



Transitionsnetværk - den moderniserede Datafordeler

10. marts 2026



Klimadatastyrelsen

Arbejdsmøde 19. februar

- Datafordeler Administration
- CPR modernisering
- VUR – opdatering på proces
- Seneste nyt om adressevælger



Klimadatastyrelsen

Tid	Emne	
10:00 – 10:15	Indflyvning og opdatering på udfasning	Linda/Asbjørn
10:15 - 10:30	Kombit guide til kommuner	Jan Kjellerup
10:30 – 10:45	Igangsætning af opsamling	Linda
10:45 – 11:00	Sidste nyt om adressevælger/Dawa	Margiy
11:00 -11:15	Pause	
11:15 – 12:00	CPR & BBR Hændelser	NetCompany
12:00 – 12:45	Opsamling og drøftelse af konkrete anvenderbehov for yderligere støtteindsatser	Linda
12:45 – 13:00	Afrunding	Linda

Opdatering om paralleldrift

Den moderniserede Datafordeler er sat i drift 4. december, og der er nu en periode med paralleldrift.

På baggrund af dialog med anvendere har Klimadatastyrelsen besluttet, at det er nødvendigt at forlænge paralleldriften og den løbende udfasning af ikke moderniserede tjenester og funktioner på Datafordeleren.

Det betyder, at Klimadatastyrelsen arbejder på at rykke den nuværende udfasningsplan for ikke-moderniserede tjenester på Datafordeleren, så for:

- Planlagt udfasning af tjenester 1. april 2026: Udfasningen af Rasterdata filudtræk fastholdes i april. Men udfasning af WFS på Snowflake rykkes til medio 2026.
- Planlagt udfasning af resterende ikke-moderniserede tjenester 30. juni 2026: Sammenstillede WFS-tjenester på Geoserver, REST-tjenester, WMS/WMTS/WCS på ikke-moderniseret brugerstyring, Hændelser og Filudtræk rykkes til ultimo 2026.

Der vil således fortsat være tale om en gradvis udfasning, og alle tjenester vil fortsat blive udfaset i løbet af 2026.

KDS arbejder i dialog med centrale anvendere på at lægge den endelige opdaterede plan for paralleldriften og fastsætte datoerne for udfasning. En opdateret udfasningsplan forventes at være tilgængelig ultimo marts, og anvenderne vil blive orienteret via kds.dk og datafordeler.dk.

For mere information om hvordan man kommer godt i gang med de moderniserede tjenester læs på: <https://datafordeler.dk/vejledning/>

Læs også om Klimadatastyrelsens netværk for anvendere, der skal overgå til de moderniserede tjenester: <http://datafordeler.dk/transition>"



2. Seneste nyt fra Dawa

V. Karen



Klimadatastyrelsen



3. Guide til kommuner

V. Jan/ Kombit



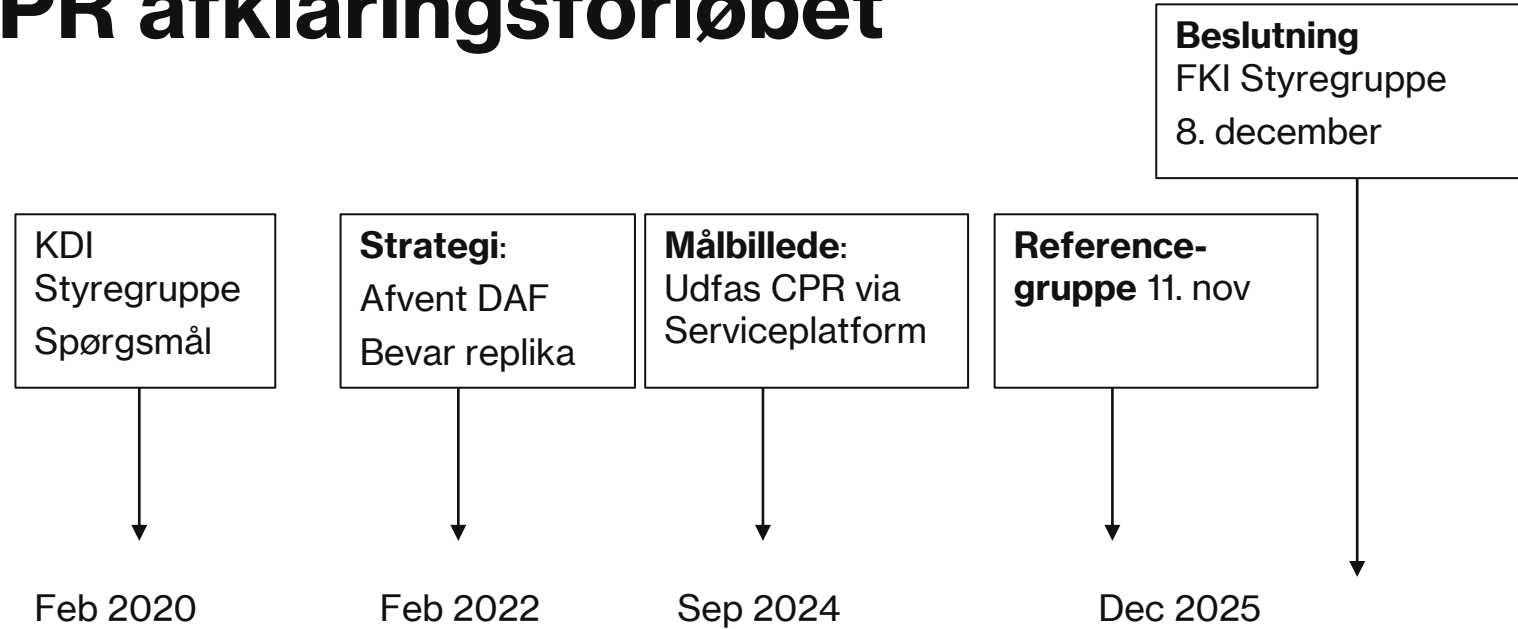
Klimadatastyrelsen

CPR udfasning

Hvordan man flytter alle kommuner over på datafordeleren.



CPR afklaringsforløbet



Transition



Den nye, moderne datafordeler

- Datafordeleren er moderniseret
- Udstilling via GraphQL – Dataminimering
- Understøttelse af CPR2 – en opdateret datamodel
- Nedbringelse af latency til "nær-realtid" fra 2027
- Grunddataprogrammet, politisk forankret og godkendt af flere ministerier, har besluttet, at Datafordeleren er fremtidige fælles distributionsplatform.
- Der er kommunalt investeret i grunddataprogrammet.

Aktstykke 16, Finansministeriet

– Folketingstidende 2019-20:

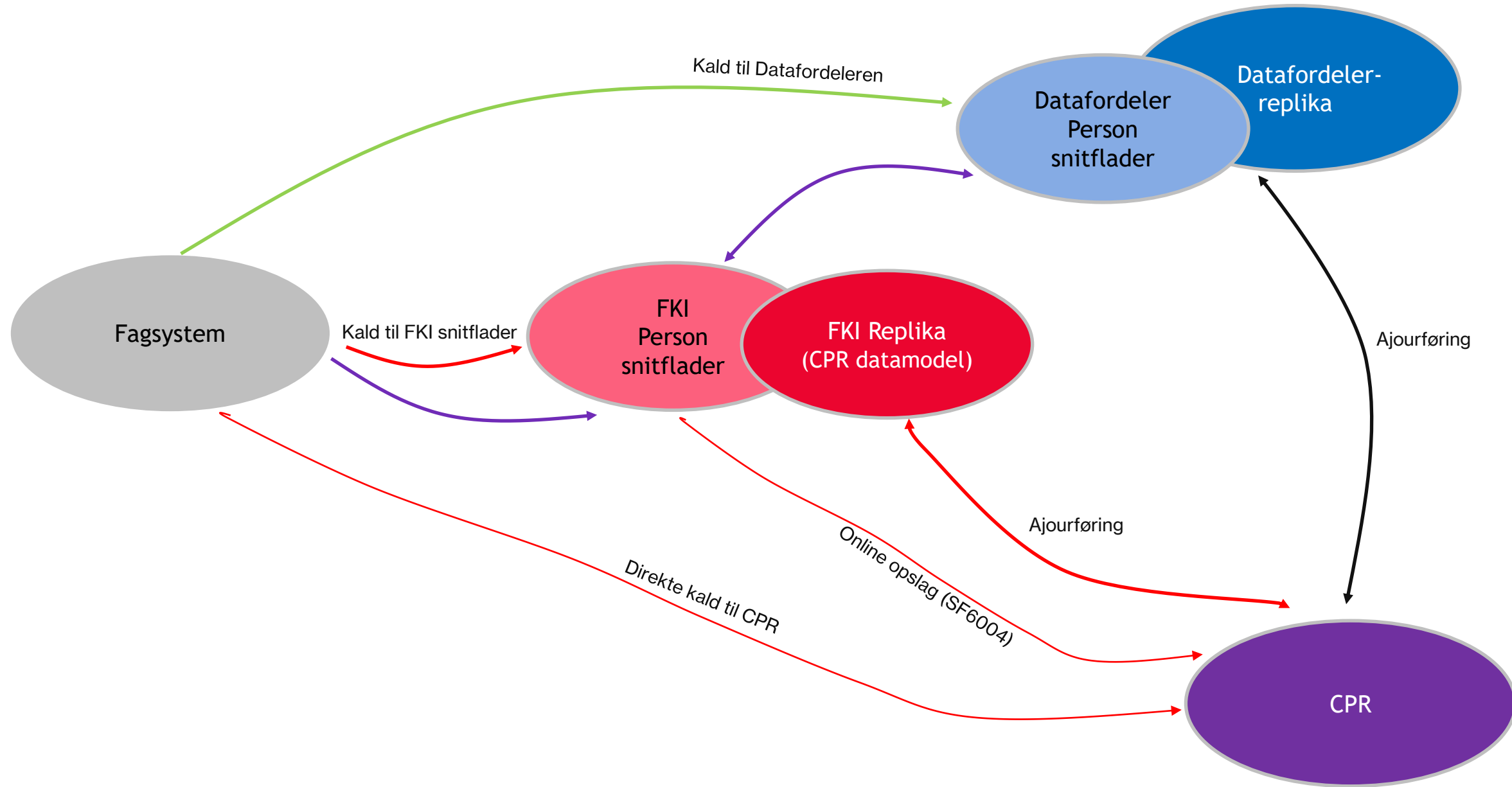
Paralleldrift i grunddataprogrammet

Anvenderne skal sikres en sikker og stabil overgang til grunddata via Datafordeleren. For at nedbringe risici ved anvendernes overgang til grunddata via Datafordeleren har grunddataprogrammet planlagt en **paralleldriftsperiode**, hvor data fra en række registre udstilles i de tidligere formater parallelt med, at Datafordeleren udstiller de nye grunddata.

De fleste registre har aftalt 12 måneders paralleldrift, mens CVR og **CPR opererer med paralleldrift i flere år.**

**HERUDOVER INGEN POLITISKE
HOLDEPUNKTER**

Hvordan kalder fagsystemer persondata?



Transition

Type 1 – anvendere af simple opslag

- Skal ændre få kald
- Minimale transaktionsomkostninger
- Står for ca. 80 % af trafikken

Type 2 - Store systemer – flere integrationer

- Projekter i omegnen af 0,5 mio. pr. system.
- Kan lægges i udbud og konkurrenceudsættes.

- Vi ikke modtaget reaktioner fra anvenderne (ud over KOMBIT-internt)
- Vi må formode, at de fleste bare taget situationen til efterretning og ændrer deres kald.
- En række anvendere går uden om KOMBIT alligevel - måske fordi Datafordeleren er fremtiden ?



Snitflader, der bliver udfaset

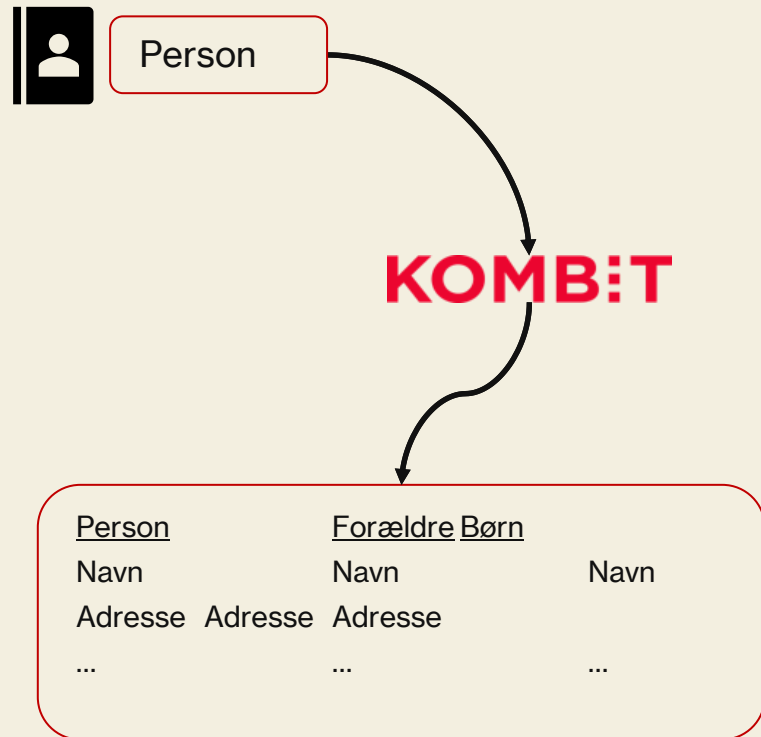
- SF1320_A - CPR – Hændelser
- SF1520 - CPR replika opslag
- SF1525 - Persondata Samling Hent
- SF6001 - ADRSOG1 - CPR Opslag (lokal)
- SF6002 - CPR Abonnement
- SF6005 - Familie - CPR Opslag (lokal)
- SF6006 - Familie+ - CPR Opslag (lokal)
- SF6009 - STAM - CPR Opslag (lokal)
- SF6010 - STAM+ - CPR Opslag (lokal)



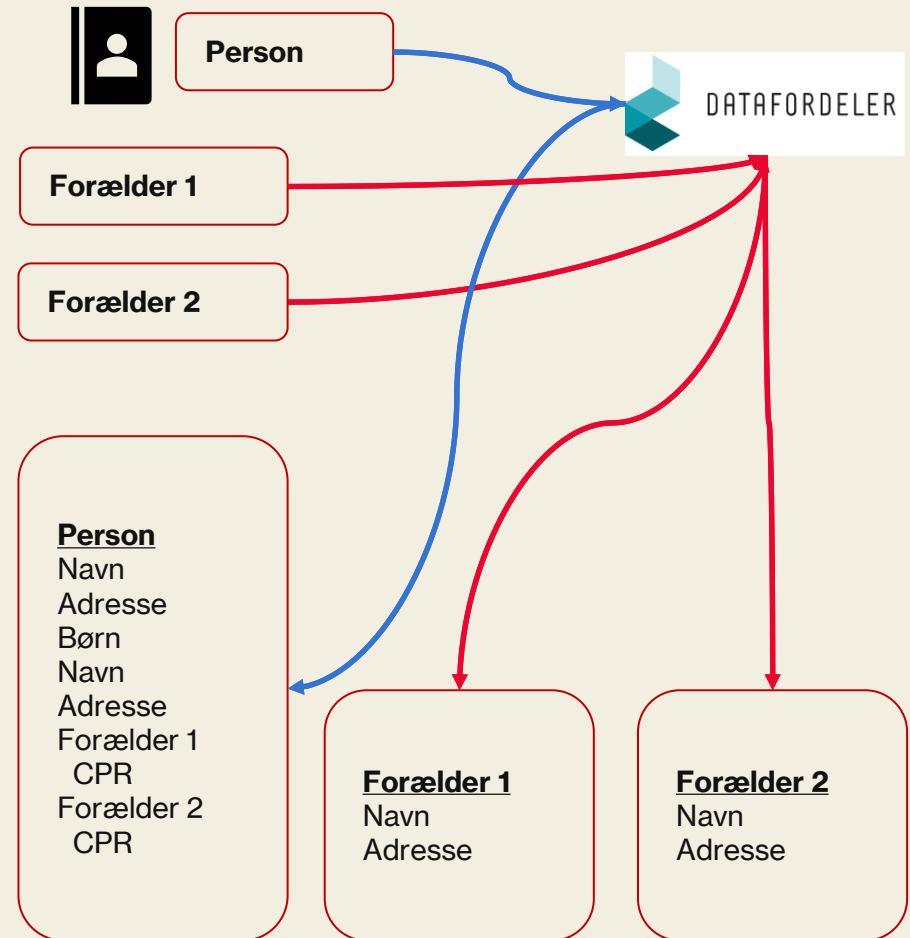
 **CPRPublicSector GraphQL** 

TJENESTE	SERVICE	BESKRIVELSE
Webservice	GraphQL	Hvis du ønsker at tilgå CPR-data hos Datafordeleren via GraphQL på vegne af en offentlig myndighed, skal du først have adgang til CPR Public Sector-tjenesten. Med godkendt adgang til CPR Public Sector-tjenesten, har du mulighed for at tilgå entiteten CPRCustom_PublicSectorPerson via GraphQL. Denne specialudviklede entitet giver mulighed for at tilgå en række af de almindelige entiteter fra CPR-registret og sammenstille disse, så du ikke selv behøver at tage stilling til hvordan data hænger sammen. Bemærk at du med denne tjeneste har mulighed for at hente alle data ,som CPR har tilgængeligt ,og der derfor er et dataminimeringsansvar, som ligger hos anvenderen, i kun at hente den data, der er nødvendigt for at udføre sin funktion i den tilknyttede myndighed.
ADGANG	FILFORMAT	
Offentlig myndighed	JSON	

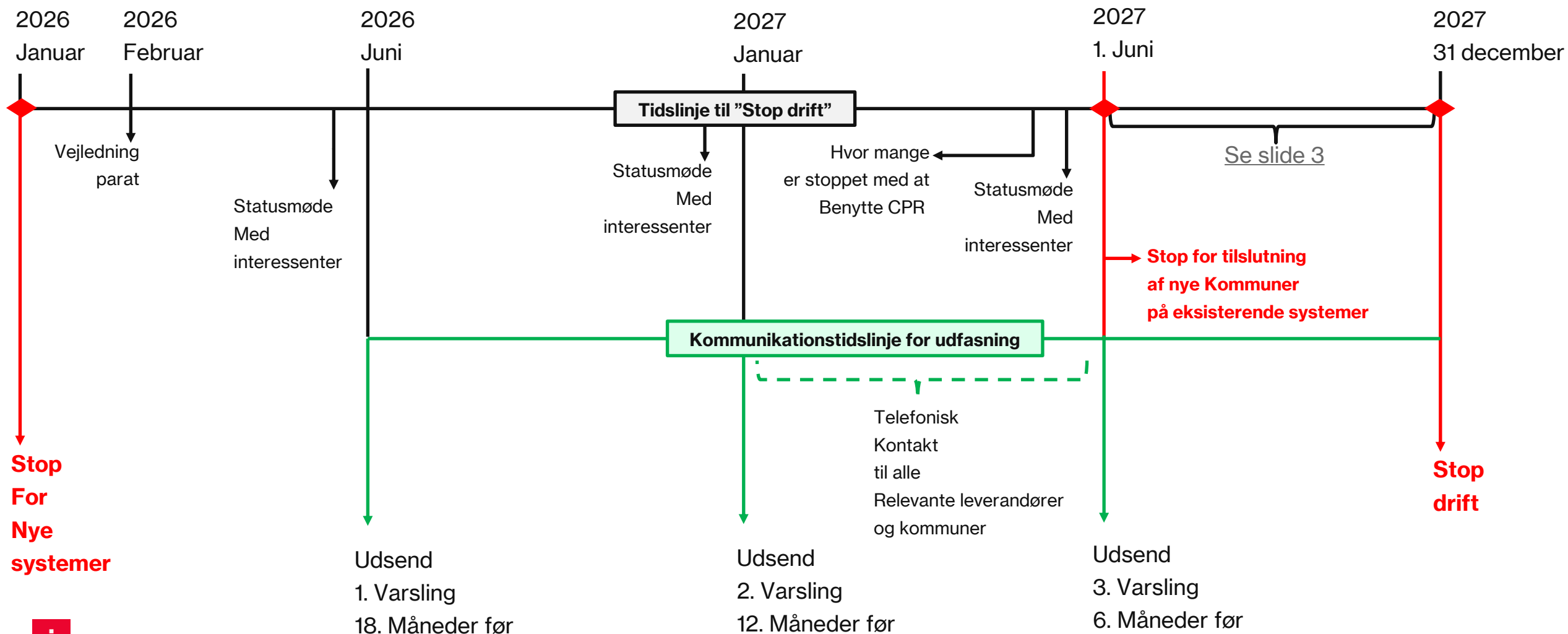
SOAP



GraphQL



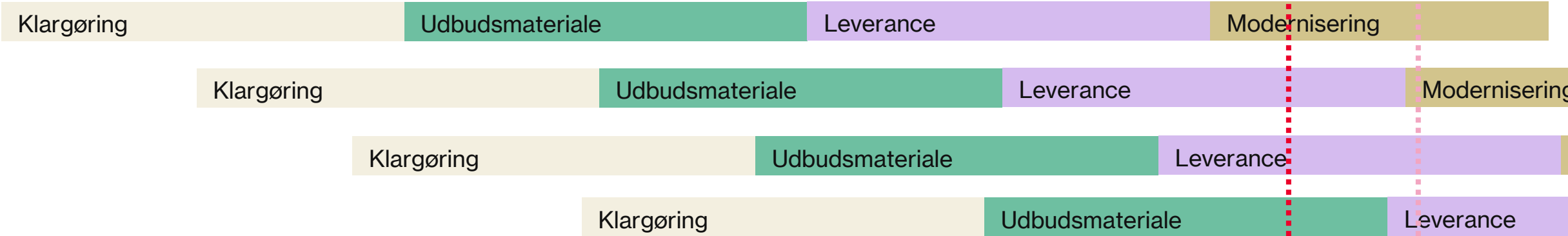
Tidslinje for CPR udfasning – fuld tidslinje



"Uformel" kommunikation sendt januar 2026

KOMBIT udbudsplan

UDFASNING
AF CPR



Ny SKI 02.19



Hvad er det folk spørger om?

- Lige nu benytter vores Gateway samme "API-key" for alle medlemmer, er det ok eller skal vi dele trafikken op på specifikke medlemmers api-keys?
- Skal vi skifte til o-auth?
- Er der nogen begrænsninger på mængden af trafik vi må sende efter Datafordeleren?
- Hvilket format (CSV-file eller JSON liste)?
- **GraphQL samlinger**, Hvad menes der med samlinger?
- **CPR Hændelser**, Hvordan kommer en hændelse?
- **CPR testdata**, Findes der en liste på DAF , der beskriver alle felter?
- Findes der en måde, hvor man kan lave direkte opslag i CPR
- Hvad med revisionlog



Udfordringer

- Vi har ca 14 KOMBIT-løsninger og ca. 185 eksterne
- Forståelse af, hvor KOMBIT ikke længere skal udstille CPR
- Manglende viden om GraphQL
- Manglende finansiering til omlægning
- Udbudsmæssige (planmæssige) udfordringer
- Helt ny måde at tænke hændelseslogik på
- Kodeeksempler – gøre det enkelt
- Dokumentation på DAF er ikke opdelt og opmærket tilstrækkeligt





4. Hændelser



Klimadatastyrelsen

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 (DHMHøjdekurver)
 - 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

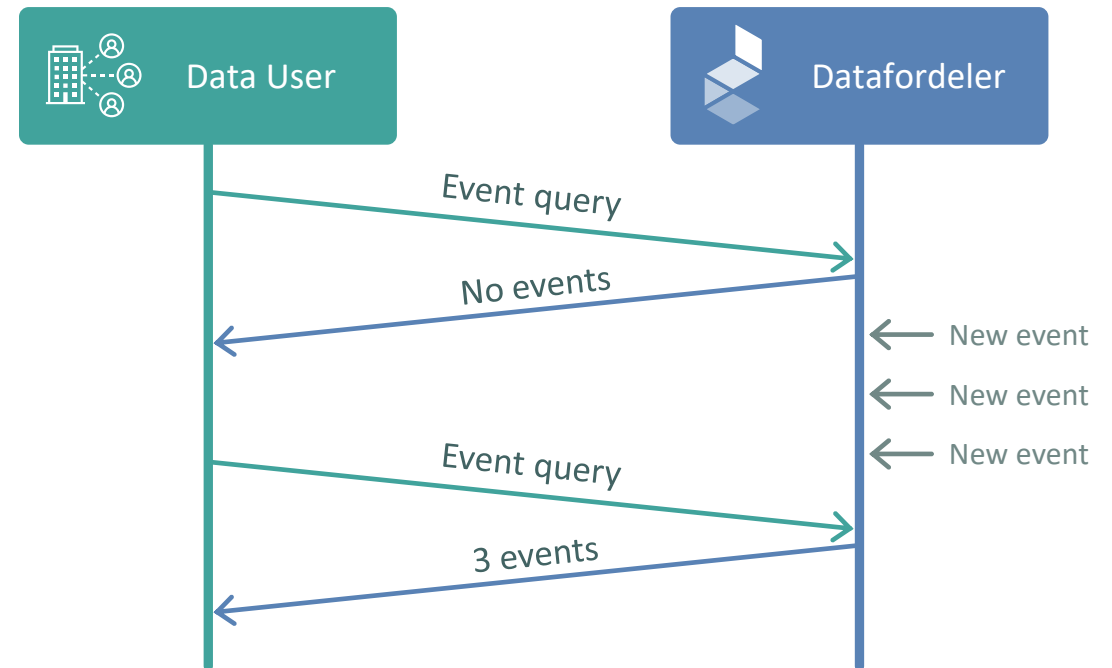
Hændelsesentiteter – Data

- 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

<Register>_Event	
-	datafordelerOpdateringstid: DateTime
-	datafordelerRegisterImportSequenceNumber: int
-	entityName: String
-	eventAction: char(1)
-	eventId: long
-	fromFailedImport: bool
-	modelVersion: String
-	object_datafordelerRowId: UUID
-	object_datafordelerRowVersion: int
-	object_id: String [0..1]
-	object_registreringFra: DateTime [0..1]
-	object_registreringTil: DateTime [0..1]
-	object_status: String [0..1]
-	object_virkningFra: DateTime [0..1]
-	object_virkningTil: DateTime [0..1]

Hændelsesentiteter – Queries

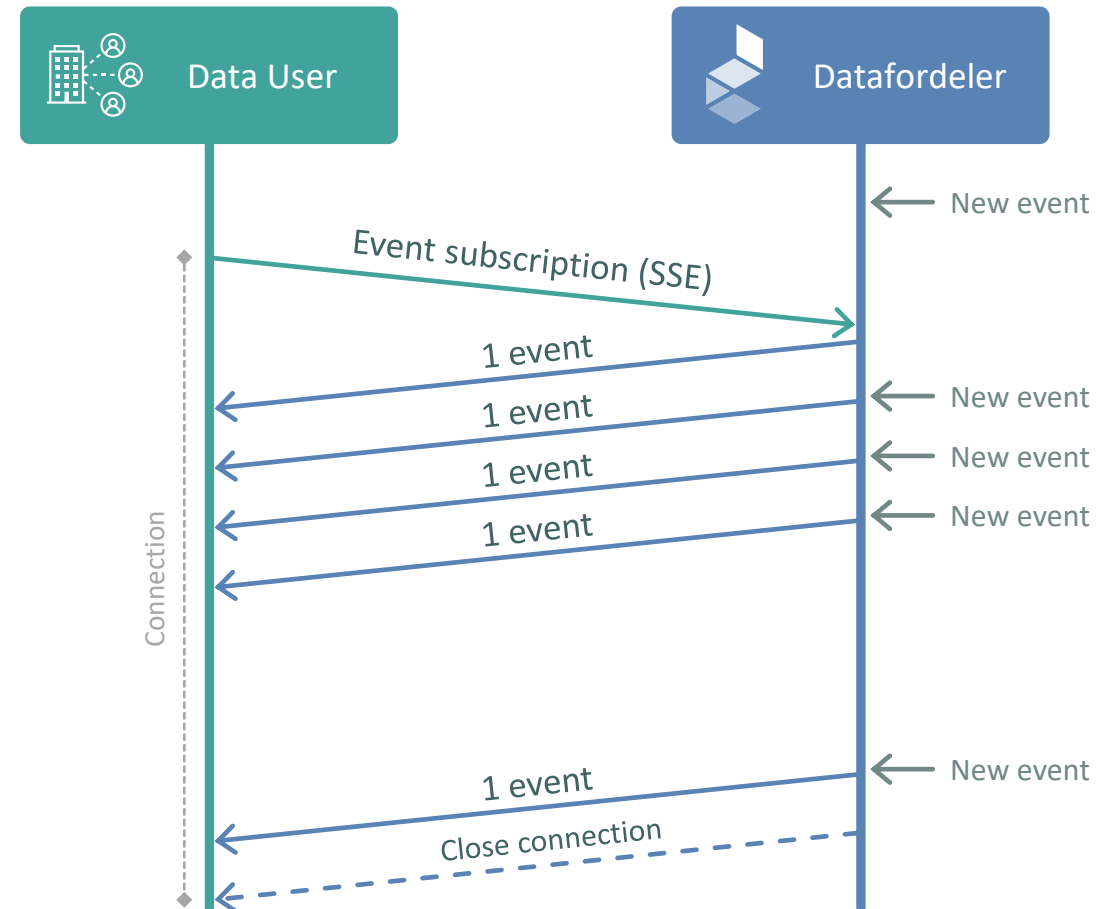
- Kan kun hentes via GraphQL
 - Udstilles af de entitetsbaserede GraphQL-tjenester



Hændelsesentiteter – Subscriptions

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

GraphQL subscription With Server-Sent Event (SSE)



Netcompany

Register Import Status

Register Import Status – GraphQL

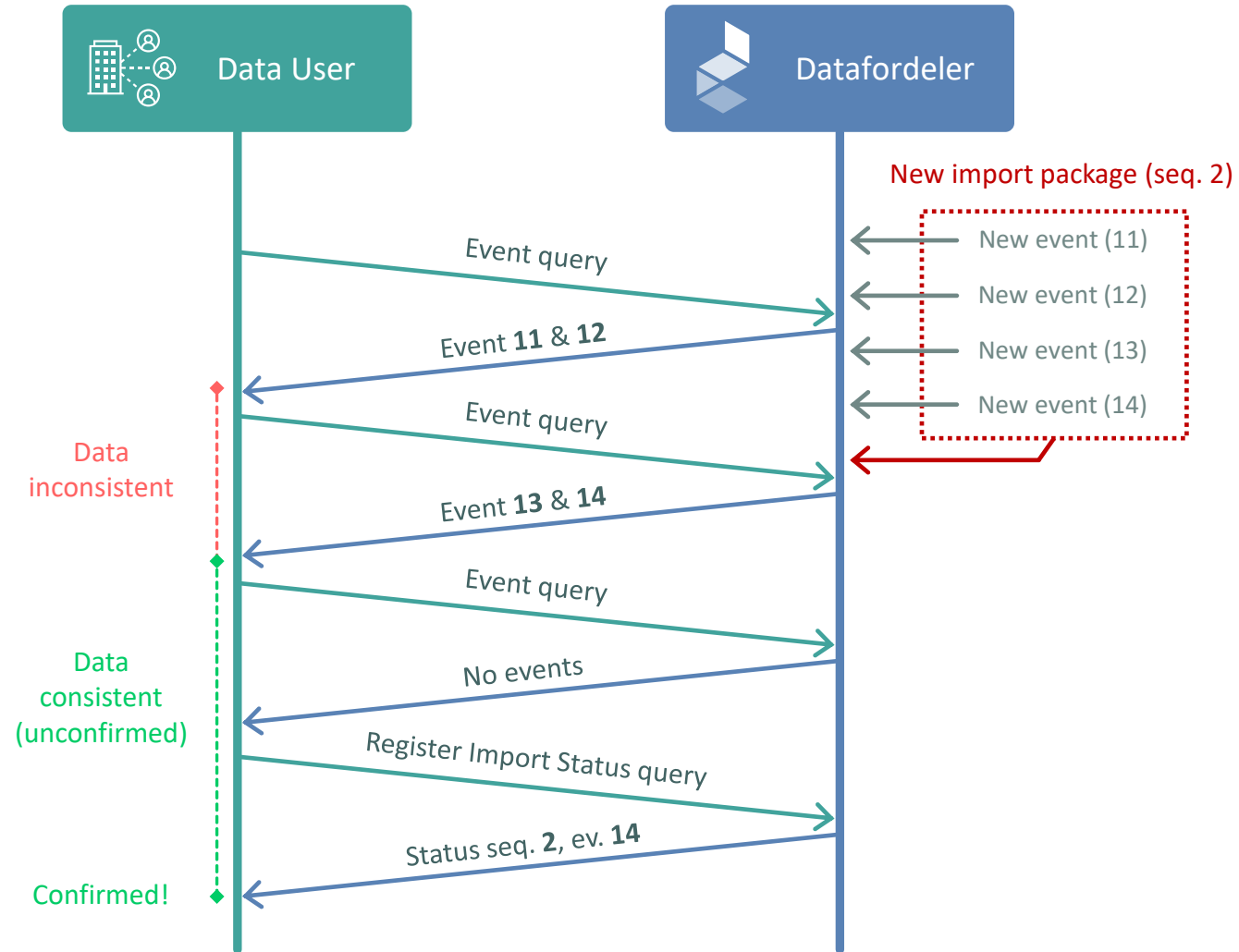
- Ens med hændelser, hvor det udstilles som en entitet i GraphQL
- Opdateres, når en datapakke er indlæst færdig og kan derfor bruges til at sikre dataintegritet

DAF_RegisterImportStatus
- lastEventId: long
- lastSequenceNumber: int
- lastUpdated: DateTime

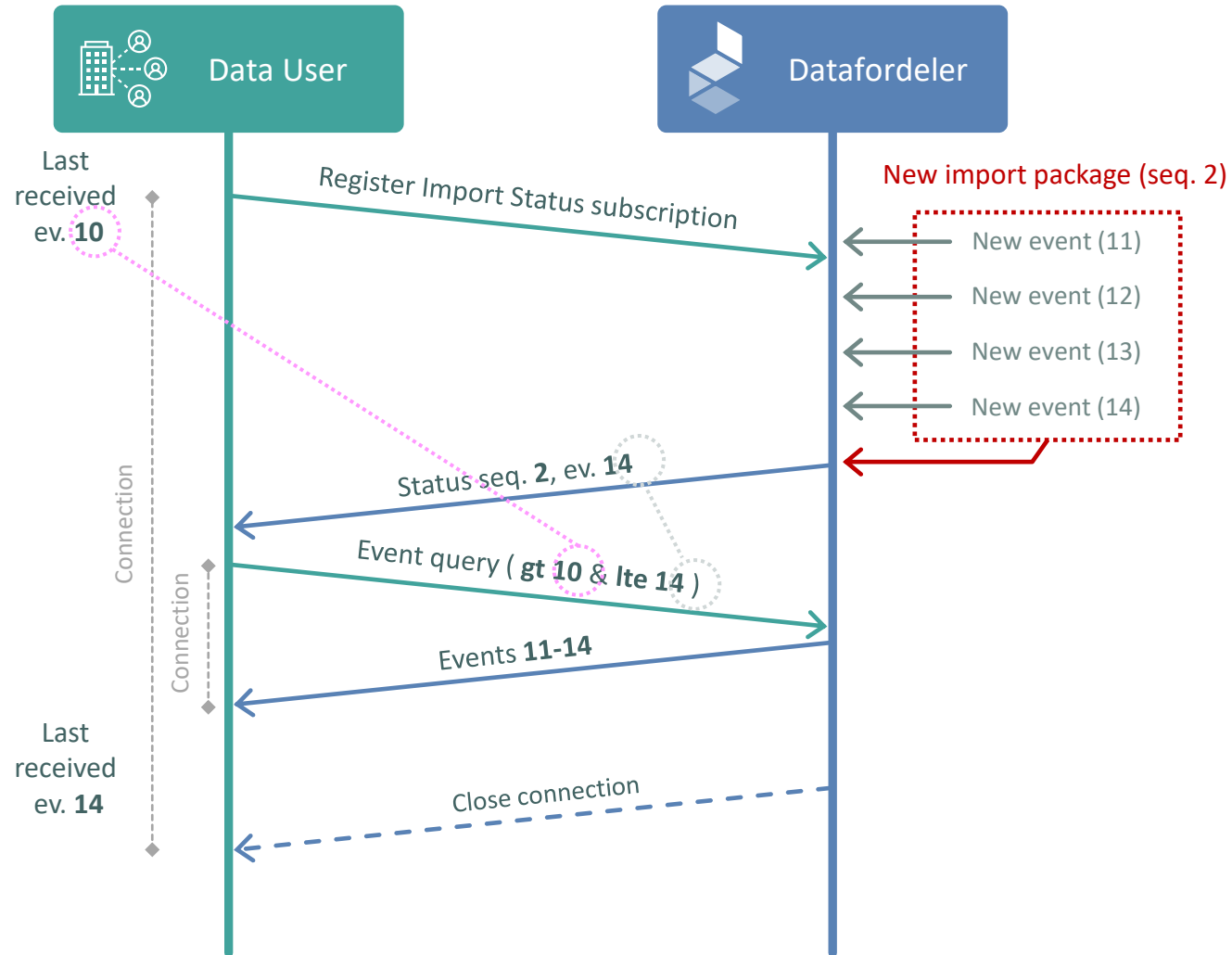
```
subscription {  
  DAF_RegisterImportStatus {  
    lastEventId  
    lastSequenceNumber  
    lastUpdated  
  }  
}
```

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  
  }  
}
```

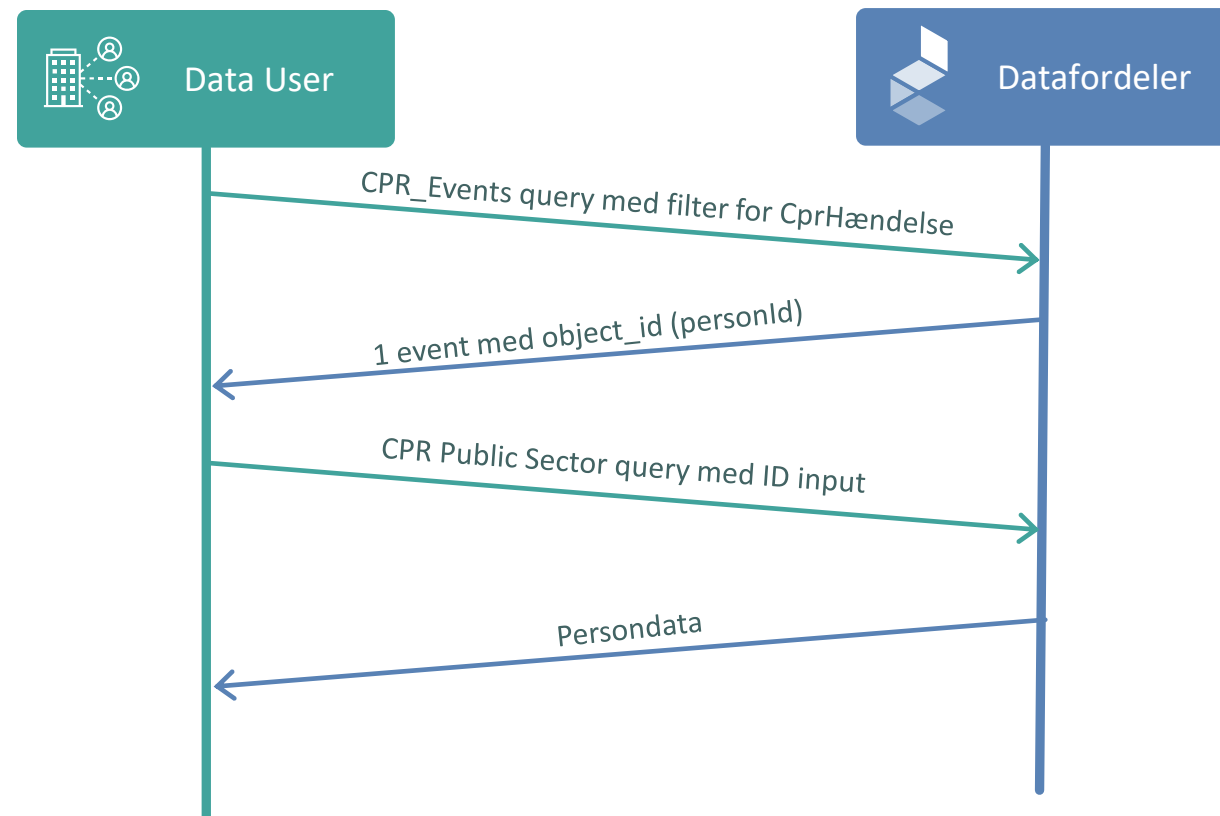
Eksempel på GraphQL-query med forretningshændelsesfiltrering

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) – Kombinerer datanære og forretningshændelser

- Derimod, hvis man har adgang til CPR Public Sector (og dermed forretningshændelser) 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 forretningshændelse, men **ikke** hvilken en



LIVE DEMO

CPR Events Eksempel – Hent hændelser

■ Med EventID:

Request	Response
<pre>1 Run > 2 query { 3 CPR_Events(4 where: { 5 entityname: { eq: "Cprhaendelse" }, 6 eventid: { gt: 550 } } 7) { 8 nodes { 9 entityname 10 eventid 11 object_id 12 } 13 } 14 }</pre>	<pre>1 { 2 "data": { 3 "CPR_Events": { 4 "nodes": [5 { 6 "entityname": "Cprhaendelse", 7 "eventid": 6297086, 8 "object_id": [REDACTED] 9 }, 10 { 11 "entityname": "Cprhaendelse", 12 "eventid": 6297088, 13 "object_id": [REDACTED] 14 }, 15 {</pre>

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

■ Med Opdateringstid:

Request	Response
<pre>1 Run > 2 query { 3 CPR_Events(4 where: { 5 entityname: { eq: "Cprhaendelse" } 6 datafordelerOpdateringstid: { gt: "2020-01-01T12:01:01Z" } 7 } 8) { 9 nodes { 10 entityname 11 object_id 12 datafordelerOpdateringstid 13 } 14 } 15 }</pre>	<pre>1 { 2 "data": { 3 "CPR_Events": { 4 "nodes": [5 { 6 "entityname": "Cprhaendelse", 7 "object_id": [REDACTED] 8 "datafordelerOpdateringstid": "2025-11-03T10:56:58.786599Z" 9 }, 10 { 11 "entityname": "Cprhaendelse", 12 "object_id": [REDACTED] 13 "datafordelerOpdateringstid": "2025-11-03T10:56:58.857054Z" 14 }, 15 {</pre>

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

CPR Events Eksempel – Hent ny Persondata

■ Med forretningshændelse:

```
Request
Run >
query {
  CPRCustom_PublicSectorPerson(
    input: {
      id: { eq: [REDACTED] }
      cprhaendelse: { forretningshaendelse: { eq: "CPRADRESSE" } }
    }
  ) {
    nodes {
      beskyttelser {
        beskyttelsestype
        status
        virkningfra
        virkningtil
      }
      adresseoplysninger(where: { status: { eq: "aktuel" } }) {
        cprAdresse {
          vejadresseringsnavn
        }
        status
      }
    }
  }
}

Response
200 status
{
  "data": {
    "CPRCustom_PublicSectorPerson": {
      "nodes": [
        {
          "beskyttelser": [],
          "adresseoplysninger": [
            {
              "cprAdresse": {
                "vejadresseringsnavn": [REDACTED]
              },
              "status": "aktuel"
            }
          ]
        }
      ]
    }
  }
}
```

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

■ Uden forretningshændelse:

```
Request
Run >
query {
  CPRCustom_PublicSectorPerson(
    input: {
      id: { eq: [REDACTED] }
    }
  ) {
    nodes {
      beskyttelser {
        beskyttelsestype
        status
        virkningfra
        virkningtil
      }
      adresseoplysninger(where: { status: { eq: "aktuel" } }) {
        cprAdresse {
          vejadresseringsnavn
        }
        status
      }
    }
  }
}

Response
200 status
{
  "data": {
    "CPRCustom_PublicSectorPerson": {
      "nodes": [
        {
          "beskyttelser": [],
          "adresseoplysninger": [
            {
              "cprAdresse": {
                "vejadresseringsnavn": [REDACTED]
              },
              "status": "aktuel"
            }
          ]
        }
      ]
    }
  }
}
```

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](https://netcompany.com)



Tak for i dag! Vi ses den 21. april '26

