

Hændelser på datafordeleren

En kort introduktion til begreber og anvendelsesmuligheder

SDFE

version 1.0

24. oktober 2016

Indhold

Indledning.....	2
Overordnet arkitektur	2
Hændelsesbegrebet.....	3
Forretningsmæssige og datanære hændelser.....	3
Hændelsesbeskeder	3
Objektbegrebet	4
Databaserepræsentation.....	4
Dataopdateringer	5
Abonnementer	6
Filter.....	6
Sikkerhedsmodel	6
Persondata.....	6
PUSH af hændelser	6
PULL af hændelser	7
Eksempel.....	7
Anvendelse af hændelser	10
Brugscenarier	10
Hændelser vs. deltafiler.....	10
Andre abonnementer	10
Abonnementer på filudtræk.....	11
Abonnement på Atom feeds	11
Resumé	11
Referencer	12

Indledning

Dette dokument har til hensigt at give en introduktion til begrebet hændelser, som det er implementeret på datafordeleren. Derudover skitseres mulige anvendelser. For oplysninger om hvilke hændelser og data, der er tilgængelige på datafordeleren samt vilkår for anvendelse henvises til <http://datafordeler.dk>

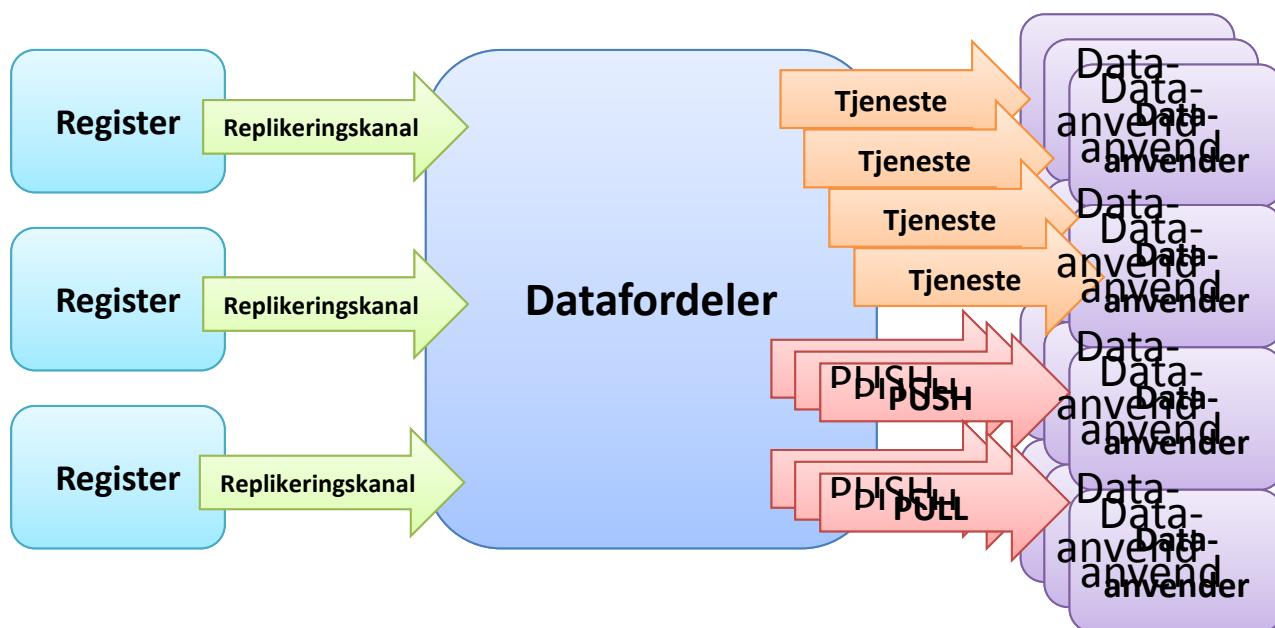
For teknisk dokumentation af snitfladerne til datafordeleren, og hvordan man tilgår dem, henvises desuden til [DAF]. For generel information om hændelsesbaseret eller beskedbaseret arkitektur (på engelsk: Event-driven architecture, EDA) henvises til [EDA].

I de følgende afsnit introduceres først den overordnede arkitektur, hvorefter hændelsesbegrebet og hændelsesbeskeder beskrives. Herefter beskrives objekter og deres repræsentation i databaser med dobbelthistorik, dataopdateringer samt abonnemeter. Et gennemgående eksempel fremstiller det samlede flow fra register til dataanvender, og endelige kommer vi ind på mulige anvendelser af hændelser samt andre former for abonnemeter på datafordeleren end hændelsesabonnemeter.

Overordnet arkitektur

Figur 1 viser den overordnede arkitektur for distribution af data via datafordeleren. Data bliver oprettet og vedligeholdt i *grunddataregistre* og løbende kopieret til datafordeleren via *replikeringskanaler*.

Dataanvendere kan hente data fra datafordeleren gennem forskellige former for tjenester, herunder filudtræk. Udover selve data, distribuerer datafordeleren også *hændelsesbeskeder*. For at modtage hændelsesbeskeder skal en dataanvender oprette et *abonnement* på den type af hændelsesbeskeder, der ønskes. Ved opsætning af abonnemeter er der mulighed for at filtrere yderligere i hændelsesbeskederne på baggrund af deres indhold, og der vælges mellem at få tilsendt hændelsesbeskeder (PUSH scenarie) eller selv hente hændelsesbeskeder på datafordeleren (PULL scenarie).



Figur 1: Distribution af data via datafordeleren.

Hændelsesbegrebet

En hændelse er, når der sker noget. I et grunddataregister betyder det, at der sker en ændring af data. En hændelse kan således være resultatet af et trin i en forvaltningsproces (også omtalt som en forretningsproces), som opdaterer data i et register.

I daglig tale skelner man ofte ikke skarpt mellem hændelser og hændelsesbeskeder, men ofte benyttes termene synonymt. Dette dokument tilstræber at afspejle den typiske sprogbrug omkring hændelser.

Forretningsmæssige og datanære hændelser

På datafordeleren er der mulighed for to former for hændelser – forretningsmæssige og datanære.

Et register kan selv danne hændelsesbeskeder på baggrund af *forretningsmæssige hændelser* (fra forvaltningsprocesser i registeret) og sende dem til datafordeleren via en særskilt replikeringskanal til videre distribution til abonnenter.

Datanære hændelser dannes af datafordeleren på baggrund af dataopdateringer modtaget på replikeringskanaler. Dannelse og distribution af en hændelsesbesked er derfor bundet til en opdatering af data. Herved undgår man situationer, hvor man fx udsender (forretningsmæssige) hændelsesbeskeder om ændringer i data, men alligevel ikke får gennemført en opdatering af data, som er tilgængelige på datafordeleren.

For en dataanvender er der i princippet ikke forskel på disse former for hændelser. De har samme format, som beskrives i et senere afsnit, mens indholdet af hændelsesbeskederne selvsagt kan variere.

I det følgende koncentrerer vi os om datanære hændelser.

Hændelsesbeskeder

En hændelsesbesked har en beskedtype. For datanære hændelser er beskedtyper dannet ved at sammensætte objekttype og operation på objektet. Operationen på objektet kan være Create, Update eller Delete. Et eksempel på en beskedtype er `KommuneinddelingUpdate`, der benyttes til Update-hændelser for objekttypen `Kommuneinddeling`.

Hændelsesbeskeder har et fast format mht. struktur, men vil være tilgængelige i XML og JSON. Det fælles beskedformat for hændelsesbeskeder, der benyttes i Grunddataprogrammet, er beskrevet i [Grunddatabesked].

Figur 2 viser den overordnede struktur i hændelsesbeskeder.

En hændelsesbesked indeholder en beskedkuvert og beskeddata. Beskedkuverten indeholder blandt andet filtreringsdata, der kan benyttes til filtrering af hændelsesbeskeder i forbindelse med abonnementer. Filtreringsdata indeholder en eller flere objektregistreringer, der indeholder feltet `Objekthandling`. Praksis omkring brug af dette felt er, at registrene kan benytte det til at overlevere ekstra information omkring hændelsen, som ikke er direkte afledt af selve data. Det kan fx være en label for den forretningsmæssige årsag til hændelsen eller en liste med navne på de felter, der er ændret ved dataopdateringen. Ved opsætning af abonnementer kan disse ekstra informationer i feltet `Objekthandling` udnyttes til filtrering.



Figur 2: Struktur for indhold i hændelsesbeskeder.

Objektbegrebet

For at forstå dannelsen af datanære hændelser på datafordeleren er det nyttigt at forstå, hvordan objekter repræsenteres i databaser og kopieres fra registre til datafordeleren via replikeringskanaler.

I datamodellering skelner man ofte mellem konkrete (fysiske eller konceptuelle) objekter/entiteter og deres repræsentation i form af forvaltnings- eller forretningsobjekter. I dette dokument omtaler vi for nemheds skyld blot forvaltnings- eller forretningsobjekter som objekter, idet fokus ikke er på modellering og samspil med virkeligheden, men udelukkende på repræsentationen og operationer i systemerne.

Databaserepræsentation

Objekter repræsenteres i databasen med rækker. Det betyder at ét objekt og dets historik er repræsenteret i databasen med en eller flere rækker. I Grunddataprogrammet indeholder data bitemporale egenskaber (dobbelthistorik). Dvs. at en række indeholder information om registreringstid og virkningstid i form af tidsintervaller. *Registreringstid* angiver, hvornår informationerne om objektet var kendt. *Virkningstid* angiver, hvornår informationerne er gyldige i forhold til virkeligheden.

En ændring i et objekt kan betyde ændringer i flere rækker i databasen. Oprettelse af et objekt bevirker oprettelse af en ny databaserække. Ændring af et objekt med bitemporale egenskaber kan fx bevirke, at der oprettes en ny databaserække med registreringFra = t og registreringTil = null, mens databaserækken for seneste forekomst af objektet får sat registreringTil = t. Sletning af et objekt udføres som regel som "soft-delete" ved at sætte registreringTil = t. Sletning kan dog også ske ved at virkningstiden sættes.

Oprettelse og opdatering (fejlrrettelse) af et objekt er illustreret i figur 3. Her oprettes et objekt først ved at oprette en ny række, hvorefter objektet opdateres ved at ændre registreringstid i rækken og oprette en ny række med det rettede data. I eksemplet er der blot tale om en fejlrrettelse, dvs. der rettes blot en stavefejl og der ændres ikke i virkningstid. Hvis der havde været tale om en decideret ændring af stavemåde, ville

det have givet anledning til to nye række for at holde styr på virkningstiden for den gamle og den nye stavemåde. Attributter med status, der angiver, hvor objektet er i sin livscyklus, samt registrerings- og virkningsaktører er ikke taget med i eksemplet.

Oprettelse af objekt - ny række i databasen:

ID	UUID	Data	Reg. Fra	Reg. Til	Virk. Fra	Virk. Til
1	xxxx	Grøndahl	01-09-2016	-	01-10-2016	-

Opdatering af objekt (fejlrettelse) - ændring i eksisterende række samt ny række i databasen:

ID	UUID	Data	Reg. Fra	Reg. Til	Virk. Fra	Virk. Til
1	xxxx	Grøndahl	01-09-2016	07-09-2016	01-10-2016	-
2	xxxx	Grøndal	07-09-2016	-	01-10-2016	-

Figur 3: Oprettelse og opdatering af objekt (fejlrettelse).

For en nærmere gennemgang af bitemporalitet, som det benyttes i Grunddataprogrammet, henvises til [Bitemporalitet] og [Dobbelthistorik]. Derudover er krav til bitemporalitet i datamodellering beskrevet i modelreglerne for Grunddataprogrammet [Modelregler].

Dataopdateringer

Registrene overfører data via replikeringskanaler til datafordeleren i XML¹. Når et grunddataregister sender dataopdateringer til datafordeleren, svarer de til opdateringer af rækker i databasen. En databaserække kan oprettes, opdateres og slettes.

Registeret har mulighed for at lade en dataopdatering (operation på en databaserække) danne en datanær hændelse. Registeret kan triggere en hændelse med en type svarende til operationen på objektet set fra et forvaltningsmæssigt synspunkt, og beskedtypen følger således ikke nødvendigvis dataopdateringens type. En oprettelse af en databaserække kan triggere en Update-hændelse. En opdatering af en databaserække kan triggere en Delete-hændelse. En dataopdatering behøver ikke nødvendigvis danne en datanær hændelse. Ved en opdatering af et objekt, som omfatter oprettelse og opdatering af flere rækker i databasen, kan registeret således lade en enkelt af dataopdateringerne triggere dannelsen af en datanær hændelse svarende til den forretningsmæssige ændring af objektet, mens de resterende dataopdateringer ikke giver anledning til hændelser.

¹ Der findes også replikeringskanaler til overførsel af rasterdata, fx skærmbort i billedfiler, men her betragter vi kun registerdata i relationelle databaser.

Dataanvendere vil kun se de hændelsesbeskeder, der bliver dannet til abonnementerne, og behøver ikke forholde sig til de ændringer og oprettelser af rækker i databasen, der ligger til grund. Registeret har taget stilling til dannelsen af de datanære hændelser, og dataanvenderne får oplysninger gennem hændelsesbeskeder og selve data udstillet i tjenester, herunder filudtræk.

Abonnementer

En dataanvender skal have oprettet et abonnement, før det er muligt at få tilsendt eller hente hændelsesbeskeder. Et abonnement er tilknyttet en adgang til data (gennem en tjenestebruger), og begge dele oprettes i selvbetjeningsportalen.

Filter

Det er muligt at opsætte et filter til et abonnement, således at abonnementet kun får hændelsesbeskeder, der opfylder betingelserne i filteret. Der kan filtreres på de felter, der er i filtreringsdata i hændelsesbeskeden. I filteret kan en række operatorer benyttes til at sammenligne indhold i felterne med konstante værdier.

For feltet Objekthandling er det som noget specielt muligt at filtrere på, om det indeholder et bestemt ord. Hvis Objekthandling benyttes af registeret til at angive en liste med navne på ændrede felter, kan en dataanvender således opsætte et filter til kun at modtage hændelser, der omhandler ændringer i et bestemt felt.

Sikkerhedsmodel

Nogle hændelser kræver, at registeret giver adgang til at dataanvenderen kan abonnere på dem. Dataanvendere kan anmode om adgang til hændelser i selvbetjeningsportalen.

Persondata

Datafordeleren distribuerer persondata fra CPR-kontoret, og de lovmæssige restriktioner omkring udlevering af persondata til private dataanvendere gælder også hændelser om persondata. Offentlige dataanvendere er ikke underlagt samme restriktioner for udlevering af persondata som private dataanvendere, men de er til gengæld underlagt andre krav om behandling af persondata. For standardvilkår for brug af CPR henvises til [CPRVilkårOff] og [CPRVilkårPrivat].

PUSH af hændelser

For at datafordeleren kan sende hændelsesbeskeder til en dataanvender, skal dataanvenderen implementere en tjeneste, der kan modtage hændelsesbeskeder. Tjenesten skal følge OData v4.0 protokollen og have et HTTPS-end-point, som datafordeleren vil tilgå med et FOCES certifikat.

Ved PUSH bliver hændelsesbeskederne sendt til dataanvenderen umiddelbart efter, de er dannet. Hvis det ikke lykkes at aflevere hændelsesbeskederne, vil datafordeleren forsøge igen én gang i timen, indtil hændelsesbeskederne er leveret, eller der er gået 120 timer, hvorefter abonnementet vil blive deaktiveret og hændelsesbeskederne slettet.

For nærmere beskrivelse henvises til [DAF].

PULL af hændelser

Dataanvenderen kan hente hændelsesbeskeder ved at kalde en tjeneste på datafordeleren svarende til det opsatte abonnement.

Ved PULL er det op til dataanvenderen at hente hændelsesbeskederne, som vil være tilgængelige umiddelbart efter, de er dannet. Hændelsesbeskederne vil være tilgængelige for afhentning i 120 timer, hvorefter de slettes.

For en beskrivelse af, hvordan man opretter et abonnement på hændelser og henter hændelsesbeskeder, henviser vi til [QuickGuide] og [DAF].

Eksempel

I dette afsnit vil vi gennemgå flowet fra opdatering til udsendelse af hændelser med et eksempel. Noget af eksemplet illustrerer, hvordan opdateringer fra et register styrer dannelsen af datanære hændelser. Det er en proces som ikke er synlig for dataanvenderen, men er taget med her af hensyn til fuldstændighed og introduktion til registre, som har behov for at forstå hele flowet. I eksemplet betragtes en simpel fejlrettelse – andre opdateringer kan give anledning til mere end én ny række i databasen. Bemærk at eksemplet kan afvige fra den endelige opsætning af det konkrete register på datafordeleren.

Flowet starter ved at data bliver ændret i et register. Det kan fx være, at der er blevet konstateret en fejl i geometrien for en kommune og den bliver rettet – dvs. der sker en fejlrettelse i geometrien for en KommuneInddeling i DAGI-registreret. Denne opdatering af data bevirker, at der bliver sendt en opdatering til datafordeleren.

Da der er bitemporalitet i data, bliver opdateringen foretaget ved, at den eksisterende forekomst af kommunen bliver gjort historisk, og der bliver oprettet en ny forekomst med den opdaterede geometri. Det sker i databasen ved at en eksisterende række får ændret RegistreringTil fra null til dags dato, og der bliver oprettet en ny række med RegistreringFra = dags dato, RegistreringTil = null og opdateret geometri i data.

Disse ændringer bliver sendt til datafordeleren i XML med et element for oprettelsen og et element for ændringen (eksemplet er forkortet af hensyn til overskuelighed og kan afvige fra den endelige implementering):

```
<?xml version='1.0' encoding='UTF-8'?>
<cmn:DatafordelerDataDelivery ... >
<cmn:Header>
...
</cmn:Header>
  <cmn:Data>
    <cmn:DataEntity>
      <cmn:Action>Create</cmn:Action>
      <cmn:CreateEvent>UpdateEvent</cmn:CreateEvent>
      <dagi:Kommuneinddeling>
        <dagi:objectid>20059189</dagi:objectid>
        <dagi:feltliste>geometri</dagi:feltliste>
        ...
      <dagi:id_namespace>http://data.gov.dk/dagi</dagi:id_namespace>
      <dagi:id_lokalId>10389119</dagi:id_lokalId>
    </cmn:DataEntity>
  </cmn:Data>
</cmn:DatafordelerDataDelivery >
```

```

    ...
    <dagi:landekode>DK</dagi:landekode>
    <dagi:skala>1:10.000</dagi:skala>
    <dagi:geometri> ... </dagi:geometri>
    ...
  </dagi:Kommuneinddeling>
</cmn:DataEntity>
<cmn:DataEntity>
  <cmn:Action>Update</cmn:Action>
  <cmn:CreateEvent>NoEvent</cmn:CreateEvent>
  <dagi:Kommuneinddeling>
    <dagi:objectid>10059189</dagi:objectid>
    <dagi:registreringTil>2016-08-07T02:10:00.000000</dagi:registreringTil>
  </dagi:Kommuneinddeling>
</cmn:DataEntity>
</cmn:Data>
</cmn:DatafordelerDataDelivery>

```

Den eksisterende række bliver opdateret og gjort historisk – dette skal ikke udløse en hændelse. Oprettelsen af en ny række med opdateret data bliver derimod brugt til at trigge dannelse af en datanær hændelse ved at angive elementet *<CreateEvent>* med værdien *UpdateEvent*. Da der set fra et forretningsmæssigt perspektiv er tale om en opdatering af data, dannes der en Update-hændelse.

Dette er et eksempel på, at typen af den datanære hændelse ikke er bestemt af databaseoperationen, men at det alene er registreret, der styrer beskedtypen ved et element i opdateringen.

Der dannes en datanær hændelsesbesked med beskedtypen *KommuneInddelingUpdate*. Det er i forvejen konfigureret, hvilke informationer, der skal fyldes i en hændelsesbesked af denne type. Informationerne hentes typisk fra dataopdateringen selv.

Dataopdateringer kan indeholde oplysninger, som ikke er en del af de data, der gemmes og udstilles af datafordeleren, men som udelukkende benyttes til at danne hændelsesbeskeder. Elementet *feltliste* i dataopdateringen indeholder en liste af navne på felter, som er ændret ved dataopdateringen. Ved ændring af geometrien for en kommune, indeholder opdateringen "geometri" i elementet feltliste.

Som dataanvender skal man i forvejen have sat et abonnement op for at få den hændelsesbesked, der bliver dannet. Hvis man i sit abonnement har valgt at hente hændelsesbeskeder (PULL-scenariet), vil hændelsesbeskeden blive gjort tilgængelig i en RESTful tjeneste med følgende signatur:

<https://services.datafordeler.dk/system/EventMessages/1.0.0/custom?datefrom=2016-10-06&dateto=2016-10-07&format=json&username=xxx&password=yyy>

Ved at kalde tjenesten, kan man hente hændelsesbeskederne for en given periode for de abonnementer, der er sat op for dataanvenderen. Det er først her, at dataanvenderen kommer i kontakt med hændelserne – dataopdateringerne fra registeret til datafordeleren er ikke direkte synlige for dataanvenderen, men bliver afspejlet i de genererede hændelser.

En hændelsesbesked dannet ud fra opdateringen i eksemplet kunne se ud som følger (eksemplet er forkortet af hensyn til overskuelighed og kan afvige fra den endelige implementering):

```
<?xml version="1.0" encoding="utf-16"?>
```



```

<Haendelsesbesked xmlns:urn="urn:dk:grunddata:1.0.0" xmlns="urn">
  <Beskedversion>1.0</Beskedversion>
  <BeskedId>f90c19c0-39da-45f2-9093-fd6afe4335ca</BeskedId>
  <Beskedkuvert>
    <Filtreringsdata>
      <Beskedtype>KommuneinddelingUpdate</Beskedtype>
      <BeskedansvarligAktoer />
      <RelateretObjekt />
      <Objektregistrering>
        <Registreringsaktoer>Systemetablering</Registreringsaktoer>
        <Registreringstid>2016-08-07T02:10:00.000000</Registreringstid>
        <Status>vedtaget</Status>
        <ObjektansvarligAktoer>Social- og Indenrigsministeriet</ObjektansvarligAktoer>
        <ObjektId>20059189</ObjektId>
        <Objekttype>Kommuneinddeling</Objekttype>
        <Objekthandling>geometri</Objekthandling>
        <Opgaveemne>52.20.10.15</Opgaveemne>
        <RegistreringsId>10389119,2016-08-07T02:10:00.000000</RegistreringsId>
        <Stedbestemmelse>
          <StedbestemmelseReference>MULTIPOLYGON (((738763.98 6169447.66 ... )))
          </StedbestemmelseReference>
        </Stedbestemmelse>
      </Objektregistrering>
      <TvaergaaendeProces>systemetablering</TvaergaaendeProces>
    </Filtreringsdata>
    <Leveranceinformation>
      <Dannelsestidspunkt>2016-08-07T07:01:24.5270830+02:00</Dannelsestidspunkt>
      <Kildesystem>http://data.gov.dk/id/itsystem/basicdata#d45f25737a3972bd68
ec2d9f043bb7e479fc1b6f</Kildesystem>
      <Sikkerhedsklassificering>http://data.gov.dk/vocabulary/security/
confidentiality#NonConfidential</Sikkerhedsklassificering>
    </Leveranceinformation>
  </Beskedkuvert>
  <Beskeddata>
    <Objektreference>20059189</Objektreference>
  </Beskeddata>
</Haendelsesbesked>

```

Når hændelsesbeskeden hentes (PULL-scenariet), vil den være pakket ind i et format, der kan indeholde flere hændelsesbeskeder:

```

<?xml version="1.0" encoding="utf-8"?>
<ArrayOfEnvelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Envelope>
    <Id>1878</Id>
    <Message>
      <?xml version="1.0" encoding="utf-16"?>
      <Haendelsesbesked xmlns:urn="urn:dk:grunddata:1.0.0" xmlns="urn">
        <Beskedversion>1.0</Beskedversion>
        <BeskedId>f90c19c0-39da-45f2-9093-fd6afe4335ca</BeskedId>
        ...
      </Haendelsesbesked>
    </Message>
    <Format>Xml</Format>
    <Timestamp>2016-08-07T07:02:17.0601880+02:00</Timestamp>
  </Envelope>
</ArrayOfEnvelope>

```

Når hændelsesbeskeden sendes til dataanvenderen (PUSH-scenariet), vil den tilsvarende være pakket ind i et JSON-format.

Anvendelse af hændelser

Hændelser benyttes til at binde processer på tværs af grunddataregistrene sammen. Når en forvaltningsproces opdaterer data i ét grunddataregister, kan det igangsætte en forvaltningsproces for opdatering af data i et andet grunddataregister.

Brugsscenarier

Hvis du selv har data, som er relateret til grunddata eller afledt af grunddata, vil de løbende skulle opdateres som konsekvens af ændringer i grunddata. Her kan hændelser benyttes til at trigge opdatering af dine data.

Et simpelt eksempel på data afledt af grunddata kunne være et såkaldt heatmap for matrikulære aktiviteter. Dvs. et kort, hvor farven for et område afhænger af antallet af udstykninger inden for en given periode, fx det seneste år. Dette kort kan genereres ud fra matrikulære data med historiske oplysninger. I stedet for at generere kortet gentagne gange, kunne man vedligeholde datagrundlaget for kortet ud fra ændringer, som der løbende kommer hændelser om.

Hændelser vs. deltafiler

En del dataanvendere har i dag lokale kopier af data liggende i deres systemer og opdaterer disse "skyggeregistre" ved hjælp af såkaldte deltafiler, dvs. filer med ændringer i data. Hændelser indeholder oplysninger om, at data er opdateret, og kan derfor i princippet benyttes til at trigge opdatering eller vedligehold af lokale data hos en dataanvender.

Datafordeleren danner ikke deltainformation i forbindelse med datanære hændelser. Som tidligere omtalt, vil nogle registre sende oplysninger om opdaterede felter i feltet Objekthandling og der kan medsendes oplysninger i beskedata, men det vil som udgangspunkt være op til at dataanvenderen at hente tidligere og nye versioner af objekter, hvis der er behov for at sammenligne. I hændelsesbeskederne angives typisk referencer til objekter i form af deres id'er.

Data med bitemporale egenskaber kan udstilles i tjenester på en sådan måde, at man kan fremsøge data for bestemte tidspunkter og tidsperioder. Med sådanne tjenester vil det derfor være muligt at fremsøge historiske data, således at man kan få historik for data uden at have abonneret på hændelser i en given periode. Registerne bestemmer hvilke tjenester, der skal udbydes på datafordeleren, og det er således op til registrene at understøtte behov for historik i tjenester og eventuelle særlige tjenester.

Andre abonnementer

Der er andre former for abonnementer end hændelsesabonnementer på datafordeleren, og det kan give anledning til forvirring, når der blot tales om "abonnementer". Det er muligt at opsætte abonnementer på filudtræk og at abonnere på opdateringer af tjenester ved hjælp af Atom feeds.

Abonnementer på filudtræk

Dataanvendere kan oprette abonnement på et filudtræk, der udvælger en delmængde af data og dannes ved et angivet tidsinterval. Når der er genereret et nyt filudtræk (med nye data), får dataanvenderen en e-mail med link til filudtrækket. Filudtrækket kan hentes via FTP, SFTP eller en tjeneste på datafordeleren, alt efter hvordan man identificerer sig (brugernavn/password, certifikater eller SSH2-nøgler) og sikkerhedsniveau for data.

Abonnement på Atom feeds

Dataanvendere har mulighed for at blive adviseret, når en ny version af en given tjeneste publiceres. I selvbetjeningsportalen findes oplysninger om Atom feeds for tjenester under detaljevisningen – her kan man finde og kopiere webadressen på tjenestens Atom feed. Man kan herefter oprette et feed-abonnement ved fx at tilføje et nyt feed i Microsoft Outlook eller en anden applikation med angivelse af webadressen på tjenestens Atom feed.

Når man har sat et feed-abonnement op for en given tjeneste, bliver man adviseret, når en ny version af tjenesten publiceres. Det sker ved, at den applikation (feed reader), man benytter, ved jævne mellemrum tjekker for nyt indhold i Atom feed'et på datafordeleren.

Resumé

Data fra grunddataregistre bliver løbende kopieret til datafordeleren via replikeringskanaler og derefter distribueret gennem forskellige former for tjenester. Datafordeleren distribuerer også hændelsesbeskeder. Såkaldte datanære hændelser dannes af datafordeleren på baggrund af dataopdateringer, hvorefter de bliver tilgængelige for dataanvendere, der på forhånd har oprettet et abonnement på hændelserne.

Objekter repræsenteres med rækker i databaser og det er opdatering af rækker, der kan trigge dannelsen af datanære hændelser. Hændelsesbeskederne har en fast struktur og indeholder blandt andet informationer, som der kan filtreres på. Når man opretter et abonnement, er det således muligt at opsætte et filter. Derudover kan man få tilsendt hændelsesbeskeder (PUSH) eller hente dem i en tjeneste (PULL).

Flowet for hændelser starter med en opdatering af data, der sendes fra et register til datafordeleren. Dataopdateringen er i en XML, hvor der for hver opdatering af en række i databasen er angivet, om den skal trigge dannelsen af en datanær hændelse. Hvis der dannes en hændelse, vil abonnenter fx kunne hente den i XML-format i en tjeneste.

Hændelser benyttes til at binde processer på tværs af grunddataregistrene sammen, men vil også kunne benyttes af andre til fx at igangsætte opdateringer af afledte eller relaterede data. Datafordeleren danner ikke delt information i forbindelse med datanære hændelser, men data med bitemporale kan udstilles i tjenester således, at det vil være muligt at fremsøge data for bestemte tidspunkter og tidsperioder.

Udover hændelsesabonnementer er der også abonnementer på filudtræk og man kan tale om at abonnere på atom feeds på datafordeleren.

Referencer

[Bitemporalitet] *Bitemporalitet – Proof of concept.*

<http://arkitekturguiden.digitaliser.dk/sites/default/files/ctools/bitemporalitet-v1.2.pdf>

[CPRVilkårOff] <https://cpr.dk/kunder/offentlige-myndigheder/udtraek/standardvilkaar-for-offentlige-myndigheders-adgang-til-cpr/>

[CPRVilkårPrivat] <https://cpr.dk/kunder/private-virksomheder/personnummerudtraek/standardvilkaar-for-dataleverancer-fra-cpr-til-private>

[DAF] *Anvendelse af tjenester og hændelser på Datafordeleren. Version 1.4.* http://grunddata-ejendom-adresse.dk/file/619801/daf_tekniskdok_tilslutningforanvendelseaftjenesteroghaendelser_v.1.4.docx eller <http://datafordeler.dk/media/2916423/tilslutning-for-anvendelse-af-tjenester-og-haendelser-paa-datafordeleren-v1-4.pdf>

[Dobbelthistorik] *Anvendelse af dobbelthistorik i GD2 – Implementerings regler og eksempler på dobbelthistorik.* http://grunddata-ejendom-adresse.dk/file/557241/dobbelthistorik_adresseprogrammet.pdf

[EDA] <http://arkitekturguiden.digitaliser.dk/hvad-er-eda>

[Grunddatabesked] *Beskedformat, Grunddatabesked, version 1.0.* <http://data.gov.dk/grunddatabesked/grunddatabesked.pdf>

[Modelregler] Digitaliseringsstyrelsen. *Modelregler for Grunddata.* <http://arkitekturguiden.digitaliser.dk/grunddata-modelregler>

[Quickguide] *Quick guide til hændelser. Version 2.* http://grunddata-ejendom-adresse.dk/file/624021/daf_brugervejledning_introtilhaendelser_v.2.docx