Netcompany

Moderniserede hændelser

Klimadatastyrelsen (KDS)
Datafordeleren (DAF)

10-03-2026

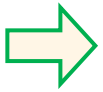
Agenda

- Introduktion
- Hændelsesentiteter
- Register Import Status-entiteten
- Hændelseskolonner på registerentiteter
- Forretningshændelser (CPR)
- Live demo

Introduktion – Gevinster ved moderniserede hændelser

Nuværende Datafordeler

- ÷ Varierende udstilling af hændelser fra register til register
- ÷ Hændelser kun tilgængelig for en brøkdel af entiteter
- ÷ Kompleks 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
- ÷ Kompleks og forældet teknisk løsning



Moderniserede Datafordeler

- \+ Ensartet udstilling af hændelser, uanset register
- \+ Hændelser for alle entiteter 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 ”long polling” som kan bruges til samme formål

Introduktion – Hvad er hændelser?

- De nye hændelser er kun datahændelser
  - Der genereres altså 1 hændelse for hver data-operation i databasen, når registret indlæser data
  - Dette er forskelligt fra tidligere, hvor hændelserne blev defineret af registret ved dataindlæsning
- Hændelser giver brugerne information om:
  - Hvilken data er opdateret
  - Hvornår den er opdateret
  - Hvordan den er opdateret
- Denne information kan bruges af anvendere til at holde deres kopiregister opdateret i nær realtid

- Hentning og anvendelse af hændelser beskrives nærmere i kommende slides

Introduktion – Hvor findes hændelser?

- Hændelsesentiteter
  - GraphQL
- Register Import Status-entiteten
  - GraphQL
  - Fildownload metadata (kun sekvensnummer for den sidste indlæste pakke)
- Nye felter på registerentiteter (rowVersion, rowId og registerImportSequenceNumber)
  - GraphQL
  - Fildownload

- Hændelser er ikke tilgængelige for disse register, på grund af sjælden indlæsning:
  - Historiske kort og data (HISTKORT)
  - Danmarks Højdemodel – Højdekurver (DHMHoejdekurver)
  - Danmarks Højdemodel – Oprindelse (DHMOprindelse)

Introduktion – Hvordan gives der adgang til hændelser?

- Hændelsesentiteter:
  - Du kan automatisk se hændelser for de entiteter du har adgang til
  - Undtagelse: CPR Public Sector giver også adgang til at se alle hændelser for CPR
- Register Import Status-entiteten
  - Alle anvendere kan tilgå entiteten

Netcompany

Hændelses-felter på registerentiteter

Hændelses-felter på registerentiteter

- Der er tilføjet 3 nye felter til alle registerentiteter
  - Disse felter indeholder metadata for en datarække
- Felterne bruges til at identificere rækker og opdateringer der er sket
- Udstilles i både GraphQL og Fildownload

Attribut	Type	Beskrivelse
datafordelerRowId	UUID	Unik identifikator for en række. Genereret af DAF.
datafordelerRowVersion	Int	Rækkeversion. Begynder på 1 og øger hver gang rækken bliver opdateret.
datafordelerRegisterImportSequenceNumber	Int	Sekvensnummer for den dataindlæsningspakke der senest opdaterede rækken

Hændelses-felter på registerentiteter

- RowID er særligt nyttig for bitemporale entiteter for at vide præcis, hvilken række der skal opdateres i en anvenders kopiregister
  - Dette kan være svært for anvender at udlede, da f.eks. delta-downloads indeholder flere rækker med samme register-ID ved en bitemporal opdatering
  - F.eks. opdateres den gamle historiske række, og en ny aktiv indsættes – disse vil have samme register-ID men forskellige RowID

```
query {
  BBR_Bygning(where: { datafordelerRowId: { eq: “a267e432-273e-471b-8f33-c07dd774392c” } }) {
    nodes {
      datafordelerRegisterImportSequenceNumber
      datafordelerRowId
      datafordelerRowVersion
      husnummer
      kommunekode
    }
  }
}
```

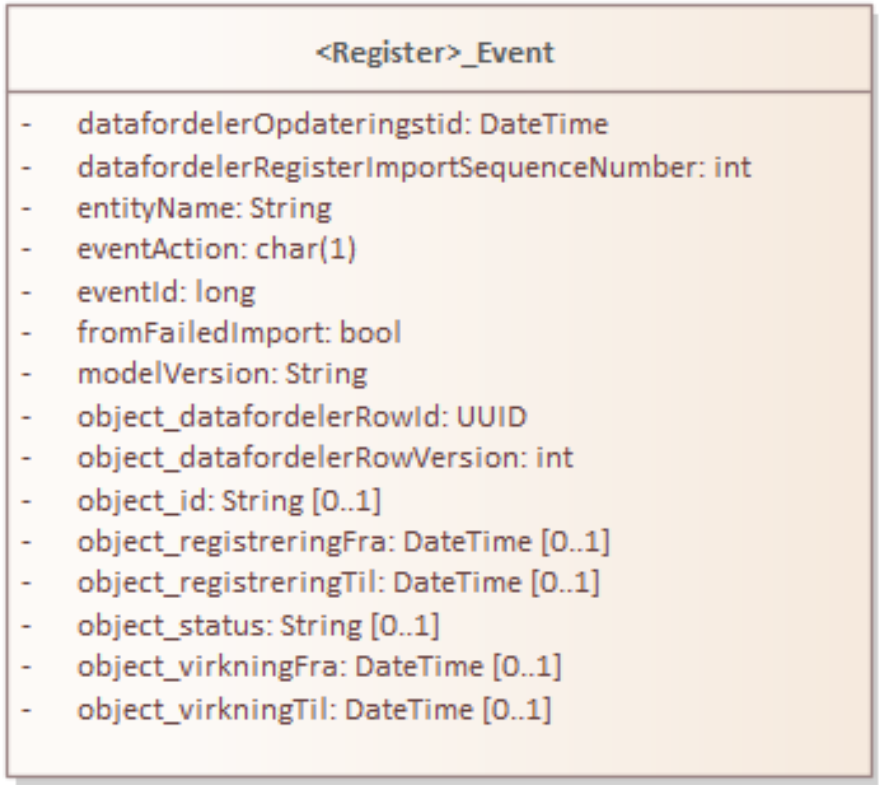
Eksempel på GraphQL-query for at hente data med rowId-filtrering

Netcompany

Hændelsesentiteter

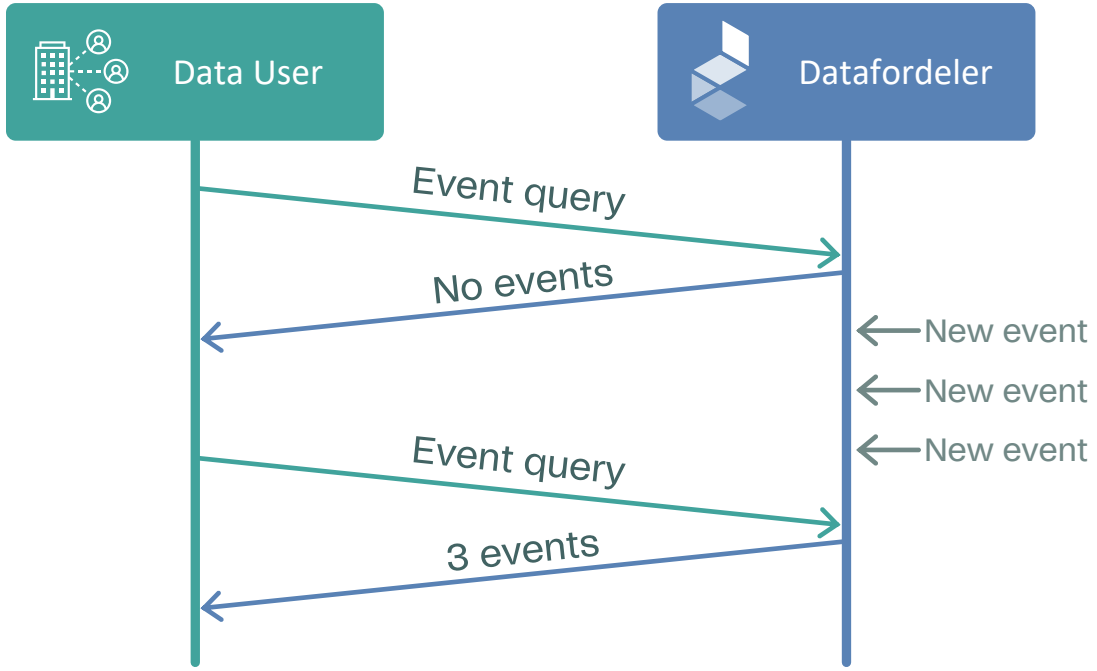
Hændelsesentiteter – Data

- Det mest centrale for hændelser
- Der oprettes en hændelses-række for hver dataændring, der sker i et register
- Alle ”object_” felt mapper til den relateret entitet



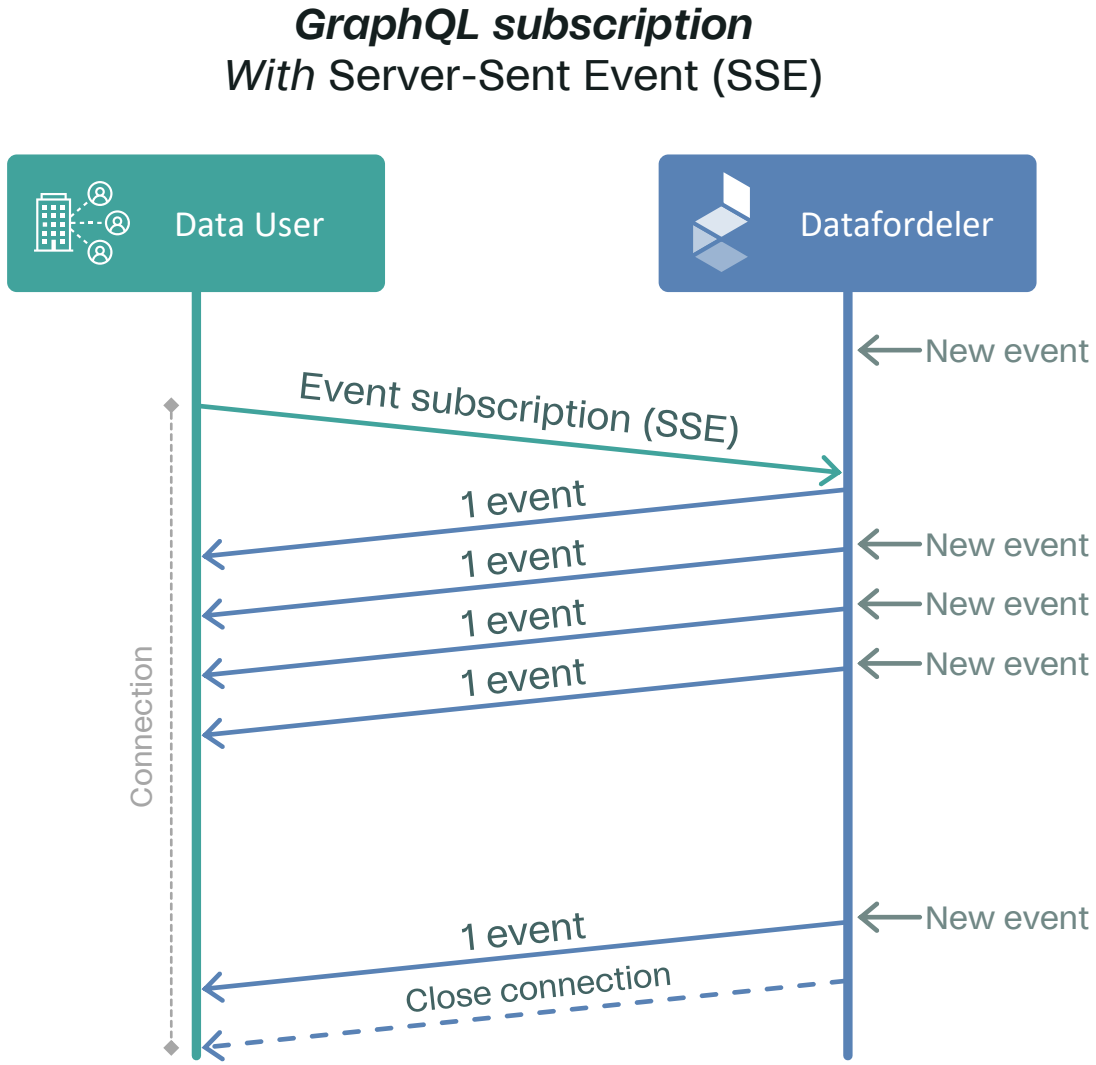
Hændelsesentiteter – Queries

- Kan kun hentes via GraphQL
  - Udstilles af de entitetsbaserede GraphQL-tjenester



Hændelsesentiteter – Subscriptions

- Det er også muligt at oprette et abonnement for automatisk at hente nye hændelser
  - Dette sker også i de entitetsbaserede GraphQL-tjenester

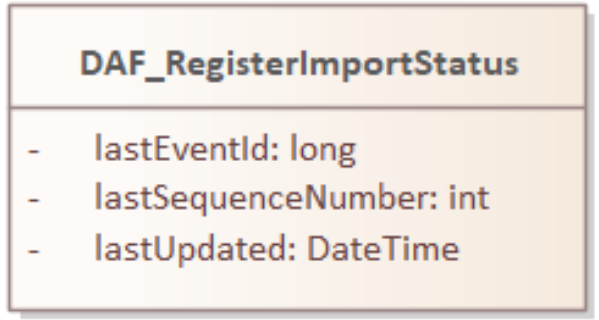


Netcompany

Register Import Status

Register Import Status – GraphQL

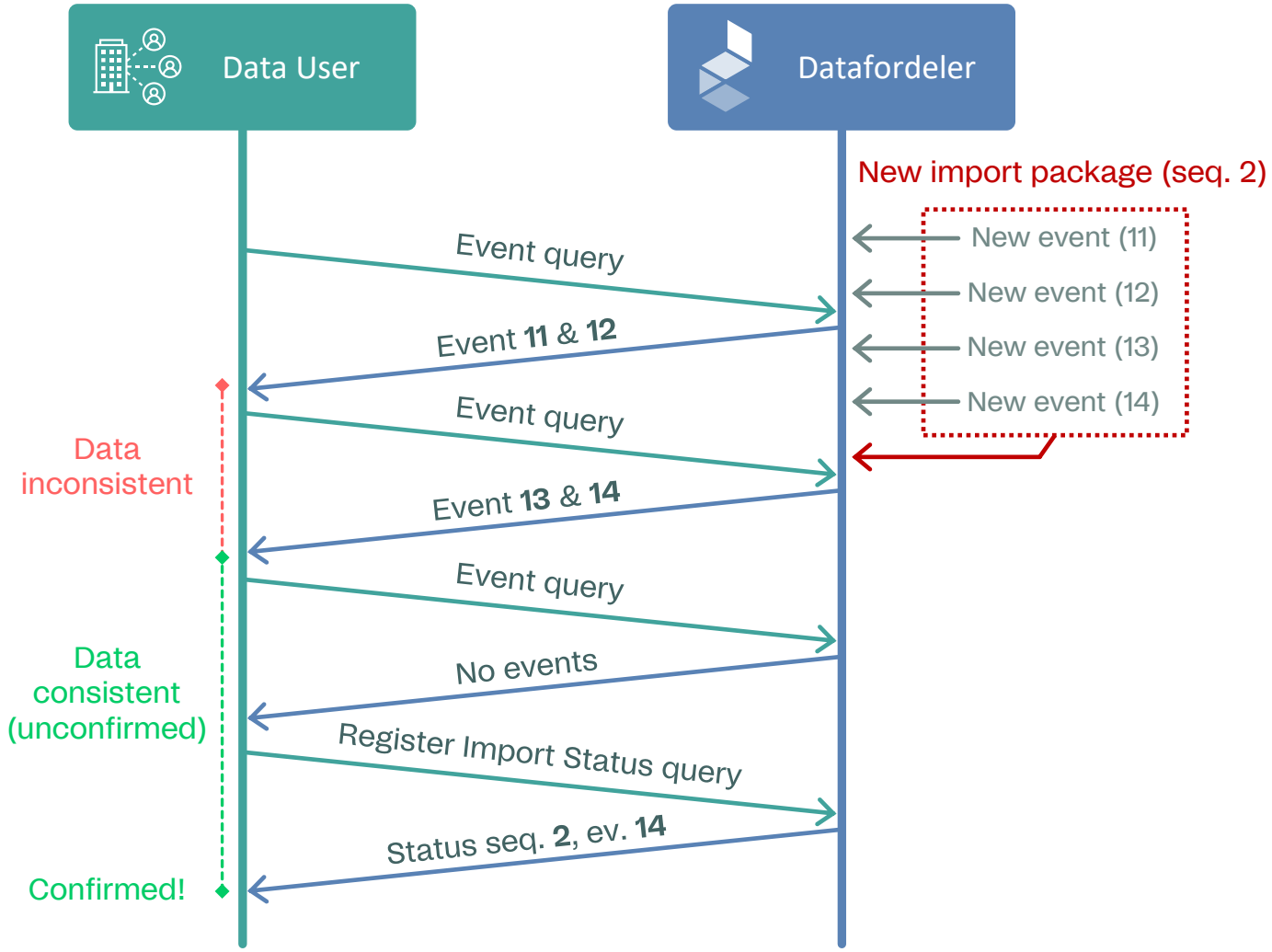
- Ens med hændelser, hvor det udstilles som en entitet i GraphQL
- Opdateres, når en datapakke er indlæst færdig og kan derfor bruges til at sikre dataintegritet



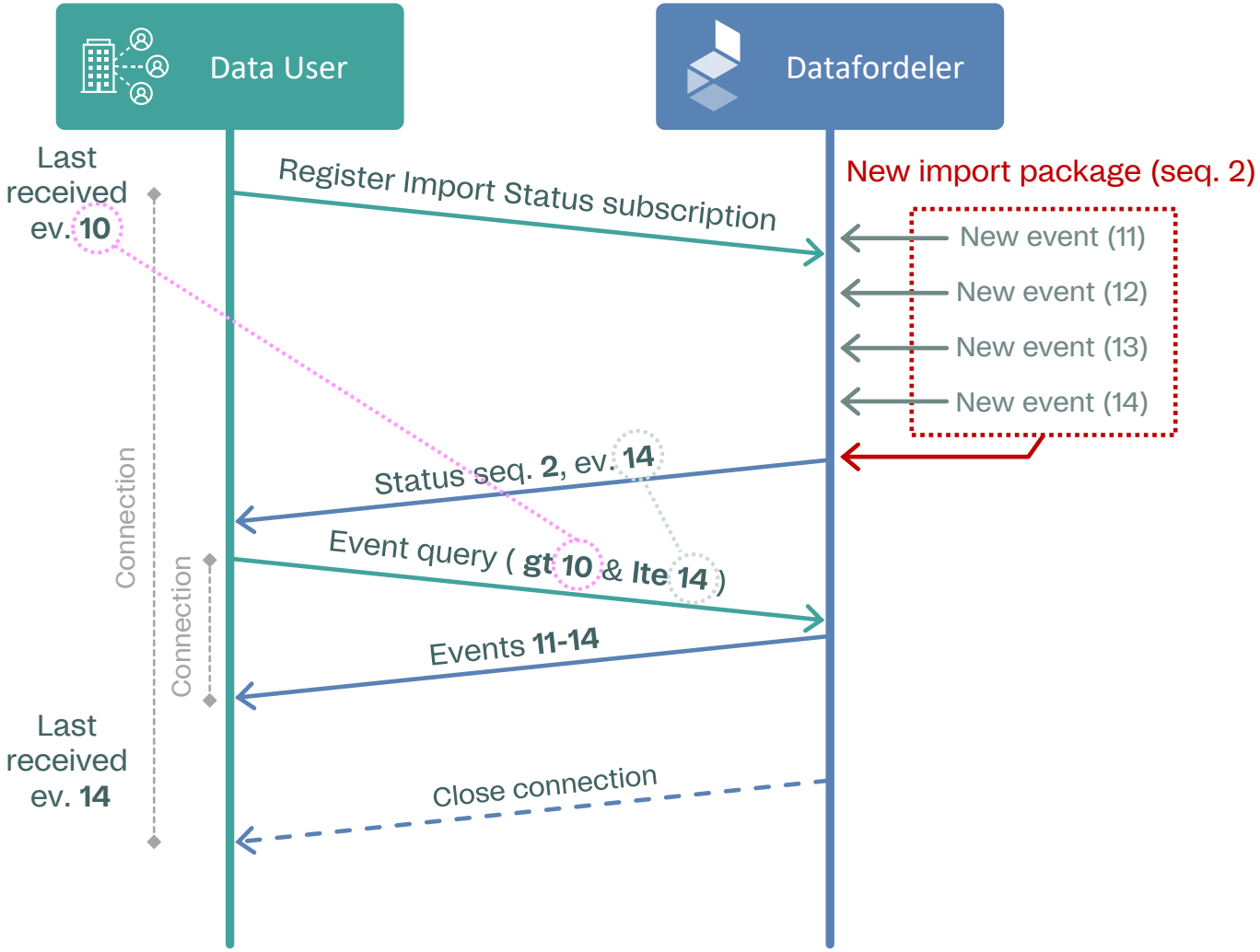

```
subscription {
  DAF_RegisterImportStatus {
    lastEventId
    lastSequenceNumber
    lastUpdated
  }
}
```

Eksempel på GraphQL-subscription for at hente Register Import Status

Register Import Status – Eksempel flow for dataintegritet



Register Import Status – Eksempel flow med hændelser



Register Import Status – Fildownload

- Det er, udover ved GraphQL, også muligt at finde Register Import Status i Fildownload-metadata
- Dette er dog begrænset til at kunne se det sidste sekvensnummer, der er brugt til at generere et fildownload
- Udstilles under GetAvailableFileDownloads-endepunktet


```
{
  "fileName": "BBR_V3_FordelingAfFordelingsareal_0480_TotalDownload_csv_Current_556.zip",
  "register": "BBR",
  "entityName": "FordelingAfFordelingsareal",
  "frequency": "Generated from 23:00 UTC on Friday",
  "typeOfDownload": "TotalDownload",
  "typeOfData": "Current",
  "version": "3",
  "generationNumber": 556,
  "pointInTime": "2026-02-27T23:03:28.836162Z",
  "generationTime": "2026-02-27T23:17:15.975907Z",
  "expirationDate": "2026-03-06T23:17:15.975907Z",
  "containedFileFormat": "csv",
  "outputFileFormat": "zip",
  "municipalityCode": "0480",
  "registerImportSequenceNumber": 6648514,
  "md5Hash": "efa97ec940a49a8ae058874db2a8fcfb",
  "fileSizeInBytes": 2183
},
```


Eksempel på udstillet sekvensnummer i GetAvailableFileDownloads

Netcompany

Forretningshændelser (CPR)

Forretningshændelser (CPR) – Intro

- CPR har som tidligere også egne forretningshændelser
- Disse er ikke blevet opdateret ifm. moderniseringen
  - Udover at de nu hentes via GraphQL og Fildownload i stedet for de gamle tjenester
- Anbefalingen er, at CPR Custom-tjenesterne bruges til at hente CPR-data, hvor anvender kan filtrere på forretningshændelser

Forretningshændelser (CPR) – CPR Custom

- Både CPR Private og Public Sector har valgfri filtrering for forretningshændelser
- Disse tjenester kan altså bruges separat fra CPR_Events, hvis man kun er interesseret i forretningshændelser
  - Præcis som før

```
query {
  CPRCustom_PrivateSectorPerson(
    input: {
      foedselsdato: "1900-01-01"
      navn: { fornavne: "Test", efternavn: "Person" }
      haendelse: { haen: "CPRADRESSE" }
    }
  ) {
    ... fields
    }
}
```

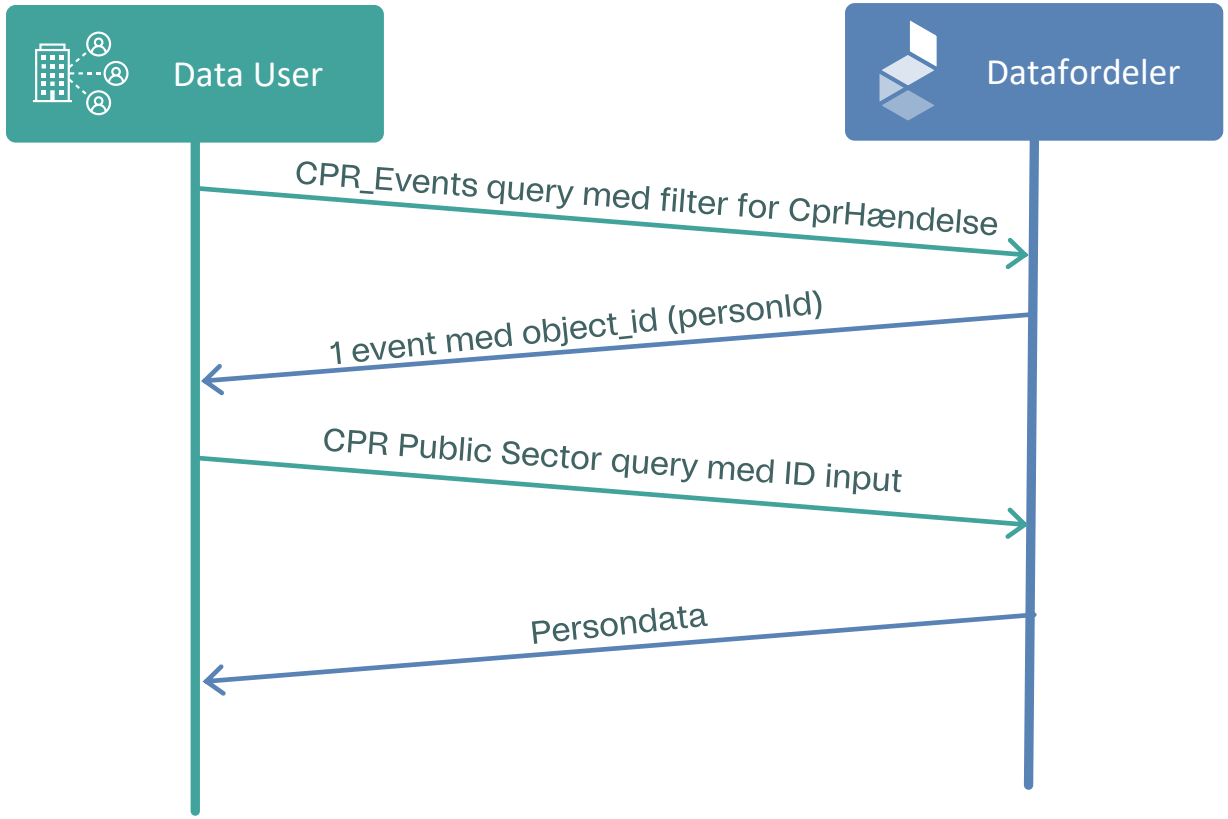
Eksempel på GraphQL-query med forretningshændelsesfiltrering

Forretningshændelser (CPR) – Versus datanære hændelser

- Det er vigtigt at forstå forskellen på, at de hændelser, der er præsenteret i tidligere slides, er datanære hændelser og ikke indeholder information om, hvilken forretningshændelse en dataopdatering relaterer til.
- Der er derfor begrænsede muligheder for at kombinere dem

Forretningshændelser (CPR) – Kombinera datanære og forretningshændelser

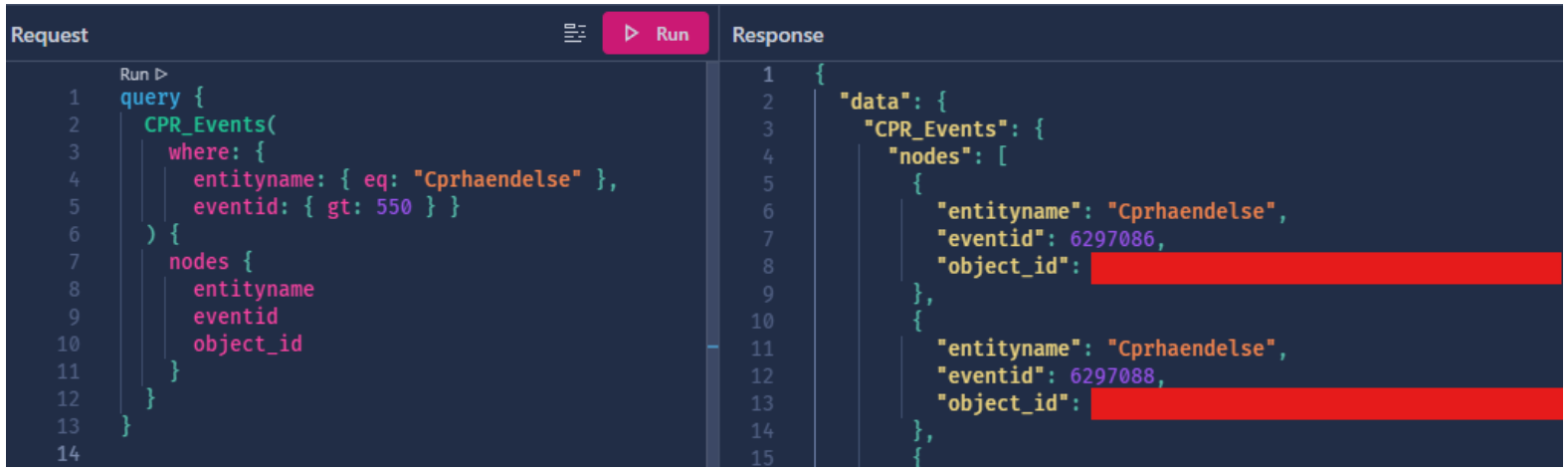
- Derimod, hvis man har adgang til CPR Public Sector (og dermed forrettningshændelser) kan man kombinere dem med de datanære hændelser (CPR_Events)
- Dette giver et flow som vist til højre
- Brugere er dog altid nødt til at lave en forespørgsel til CPR Public Sector
  - Hvis der ikke er en relevant hændelse, returneres ingen data
- CPR_Events kan kun give information om, at der er sket en forrettningshændelse, men **ikke** hvilken en



LIVE DEMO

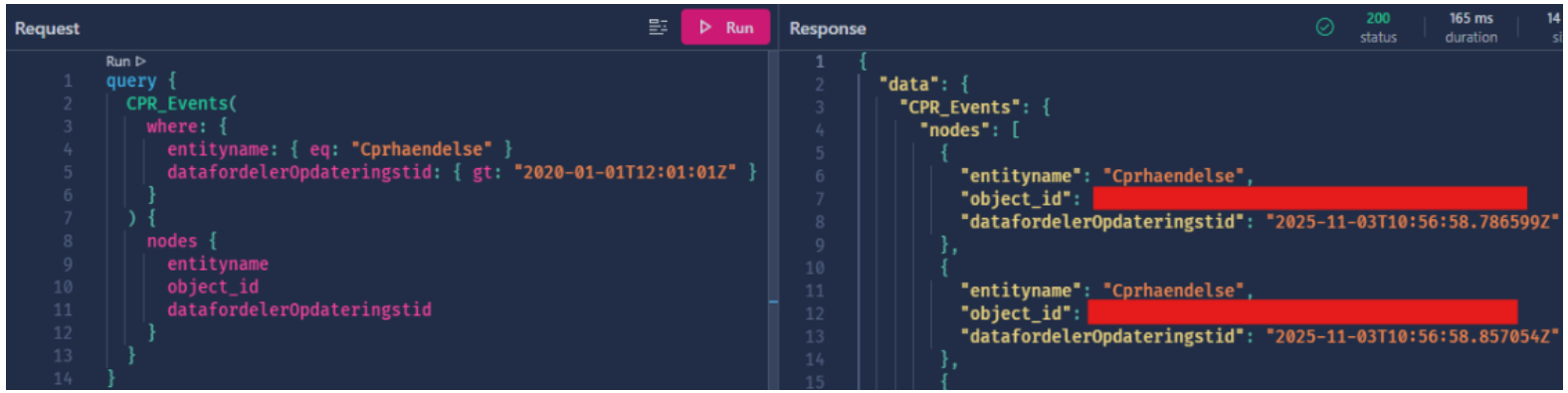
CPR Events Eksempel – Hent hændelser

- Med EventID:



Eksempel på GraphQL-query med Event ID filtrering (beskyttet data ikke vist)

- Med Opdateringstid:



Eksempel på GraphQL-query med Opdateringstid-filtrering (beskyttet data ikke vist)

CPR Events Eksempel – Hent ny Persondata

- Med forretningshændelse:

Eksempel på opfølgende GraphQL-query til CPR Public Sector Person (beskyttet data ikke vist)

- Uden forretningshændelse:

Eksempel på opfølgende GraphQL-query til CPR Public Sector Person (beskyttet data ikke vist)

Netcompany

Opsamling og spørgsmål

Netcompany

Marcus Winding Quistgaard
Lead Software Engineer
mwq@netcompany.com

Netcompany.com