruger CQL-filer for at filtrere veje der har "Kongens" i vejnavnet

Forbundet til BBR og DAR services i QGIS – Viser automatisk hvilke lag (entiteter) der er tilgængelige





Brugerscenarie, QGIS – Hent nuværende bygnings- og vejdata



Feature	Value
navngivenvej_current	
name	NULL
▶ (Derived)	
▶ (Actions)	
id	navngivenvej_current.fid-7826650a...
metadatatype	NULL
description_href	NULL
description_title	NULL
description_nilreason	NULL
description	NULL
descriptionreference_href	NULL
descriptionreference_title	NULL
descriptionreference_nilreason	NULL
identifier_codespace	NULL
identifier	NULL
name	NULL
location_location	NULL
forretningshændelse	20
forretningshændelse_nil	false
forretningsområde	10.00.00.00
forretningsområde_nil	false

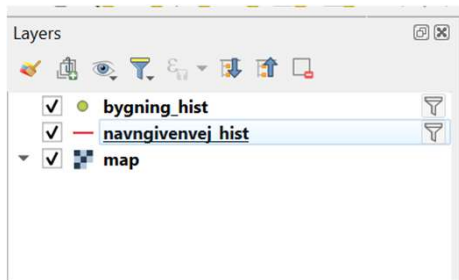
*NavngivenVej_current view -
Alle data og attributter for
Kongensvej*

Feature	Value
bygning_current	
id_namespace	http://data.gov.dk/bbr/bygning
▶ (Derived)	
▶ (Actions)	
forretningshændelse	Bygning
forretningsområde	54.15.05.05
forretningsproces	4
id_namespace	http://data.gov.dk/bbr/bygning
id_lokalld	524343a5-f27b-4530-9e4d-79f37a...
kommunekode	0101
registreringFra	21/11/2018 10:31:19 (Romance Sta...
registreringsaktør	BBR
registreringTil	NULL
virkningFra	21/11/2018 10:31:19 (Romance Sta...
virkningsaktør	Registerfører
virkningTil	NULL
status	6
byg007Bygningsnummer	17
byg021BygningensAnvendelse	313
byg024AntalLejlighederMedKøkken	NULL
byg025AntalLejlighederUdenKøkken	NULL

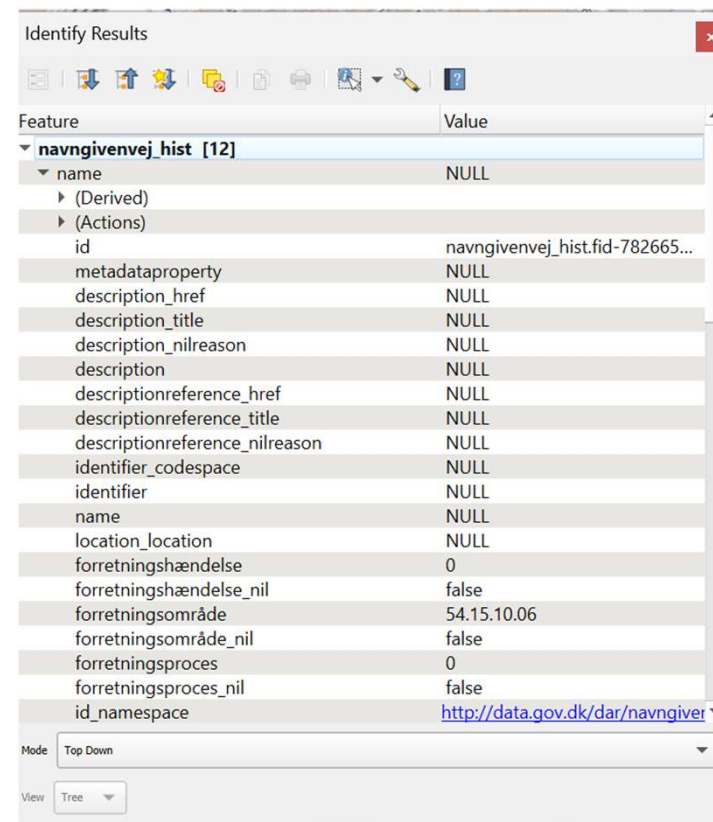
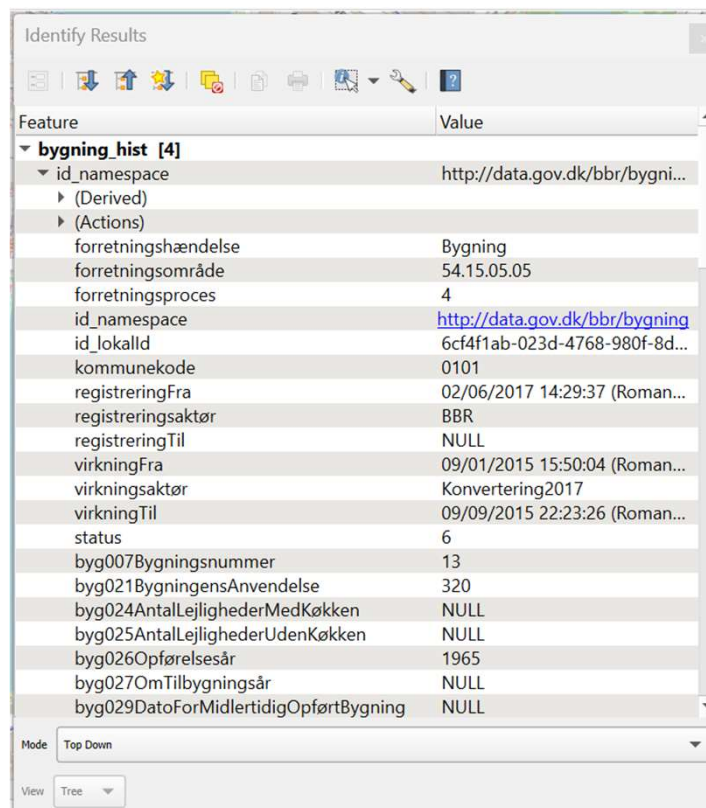
*Bygning_current view - Alle
data og attributter for en
specifikt filtreret bygning*



Brugerscenarie, QGIS – Hent bygnings- og vejdata for al historik



View i QGIS af nuværende lag som vises på kort



Entitetsbaserede Fildownload med geografisk kommuneopdelt indhold

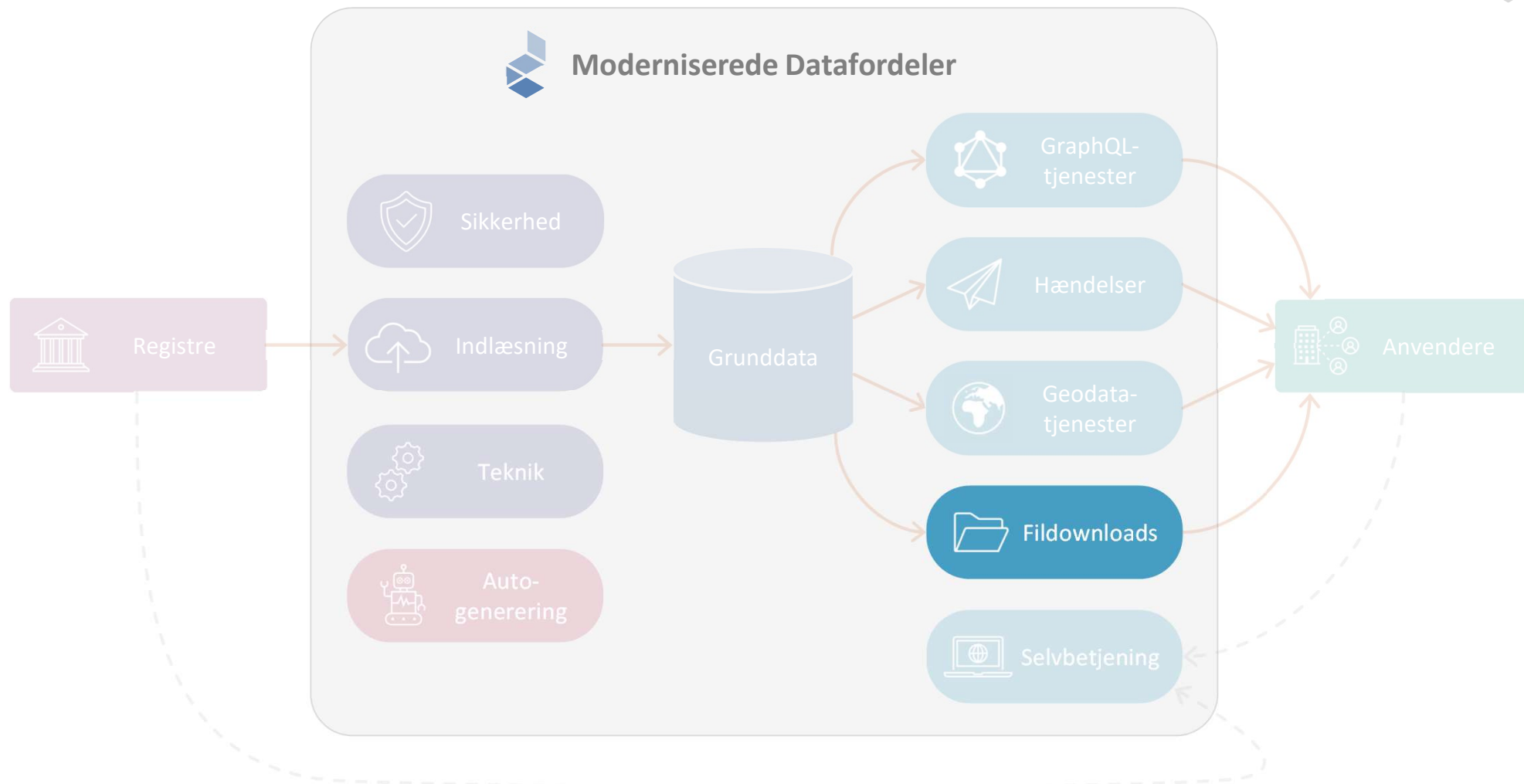
v. Kristian Kjølstad Jørgensen, Product Owner KDS

v. Marcus Norton Quistgaard, Team Lead Netcompany



DATAFORDELER

Funktionelle og tekniske domæner





Geografisk filtrering



Nuværende Datafordeler

- ÷ Predefineret GeoDanmark Vektor og Matriklen
- ÷ Brugerdefinerede/specialbyggede filudtræk



Moderniserede Datafordeler

- + Kun registre og entiteter med geografisk information
- + Filtrere registerdata ud fra:
 - Kommunekodeog/eller
 - Kommunegeometrier
 - ⇒ Geografisk geometri der intersecter med kommunegeometrien (GeoDanmark Vektor) associeret med den pågældende kommunekode
- + Filformater der understøttes:
 - JSON
 - GML
 - GPKG
 - CSV
- + Ugentlig total fildownload
- + Alle aktuelle data

Fildownloads-tjenester

Sådan finder du filer med geografisk indhold

- Ingen ændringer til query parametre fra UL2
 - Geografiske fildownloads findes kun på v2.0 af tjenesten eller nyere
- Ny "municipalityCode" felt indikerer geografisk indhold
 - Værdien kan være "null", hvis indholdet ikke kommer fra en bestemt kommune
 - Ellers svarer værdien til en kommunekode
 - Høje-Taastrup har kommunekode – 0169
 - Kommunekoden indgår også i filnavnet



Geografisk JSON filer

```
[
  {
    "fileName": "BBR_V2_Bygning_0169_TotalDownload_json_Current_269.zip",
    "register": "BBR",
    "entityName": "Bygning",
    "frequency": "Generated on the night to Monday",
    "typeOfDownload": "TotalDownload",
    "typeOfData": "Current",
    "version": "2",
    "generationNumber": 269,
    "pointInTime": "2025-05-01T01:11:02.227915Z",
    "generationTime": "2025-05-10T06:54:51.498499Z",
    "expirationDate": "2025-05-17T06:54:51.498499Z",
    "containedFileFormat": "json",
    "outputFileFormat": "zip",
    "municipalityCode": "0169"
  }
]
```

GET	https://api.datafordeler.dk/FileDownloads/v2.0/GetAvailableFileDownloads?register=BBR&version=2&entity=Bygning&apiKey= {{APL_KEY}}
Params	Authorization Headers (7) Body Scripts Settings
Query Params	
☒	Key
☒	Value
☒	register
☒	BBR
☒	version
☒	2
☒	entity
☒	Bygning
☒	apiKey
☒	{{APL_KEY}}



Fildownloads-tjenester

Sådan downloader du filer med geografisk indhold

- Geografiske filer kan også downloades ved brug af et filnavn via GetFile
- Her downloades data om Bygninger i Odense kommune fra BBR registreret

GET ▼ https://api.datafordeler.dk/FileDownloads/v1.0/GetFile?filename=BBR_V2_Bygning_0461_TotalDownload_json_Current_263.zip&apiKey= {{API_KEY}}

Params Authorization Headers (7) Body Scripts Settings

Query Params

☒	Key	Value
☒	filename	BBR_V2_Bygning_0461_TotalDownload_json_Current_263.zip
☒	apiKey	{{API_KEY}}



Fildownloads-tjenester

Sådan downloader du filer med geografisk indhold

- Alternativt kan alt data for et enkelt register downloades for en bestemt kommune
- Her downloades alt data GEODKV registreret har om Dragør kommune

GET ⌵ https://api.datafordeler.dk/FileDownloads/v1.0/GetFile?register=GEODKV&version=2&municipalityCode=0155&apiKey= {{API_KEY}}

Params Authorization Headers (7) Body Scripts Settings

Query Params

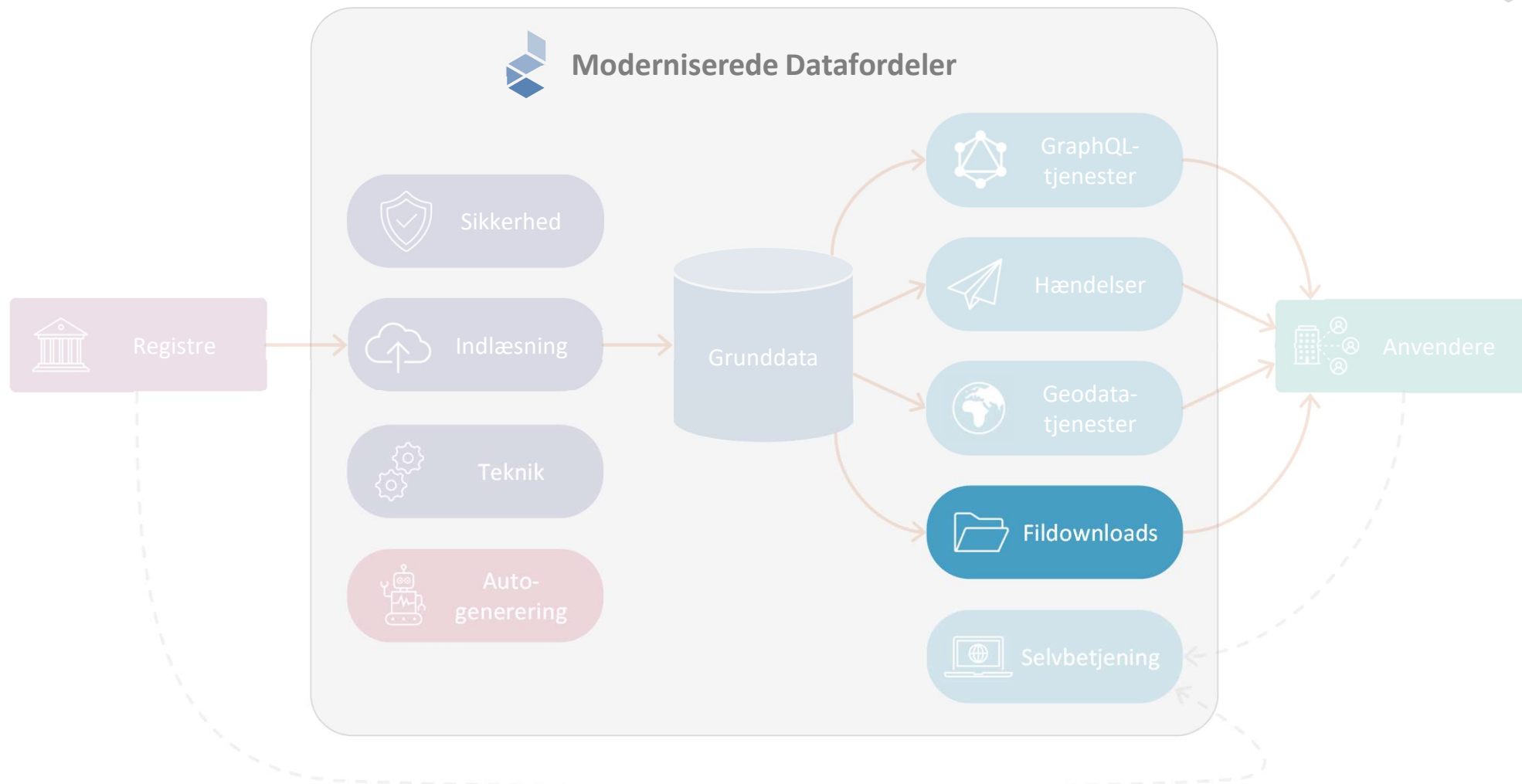
☒	Key	Value
☒	register	GEODKV
☒	version	2
☒	municipalityCode	0155
☒	apiKey	{{API_KEY}}

Flere rasterdata fildownloadendepunkter

v. Kristian Kjølstad Jørgensen, Product Owner KDS

v. Marcus Norton Quistgaard, Team Lead Netcompany

Funktionelle og tekniske domæner



Nye fildownload endepunkter

Moderniserede Datafordele

- + GetAvailableRasterFileDownloads/GetAvailablePointCloudFileDownloads
 - Skaber overblik over de raster samt PointCloud filer, der er mulig at hente
- + GetAvailableRasterFileDownloadsOverview/ GetAvailablePointCloudFileDownloadsOverview
 - Skaber overblik over de datasæt, der er tilgængelige (f.eks. Terraen samt Overflade i DHM og dæmpet/ikke dæmpet i Skærmkort inkl. zoom-niveauer)
 - se mere heri i "Transitionsguide for fildownload samt etablering og vedligeholdelse af kopiregister"
- + GetSkaermkortFiles
 - Henter samtlige zoomniveauer af Skærmkort for angivet datasæt
- + GetAllSkaermkortFiles
 - Henter alt Skærmkort for begge datasæt. Samtlige zoomniveauer for hele landet
- + 10x10 Tiles
 - Samler 1x1 tiles til 10x10 tiles for DHM: Terraen, Overflade, TerraenSkygge, OverfladeSkygge, Terraen2015 og Overflade2015 – se mere heri i "Transitionsguide for fildownload samt etablering og vedligeholdelse af kopiregister"



Hent overblik over tilgængelige rasterdata filer



<https://api.datafordeler.dk/FileDownloads/v1.0/GetAvailableRasterFileDownloadsOverview>

GetAvailableRasterFileDownloadsOverview eller **GetAvailablePointCloudFileDownloadsOverview**

```
[
  {
    "fileFormat": "tif",
    "register": "dhm",
    "dataSetName": "Bluespot2007",
    "version": "1",
    "dataDeliveryNumber": 1,
    "recordCount": 656
  },
  {
    "fileFormat": "tif",
    "register": "skaermkort",
    "dataSetName": "DaempetS07",
    "version": "1",
    "dataDeliveryNumber": 1,
    "recordCount": 171828
  },
  {
    "fileFormat": "tif",
    "register": "skaermkort",
    "dataSetName": "DaempetS1",
    "version": "1",
    "dataDeliveryNumber": 1,
    "recordCount": 27347
  },
  {
    "fileFormat": "tif",
    "register": "skaermkort",
    "dataSetName": "DaempetS12",
    "version": "1",
    "dataDeliveryNumber": 1,
    "recordCount": 3408
  },
  {
    "fileFormat": "tif",
    "register": "skaermkort",
    "dataSetName": "DaempetS1548",
    ..
  }
]
```



Hent liste af tilgængelige rasterdata filer



<https://api.datafordeler.dk/FileDownloads/v1.0/GetAvailableRasterFileDownloads?Register=DHM&minX=50000&maxX=60700&minY=61900&maxY=70000&FileFormat=ECW&Version=1&DataSetName=Overflade&PageNumber=1>

GetAvailableRasterFileDownloads eller
GetAvailablePointCloudFileDownloads

Filter for register, fx
DTK eller DHM

Afgrænsningsbox,
minX, minY, maxX,
maxY

Filter for filformat,
enten ECW, GeoTIFF
for Raster eller LAZ,
LAS for Point Cloud

Filter for version

Filter for specifikt
datasæt, fx Overflade
eller SkaermkortS3

Bruges til paginering

Resultat:

```
[
{
  "fileFormat": "tif",
  "id": "55a45425-e8c0-46d4-850b-d301b66ddb11",
  "dataSetName": "Overflade",
  "register": "dhm",
  "version": "1",
  "dataDeliveryNumber": 1,
  "FileName": "DHM_OVERFLADE_1km_6049_684.tif",
  "generationTime": "2024-03-06T22:10:50.123456Z",
  "boundingBox": {
    "minx": 50000,
    "maxX": 60700,
    "minY": 61900,
    "maxY": 70000
  },
  "fileSizeInBytes": 100708
},
]
```

```
{
  "fileFormat": "tif",
  "id": "55a45425-e8c0-46d4-850b-d301b66ddb12",
  "dataSetName": "Overflade",
  "register": "dhm",
  "version": "1",
  "dataDeliveryNumber": 1,
  "FileName": "DHM_OVERFLADE_1km_6049_685.tif",
  "generationTime": "2024-03-06T22:10:50.123456Z",
  "boundingBox": {
    "minx": 50000,
    "maxX": 60700,
    "minY": 61900,
    "maxY": 70000
  },
  "fileSizeInBytes": 100708
}
]
```



Hent skærmkortfiler i samtlige zoomniveauer

[https://api.datafordeler.dk/FileDownloads/v1.0/GetSkaermkortFiles?
minX=520000&maxX=540000&minY=6160000&maxY=6180000&
Version=1&DataSet=Klassisk&DataDeliveryNumber=1](https://api.datafordeler.dk/FileDownloads/v1.0/GetSkaermkortFiles?minX=520000&maxX=540000&minY=6160000&maxY=6180000&Version=1&DataSet=Klassisk&DataDeliveryNumber=1)

Afgrænsningsbox,
minX, minY, maxX,
maxY

Filter for version

Enten **Klassisk** eller
Daempet

Filter for
dataleverencenummer

Recent download history



SKAERMKORT_klassisk_520000_6160000
-540000_6180000.zip
64.5 KB • Resuming...

Resultat:

Henter samtlige zoomniveauer af Skærmkort for angivet datasæt.

[https://api.datafordeler.dk/FileDownloads/v1.0/GetAllSkaermkortFiles?
Version=1&DataDeliveryNumber=1](https://api.datafordeler.dk/FileDownloads/v1.0/GetAllSkaermkortFiles?Version=1&DataDeliveryNumber=1)

Filter for version

Filter for
dataleverencenummer

Recent download history



SKAERMKORT.zip
15.3 MB • Resuming...



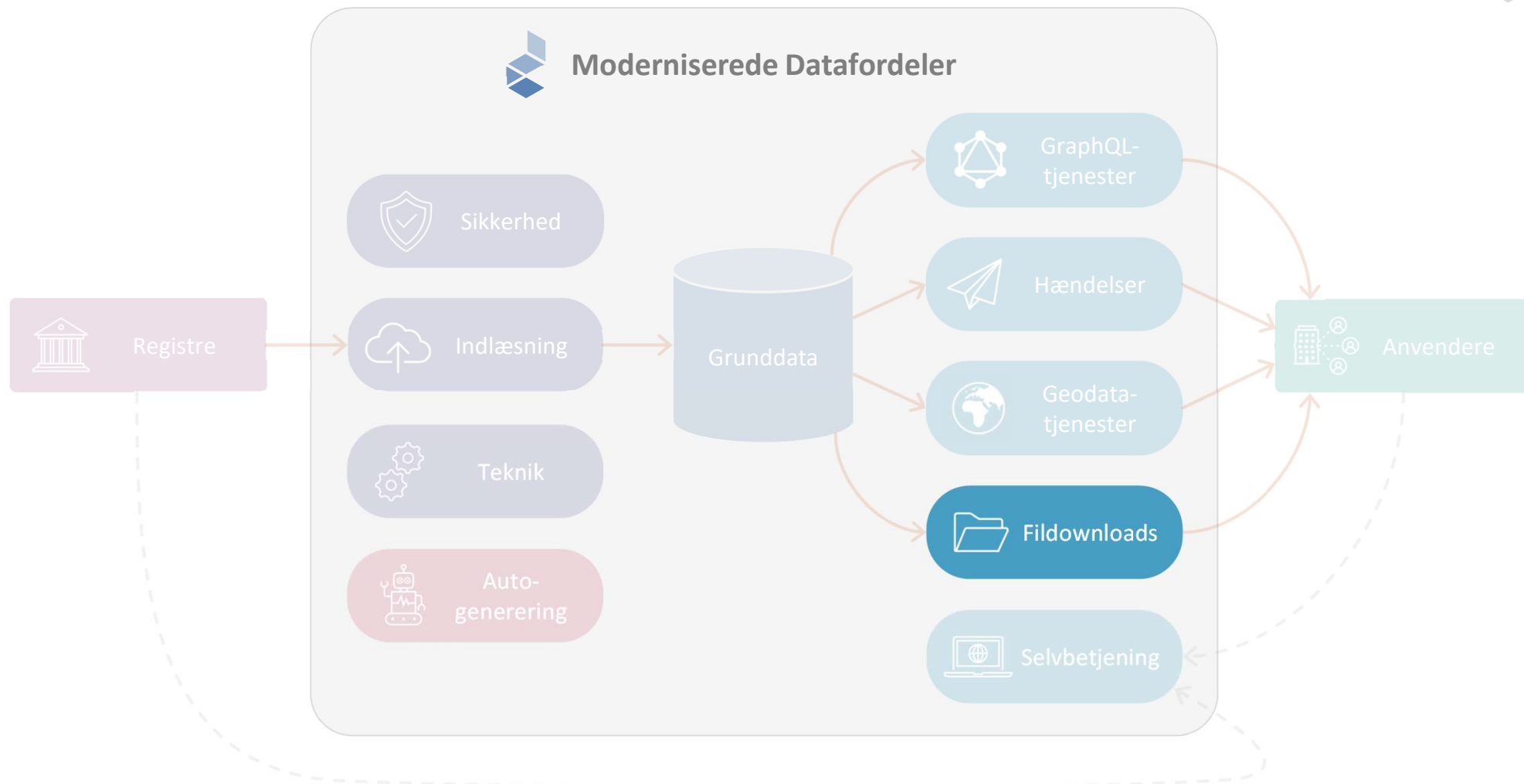
Resultat:

Henter alt Skærmkort for begge datasæt. Samtlige zoomniveauer for hele landet.

MD5 checksum på Fildownload af tabulærdata

v. Kristian Kjølstad Jørgensen, Product Owner KDS

Funktionelle og tekniske domæner





MD5 checksum på Fildownload



Nuværende Datafordeler

- ÷ Hverken filstørrelse eller MD5 checksum er tilføjet til filer
- ÷ Ingen måde at vide om Fildownload er fejlet



Moderniserede Datafordeler

- + /GetAvailableFileDownloads returnerer nu
 - MD5 checksum på den downloadede zip-fil
 - Filstørrelse i bytes på den downloadede zip-fil



MD5 checksum på Fildownload

- Ingen ændringer til query parametre fra UL2 i forbindelse med kald til endpointet /GetAvailableFileDownloads
- Metadata inkluderer nu:
 - md5Hash
 - fileSizeInBytes

GET

https://api.datafordeler.dk/FileDownloads/GetAvailableFileDownloads?format=GML®ister=DHMHoejdekurver&entity=Kote2_5&version=1

Params

Authorization

Headers (9)

Body

Scripts

Settings

Query Params

☒ Key	Value
☒ format	GML
☒ register	DHMHoejdekurver
☒ entity	Kote2_5
☒ version	1



MD5 checksum & filstørrelse

```
[
{
  "fileName": "DHMHoejdekurver_V1_Kote2_5_TotalDownload_gml_Temporal_75.zip",
  "register": "DHMHoejdekurver",
  "entityName": "Kote2_5",
  "frequency": "Generated on the night to Monday",
  "typeOfDownload": "TotalDownload",
  "typeOfData": "Temporal",
  "version": "1",
  "generationNumber": 75,
  "pointInTime": "2025-06-14T03:01:23.141677Z",
  "generationTime": "2025-07-28T21:12:19.092543Z",
  "expirationDate": "2025-08-04T21:12:19.092543Z",
  "containedFileFormat": "gml",
  "outputFileFormat": "zip",
  "municipalityCode": null,
  "registerImportSequenceNumber": null,
  "md5Hash": "601417fefbd4a1e1593500e3d859cc55",
  "fileSizeInBytes": 17113945
}
]
```

Ekstern test

v. Kaja Hansen, Testmanager KDS



DATAFORDELER



Mulighed for ekstern test af UL3 (I)

Test vil være mulig i perioden **22. september til 24. oktober 2025**

Testaktiviteterne omfatter leverancerne i Datafordelerens udviklingsleverance 3:

- Fleksibel opslagslogik til GraphQL-tjenester
- Specielle GraphQL-tjenester til EJF
- Specielle GraphQL-tjenester til CPR
 - Testes i testperioden af registeret
 - Ekstern test kan ske på test04 efter idriftsættelse
- Entitetsbaserede hændelser
 - Kan testes i begrænset omfang, da det afhænger af dataopdateringer på testmiljøet



Mulighed for ekstern test af UL3 (II)

- Entitetsbaserede WFS-tjenester
 - BBR og DAR entiteter er blevet tilføjet
- Entitetsbaserede fildownloads med geografisk defineret indhold (kommuneopdelte) for BBR, DAR, DAGI, DHM Højdekurver, GeoDKV og MAT
 - Et begrænset udvalg kan testes
- Flere rasterfildownloads tjenester
 - Hent liste af tilgængelige rasterdata og punktsky filer
 - Hent skærmkortfiler i samtlige zoomniveauer
 - 10x10km tiles for DHM

Mulighed for ekstern test af UL3 (III)



Der er tale om en eksplorativ test, hvor man som anvender kan teste i forhold til eget behov for anvendelse af services og data.

Test foregår på Datafordelerens testmiljø Test06 og forudsætter, at man har oprettet en bruger(e) på Test 06 i Datafordeler Administration:

- <https://test06-portal.datafordeler.dk/hjem>

Se guide til brugeroprettelse:

- <https://confluence.sdfi.dk/pages/viewpage.action?pageId=187105448>

Information om den eksterne test findes her:

- <https://datafordeler.dk/vejledning/modernisering/ekstern-test-udviklingsleverance-3-ul3/>

Mulighed for ekstern test af UL3 (IV)



Test kan tage udgangspunkt i de guides, som vil blive tilføjet undersiden om test, når testperioden starter:

Entitetsbaserede og fleksibel opslagslogik (GraphQL)

- Brugervejledning - Transitionsguide for entitetsbaserede GraphQL-tjenester
- Brugervejledning Transitionsguide for Fleksibel Opslagslogik
- Brugervejledning - Udvalgte REST-tjenester omsat til GraphQL-queries
- Transition mellem REST-services og GraphQL-services

Hændelser (GraphQL)

- Brugervejledning - Hændelse anvenderguide
- Brugervejledning - Vedligeholdelse af kopiregister med hændelser

Entitetsbaserede WFS

- Brugervejledning: Transitionsguide for entitetsbaserede WFS-tjenester

Fildownload

- Brugervejledning - Transitionsguide for fildownload samt etablering og vedligeholdelse af kopiregister
- Brugervejledning - Oprettelse af kopiregister for fildownload

Transition til den Moderniserede Datafordeler

v. Linda Strøm Lundum, Implementeringsansvarlig KDS



Vi skifter gear i implementeringsstøtte

- To primære elementer

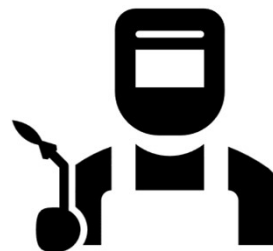


1. Tværgående KDS Implementeringsteam etableret

- Bygger på den eksisterende støtte indsats
- Udvikling og Drift & Forvaltning – 15 personer involveret

2. Transitionsnetværk for den moderniserede Datafordeler

- Operationelt fokuseret netværk, der støtter anvenderes overgang som består af:
 - Planlægning
 - Implementering i egne systemer & processer
 - Ibrugtagning
- Arbejdsgrundlag for netværket er godkendt
 - Kick off oktober 2025 (er planen)





Støtte indsatser

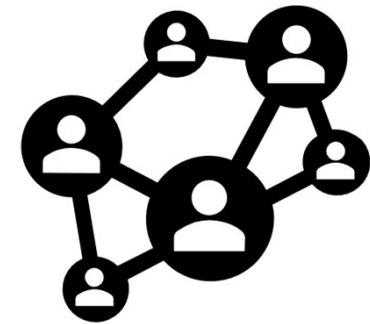
I forbindelse med idriftsættelsen af udviklingsleverance 3 vil anvendere kunne gøre brug af følgende hjælpeværktøjer og støtteformater:

- Generelle informations- og temamøder – optages og gøres tilgængelige via Datafordeler.dk
- Specifikke webinarer – optages og gøres tilgængelige via Datafordeler.dk
- Dokumentation af de moderniserede funktioner via Datafordeler.dk, herunder Transitionsguides inklusive hands-on eksempler på oversættelse af REST-kald til GraphQL-forespørgsler
- Deltagelse i ekstern test
- Adgang til testmiljøer hvor anvendere kan afprøve fx GraphQL-tjenester
- [DAF2 postkasse](#) – kanal for opklarende spørgsmål



Klimadatastyrelsen vil etablere og facilitere et Transitionsnetværk

- Fremme en fælles forståelse
- Nedbryde barrierer
- Facilitere vidensdeling
- Styrke samarbejdet
- Maksimere værdiskabelsen
- Bringe de rette kompetencer i spil





Klimadatastyrelsen



Cost-beregning og performance

- Forespørgsler placeres indenfor en af følgende tre kategorier
 1. Simple (100ms)
 2. Almindelig (1.000ms)
 3. Komplex (2.500ms)
- Følgende anvendelse er de mest indflydelsesrige faktorer ift. beregning af cost:
 - Antal resultater, der forespørges om
 - Filtrering på geometrier
 - Generel filtrering
 - Join-dybde



Cost-beregning og performance

- Cost beregningen anvendes til at anskueliggøre kompleksiteten af kald til en Datafordeler-tjeneste.
- Cost-beregningen anvendes bl.a. til at kunne indplacere kald i de tre kategorier med tilhørende svartider.
- Cost-beregningen kan også anvendes til at beskytte Datafordeleren mod eventuel uhensigtsmæssig brug – og dermed sikre stabil drift og performance for alle anvendere.
- Cost vs. performance vurderes på baggrund af den eksterne test og overtagelsesprøven, hvorefter det evalueres, om der skal sættes en cost grænse.
 - Cost skal fortsat understøtte forespørgsler, der passer til anvenderes forretningsbehov.

Afrunding

v. Natalia Hansen, Projektleder KDS



DATAFORDELER