GD1/GD2 - Retningslinjer for udarbejdelse af DLS

Version: 0.36

Status: Udkast

Oprettet: 17-07-2015

Versionshistorik

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Dato | Ændring | Initialer |
| 0.1 | 30-06-2015 | Dokument oprettet og afsnit 3.2 navngivning af services samt afsnit 4.2 navngivning af hændelser, beskrevet. | S&D-KH ogS&D-RSP |
| 0.2 | 06-07-2015 | Afsnit 4.1 generelle kvalitetskrav (hændelsesbeskrivelser), afsnit 4.3 Integrationsmønstre (hændelser), afsnit 5.3 forretningsmæssig beskrivelse af hændelser samt afsnit 5.4 DLS skabelon til hændelser, beskrevet  | S&D-RSP |
| 0.3 | 08-07-2015 | Hændelsesbeskrivelse justeret i forhold til attributten objektAktion, samt mindre præciseringer.Afsnit 3.2: HentÆndringer og HentSimpel er tilføjet som mulig servicehandling.Afsnit 2 Udstillingsmodeller, beskrevetAfsnit 3.1, 3.3, 3.4 og tilhørende skabeloner i afsnit 5.2 beskrevetAfsnit 5.4: Mindre præciseringer, afsnittet er fortsat under udarbejdelse. | S&D-KH ogS&D-RSP |
| 0.35 | 17-07-2015 | Afsnit 2.2: De tre attributter til forretningsmæssig kontekst er ændret til datatype ”enumeration”, de var før ”CharacterString”Afsnit 5.2 og 5.3: I UUID referencer er ændret til ID (UUID eller entydig numerisk ID som URI) samt mindre ID/UUID relaterede rettelser. | S&D-RSP |
| 0.36 | 17-09-2015 | Afsnit 1. Indledning blev ændret, så scope blev præciseret |  |

Indholdsfortegnelse

1. Indledning 4

1.1 Dokumentets formål 4

1.2 Dokumentets status 4

1.3 Relaterede dokumenter 4

2. Udstillingsmodeller 5

2.1 Generelle kvalitetskrav 5

2.2 Standardattributter (modelregler) 5

2.3 Navngivning af begreber 6

2.4 Anvendelse af kopitabeller (begreber) 7

2.5 Dobbeltregistreringer 8

2.6 Modellering af kommuner 8

2.7 Opdateringsfrekvens på Datafordeleren 8

2.7.1 Tæt samarbejdende grunddataregistre. 8

2.7.2 Grunddataregistre med begrænset tidsmæssig afhængighed. 9

2.7.3 Grunddataregistre uden tidsmæssig afhængighed. 9

3. Servicebeskrivelser 10

3.1 Generelle kvalitetskrav 10

3.2 Navngivning af services 11

3.3 Services og dobbelthistorik 12

3.4 Sikkerhed og sikkerhedsroller 13

4. Hændelsesbeskrivelser 15

4.1 Generelle kvalitetskrav 15

4.2 Navngivning af hændelsesbeskeder 16

4.3 Integrationsmønstre 16

4.3.1 Generering af hændelser 16

4.3.2 Abonnering på hændelser 16

4.3.3 Forretningsdata i hændelsesbeskeder 17

5. Vejledning til skabeloner 18

5.1 Forretningsmæssig beskrivelse af udstillingsmodeller 18

5.2 Forretningsmæssig beskrivelse af services 18

5.2.1 Overordnet servicebeskrivelse 18

5.2.2 Beskrivelse af servicemetoder 19

5.2.3 Generisk skabelon til \*Opret servicemetoder 20

5.2.4 Generisk skabelon til \*Opdater servicemetoder 21

5.2.5 Generisk skabelon til \*Hent servicemetoder 22

5.2.6 Generisk skabelon til \*HentSimpel servicemetoder 23

5.2.7 Generisk skabelon til \*HentÆndringer servicemetoder 24

5.3 Forretningsmæssig beskrivelse af hændelser 26

5.3.1 Den oprindelige hændelsesskabelon 26

5.3.2 Det fællesoffentlige beskedformat 28

5.3.3 Den nye hændelsesskabelon 30

5.3.4 Sikkerhedsklassifikationer 31

5.3.5 Forberedende registeraktiviteter, inden udfyldelse af hændelsesskabelonen 31

5.4 DLS skabelon til hændelser 32

5.4.1 Register-genererede hændelser 32

5.4.2 Datafordeler-genererede hændelser 33

5.4.3 Hvilke opgaver giver de Datafordeler-generede hændelser til registrene 36

5.4.4 Hvornår skal der genereres Datafordeler-genererede hændelser 37

5.5 DLS skabelon 38

# Indledning

## Dokumentets formål

Dokumentet har til formål at give en GD1/GD2 fælles forretningsmæssig beskrivelse af de data og de tjenester (services **og hændelsesbeskeder**), som projekterne kan**skal** bruge som udgangspunkt for udfyldelsen af DLS (dataleverancespecifikation) **og tilsvarende specifikationer til brug for de direkte integrationer mellem grunddataregistrene**. Beskrivelsen er etableret som et kvalitetsunderstøttende dokument, som bør bruges til at tjekke projekternes egne antagelser og de præmisser **på et forretningsmæssigt niveau** – skrevne og uskrevne – som var kendte i tidsrummet op til 1. august 2015 **forbindelse med udfyldelse af forretningsmæssige beskrivelser af data og tjenester (services og hændelsesbeskeder) juli og august 2015.**

De første udgaver af dokumentet er udarbejdet primo juni som grundlag for registerprojekternes leverancer af forretningsmæssige beskrivelser, som skal afleveres til GD1/GD2 sekretariatet senest 7. august 2015.

På grund af sommerferieperioden har det ikke været muligt at få afstemt og godkendt alle anbefalinger og beslutninger, som dette dokument indeholder. Der er derfor i første omgang tale om retningslinjer til brug for leverancerne 7. august. Skulle der komme forbedringsforslag, ændringer til DLS-skabelon mv. må vi indarbejde disse ændringer i dokumentet og i de forskellige beskrivelser efterfølgende.

## Dokumentets status

Dokumentet i denne version 0.35 er oprettet med den påtænkte hovedstruktur – om end der nok vil komme mindre justeringer til denne. Der er i version 0.2 foretaget en ændring i strukturen, med oprettelse af afsnittet ”DLS skabelon til hændelser”

Udover selve strukturen er der beskrevet en standard for navngivning af services, servicemetoder og hændelsesbeskeder. Derudover er alt omkring hændelser beskrevet, dog med enkelte udeståender, der afventer afklaring fra GD7 og en opdatering af DLS skabelonen. Udeståenderne fremstår med gul markering i afsnit 5.4.2. Øvrige gule markeringer er noter om kommende indhold i afsnit, der ikke er beskrevet endnu.

Bemærk at udeståenderne omkring hændelser, kun er i forhold til DLS skabelonen. Den nye skabelon til den forretningsmæssige beskrivelse af hændelser, er udarbejdet, jf. afsnit 5.3, og forventes ikke ændret yderligere.

## Relaterede dokumenter

For yderligere information om implementering af dobbelthistorik, henvises der til dokumentet ”Forretningsregler for implementering af dobbelthistorik”, udarbejdet af GD2 programsekretariatet.

En oversigt over de anvendte begreber i GD1 og GD2 udstillingsmodeller findes i dokumentet ”GD1-GD2 Begrebsoverblik, version 0.1”.

Dette dokument er fremsendt separat.

# Udstillingsmodeller

## Generelle kvalitetskrav

Grunddataprogrammet ønsker at give brugerne (myndigheder, virksomheder m.fl.) en samlet og sammenhængende model over de udstillede data. Det betyder, at man som bruger oplever sammenhæng på tværs af forretningsdomænerne, og at grunddatamodellen fremstår som et sammenhængende hele med en ensartet begrebsanvendelse, ensartet modellering af generelle egenskaber mv.

Dette skal sikres selvom data i grunddatamodellen er sammensat af udstillingsmodeller fra forskellige forretningsdomænerne i grunddataprogrammet.

Derfor skal udstillingsmodellerne efterleve de ”fælles spilleregler” beskrevet i dokumentet ”Modelregler for grunddata” udgivet af Digitaliseringsstyrelsen.

Som hjælp hertil arbejder Grunddatasekretariatet på at etablere en ”modeltjekker”, som foreløbig findes i en beta-version: <http://data.gov.dk/modeltjekker>. Denne kan med fordel anvendes til sikring af de mest trivielle kontroller, som ellers kan være ret så tidskrævende.

## Standardattributter (modelregler)

Modellen skal på de enkelte modelobjekter indeholde en ”id”, dvs. en unik identifikation af objektet, en ”status”, hvor værdisættet vil være domænespecifikt/specifikt for det enkelte begreb, registreringer i relation til dobbelthistorik samt registreringer i relation til den forretningsmæssige kontekst.

**Registrering af dobbelthistorik:**

Dobbelthistorik handler om, at objekter registreres versioneret, således der over tid altid er styr på et objekts aktuelle og historiske informationsindhold. Dette i form af en registrering af en virkningsperiode og en registreringsperiode på den enkelte forekomst.

| Attribut | Type | Beskrivelse |
| --- | --- | --- |
| registreringFra | DataTime | Tidspunktet hvor registreringen er foretaget |
| registreringTil | DataTime | Tidspunktet hvor en ny registrering er foretaget på dataobjektet, og hvor denne version således ikke længere er den seneste. Værdien kan være tom. |
| registreringsaktør | CharacterString | Den aktør der har foretaget registreringen.Værdisæt er domænespecifikt. |
| virkningFra | DataTime | Tidspunktet hvor registreringen har virkning.Virkningsperioden inkluderer dette tidspunkt. |
| virkningTil | DataTime | Tidspunktet hvor registreringens virkning ophører. Virkningsperioden stopper umiddelbart før dette tidspunkt. Værdien kan være tom. |
| virkningssaktør | CharacterString | Den aktør der har afstedkommet objektets virkning. Værdisæt er domænespecifikt. |

**Registrering af forretningsmæssig kontekst:**

De enkelte grunddatabegreber anbefales i modelreglerne modelleret således, at dataobjektet kommer til at indeholde informationer, som kan forbedre kvaliteten af hændelsesbeskeder, der udsendes i forbindelse med opdatering af dataobjektet. Disse informationer omfatter den forretningsmæssige kontekst, hvori dataobjektet opdateredes, samt den bagvedliggende forretningsmæssige årsag til opdateringen.

GD1 og GD2 følger denne anbefaling, hvilket betyder, at følgende attributter skal registreres på de enkelte forekomster:

| Attribut | Type | Beskrivelse |
| --- | --- | --- |
| forretningsområde | Enumeration | Den del af den offentlige forretning der håndterer hændelsen og derved udvirker ændringen i data.Indeholder en FORM-nøgle. |
| Forretningsproces | Enumeration | Den manuelle eller IT-understøttede proces hvori forretningsområdet håndterer hændelsen. |
| forretningshændelse | Enumeration | Den begivenhed i virkeligheden som udløste ændringen i data. |

Det er de enkelte registres ansvar at definere udfaldsrummet for disse 3 enumerationer, da enumerationerne er registerspecifikke.

NB! Bemærk her at der er uoverensstemmelse mellem navngivning i modelregler og så de navne der anvendes i den forretningsmæssige DLS skabelon til hændelser jf. nedenstående:

| Navngivning i ”Modelregler” | Navngivning i DLS skabelon |
| --- | --- |
| forretningsområde | opgaveemne |
| Forretningsproces | tværgåendeProces |
| forretningshændelse | objektHandling |

Ovennævnte attributter i relation til dobbelthistorik og forretningsmæssig kontekst kan sammen med ”id” og ”status” med fordel modelleres i en generisk klasse, som inkluderes i de enkelte modelbegreber.

## Navngivning af begreber

Begreber og navngivningen af disse optræder i forskellige modeller:

* **Registres udstillingsmodel.**Begreber i udstillingsmodeller skal være entydigt navngivet på tværs af alle udstillingsmodeller. Reglerne herfor beskrives nedenfor.
* **Registres egen model.**Her er det helt op til det enkelte registerprojekt, hvorledes begreber navngives.
Årsagen hertil er bl.a., at en del registre er eksisterende registre med en eksisterende navngivning, hvor det vil være for omkostningstungt at ændre navngivning i datamodel, kode, brugergrænseflader mv.
Der er dog ét navngivningskrav til disse registermodeller: Når begrebet udstilles i en ajourføringsservice, som skal anvendes af et eksternt system, skal begrebet navngives som begrebet er modelleret i udstillingsmodellen.

Begreber i udstillingsmodeller skal kunne genbruges på tværs af grunddata for at sikre sammenhæng og undgå redundant vedligeholdelse af data. Der er derfor behov for i grunddataprogrammet entydigt at beslutte navngivningen af begrebet, samt hvem der er autoritativt register for begrebet.

 I den forbindelse skal begreberne i nogle tilfælde vurderes i et bredere perspektiv end blot GD1/GD2. Placeres ansvaret for et begreb i et GD1/GD2 register, følger hermed også ansvaret for registrering og udstilling af stamdata i relation til det pågældende begreb.

Ved navngivningen skal registerprojektet forholde sig til, om begrebet er entydigt nok – eksempelvis findes begrebet ”Sag” i flere registre men med hver deres betydning og informationsindhold. Her kan der med fordel anvendes et prefix, fx "MatrikulærSag".

Begrebet kan også være et udtryk for, at registret registrerer supplerende information til et begreb hørende hjemme i et andet register – fx registrerer Danske Stednavne på en række områder stednavne til andre begreber. Her skal der ud fra den forretningsmæssige kontekst vurderes, om det giver bedst mening at forsyne disse begreber med et prefix eller et suffix.

Eksempel: Et begreb som indeholde et stednavn til en Bygning kan med fordel navngives enten med et prefix – fx ”StednavnTilBygning” – eller et suffix – fx ”Bygningsstednavn”.

## Anvendelse af kopitabeller (begreber)

Nogle registre har udtrykt et behov for at kunne opbevare kopitabeller af andre registre i eget register – og dermed også for at kunne indeholde en kopi af begrebet i egen model.

Det anbefales stærkt at undgå disse kopier, hvis det er muligt.

Er der et behov herfor, skal forskellige regler overholdes:

* **Behovet skal dokumenteres og godkendes i delprogrammet.**Der kan være gode argumenter for anvendelse af kopitabeller. Disse nedskrives sammen med en beskrivelse af, hvorledes tabellen holdes opdateret, og danner grundlaget for delprogrammets godkendelse af kopitabellen.
Det er vigtigt for delprogrammerne at have denne synlighed – både i relation til de forskellige tværgående test og i forhold til et evt. senere behov for fejlsøgning. Det er her vigtigt at vide, hvis registret anvender en forsinket kopi i stedet for de autoritative grunddata.
* **Kopitabeller/begreber udstilles ikke.**På datafordeleren er der kun de autoritative grunddata begreber, hvorfor kopibegreber ikke må indgå i udstillingsmodellen.
I udstillingsmodellen skal evt. kopibegreber og relationer til disse erstattes af relationer til de autoritative grunddata begreber.
* **Tabellen må ikke opdateres i eget register.**Med andre ord – de autoritative grunddata må ikke overskrives.
* **Kopitabellen skal opdateres gennem de data, der udstilles på Datafordeleren.**En af grundideerne i gruinddataprogrammet er at undgå disse mange punkt-til-punkt integrationer ved at et register henter data fra andre registre gennem Datafordeleren.
En måde at holde registret opdateret på er at abonnere på de hændelser, som Datafordeleren udstiller i relation til den pågældende tabel.

## Dobbeltregistreringer

Der er i den tværgående kvalitetssikring identificeret et par dobbeltregistreringer – oplysninger som registreres i mere end ét register og som samtidig udstilles på Datafordeleren fra begge. Det drejer sig om ”etageangivelse” (Matriklen og BBR med hver deres værdisæt for etage) og ”ejerforholdskode” (Ejerfortegnelsen og BBR). Der kan være andre eksempler.

Der kan være gode argumenter for disse registreringer og måske er det forskellige ting, som kaldes det samme. Disse argumenter nedskrives sammen med en angivelse af, om oplysningen medtages i udstillingsmodellen.

Argumenterne danner grundlaget for delprogrammets godkendelse heraf, og skal samtidig kunne anvendes over for anvendere, såfremt en anvender måtte have spørgsmål til de forskellige registreringer af tilsyneladende samme information.

## Modellering af kommuner

Modellering af kommuner er under afklaring.

Kommuner er p.t. i registrene og deres udstillingsmodeller modelleret meget forskelligt – interne tabeller, referencer til eksterne (ikke eksisterende) registre, referencer til DAGI mv.

En del forventer DAGI som det autoritative register, men dette bliver næppe tilfældet.

I grunddataprogrammets modelregler forventes dette implementeret i en ”Organisationskomponent” – dels under kommunernes rammearkitektur, dels i en tilsvarende fællesoffentlig komponent. Sidstnævnte er p.t. ikke planlagt rettidigt ift. GD1/GD2 behov.

Den autoritative tabel er Indenrigsministeriets register med 4 cifrede kommunekoder og officielle kommunenavne.

Som nævnt pågår en afklaring i relation til hvor på Datafordeleren disse kommunekoder etableres. Indtil denne afklaring er på plads, anbefales det, at projekterne refererer til et eksternt begreb indeholdende Indenrigsministeriets register.

Den praktiske implementering kan så eventuelt vise sig at blive, at det enkelte register selv må implementere denne kommunetabel internt i registret, men det må afklares senere.

## Opdateringsfrekvens på Datafordeleren

Når data opdateres i de enkelte registre, er det vigtigt disse opdateringer overføres til Datafordeleren, så der kan udsendes hændelsesbeskeder til diverse abonnenter, og således anvendere ikke kommer til at træffe beslutninger på forældede data.

Som udgangspunkt er der derfor behov for en opdatering på Datafordeleren, der er ”så nær realtid som mulig”.

I praksis er dette ikke altid muligt for det enkelte register, hvorfor behovet for opdateringsfrekvens inden for GD1/GD2 defineres i nedenstående grupper:

### Tæt samarbejdende grunddataregistre.

**Omfatter**: Matriklen, Ejerfortegnelsen, BBR og DAR.

**Opdateringsfrekvens**: Ved opdatering - så nær realtid som mulig.

Her er der tale om brugere og systemer, som ofte arbejder på tværs af disse registre. Når man arbejder i ét register, læses egne data i registret, mens andre registres data læses via Datafordeleren. De enkelte registre opdaterer Datafordeleren individuelt, så er der meget forskel i opdateringsfrekvensen, vil dette samarbejde fejle og udstillede data risikerer at blive inkonsistente.

Eksempler:

1. BBR opretter en BPFG via en service i Matriklen, som i samme moment opretter ejerskabet i Ejerfortegnelsen. Der er tale om en sammenhængende opdatering i tre registre, og overføres disse opdateringer ikke nogenlunde samtidigt til Datafordeleren, vil der opstå inkonsistens i data der – fx ved at der optræder ejerskaber til ejendomme. Som endnu ikke er udstillet på Datafordeleren.
2. En Enhed i BBR skal altid have en tilknyttet Adresse i DAR. Overføres BBR til Datafordeleren, mens overførslen af DAR først sker senere, vil der på Datafordeleren blive udstillet enheder uden en adresse.

### Grunddataregistre med begrænset tidsmæssig afhængighed.

**Omfatter**: DAGI.

**Opdateringsfrekvens**: Flere gange dagligt – evt. ved opdatering.

Her er der en afhængighed til DAR – eksempelvis når der ændres i et Afstemningsområde.

Denne afhængighed er etableret gennem Datafordeleren ved at DAR opdateres, når DAGI opdateres på Datafordeleren og udsender en hændelsesbesked herom. Der opstår derfor ikke inkonsistente data på Datafordeleren – lige bortset fra det korte tidsrum frem til at DAR har behandlet hændelsen og opdateret Datafordeleren som følge heraf.

Frekvensen kunne være daglig opdatering, men det holder næppe forretningsmæssigt, at fx en fejlretning omkring et Afstemningsområde op til et valg først får gennemslagskraft dagen efter på Datafordeleren og dermed også i DAR.

### Grunddataregistre uden tidsmæssig afhængighed.

**Omfatter**: Danske Stednavne

**Opdateringsfrekvens**: Dagligt.

Her er der i forhold til opdatering på Datafordeleren ikke afhængigheder til andre registre – der er ikke andre registre som skal opdateres i samme sammenhæng.

Derfor kan opdateringsfrekvensen (fx daglig opdatering) fastlægges ud fra det forretningsmæssigt behov i Danske Stednavne uden behov for afstemning med andre grunddataregistre.

# Servicebeskrivelser

## Generelle kvalitetskrav

Kvalitetsmålet med servicebeskrivelser er dels

* at danne grundlag for at registrenes leverandører kan udvikle de tekniske specifikationer, der kræves i forbindelse med DLS’en
* og dels at danne grundlag tværgående kvalitetssikring mellem registerprojekterne og eventuelle øvrige anvendere, som har behov for servicebeskrivelser på et forretningsmæssigt niveau

De skabeloner, der fremgår af afsnit 5.2, er baseret på tidligere gode erfaringer overfor leverandører. Skabelonerne er stort set magen til de skabeloner, registrene har anvendt i deres løsningsarkitekturer. Største forskel er at alle attributter nu skal defineres.

 Som det fremgår af afsnit 3.2, er der defineret 7 typer af servicemetoder: Opret, Opdater, Slet, Hent, HentAktuel, HentSimpel, HentÆndringer. De fire varianter af Hent, kræver en nærmere forklaring.

* ”Hent”, er den komplette metode, der kan anvendes til alle formål. Man kan angive dobbelthistorik parametre, der gør det muligt at hente præcis den eller de forekomster man ønsker, uanset om forekomster er historiske, nutidige, eller fremtidige. Metoden henter også alle relevante relaterede objekter, eksempelvis vil \*BfeHent også returnere tilhørende Jordstykker.
* ”HentAktuel”, er den variant af ”Hent”, hvor der ikke kan angives dobbelthistorik parametre, metoden vil således altid returnere den eller de forekomster med aktiv registreringstid og virkningstid på tidspunktet for anvendelsen af metoden.
* ”HentSimpel”, er en reduceret udgave af ”Hent”, der kun returnerer attributter tilhørende hovedobjektet. Eksempelvis vil \*BfeHentSimpel ikke returnere Jordstykke og dette objekts attributter, men kun de attributter der hører direkte til BFE objektet.
* ”HentÆndringer”, er en særlig hent-metode, der er relateret til hændelser. Grundet begrænsningen i hvilke data, der kan vedlægges hændelser, samt at hændelser slettes efter 120 timer, jf. afsnit 4.3. Der er derfor identificeret et behov for at anvendere enten kan:
	+ hente oplysning om hvilke attributter, der er ændret, i forbindelse med modtagelsen af en hændelse
	+ eller, i tilfælde af at anvenderens egne data er kommet ud af synk med Datafordeleren, hente ændringerne mellem anvenderens egen kopi og den nyeste (eller en hvilken som helst anden) forekomst på Datafordeleren
	+ eller at anvenderen kan hente ændringer med en fremtidig virkningstid

Alle registre skal som minimum udstille en ”Hent” metode pr. hovedobjekt, således at anvendere kan hente hele registrets udstillingsmodel via en eller flere ”Hent” metoder. Der er således ikke krav til at udstille ”Hent” metoder til alle underobjekter, så længe disse objekter kan hentes via et hovedobjekt.

Det anbefales at der udstilles en ”Hent” metode for hvert objekt, der dannes hændelser for. Ligeledes anbefales det at der udstilles en ”HentÆndringer” for hvert af disse objekter.

”HentAktuel”, skal kun udstilles, hvis registret skønner det nødvendigt af hensyn til datastrukturer eller performance.

”HentSimpel” er tænkt som en performancemæssig hurtig service, der kan anvendes i brugergrænseflader, eksempelvis til at hente beliggenhedsadresse, når brugeren har indtastet BFE nummer og forlader feltet. De hyppigst brugte objekter, bør have en ”HentSimpel” metode.

## Navngivning af services

Dette afsnit beskriver reglerne for navngivning af services og tilhørende servicemetoder (også omtalt som ”serviceoperationer”).

I navngivningen undlades brugen af underscore (”\_”) eller andre special tegn. Årsagen hertil er, at dette i nogle systemer/værktøjer giver problemer med at benytte servicen.

Derfor er navngivningen fastlagt som en tekststreng, hvor store begyndelsesbogstaver benyttes til at adskille <objekt> og <handling>

Services navngives efter følgende syntaks:

**<GD delprogram><Register><Kategori><Type><Objekt>**

Servicemetoder navngives efter følgende syntaks:

**<GD delprogram><Register><Kategori><Type><Objekt><Handling>**

| Syntaks | Udfaldsrum | Forklaring |
| --- | --- | --- |
| GD delprogram | EJDADR | Obligatorisk forkortelse for hvilket delprogram, servicen hører til. EJD = GD1, ADR = GD2 |
| Register | admebpvs | Obligatorisk forkortelse for hvilket register, der er ansvarlig for servicen.a = DAR (Adresseregister), d = DAGI, m = Matriklen, e = Ejerfortegnelsen, b = BBR, p = CPR (Personer) , v = CVR (Virksomheder), s = Danske Stednavne |
| Kategori | AU | Obligatorisk forkortelse for hvilken slags service/hændelse, der er tale om.A = Ajourføring, U = Udstilling, H = Hændelse |
| Type | soaprestwmswfsfil | Valgfri angivelse af hvilken teknologisk type service, der er tale om. SOAP antages, hvis der ikke angives nogen type. |
| Objekt |  | Obligatorisk angivelse af navnet på det hovedobjekt fra begrebsmodellen, som servicen anvendes til.Sammenstillede services navngives med sigende navne, sammensat af de vigtigste hovedobjekter i servicen. |
| Handling | OpretOpdaterSletHentHentAktuelHentSimpel HentÆndringer | Angiver hvad metoden anvendes til. ”Hent” anvendes til alle former for læsning af data – uanset om der angives en eksakt nøgle eller søge kriterier hhv. om resultatet er en enkelt forekomst eller en liste.Ved ”Hent” kan angives et virkningstidspunkt og/eller et registreringstidspunkt, således det er muligt at hente historiske data samt evt. data med fremtidig virkning.Er der behov for en servicemetode, der altid kun returnerer aktuel registrering og virkning navngives denne som ”HentAktuel”. Der kan ikke angives virkningstid og registreringstid som input parameter ved ”HentAktuel” metoder.”HentSimpel” er tænkt som en reduceret udgave af ”Hent”, der kun henter attributterne til hovedobjektet. Metoden skal primært bruges i forbindelse med brugergrænseflader. ”HentÆndringer” er relateret til hændelser og skal anvendes til at hente ændringer mellem to angivne forekomster af et objekt. |

Eksempler på standardens anvendelse på servicenavne:

* EJDmUBestemtFastEjendom
* EJDeAEjerskifte
* ADRaUrestAdresse
* ADRdUwfsAdministrativInddeling
* EJDeUNavngivenEjerTilEjendom (sammenstillet service CPR/CVR + EJF + MU)

Eksempler på standardens anvendelse på servicemetodenavne:

* EJDmABygningPaaFremmedGrundOpret
* EJDeUEjerskifteHent
* ADRaANavngivenVejOpdater
* ADRdUwfsAdministrativInddelingSingleHent
* ADRsUStednavnTilBygningHentAktuel
* EJDmUBestemtFastEjendomHentÆndringer

## Services og dobbelthistorik

Alle services skal understøtte dobbelthistorik ved at give mulighed for at oprette og opdatere forekomster, uanset om deres virkningstid, i forhold til dags dato, er historik, nutidig eller fremtidig. Ligeledes skal det være muligt at hente forekomster ved at angive en eller flere af dobbelthistorik parametrene.

Det er vigtigt at servicemetoderne, på tværs af registrene, har en ensartet måde at behandle dobbelthistorikken på, således at anvendere kun behøver at lære en måde at håndtere dobbelthistorik på.

For at sikre denne ensartethed, er der i afsnit 5.2 angivet generiske skabeloner til \*Opret, \*Opdater, \*Hent og \*HentÆndringer servicemetoder.

Generelt for disse generiske skabeloner, gælder det at dobbelthistorik parametrene altid skal returneres, således at anvenderen aldrig bliver i tvivl om hvilken forekomst de arbejder med.

Af hensyn til betjeningen og forståelsen af servicemetoderne, er der i de generiske skabeloner, defineret nogle default værdier, der sættes, hvis anvenderen ikke angiver alle de styrende dobbelthistorik parametre (UUID, registreringstid, virkningstid, status).

Dobbelthistorikken skal naturligvis også implementeres ensartet på Datafordeleren og dermed også i registrene, hvis servicemetoderne og specielt sammenstillet servicemetoder skal fungerer ensartet.

Der er udarbejdet implementeringsregler til dobbelthistorik i forbindelse med Matriklens og Ejerfortegnelsens udbudsmaterialer, samt en implementeringsvejledning udarbejdet til GD2. Alle tre regelsæt er enslydende, hvorfor det forventes at dobbelthistorik implementeres ensartet. Der henvises til sidstnævnte vejledning, da denne er tilgængelig for alle parter, jf. afsnit 1.3.

## Sikkerhed og sikkerhedsroller

Sikkerheden i services på Datafordeleren styres via to mekanismer:

1. Adgang til services generelt. En service kan enten være frit tilgængelig, eller kræve tildelt adgang. Tildelt adgang betyder at serviceejeren skal godkende anvenderen, før denne kan bruge servicen.
2. Systemrettigheder i forbindelse med servicemetoder. Alle metoder, der ikke er frit tilgængelige, hvilket betyder alle metoder, der kan indeholde ikke-offentlige data, eller data på sikkerhedsniveau 2, 3 eller 4, jf. afsnit 5.3.4, kræver specifikke sikkerhedsroller for at fungerer. Disse sikkerhedsroller skal specificeres af serviceejerne og være tilpas detaljeret til at understøtte alle anvendere, såvel interne som eksterne. Frit tilgængelige services indeholder pr. definition kun data med sikkerhedsniveau 1, som er offentlige data.

Til alle de services, som registret udstiller til brug for andre systemer (brugervendte systemer eller andre serviceanvendere) defineres der en systemrolle. Denne skal indmeldes i Grunddataprogrammets systemrollekatalog sammen med et certifikat knyttet til anvenderen.

En systemrolle kan omfatte adgang til flere services/servicemetoder, men kan også være målrettet én enkelt servicemetode.

Der defineres ikke nogle brugerroller til udstillingsservices.

Systemrollerne navngives med et forretningsmæssigt signende navn, efterfulgt af en kort forretningsmæssig beskrivelse. Når sikkerhedsrollerne defineres, er det vigtigt at forholde sig til om de enkelte services/servicemetoder, kan hente data fra flere sikkerhedsniveauer. Sikkerhedsrollerne til services/servicemetoder skal defineres efter den højst mulige sikkerhedsklassifikation, dvs. en service/servicemetode, der KUN henter offentlige data, kan have en sikkerhedsrolle svarende til sikkerhedsniveau 1, men hvis en servicemetode KAN hente data med sikkerhedsklassifikation 2, skal der kræves en sikkerhedsniveau 2 rolle.

*Eksempel på systemroller, med adskillelse på sikkerhedsniveau:*

* ***ForespørgSagsdata*** *Giver rettighed til at forespørge på alle BBR sagsdata på Datafordeleren, uanset sikkerhedsklassifikation*
* ***ForespørgOffentligtligeSagsdata*** *Giver rettighed til at forespørge på BBR sagsdata på Datafordeleren, der kun omfatter sikkerhedsniveau 1 data.*

Alle registerenes systemroller samles i nedenstående rolle-rettigheds matrice, der for forståelsens skyld er udfyldt med ovenstående eksempel:

|  |  |
| --- | --- |
| **Roller****Rettigheder** | **Systemroller** |
| ForespørgeSagsdata | ForespørgeOffentligeSagsdata | … | … | … |
| Læse BBR Sag, sikkerhedsniveau 1 | **+** | **+** |  |  |  |
| Læse BBR Sag, sikkerhedsniveau 2-4 | **+** |  |  |  |  |
| … |  |  |  |  |  |

# Hændelsesbeskrivelser

## Generelle kvalitetskrav

En række processer i GD1 og GD2 foregår i sammenhæng understøttet af forskellige systemer og håndteret af forskellige aktører i relation til den enkeltes ressort ansvarsområde. Derfor er der et stort behov for at kommunikere forretningshændelser via beskeder mellem de forskellige it-løsninger og aktører. Dette er en afgørende forudsætning for at kunne sikre en forretningsmæssig sammenhæng i processer og de dertil hørende data registreret i forskellige registre, og dermed en afgørende forudsætning for en effektiv digitalisering.

Hændelsesbeskeder anvendes på forskellig måde – nogle som grundlag for at hente yderligere data, som beskeden refererer til, andre anvendes i sig selv.

**NB!** Ved definitionen af den enkelte hændelsesbesked er det derfor vigtigt at registerprojekterne gør sig klart, hvilket af disse to mønstre, der forventes anvendt af anvenderen. Skal alle data medsendes som payload eller leveres der kun nøgler som grundlag for en efterfølgende hentning af data.

Nogle beskeder vil formentlig kunne behandles digitalt, mens andre vil kræve et advis sendt til en medarbejder. Når beskeden (i form af et advis) rammer en medarbejder, skal dette ske på rette tid og sted på en måde, så medarbejderen ikke drukner i for mange, for detaljerede og ikke relevante beskeder.

**NB!** Det er her vigtigt at registerprojekterne sikrer, at det bliver så enkelt og brugervenligt som muligt ved at beskeder etableres med et forretningsmæssigt sigende indhold, således at anvenderne nemt kan forstå og agere på baggrund af disse.

Kvalitetskravene til hændelsesbeskrivelserne skal sikre:

* at registrene, med udgangspunkt i den forretningsmæssige hændelsesbeskrivelse, er i stand til at definere registrets hændelser overfor Datafordeleren i DLS’en, som derefter vil være i stand til at generere hændelserne
* at registrenes, på baggrund af hinandens forretningsmæssige hændelsesbeskrivelser, kan få de tværgående forretningsmæssige processer til at fungere
* at alle anvendere, såvel grunddataregistre som øvrige anvendere, med udgangspunkt i de forretningsmæssige hændelsesbeskrivelser, kan forestille sig de abonnementer på hændelser, de har behov for – uden at skulle modtage mange irrelevante hændelsesbeskeder

Kvalitetskravene søges opfyldt ved anvendelse af en entydig navnestandard, som beskrevet i afsnit 4.2; nogle anbefalinger omkring sammenhængen mellem hændelser og services, som beskrevet i afsnit 4.3; samt stringent anvendelse af den fælles hændelsesskabelon, hvor klassifikationerne, der skal anvendes som abonnementskriterier, defineres med anvenderne for øje. Skabelonen og klassifikationerne er nærmere beskrevet i afsnit 5.3.

## Navngivning af hændelsesbeskeder

Dette afsnit beskriver reglerne for navngivning af hændelser.

Hændelser navngives efter følgende syntaks:

**<GD delprogram><register><kategori><objekt>**

| Syntaks | Udfaldsrum | Forklaring |
| --- | --- | --- |
| GD delprogram | EJDADR | Obligatorisk forkortelse for hvilket delprogram, servicen hører til. EJD = GD1, ADR = GD2 |
| Register | admebpvs | Obligatorisk forkortelse for hvilket register, der er ansvarlig for hændelsesbeskeden.a = DAR (Adresseregister), d = DAGI, m = Matriklen, e = Ejerfortegnelsen, b = BBR, p = CPR (Personer) , v = CVR (Virksomheder), s = Danske Stednavne |
| Kategori | H | Obligatorisk forkortelse for hvilken slags service/hændelse, der er tale om.A = Ajourføring, U = Udstilling, H = Hændelse |
| Objekt |  | Obligatorisk angivelse af navnet på det hovedobjekt fra begrebsmodellen, som hændelsen vedrører. |

Eksempler på hændelsesnavne:

* EJDmHBestemtFastEjendom
* EJDeHEjerskifte
* ADRaHAdresse
* ADRaHNavngivenVej

##  Integrationsmønstre

### Generering af hændelser

GD1 og GD2 har aftalt med GD7, at Datafordeleren genererer registrenes forretningsmæssige hændelser, på baggrund af hændelsesinformationer, som registrene skal medsende, når de opdaterer data på Datafordeleren.

Teknisk – og definitionsmæssigt – foregår dette ved at anvende ”CreateEvent” i DLS’en, hvilket er nærmere beskrevet i afsnit 5.4.1.

### Abonnering på hændelser

Datafordeleren understøtter både ”push” og ”pull” af hændelser. GD1 og GD2 programledelserne anbefaler at ”push” modellen anvendes, da det er denne metode, der har været tænkt på under udarbejdelse af Målarkitekturen.

**Push:**

Push betyder at Datafordeleren kalder en REST-service hos abonnenten, hvor hændelsesbeskeder afleveres. For at kunne anvende push, skal abonnenten derfor stille en REST-service til rådighed for Datafordeleren, der både understøtter JSON og XML.

Såfremt abonnentens service ikke er tilgængelig, sender Datafordeleren en advisering til abonnenten på email, og forsøger herefter, en gang i timen i 120 timer at fremsende hændelserne. Hvis abonnentens service ikke fungerer i 120 timer deaktiveres abonnementet og hændelsesbeskederne slettes. Abonnenten modtager en advisering om dette på email.

Når abonnentens service fungerer igen, skal abonnementet genaktiveres. De manglende hændelsesbeskeder kan ikke genskabes.

**Pull:**

Pull betyder at Datafordeleren lægger alle abonnentens hændelsesbeskeder i en postkasse, som abonnenten kan tømme, via en service på Datafordeleren. Hændelsesbeskederne slettes automatisk efter 120 timer.

**Backup løsning ved manglende hændelser, efter 120 timers inaktivitet eller fejl:**

Abonnentens eneste mulighed for at hente ændringer i den manglende periode, er via en service. Dette kræver dog at de enkelte registre stiller en sådan service til rådighed, hvor abonnenten kan angive start og slut registreringstider og få returneret alle ændringer i den pågældende periode.

### Forretningsdata i hændelsesbeskeder

Det er muligt at vedlægge forretningsdata i hændelser, via objektet ”ObjektData”. Alt data, der indgår i den dataopdatering, der generer hændelsen kan vedlægges. Derudover kan der vedlægges data, der allerede ligger på Datafordeleren, dog begrænset til gældende forekomst på tidspunktet for generering af hændelsen. Ligeledes er der mulighed for at vedlægge konstanter, det vil sige en tekst, registeret angiver.

Dette betyder blandt andet at der ikke kan vedlægges ”før” værdier i hændelserne, medmindre registrene angiver før-værdierne, i forbindelse med dataopdateringen.

Såfremt registrene ikke vedlægger før-efter data i hændelserne, anbefales det at registrene stiller services til rådighed, som kan anvendes til at hente ændringer på et givent objekt, samt services til at hente de objekter, der genereres hændelser på – sidstnævnte bør stilles til rådighed under alle omstændigheder.

Eksempelvis:

Hændelse: EDJmHBestemtFastEjendom

Service: EJDmUBestemtFastEjendomHent (med angivelse af et sæt dobbelthistorik parametre)

Service: EJDmUBestemtFastEjendomHentÆndringer (med angivelse af to sæt dobbelthistorik parametre)

Afsnit 5.2.5 og afsnit 5.2.6 indeholder generiske definitioner af \*Hent og \*HentÆndringer servicemetoder.

# Vejledning til skabeloner

## Forretningsmæssig beskrivelse af udstillingsmodeller

## Forretningsmæssig beskrivelse af services

Skabelonerne for forretningsmæssig beskrivelse af services, følger skabelonerne, der blev anvendt i løsningsarkitekturerne. Dog er der tilføjet en konsistent anvendelse af dobbelthistorik, jf. afsnit 3.3 samt en angivelse af alle attributter.

Udover de generelle skabeloner til services og servicemetoder i afsnit 5.2.1 og afsnit 5.2.2, er der udarbejdet generiske skabeloner til metoderne: \*Opret, \*Opdater, \*Hent, \*HentSimpel og \*HentÆndringer.

### Overordnet servicebeskrivelse

De enkelte services skal beskrives på en ensartet måde på tværs af systemerne, så der sikres en fælles forståelse for, hvilke metoder de forskellige services indeholder, og hvilken funktionalitet der tilbydes i de forskellige metoder.

Da alle systemerne skal udvikles parallelt, bliver servicebeskrivelserne afgørende for at sikre, at systemerne kan integrerer med hinanden på det ønskede niveau, med den rette dataudveksling.

Services skal på et overordnet niveau beskrives i følgende skabelon:

|  |
| --- |
| **Navn:** *navn på egen service, eksempelvis: EJDbABygningAjourfoer* |
| **Formål:** *Formålet med servicen beskrives kort, eksempelvis: Formålet med servicen er at oprette og vedligeholde Bygning og Teknisk anlæg.* |
| **Understøttede processer:***Der beskrives kort hvilke processer servicen skal understøtte, eksempelvis: Understøtter processen med udarbejdelse af teknisk dokumentation for Bygninger i BBR* |
| **Liste over metoder:***Komplet liste af de metoder servicen indeholder, eksempelvis:** *BygningOpret*
* *BygningOpdater*
* *BygningNedlæg*
* *BygningGenaktiver*
 |
| **Service informationsmodel:***En komplet liste af hvilke begreber fra begrebsmodellen, der anvendes i servicen, som input- og outputparametre:** *BBR Sag*
* *Bygning*
* *Teknisk anlæg*
 |
| **Service Level Agreement (SLA):***Udfyldes med krav og forventninger til SLA parametre, eks:** *Svartider:*
	+ *opret/opdater metoder < 1 sekund*
	+ *hent metoder < 0,5 sekund*
* *Oppetid:*
	+ *Mandag – fredag, 7.00 – 19.00: 99 %*
	+ *Øvrig: 96 %*
* *Support tilgængelighed:*
	+ *Mandag – fredag, 9.00 – 15.00*
 |

### Beskrivelse af servicemetoder

Alle metoder i en service, skal enkeltvis beskrives i følgende skabelon:

|  |
| --- |
| **Navn:** *Navn på servicemetode, eksempelvis: BygningOpret* |
| **Formål:** *Forretningsmæssig beskrivelse af metodens formål, eksempelvis: Opretter en Bygning eller et teknisk anlæg i BBR.* |
| **Input parametre:***En komplet liste af mulige input parametre, eksempelvis:**Obligatorisk:** *JordstykkeID*
* *Bygningsnummer*
* *Anvendelse*
* *Sikkerhedsklassifikation*
* *Bevaringsværdig*
* *Fredning*
* *Bygningsarealer*

*Valgfri:** *VirkningstidFra*
* *VirkningstidTil*
* *Status*
* *Opførelsesår*
* *Varmeinstallation*
* *Opvarmningsmiddel*
 |
| **Output parametre:***En komplet liste af mulige output parametre, eksempelvis:**Liste indeholdende:**Obligatorisk:** *ID*
* *RegistreringstidFra*
* *RegistreringstidTil*
* *RegistreringsAktør*
* *VirkningstidFra*
* *VirkningstidTil*
* *VirkningsAktør*
* *Status*

*Valgfri:** *Bygningnummer*
 |
| **Returkoder:***Komplet liste over mulige returkoder, eksempelvis:** *OK*
* *VirkningstidTil må være mindre end VirkningstidFra*
* *Anvendelseskode skal angives*
* *Ukendt Jordstykke*
 |
| **Præbetingelser:***Beskrivelse af eventuelle forretningsmæssige forudsætninger for at metoden kan fungerer korrekt, samt en beskrivelse af default værdier til dobbelthistorik parametrene, hvis disse ikke angives eksplicit.**Eksempelvis: Jordstykke, bygningen er opført på, skal eksisterer på Datafordeleren.**Dobbelthistorik: Hvis der ikke angives ”Status”, anvendes værdien <Projekteret>, hvis der ikke angives ”VirkningstidFra”, anvendes tidspunktet for servicekaldet, hvis der ikke angives ”VirkningstidTil”, anvendes NULL (uendelig).* |
| **Postbetingelser:***Beskrivelse af eventuelle forretningsmæssige konsekvenser af at metoden blev udført uden fejl, eksempelvis ”Ny Bygning oprettet”* |
| **Sikkerhed:***En beskrivelse af hvilket sikkerhedsrolle(r) der kræves for at anvende metoden, eksempelvis:** *BBR\_UPDATE\_BYGNING*

*Sikkerhedsrollen styrer, at det kun er brugere med den korrekte rolle, der får mulighed for at oprette bygninger.* |

### Generisk skabelon til \*Opret servicemetoder

Nedenstående skabelon er forud udfyldt med de input- og outputparametre, returkoder og præbetingelser, der altid skal indgå en i \*Opret servicemetode. Når der er anvendt \* i skabelonen, indikerer det at registrene skal tilføje relevante oplysninger.

|  |
| --- |
| **Navn:** \*Opret |
| **Formål:** Opretter \* |
| **Input parametre:**Obligatorisk:* Registreringsaktør
* VirkningstidFra
* Virkningsaktør
* Status
* \*

Valgfri:* VirkningstidTil
* \*
 |
| **Output parametre:**Obligatorisk:* \*ID *(UUID eller entydig numerisk ID som URI)*
* RegistreringstidFra
* RegistreringstidTil
* RegistreringsAktør
* VirkningstidFra
* VirkningstidTil
* VirkningsAktør
* Status
* \*

Valgfri:* \*
 |
| **Returkoder:*** OK
* Obligatoriske parametre ikke angivet
* Ukendt Status
* \*
 |
| **Præbetingelser:**Dobbelthistorik: * Hvis der ikke angives ”VirkningstidTil, anvendes NULL (uendelig) som værdi.

Forretningsbetingelser:* \*
 |
| **Postbetingelser:**\* |
| **Sikkerhed:**\* |

### Generisk skabelon til \*Opdater servicemetoder

Nedenstående skabelon er forud udfyldt med de input- og outputparametre, returkoder og præbetingelser, der altid skal indgå en i \*Opdater servicemetode. Når der er anvendt \* i skabelonen, indikerer det at registrene skal tilføje relevante oplysninger.

|  |
| --- |
| **Navn:** \*Opdater |
| **Formål:** Vedligeholder \* |
| **Input parametre:**Obligatorisk:* \*ID (anvendes til fremsøgning af den forekomst, der skal opdateres)
* ObjektStatus (anvendes til fremsøgning af den forekomst, der skal opdateres)
* Registreringsaktør (værdi på ny forekomst)
* VirkningstidFra (værdi på ny forekomst)
* Virkningsaktør (værdi på ny forekomst)
* \*

Valgfri:* Registreringstid (anvendes til fremsøgning af den forekomst, der skal opdateres)
* Virkningstid (anvendes til fremsøgning af den forekomst, der skal opdateres)
* VirkningstidTil (værdi på ny forekomst)
* Status (værdi på ny forekomst)
* \*
 |
| **Output parametre:**Obligatorisk:* \*ID
* RegistreringstidFra
* RegistreringstidTil
* RegistreringsAktør
* VirkningstidFra
* VirkningstidTil
* VirkningsAktør
* Status
* \*

Valgfri:* \*
 |
| **Returkoder:*** OK
* Obligatoriske parametre ikke angivet
* De angivne dobbelthistorik parametre giver ikke en entydig forekomst
* Ukendt ID
* Ukendt Status
* \*
 |
| **Præbetingelser:**Dobbelthistorik: * Hvis der ikke angives ”Registreringstid”, anvendes tidspunktet for servicekaldet til at finde den forekomst, der skal opdateres.
* Hvis der ikke angives ”Virkningstid”, anvendes tidspunktet for servicekaldet til at finde den forekomst, der skal opdateres.
* Hvis der ikke angives ”VirkningstidTil, anvendes NULL (uendelig) som værdi.
* Hvis der ikke angives ”Status”, anvendes ”ObjektStatus” som værdi.

Forretningsbetingelser:* \*
 |
| **Postbetingelser:**\* |
| **Sikkerhed:**\* |

### Generisk skabelon til \*Hent servicemetoder

Nedenstående skabelon er forud udfyldt med de input- og outputparametre, returkoder og præbetingelser, der altid skal indgå en i \*Hent servicemetode. Når der er anvendt \* i skabelonen, indikerer det at registrene skal tilføje relevante oplysninger.

|  |
| --- |
| **Navn:** \*Hent |
| **Formål:** Henter \* |
| **Input parametre:**Obligatorisk:* \*ID
* \*

Valgfri:* Registreringstid
* Registreringsaktør
* Virkningstid
* Virkningsaktør
* Status
* \*
 |
| **Output parametre:**Liste indeholdende nul-mange elementer.Obligatorisk:* \*ID
* RegistreringstidFra
* RegistreringstidTil
* RegistreringsAktør
* VirkningstidFra
* VirkningstidTil
* VirkningsAktør
* Status
* \* (attributter til hovedobjektet og relevante relaterede objekter)
 |
| **Returkoder:*** OK
* Obligatoriske parametre ikke angivet
* Ingen forekomster med de angivne parametre fundet
* \*
 |
| **Præbetingelser:**\* |
| **Postbetingelser:**Dobbelthistorik: * De valgfri parametre anvendes til at indsnævre antallet af forekomster, der returneres. Hvis ingen valgfri parametre angives, returneres alle forekomster; historiske, nutidige og fremtidige.

Forretningsbetingelser:\* |
| **Sikkerhed:**\* |

### Generisk skabelon til \*HentSimpel servicemetoder

Nedenstående skabelon er forud udfyldt med de input- og outputparametre, returkoder og præbetingelser, der altid skal indgå en i \*HentSimpel servicemetode. Når der er anvendt \* i skabelonen, indikerer det at registrene skal tilføje relevante oplysninger.

|  |
| --- |
| **Navn:** \*Hent |
| **Formål:** Henter attributter til \* hovedobjektet. |
| **Input parametre:**Obligatorisk:* \*ID *eller* forretningsvendt nøgle
* Registreringstid
* Virkningstid
* Status

Valgfri: |
| **Output parametre:**En forekomst, indeholdende:Obligatorisk:* \*ID
* RegistreringstidFra
* RegistreringstidTil
* RegistreringsAktør
* VirkningstidFra
* VirkningstidTil
* VirkningsAktør
* Status
* \*Forretningsvendt nøgle
* \* (kun attributter til hovedobjektet)
 |
| **Returkoder:*** OK
* Obligatoriske parametre ikke angivet
* Ingen forekomster med de angivne parametre fundet
 |
| **Præbetingelser:**\* |
| **Postbetingelser:**Dobbelthistorik: * Forekomsten findes ved:
	+ ID eller forretningsmæssig nøgle = angivet nøgle
	+ AND registreringstidFra <= Registreringstid
	+ AND registreringstidTil > Registreringstid
	+ AND virkningstidFra <= Virkningstid
	+ AND virkningstidTil > Virkningstid
	+ AND status = Status

Forretningsbetingelser:\* |
| **Sikkerhed:**\* |

### Generisk skabelon til \*HentÆndringer servicemetoder

Nedenstående skabelon er forud udfyldt med de input- og outputparametre, returkoder og præbetingelser, der altid skal indgå en i \*HentÆndringer servicemetode. Når der er anvendt \* i skabelonen, indikerer det at registrene skal tilføje relevante oplysninger.

|  |
| --- |
| **Navn:** \*HentÆndringer |
| **Formål:** Henter ændringer mellem to forekomster af \* |
| **Input parametre:**Obligatorisk:* \*ID

Valgfri:* Registreringstid\_1
* Virkningstid\_1
* Status\_1
* Registreringstid\_2
* Virkningstid\_2
* Status\_2
 |
| **Output parametre:**Liste indeholdende nul-mange elementer.Obligatorisk:* \*ID
* Forekomst\_1
	+ RegistreringstidFra
	+ RegistreringstidTil
	+ RegistreringsAktør
	+ VirkningstidFra
	+ VirkningstidTil
	+ VirkningsAktør
	+ Status
	+ \*
* Forekomst\_2 (repeterende, hvis der er tilføjet historik, eller der findes 2 samtidige versioner)
	+ RegistreringstidFra
	+ RegistreringstidTil
	+ RegistreringsAktør
	+ VirkningstidFra
	+ VirkningstidTil
	+ VirkningsAktør
	+ Status
	+ \*
 |
| **Returkoder:*** OK
* Obligatoriske parametre ikke angivet
* De angivne dobbelthistorik parametre giver ikke en entydig forekomst
* Mindre end 2 forekomster med de angivne parametre fundet
* \*
 |
| **Præbetingelser:**Dobbelthistorik: * Hvis der ikke angives nogle valgfri parametre, anvendes følgende:
	+ Registreringstid\_1 = Virkningstid\_1 = tidspunktet for servicekaldet
 |
| **Postbetingelser:**Dobbelthistorik: * Forekomst\_1 findes med følgende parametre:
	+ ID = angivet ID
	+ AND registreringstidFra <= Registreringstid\_1
	+ AND registreringstidTil > Registreringstid\_1
	+ AND virkningstidFra <= Virkningstid\_1
	+ AND virkningstidTil > Virkningstid\_1
	+ AND status = Status\_1 (hvis Status\_1 er angivet)
	+ Hvis Status\_1 ikke er angivet og der eksisterer flere samtidige versioner, der opfylder ovenstående, returneres returkoden ”De angivne dobbelthistorik parametre giver ikke en entydig forekomst”
* Hvis ikke forekomst\_2 er angivet, findes den med følgende parametre:
	+ registreringstidTil (forekomst\_2) = registreringstidFra (forekomst\_1)
* Hvis forekomst\_2 er angivet, findes den med følgende parametre:
	+ ID = angivet ID
	+ AND registreringstidFra <= Registreringstid\_2
	+ AND registreringstidTil > Registreringstid\_2
	+ AND virkningstidFra <= Virkningstid\_2
	+ AND virkningstidTil > Virkningstid\_2
	+ AND status = Status\_2
 |
| **Sikkerhed:**\* |

## Forretningsmæssig beskrivelse af hændelser

Registerprojekterne har i forbindelse med udarbejdelse af løsningsarkitekturerne, defineret deres forretningsmæssige hændelser i nedenstående fælles skabelon, der blev udarbejdet med udgangspunkt i EDA-referencearkitekturen version 0.4, som DIGST fik udarbejdet.

Efterfølgende har DIGST, i samarbejde med GD1, GD2 (MBBL og GST) og KOMBIT udarbejdet et fællesoffentligt beskedformat for hændelser. Dette format er ikke helt magen til de definitioner, der fremgår af EDA-referencearkitekturen version 0.4.

Det har derfor været nødvendigt at opdatere skabelonen til beskrivelse af hændelser for GD1 og GD2. Nærværende afsnit beskriver forskellene mellem skabelonerne og giver en vejledning til, hvordan den nye skabelon udfyldes.

### Den oprindelige hændelsesskabelon

Nedenstående er en gengivelse af den oprindelige skabelon til beskrivelse af forretningsmæssige hændelser, inklusiv mulighed for filtrering af hændelsesbeskeder i forbindelse med opsætning af abonnementer. Dog er vejledningen tilrettet, så den er i overensstemmelse med de afklaringer, der er foretaget siden skabelonen blev udarbejdet. Der kan filtreres på alle attributter, bortset fra eventuelle medsendte forretningsdata.

Filtreringsfelterne, er de generelle egenskaber for alle beskeder, jf. nedenstående.

|  |  |
| --- | --- |
| Hændelsesnavn: | Navngivning jf. GD1/GD2 standard, fx ”EJDmHBestemtFastEjendom” |

|  |
| --- |
| Teknisk del af hændelsesbesked |
| Standardiseret teknisk information relateret til det fælles beskedformat med information om afsendende system, tidspunkt, routinginformation, sikkerhed, signering mv.Ikke relevant for den logiske specifikation – bortset fra evt. sikkerhedskrav (roller mv). |
| Sikkerhed | Angivelse af evt. sikkerhedskrav ift. krav om bestemte roller og evt. krav i relation til en dataafgræsning. |

|  |
| --- |
| Generelle egenskaber for alle beskeder |
| Indeholder standardinformationer som beskedmodtager kan forvente i alle beskeder – herunder en række informationer fastlagt af de generelle modelleringsregler (aktør, virkningstid etc.).Informationer vil ikke alle være udfyldt altid – det skal give mening i den konkrete kontekst.I relation til en logisk beskrivelse af hændelser i GD1/GD2 skal der tages stilling til nedenstående. |
| Objekt type: | Her angives de entydige navne fra den fællesoffentlige model for hovedobjektet eller hovedobjekterne (hvis flere). |
| Objekt id(er): | Entydig ID (UUID) fra modelleringsregler på hovedobjektet eller hovedobjekterne (hvis flere – fx flere BFE ved en udstykning)  |
| Status: | Status på objektet fra modelleringsregler.  |
| Stedbestemmelse: | Geografisk afgrænsning, som hændelsen vedrører. Fx bygningens geokodning, matrikelfladen eller kommunen. |
| Forretningsområde: | Fra hvilket forretningsområde stammer hændelsen – fx FORMSvarer til modelleringsregler. |
| Opgave/aktivitet: | Mere specifik beskrivelse af hændelsen – hvad er det for en opgave eller aktivitet, der har udløst hændelsen. Hvis fx en abonnent kun ønsker at abonnere på udstykning, men er ligeglad med andre opgaver som arealoverførsel, er det så via opgave/aktivitet dette abonnement kan præciseres? Vis ikke hvor er information så? |

|  |
| --- |
| Forretningsdata |
| Indeholder øvrige informationer som medsendes - angivet som objekttyper eller attributter. Der kan ikke abonnement filtreres på dette data, da indholdet ikke vil være kendt af beskedfordeleren. |
| Navn på en attribut eller på et objekt | Det forretningsmæssige data, der sendes sammen med beskeden. Det vil sige data, der alternativt kan fremsøges via identifikationen fra ”objekttype”.Specielt skal der her tages stilling til, hvorledes evt. øvrige objekter eller objekttider kommunikeres til modtageren. Fx hvorledes sagsid og/eller sagsdata medsendes en udstykningssag omfattende et antal BFE-numre angivet under ”Objekt id(er)”. |

### Det fællesoffentlige beskedformat

Baggrunden for den nye skabelon er, som tidligere omtalt, det aftalte fællesoffentlige beskedformat. Formatets informationsmodel er illustreret nedenfor[[1]](#footnote-2).



En stor del af informationsmodellen er af teknisk karakter og er således kun relevant for de leverandører, der skal udvikle funktionalitet til at modtage hændelser.

I forhold til den forretningsmæssige specifikation af hændelser og sammenhængen til den gamle hændelsesskabelon, er følgende elementer relevante:

* Objektet ”Objektdata”
* Attributten ”sikkerhedsklassifikation” tilhørende objektet ”Leveranceinformation”
* Attributterne ”beskedtype”, ”tværgåendeProces” tilhørende objektet ”Filtreringsdata”
* Attributterne ”objektID”, ”objekttype”, ”status”, ”objekthandling”, ”opgaveemne” tilhørende objektet ”Objektregistrering”
* Objektet ”Stedbestemmelse”
* Objektet ”RelateretObjekt”

Derudover er der, af hensyn til abonnementsfiltrering, behov for en yderligere attribut: objektAktion, der fortæller om hændelsen omhandler en oprettelse, opdatering, nedlæggelse, genaktivering eller sletning.

Denne ekstra attribut ”objektAktion”, antages at eksisterer i de følgende beskrivelser.[[2]](#footnote-3)

Mappes dette til den gamle skabelon, ser det således ud:

|  |  |
| --- | --- |
| Hændelsesnavn: | Beskedtype (defineres som objekttype + objektAktion) |

|  |
| --- |
| Teknisk del af hændelsesbesked |
| Standardiseret teknisk information relateret til det fælles beskedformat med information om afsendende system, tidspunkt, routinginformation, sikkerhed, signering mv.Ikke relevant for den logiske specifikation – bortset fra evt. sikkerhedskrav (roller mv). |
| Sikkerhed | sikkerhedsklassifikation |

|  |
| --- |
| Generelle egenskaber for alle beskeder |
| Indeholder standardinformationer som beskedmodtager kan forvente i alle beskeder – herunder en række informationer fastlagt af de generelle modelleringsregler (aktør, virkningstid etc.).Informationer vil ikke alle være udfyldt altid – det skal give mening i den konkrete kontekst.I relation til en logisk beskrivelse af hændelser i GD1/GD2 skal der tages stilling til nedenstående. |
| Objekt type: | objekttype |
| Objekt id(er): | objektID |
| Status: | status |
| Stedbestemmelse: | stedbestemmelse, enten som reference til et geoobjekt på DAF, en DAGI inddeling eller et selvstændigt geoobjekt |
| Forretningsområde: | Opgaveemne (FORM)  |
| Opgave/aktivitet: | Objekthandling |

|  |
| --- |
| Forretningsdata |
| Indeholder øvrige informationer som medsendes - angivet som objekttyper eller attributter. Der kan ikke abonnement filtreres på dette data, da indholdet ikke vil være kendt af beskedfordeleren. |
| Navn på en attribut eller på et objekt | Objektdata |

Da det nye format indeholder en yderligere attribut og et yderligere objekt, som kan være relevante i forbindelse med GD1 og GD2 hændelser: ”tværgåendeProces” samt ”RelateretObjekt”, skal disse tilføjes til skabelonen.

### Den nye hændelsesskabelon

\* angiver obligatoriske attributter

|  |
| --- |
| Overordnede beskedindhold (filtreringsdata) |
| Identifikation af selve beskeden, samt beskedens sikkerhedsklassifikation. |
| \*Beskedtype(Hændelsesnavn) | Navngivning jf. GD1/GD2 standard, fx ”EJDmHBestemtFastEjendom”.Hvis der er forskel på indholdet af hændelsesbeskederne afhængig af objektAktion, skal der udfyldes flere skabeloner med objektAktion som suffiks, fx ”Oprettet”, ”Opdateret”, ”Nedlagt”, ”Genaktiveret”, ”Slettet” |
| TværgåendeProces | En arbejdsgang, som involverer en række aktører, som forårsagede hændelsesbeskeden.TværgåendeProces skal være en værdi fra en fast registerspecifik klassifikation. |
| \*Sikkerheds-klassifikation: | Angivelse påkrævet sikkerhedsniveau (1-4), for modtagelse af hændelsesbeskeden. Sikkerhedsniveauerne er beskrevet i afsnit 5.3.4. Sikkerhedsklassifikation er dataspecifik og kan derfor varierer for den samme beskedtype. Skabelonen udfyldes med de mulige værdier en beskedtype kan have. |

|  |
| --- |
| Generelle egenskaber (objektregistrering) |
| Indeholder de informationer som beskedmodtager kan anvende til filtrering på hændelsesabonnementer.  |
| \*Objekt type: | Her angives de entydige navne fra den fællesoffentlige model for hovedobjektet, hvis der er tale om flere Objekt typer, oprettes der en ”Generelle egenskaber”-tabel pr. Objekt type. |
| \*Objekt id: | Entydig ID (UUID) fra modelleringsregler på hovedobjektet. |
| Status: | Status på objektet fra modelleringsregler.  |
| Stedbestemmelse: | Geografisk afgrænsning, som hændelsen vedrører. Fx bygningens geokodning, matrikelfladen eller kommunen. Angives enten som reference til et geoobjekt på DAF, en DAGI inddeling eller et selvstændigt geoobjekt. |
| Opgaveemne: | Fra hvilket forretningsområde stammer hændelsen, reference til FORM (http://www.form-online.dk/) på et passende niveau. |
| Objekthandling: | Mere specifik beskrivelse af hændelsen – hvad er det for en opgave eller aktivitet, der har udløst hændelsen. Hvis fx en abonnent kun ønsker at abonnere på udstykning, men er ligeglad med andre opgaver som arealoverførsel.Opgavehandling skal være en værdi fra en fast beskedtype-specifik klassifikation. |

|  |
| --- |
| Relaterede objekter |
| Et valgfrit element, der indeholder en reference til relaterede objekter til hændelsen, fx kunne en sagsbesked samle de relaterede BFE’er, der indgår i den sag hændelsesbeskeden vedrører.En besked kan have tilknyttet nul til mange relaterede objekter. Begge attributter er obligatoriske, hvis elementet anvendes. |
| Relateret objekt type: | Her angives de entydige navne fra den fællesoffentlige model for det relaterede objekt. |
| Relateret objekt id: | Entydig ID fra modelleringsregler på det relaterede objekt. |

|  |
| --- |
| Forretningsdata (objektdata) |
| Et valgfrit element, der indeholder øvrige informationer som medsendes - angivet som objekttyper og attributter. Der kan ikke abonnement filtreres på dette data, da indholdet ikke vil være kendt af beskedfordeleren. |
| Navn på en attribut eller på et objekt | Det forretningsmæssige data, der sendes sammen med beskeden. Det vil sige data, der alternativt kan fremsøges via identifikationen fra ”objekttype”. |

### Sikkerhedsklassifikationer

1. **Ikke fortrolige data (offentlige data).**
	* Eksempler:
	Navn, adresse, fødselsdato, stilling, arbejdstelefon.
2. **Fortrolige personoplysninger (persondatalovens §6) eller Fortrolige forretningsdata.**
	* Personoplysninger:
	Private oplysninger om eksempelvis økonomi, hemmelig adresse, skatteforhold, gæld, sygedage, tjenestelige forhold og familieforhold
	* Fortrolige forretningsdata:
	Agenda til et lukket møde, Energioplysninger, budgetforslag, Udbud(tilbudsevaluering)
3. **Følsomme personoplysninger (persondatalovens §7 og §8) eller Følsomme forretningsdata.**
	* Følsomme personoplysninger:
	Racemæssig / etnisk baggrund, politisk, religiøs, eller filosofisk overbevisning, fagforeningsforhold, seksuelle forhold, helbredsmæssige forhold. Strafbare forhold, væsentlige sociale problemer, andre rent private forhold.
	* Følsomme forretningsdata:
	Forarbejde til lov, Oplysninger undtaget for aktindsigt,
4. **Særligt beskyttede oplysninger (VIP-sager, Kritiske forretningsdata)**
	* VIP sager:
	De kongelige, politikere ect
	* Kritiske forretningsdata:
	Forsvarets bygninger

### Forberedende registeraktiviteter, inden udfyldelse af hændelsesskabelonen

Som nævnt i vejledningen ovenfor, skal en del af hændelsesattributterne udfyldes med værdier fra klassifikationer. Hvert register skal derfor definere disse klassifikationer, inden de kan færdiggøre udfyldelsen af hændelsesskabelonen.

Klassifikationerne skal defineres med anvenderne for øje, da klassifikationerne bliver de primære forretningsmæssige filtreringsparametre, når anvenderne opsætter deres hændelsesabonnement.

Følgende klassifikationer skal defineres:

|  Klassifikation | Indhold |
| --- | --- |
| TværgåendeProces(hvis attributten ønskes anvendt) | Der er her tale om en registerspecifik klassifikation, det vil sige en klassifikation, der er fælles for alle registreret hændelser.*Eksempelvis:* *Udstykning, sammenlægning af Ejerlejlighed, Ændring af kommunegrænse* |
| Status | Dette er en objekttype specifik klassifikation, som indgår i dobbelthistorik parametrene. |
| Opgaveemne | Denne klassifikation er FORM, på et passende niveau, fastlagt af de enkelte registre. |
| Objekthandling | Dette er den primære klassifikation, til filtrering af hændelser. Klassifikationen er specifik for de enkelte beskedtyper.*Eksempelvis:**Adgangspunkt, Bygningsareal, Benyttelseskode, Arealoverførsel, Kystlinje* |

## DLS skabelon til hændelser

Datafordeleren understøtter to metoder til hændelser:

1. Modtagelse af komplet besked fra registrene, til direkte distribution til abonnenterne; dette kaldes i DLS’en for en forretningsmæssig hændelse. (Register-genererede hændelser)
2. Generering af hændelser på baggrund at en dataopdatering fra registrene til Datafordeleren; dette kaldes i DLS’en for en datanær hændelse. (Datafordeler-genererede hændelser)

De to benævnelser er misvisende, da den aftalte metode til generering af forretningsmæssige hændelser, er det DLS’en kalder for datanære hændelser.

Det er dog nødvendigt at bevare denne skelnen overfor Datafordelerens leverandør, af hensyn til den indgået kontrakt og DLS skabelonen. For at reducere begrebsforvirring i dette afsnit, kalder vi de to metoder for ”Register-genererede hændelser” og ”Datafordeler-generede hændelser

### Register-genererede hændelser

Da register-generede hændelser, styres fuldstændigt af registrene, skal Datafordeleren kun kende ganske få oplysninger og der skal ikke foretages nogen mapning af felter i DLS’en. Jf. DLS skabelonen ”Forretningsmæssige hændelser”, skal følgende udfyldes pr. hændelse:

|  |  |
| --- | --- |
| Felt | Forklaring |
| Beskedtype | Et entydigt navn, der er en sammenkædning af Objekttype og ObjektAktion med navnestandardens suffiks på objekttypen.*Eksempelvis: EJDbHBygningOprettet, EJDbHBygningOpdateret, EJDbHBygningNedlagt, EJDbHBygningGenaktiveret, EJDbHBygningSlettet* |
| Entitet | Hovedobjekttype hændelsen omhandler.*Eksempelvis: Bygning, Ejerlejlighed, Adresse, KommuneInddeling.* |
| Format (XML/JSON) | Angivelse af hvilket format hændelsen genereres i – XML eller JSON. |

Bemærk venligst at der altid skal udfyldes en skabelon pr. objektAktion, pr. hændelse.

### Datafordeler-genererede hændelser

De Datafordeler-generede hændelser skal udfyldes efter DLS skabelonen i DLS afsnit 8 ”Datanære hændelser”. Felterne i denne skabelon referer til beskedformatet, der er vist i afsnit 5.3.2.

Datafordeleren kan håndtere følgende syntaks til udfyldelse af felter ved generering af en hændelse:

* Load([DataField]), hvor [DataField] er et felt fra entiteten i indlæsningsmodellen.
* Publish([UUID]), hvor [UUID] er felt fra den opdaterede entitet i udstillingsmodellen.
* Constant([tekst]) eller [tekst], hvor [tekst] indsættes som konstant værdi.
* UTCNow(), indsætter aktuelle UTC dato og tid.
* NewGUID(), indsætter en ny GUID.

Bemærk at [UUID] er datamodelfeltets id (teknisk nøgle til feltnavnet), der defineres i ”Datamodel XMI skabelonen”. Ved at anvende en id, kan man omdøbe feltnavnet, uden at skulle ændre på eksempelvis hændelsesdefinitionerne.

Nedenfor er vist koblingen mellem den nye hændelsesskabelon, jf. afsnit 5.3.3, og de felter, der skal udfyldes i DLS skabelonen version 1.1 fra 06.07.2015 samt hvilken syntaks, der skal anvendes.

| DLS skabelon felt | Hændelsesskabelon felt | Forklaring | DLS syntaks |
| --- | --- | --- | --- |
| **DLS skabelon sektion: Filtreringsdata** |
| Beskedtype | Beskedtype | Navnet på hændelsen, jf. afsnit 5.3.3. | Constant([beskedtype]) |
| BeskedansvarligAktoer | - | Den aktør, der er ansvarlig for hændelsen, det vil sige registrets navn. | Constant([registernavn]) |
| **DLS skabelon sektion: RelateretObjekt (repeterede sektion)** |
| ObjektId | Relateret objekt id | Entydig ID fra modelleringsregler på det relaterede objekt. | Load([feltnavn på det felt i dataopdateringen, der indeholder ID for det relaterede objekt]) |
| Objekttype | Relateret objekt type | Her angives de entydige navne fra den fællesoffentlige model for det relaterede objekt. | Constant([objekttype på det relaterede objekt]) |
| **DLS skabelon sektion: Objektregistrering** |
| Registreringsaktoer | - | Registreringsaktør fra dobbelthistorik attributterne på objektet | Load([feltnavn på det felt i dataopdateringen, der indeholder hovedobjektets registreringsaktør]) |
| Registreringstid | - | Registreringstid fra dobbelthistorik attributterne på objektet | Load([feltnavn på det felt i dataopdateringen, der indeholder hovedobjektets registreringstidFra]) |
| Status | Status | Status fra dobbelthistorik attributterne på objektet | Load([feltnavn på det felt i dataopdateringen, der indeholder hovedobjektets status]) |
| ObjektansvarligAktoer | - | Navnet på den dataansvarlige myndighed | Constant([myndighed]) |
| ObjektId | Objekt id | ID på det objekt, der opdateres | Load([feltnavn på det felt i dataopdateringen, der indeholder hovedobjektets objektId]) |
| Objekttype | Objekttype | Hovedobjektet i hændelsen (og dataopdatering). | Constant([objekttype]) |
| Objekthandling | Objekthandling | Mere specifik beskrivelse af hændelsen – hvad er det for en opgave eller aktivitet, der har udløst hændelsen. Hvis fx en abonnent kun ønsker at abonnere på udstykning, men er ligeglad med andre opgaver som arealoverførsel. | Load([feltnavn på det felt, der indeholder klassifikationsværdien til objekt handling])Desuden skal hele udfaldsrummet for Objekthandling beskrives. |
| Opgaveemne | Opgaveemne | FORM kode for hvilket forretningsområde hændelsen stammer fra. | Constant([FORM kode]) |
| RegistreringsId | - | Anvendes umiddelbart ikke i GD1’s og GD2’s implementering af dobbelthistorik. | - |
| **DLS skabelon sektion: Stedbestemmelse** |
| StedbestemmelsesGeometri | Stedbestemmelse | Geometri, i form af et punkt eller et areal, der hører til hovedobjektet og medsendes i opdateringen. | Load([feltnavn på det felt i dataopdateringen, der indeholder geometri til stedbestemmelse]) |
| *eller* |  |  |  |
| StedbestemmelsesReference[[3]](#footnote-4) | Stedbestemmelse | Reference til eksisterende geometri i form af en DAGI inddeling | Constant([ID for den relevante DAGI-objektklasse])*Eller*Constant([httpURI link til DAGI felt])[[4]](#footnote-5)*+*Load([feltnavn på det felt i dataopdateringen, der indeholder DAGI referencen]) |
| **DLS skabelon sektion: Leveranceinformation** |
| TransaktionsId | - | Anvendes umiddelbart ikke. | - |
| Sikkerhedsklassifikation | Sikkerhedsklassifikation | Sikkerhedsklassifikation (1-4) | Constant([sikkerhedsklas-sifikationsnummer])*Eller, hvis sikkerhedsklassifikationen kan varierer inden for den samme beskedtype*Load([feltnavn på det felt i dataopdateringen, der indeholder sikkerhedsklassifikationen]) |
| **DLS skabelon sektion: Modtagerhandling** |
| Handling | - | Hvis det ønskes, at Datafordeleren skal anmode modtageren af beskeden om at kvittere for modtagelse, angives ”Kvittering” her. | Constant(”Kvittering”)*Eller*- |
| Responsmodtager | - | Udfyldes af Leverandøren ud fra URL til kvitteringsservicen og ObjektID | Constant([URL til kvitteringsservice])*Eller*- |
| **DLS skabelon sektion: Beskeddata** |
| Objektreference[[5]](#footnote-6) | Forretningsdata | Anvendes til at vedlægge et link til at hente objektets data, dvs. objektets ID prefixet med en sti til objektet på Datafordeleren. Eksempelvis et link til København Kommune: <http://www.datafordeler.dk/DAGI/Kommuneinddeling?id=0101> | - |
| *eller* |  |  |  |
| Dataskema | Forretnings-data | Sti til XML-skema og JSON skema, som beskriver strukturen i de data, der skal vedlægges som beskeddata. | Sti til dataskemaer oplyses, når DLS’en udfyldes. Datafordeleren anvender herefter skemaet til at vedlægge beskeddata til de faktiske beskeder. |
| Dataobjekt | Forretnings-data | Henvisning til skemafiler (både XML og JSON) for Dataobjektet udfyldt med mapning til udstillingmodellen.  | Navne på dataskemaer oplyses, når DLS’en udfyldes, begge skemafiler skal leveres sammen med DLS specifikationen.Datafordeleren anvender herefter skemaerne til at vedlægge beskeddata til de faktiske beskeder. |
| **Mangler pt. i DLS skabelonen, men medtages i en kommende version af DLS skabelonen[[6]](#footnote-7)** |
| - | TværgåendeProces | En arbejdsgang, som involverer en række aktører, som forårsagede hændelsesbeskeden. | Load([feltnavn på det felt, der indeholder klassifikationsværdien til den tværgående proces])Desuden skal hele udfaldsrummet for TværgåendeProces beskrives. |
| **Mangler pt. i DLS skabelonen, men er aftalt på møde mellem GD1-GD2 og GD7, 01.07.2015** |
| - | ObjektAktion | ”Oprettet”, ”Opdateret”, ”Nedlagt”, ”Genaktiveret”, ”Slettet”[[7]](#footnote-8) | Constant([objektAktion]) |

### Hvilke opgaver giver de Datafordeler-generede hændelser til registrene

Datafordeler-generede hændelser giver to opgaver til registrene:

1. Forretningsmæssig definition af hændelserne
2. Implementering af de dynamiske felter i hændelserne, i forbindelse med dataopdateringer til Datafordeleren

Den forretningsmæssige definition af hændelser, er en kendt opgave, der indebærer en endelig forretningsmæssig beskrivelse af hændelser, via skabelonen i afsnit 5.3.3.

Implementering af de dynamiske hændelsesfelter er en opgave, der ikke tidligere har været afklaret. Med den aftalte løsning til generering af hændelser via Datafordeleren, bliver det registrenes opgave – i forbindelse med dataopdateringer - at oplyse hvornår en hændelse skal genereres, samt hvilket indhold der skal være i hændelsen.

I afsnit 5.4.3 er der, i kolonnen ”DLS Syntaks”, brugt 2 af de mulige metoder til angivelse af hændelsesindhold:

**Constant:**

Disse udfyldes kun i forbindelse med DLS’en og er således statiske værdier for hændelserne. Der er ikke krav til at værdierne er udstillet på Datafordeleren.

**Load:**

Dette er de dynamiske felter i hændelserne, som varierer for hver hændelsesbesked.

Load-felterne kan opdeles i to kategorier:

**De felter der indgår i selve dataopdateringen**, eksempelvis dobbelthistorik attributterne.

* Denne kategori giver ikke nogle opgaver til registrene, som de ikke allerede har i forbindelse med dataopdateringerne.

**De felter der ikke indgår i dataopdateringen**, men skal anvendes til abonnement og filtrering af hændelsesbeskeder.

* Denne kategori, kræver at registrene, i forbindelse med deres dataopdateringer, implementerer en logik, der kan supplere dataopdateringen med yderligere elementer til anvendelse i hændelserne.

### Hvornår skal der genereres Datafordeler-genererede hændelser

Af DLS skabelonen fremgår følgende under afsnit 8 – Hændelsesbeskeder:

*Datafordeleren skaber kun datanære hændelsesbeskeder når datasektionen for den enkelte ajourføring indeholder et ’CreateEvent’-element med værdien ’1’, ’<CreateEvent>1</CreateEvent>’. Det er således op til Dataleverandøren at styre hvornår der skal skabes datanære hændelsesbeskeder.*

Dette betyder at registrene, i forbindelse med de enkelte dataopdateringer, skal angive om der skal genereres en hændelse.

Grundet virkemåden af dobbelthistorikken, er reglerne for generering af hændelser afhængig af hvilken logisk operation der sker på objektet.

Hvis der er tale om en nyoprettelse af et objekt – eller oprettelse af en ny samtidig version af et eksisterende objekt, vil dataopdateringen kun indeholde et <Action>Create</Action> element. Dette element skal have tilføjet *<CreateEvent>1</CreateEvent>* for generering af \*Oprettet hændelser.

For dataopdateringer, eksklusiv logisk sletning, vil der være tale om både <Action>Update </Action> element(er) og <Action>Create</Action> element(er).

”Create” elementerne i denne sammenhæng, vil skulle have tilføjet *<CreateEvent>1 </CreateEvent>* for generering af \*Opdateret, \*Nedlagt, \*Genaktiveret hændelser.

”Update” elementerne vil ikke give anledning til generering af hændelser, da de ”kun” sætter registreringstidTil på eksisterende forekomster på Datafordeleren.

For dataopdateringer, der er logiske sletninger af et objekt, vil der kun være tale om et <Action>Update</Action> element. Dette element skal have tilføjet *<CreateEvent>1</Create Event>* for generering af \*Slettet hændelser.

Da der ikke slettes forekomster på Datafordeleren, vil <Action>Delete</Action> ikke blive anvendt og der vil således heller ikke skulle genereres nogle hændelser.

## DLS skabelon

husk noget omkring prøvedata (fiktive data, der følger udstillingsmodellen)

1. Beskedformatet er offentliggjort på http://data.gov.dk/grunddatabesked/ [↑](#footnote-ref-2)
2. GD1-GD2 har aftalt med GD7 at attributten indføres i en kommende version af DLS skabelonen [↑](#footnote-ref-3)
3. DLS skabelonen er ikke entydig her, der skal både angives et DAGI felt i udstillingsmodellen og den værdi, der skal sammenlignes med. [↑](#footnote-ref-4)
4. Denne variant skal afklares med KMD [↑](#footnote-ref-5)
5. Denne metode til at vedlægge data, har umiddelbart ingen anvendelse i GD1-GD2, da registrene oftest har behov for at vedlægge et antal attributter. Attributterne vedlægge via dataskema+dataobjekt metoden. [↑](#footnote-ref-6)
6. Fremgår af udkast til version 1.5 af bilag 2 tjeneste skabelon (endnu ikke offentliggjort) [↑](#footnote-ref-7)
7. Disse værdier er ikke endeligt aftalt med GD7, men forventes at være mulige. [↑](#footnote-ref-8)