

# KOB

## Funktionsbeskrivelser for de fælleskommunale forretningstjenester

Januar 2012

Udført af  
Silverbullet A/S  
For og på vegne af  
KOMBIT



## Indholdsfortegnelse

Indledning.....	3
Formål med funktionsbeskrivelsen .....	3
Indhold og Metode.....	3
Ordliste/begrebsdefinitioner.....	3
De fælleskommunale støttesystemer .....	5
Klassifikation.....	5
Organisation .....	5
Part .....	5
Forretningsbehov .....	6
Klassifikation.....	6
Klassifikationstjenestens egne forretningsbehov.....	6
Fagsystemernes forretningsbehov for klassifikationstjenesten.....	7
Organisation .....	8
Organisationstjenestens egne forretningsbehov .....	8
Fagsystemernes forretningsbehov for organisationstjenesten .....	9
Part .....	10
Partstjenestens egne forretningsbehov .....	10
Fagsystemernes forretningsbehov for partstjenesten .....	12
Love og regler .....	14
Persondataloven.....	14
Sikkerhedsbestemmelserne .....	15
Mindre betydende regler og forskrifter .....	15
Forvaltningsloven .....	15
Løsningsarkitektur .....	16
Principper .....	16
Robust princip.....	17
Den tykke klient.....	17
Den anbefalede arkitektur.....	19
"Farveladen" .....	19
Arkitekturen.....	20
Driftsmodel.....	24
Forventede dataindhold i de centrale repositories.....	25

Snitfladebeskrivelserne .....	25
OIO Klassifikation.....	25
OIO Organisation .....	26
Part .....	26
Informationsmodel.....	27
Oversigt over tabeller & figurer.....	30
Tabeller .....	30
Figurer.....	30

## Indledning

KOB projektet er nu etableret med det formål at udarbejde krav og arkitektur for 4 af de støttesystemer under rammearkitekturen. Det er Klassifikation, Organisation og Beskedfordeler som i det oprindelige materiale, men desuden er Part nu blevet en del af KOB-projektet grundet dets nærhed til Organisation og lighed med Organisation og Klassifikation.

## Formål med funktionsbeskrivelsen

Formålet med funktionsbeskrivelse er at løfte tre centrale realiseringsovervejelser:

- Identifikation af forretningsbehov
- Etablering af løsningsarkitektur
- Driftsovervejelser

## Indhold og Metode

Nærværende dokument vil alene adressere de mere passive støttesystemer; Klassifikation, Organisation og Part, mens Beskedfordeleren vil blive adresseret i et andet tilsvarende dokument.

Funktionsbeskrivelsen er et målrettet arkitektprodukt til brug internt i KOMBIT blandt KOMBITs arkitekter på baggrund af system til system krav. Derfor bygger rapporten alene på krav og behov formuleret af KOMBITs arkitekter og ikke af kommunernes direkte krav og behov. Disse forventes dækket af de – af KOMBITs arkitekter – repræsenterede fagsystemer.

Rapporten vil beskrive en anbefalet løsningsarkitektur og kort kommenter på driftsmodel.

## Ordliste/begrebsdefinitioner

Begreb	Beskrivelse
<b>Repository</b>	Begrebet repository beskriver et centralt og autoritativt register, der er målrettet passiv lagring og deling af fælles data. Ved at bruge det engelske "repository" i stedet for det danske "register", følger det den normale anvendelse af repository inden for it i dagens Danmark. I denne rapport bruges begrebet "repository" for de fælles kommunale registre til opbevaring og deling af data.
<b>Støttesystem</b>	Rammearkitekturen har begrebsat visse systemer som støttesystemer. Det gælder bl.a. de systemer, der ligger under KOB projektet. Der blandes lidt rundt mellem støttesystem og tjeneste, da de grundlæggende dækker de samme løsninger. Støttesystems-begrebet er brugt i denne rapport for de systemer, der indeholder de fælleskommunale repositories.
<b>Tjeneste</b>	Begrebet tjeneste anvendes i dette dokument i betydning af et system, der leverer flere forskellige ydelser/funktionaliteter, og som anvendes af andre systemer til opslag, arkivering eller blot funktionalitetsudvidelse. Én tjeneste vil oftest bestå af mere end én service, og det er netop grunden til at vælge det danske "tjeneste" frem for det engelske "service", der ellers traditionelt er anvendt inden for it i dagens Danmark. I denne rapport beskriver begrebet "tjeneste" de ydelser støttesystemerne leverer og ikke støttesystemet selv.

**Fagsystem**

Fagsystemet er det system, der leverer den faglige arbejdsgang og interaktionen med brugerne. Fagsystemet står for sagsbehandlingsprocessen og er anvendte af støttesystemerne. Fagsystem er et bredere begreb end ydelsessystemer (som rammearkitekturen ver. 0.6 reelt er modelleret i forhold til). Fagsystemer dække alle de systemer i kommunerne, der holder konkret sagsbehandlingsprocesser og som interagerer med brugerne.

**Autoritativ kilde**

Hvor støttesystemet kan repræsentere autoritative data, er den autoritative kilde leverandør af autoritative data til støttesystemet. Og den autoritative kilde kan både være leverandør af en delmængde af dataene, eller en delmængde af de enkelte datasæt. Dvs. en leverandør kan levere vise klassifikationer og være autoritativ kilde på disse, mens andre klassifikationer kommer fra andre autoritative kilder, eller også kan en autoritativ kilde levere oplysninger på fx en person, der så efterfølgende beriges fra en anden autoritativ kilde med fx ansættelsesoplysninger.

Tabel 1 Ordliste og begrebsdefinitioner

## De fælleskommunale støttesystemer

### Klassifikation

Støttesystemet klassifikation leverer et centralt register til håndtering af fælles klassifikationer, dvs. standardiserede lister og systematikker med ensartet semantisk opbygning. Fællesbegrebet er både i forhold til deling mellem løsninger (både fagsystemer og støttesystemer) og til deling mellem kommunerne på tværs.

De standardiserede lister og systematikker er generelt meget statiske og bruges primært til at skabe fælles forståelse og sammenhænge på tværs. Alle oplysninger i Klassifikation er bitemporal i natur.

Som udgangspunkt er der ingen begrænsninger i hvilke standardiserede lister og systematikker, der kan understøttes af støttesystemet klassifikation. Det er alt lige fra de fælleoffentlige systematikker som FORM og indenrigsministeriets kontoplan til stillingsbetegnelser og fagsystemroller for den enkelte kommune.

### Organisation

Støttesystemet organisation leverer et fælles register til håndtering af de kommunale organisationer og herunder også de enkelte medarbejdere i kommunerne. Støttesystemet og dens data er primært rettet mod den enkelte kommune, da organisatoriske data og medarbejderinformationer reelt kun har intern interesse i den enkelte kommune, dog kan der være organisatoriske oplysninger, der udstilles til alle kommune og til øvrige interessenter som borger.dk.

De organisatoriske data er statiske i natur, men der forventes en regelmæssig justering af organisationerne og deres medarbejdere. Alle organisatoriske oplysninger understøtter bitemporalitet.

Som udgangspunkt er der ingen begrænsninger i hvorledes kommunernes organisation er opbygget og sammensat, og derfor understøtter Organisation enhver tænkelig organisation, lige fra den linjeorienterede ledelsesstruktur til det tværgående og løskoblede festudvalg.

### Part

Støttesystemet part leverer et centralt register til håndtering af de fælles egenskaber og data for de parter, der relateres til sager og/eller dokumenter. Parterne kan være enten person, virksomhed, forening, ejendom eller lignende juridisk/operationel entitet.

Parts data er i store træk statiske, men kan i visse henseender opdateres hyppigt. Alle partsoplysninger understøtter bitemporalitet.

Partsoplysninger vedligeholdes af respektive autoritative kilder som CPR, CVR, BBR osv., men kan også vedligeholdes fra den enkelte kommune, hvis denne opretter kommunespecifikke parter. Generelt er alle parter tilgængelig for den enkelte sagsbehandler eller det enkelte system, dog kun som reference og ikke med oplysninger om partsrelationen til konkrete sager, dokumenter eller lignende.

## Forretningsbehov

Der er grundlæggende to måde at se forretningsbehovene for en fælleskommunal forretningstjeneste på:

1. Forretningstjenestens egne forretningsbehov, dvs. de forretningsbehov, der skal til for at kunne levere de forventede ydelser til anvendelsestjenerne (fagsystemerne)
2. Fagsystemernes forretningsbehov for forretningstjenesten, dvs. de konkrete behov fagsystemet har til en leverandør af forretningstjenesten (og dens data)

## Klassifikation

### Klassifikationstjenestens egne forretningsbehov

Disse forretningsbehov er centrale for en klassifikationstjeneste, der skal kunne levere både de fælles kommunale og de kommuneindividuelle klassifikationer til relevante fagsystemer uanset hvor disse driftes.

Forretningsbehov	Uddybning
<b>Autoritativt centralt repository over klassifikationer</b>	<p>Det grundlæggende i klassifikationstjenesten er at levere fælles standardiserede lister og systematikker, og det kræver ét autoritativt og centralt repository, der kan holde:</p> <ol style="list-style-type: none"> <li>1. Fælleskommunale</li> <li>2. Kommuneindividuelle</li> <li>3. Systemspecifikke</li> </ol> <p>Klassifikationer, tillige med en sammenblanding af samme.</p>
<b>Individuelle "ejer" af klassifikationer</b>	<p>Den enkelte klassifikation har en og kun en dedikeret "ejer", der er ansvarlig for vedligehold. Det kan være en "ejer" i form af autoritative kilder som FORM, KLE osv. Eller det kan være den enkelte kommune eller det enkelte (fag)system, der "ejer" sine kommuneindividuelle/systemspecifikke klassifikationer</p> <p>Selvom der er individuelle "ejere", kan alle se alle klassifikationer (dog kan kommuneindividuelle kun ses af "kommunens systemer")</p>
<b>Vedligeholdes direkte via udstillede snitflader</b>	<p>Da klassifikationstjenesten er et centralt autoritative repository, hvor alle data skal findes, er det nødvendigt at alle data vedligeholdes direkte i det centrale repository via de udstillede snitflader:</p> <ul style="list-style-type: none"> <li>• Administrationsklient</li> <li>• Webservices (SOA)</li> <li>• Bulk import</li> </ul>
<b>Primært passiv</b>	<p>Ansvar for opdateringen er "ejers", dog kan fx KOMBIT agere på vegne af de autoritative kilder som FORM, KLE m.fl.</p> <p><b><i>Det betyder med andre ord, at klassifikationstjenesten vil antage at data er valide uanset alder eller kvalitet.</i></b></p>
<b>Udsender beskeder via Beskedfordeler</b>	<p>Når data opdateres udsender klassifikationstjenesten beskeder via Beskedfordeleren. Der udsendes to forskellige typer beskeder:</p> <ol style="list-style-type: none"> <li>1. Klasser er direkte blevet ændret</li> <li>2. Klasse er indirekte ændret ved bitemporalt periodeskifte</li> </ol>

Tabel 2 Klassifikationstjenestens egne forretningsbehov

Disse behov tegner et klart og ensartet billede af en klassifikationstjeneste, der er tilgængelig med yderst begrænset funktionalitet i én central implementering (med mulighed for logiske instanser pr. kommune).

### Fagsystemernes forretningsbehov for klassifikationstjenesten

Forretningsbehov	Uddybning
<b>Filtrering/udvælgelse af relevante data (dataindhold)</b>	Relevans kriterium defineret af fagsystem på baggrund af data i klassifikationens metadata. Dvs. i forespørgslen angiver fagsystemet hvilke data der ønskes. Fagsystemet vil ikke stå for filtreringen.
<b>Se både generelle og kommuneindividuelle data</b>	<p>Det skal være muligt ved ét opslag at hente både de fælleskommunale og kommuneindividuelle klassifikationer.</p> <p>Angivelsen af om der både ønskes fælleskommunale og kommuneindividuelle klassifikationer skal kunne angives af fagsystemet i forbindelse med forespørgslen.</p>
<b>Hente/læse data</b>	<p>Der er forskellige behov, der skal indfries i forbindelse med hent/læs af data fra klassifikationstjenesten.</p> <ul style="list-style-type: none"> <li>• Data skal være tilgængelig "hele tiden"</li> <li>• Svartiderne skal være "korte"</li> </ul> <p>Disse behov afstedkommer disse relaterede behov:</p> <ul style="list-style-type: none"> <li>• Replikadata skal findes lokalt, dvs. enten i fagsystemet eller i en lokal instans af klassifikationskomponenten <ul style="list-style-type: none"> <li>○ Caching i lokal instans af klassifikationskomponenten for at opnå opetid og gode svartider uden at kræve replikadata</li> <li>○ Evt. med initiel-caching ved opstart af lokal instans. Skal være konfigurerbar fra fagsystem</li> </ul> </li> <li>• Live opslag til centralt repository muligt, hvis svartider og opetid er acceptabelt, kan fx være ved sjældne anvendte klassifikationer</li> </ul>
<b>Asynkron data</b>	<p>Et fagsystem kan vælge at gemme data i fagsystemet, hvis det er kritiske data, der ikke bare må blive opdateret afspejlende de centrale, autoritative data.</p> <p>Dette kan også kaldes, at fagsystemet har asynkron data i forhold til det centrale og autoritative repository.</p> <p>De asynkron data vedligeholdes (semi-)manuelt i fagsystemet, idet evt. ændringerne kan have store konsekvenser.</p>
<b>Have besked ved ændringer (via Beskedfordeler)</b>	<p>Når der sker ændringer i klassifikationerne, skal fagsystemet have en besked, uanset om det er asynkron eller live data, da ændringer i klassifikationerne kan afstedkomme en handling i fagsystemet enten automatisk, semiautomatisk eller manuelt.</p> <p>Der ønskes besked om</p> <ol style="list-style-type: none"> <li>1. Direkte ændring af data på klasser uanset om det er trådt i kraft eller ej</li> <li>2. Indirekte ændring af data på klasser, når disse "træder i kraft" ved bitemporalt periodeskifte.</li> </ol>



<b>Kunne oprette og vedligeholde egne lister – Direkte og uden caching</b>	Et fagsystem skal kunne "eje" klassifikationer. Disse skal vedligeholdes direkte i det centrale, autoritative repository, og dermed uden at der caches lokalt i forhold til fagsystemet.  Der er ikke krav om 100 % opetid for dette behov.
<b>Vil helst ikke have ansvar for integration</b>	Integrationen fra fagsystem til klassifikationstjenesten skal helst ikke "belaste" fagsystemet, men være generelt tilgængelig for fagsystemet og/eller andre systemer hos samme leverandør

Tabel 3 Fagsystemernes forretningsbehov for klassifikationstjenesten

Hvor klassifikationstjenestens egne forretningsbehov peger på én central komponenten, er fagsystemets krav mere diffuse. Der er stadig behov for ét centralt og autoritativt repository, men der er behov for mere for at sikre opetid og svartider. Det kan være i form af en klient/plugin eller tilsvarende som leveres sammen med det centrale repository, eller også kan det være direkte understøttelse i fagsystemet.

## Organisation

### Organisationstjenestens egne forretningsbehov

Disse forretningsbehov er centrale for en organisationstjeneste, der skal kunne levere kommunens egne organisationsinformationer til relevante fagsystemer uanset hvor disse driftes, samt levering af fælles organisatoriske (kontakt-)data, hvor det er relevant.

Forretningsbehov	Uddybning
<b>Autoritativt centralt repository over organisationsdata</b>	Det grundlæggende i organisationstjenesten er at samle og levere alle organisatoriske data inkl. medarbejderoplysninger for den enkelte kommune, og det kræver ét autoritativt og centralt repository, der kan holde data for alle kommuner og tilgås fra alle relevante fagsystemer uanset hvor de driftes.  Organisationsdata er som udgangspunkt "privat" til den enkelte kommune, men der kan dog åbnes op for, at organisationsdata kan deles på tværs af kommunerne.
<b>Individuelle "ejer" af organisationer</b>	Den enkelte organisation har en og kun en dedikeret "ejer", der er ansvarlig for vedligehold. Den dedikerede ejer er som udgangspunkt kommunen og dennes udpegede gesandter.  Det er antageligt at de forskellige organisationer i kommunen har individuelle "ejere", der står for administration og vedligehold.
<b>Vedligeholdes direkte via udstillede snitflader</b>	Da organisationstjenesten er et centralt autoritative repository, hvor alle data skal findes, er det nødvendigt at alle data vedligeholdes direkte i det centrale repository via de udstillede snitflader: <ul style="list-style-type: none"> <li>• Administrationsklient</li> <li>• Webservices (SOA)</li> <li>• Bulk import</li> </ul>
<b>Primært passiv</b>	Ansvaret for opdateringen er "ejers".  Det betyder med andre ord, at organisationstjenesten vil antage at data er valide uanset alder eller kvalitet.

<b>Udsender beskeder via Beskedfordeler</b>	Når data opdateres udsender organisationstjenesten beskeder via Beskedfordeleren. Der udsendes to forskellige typer beskeder: <ol style="list-style-type: none"> <li>1. Organisationer er direkte blevet ændret</li> <li>2. Organisationer er indirekte ændret ved bitemporalt periodeskifte</li> </ol>
---	---

Tabel 4 Organisationstjenestens egne forretningsbehov

Disse behov tegner et billede af en organisationstjeneste, der er tilgængelig med yderst begrænset funktionalitet i én central implementering (med mulighed for logiske instanser pr. kommune og/eller pr. driftsmiljø). Da de organisatoriske data som udgangspunkt er "private" til kommunen, kan billedet for organisationstjeneste rumme muligheden for kommuneindividuelle organisationsimplementeringer, der løfter den primære forretningstjeneste, mens der udstilles fælles organisatoriske data via ét centralt repository. Den endelige model afklares i sammenhæng med de øvrige fælles forretningstjenester.

### Fagsystemernes forretningsbehov for organisationstjenesten

Forretningsbehov	Uddybning
<b>Filtrering/udvælgelse af relevante data (dataindhold)</b>	Relevans kriterium defineret af fagsystem på baggrund af data i organisationens metadata. Dvs. i forespørgslen angiver fagsystemet hvilke data der ønskes. Fagsystemet vil ikke stå for filtreringen.
<b>Se udvalgte kommuner</b>	Det skal være muligt ved ét opslag at hente enten den enkelte kommunes organisationer eller organisationer fra flere kommuner.  Angivelsen af om hvilke kommuner (eller blot hvilken kommune) skal kunne angives af fagsystemet i forbindelse med forespørgslen.
<b>Hente/læse data</b>	Der er forskellige behov, der skal indfries i forbindelse med hent/læs af data fra organisationstjenesten. <ul style="list-style-type: none"> <li>• Data skal være tilgængelig "hele tiden"</li> <li>• Svartiderne skal være "korte"</li> </ul> Disse behov afstedkommer disse relaterede behov: <ul style="list-style-type: none"> <li>• Replikadata skal findes lokalt, dvs. enten i fagsystemet eller i en lokal instans af organisationskomponenten <ul style="list-style-type: none"> <li>○ Caching i lokal instans af organisationskomponenten for at opnå opetid og gode svartider uden at kræve replikadata</li> <li>○ Evt. med initiel-caching ved opstart af lokal instans. Skal være konfigurerbar fra fagsystem</li> </ul> </li> <li>• Live opslag til centralt repository muligt, hvis svartider og opetid er acceptabelt, kan fx være ved sjældne anvendte organisatoriske oplysninger.</li> </ul>

#### Asynkron data

Et fagsystem kan vælge at gemme data i fagsystemet, hvis det er kritiske data, der ikke bare må blive opdateret afspejlende de centrale, autoritative data.

Dette kan også kaldes, at fagsystemet har asynkron data i forhold til det centrale og autoritative repository.

De asynkron data vedligeholdes (semi-)manuelt i fagsystemet, idet evt. ændringerne kan have store konsekvenser.

<b>Have besked ved ændringer (via Beskedfordeler)</b>	Når der sker ændringer i organisationerne, skal fagsystemet have en besked, uanset om det er asynkron eller live data, da ændringer organisationerne kan afstedkomme en handling i fagsystemet enten automatisk, semiautomatisk eller manuelt.  Der ønskes besked om <ol style="list-style-type: none"> <li>1. Direkte ændring af data på organisation uanset om det er trådt i kraft eller ej</li> <li>2. Indirekte ændring af data på organisation, når disse "træder i kraft" ved bitemporalt periodeskifte.</li> </ol>
<b>Vil helst ikke have ansvar for integration</b>	Integrationen fra fagsystem til organisationstjenesten skal helst ikke "belaste" fagsystemet, men være generelt tilgængelig for fagsystemet og/eller andre systemer hos samme leverandør

Tabel 5 Fagsystemernes forretningsbehov for organisationstjenesten

Både organisationstjenestens egne forretningsbehov og fagsystemets ditto udtrykker primært behov for ét centralt repository, specielt gør fagsystemets forretningsbehov om at kunne trække flere kommuners organisatoriske data ved ét kald det klart, at der skal være ét centralt repository med alle organisationsdata., men der er behov for mere for at sikre opetid og svartider. Det kan være i form af en klient/plug'in eller tilsvarende som leveres sammen med det centrale repository, eller også kan det være direkte understøttelse i fagsystemet.

## Part

### Partstjenestens egne forretningsbehov

Disse forretningsbehov er centrale for en partstjeneste, der skal kunne levere vedligeholdte og delte partsoplysninger for såvel kommune som for kommunerne på tværs.

Forretningsbehov	Uddybning
<b>Autoritativt centralt repository over partsdata</b>	<p>Det grundlæggende i partstjenesten er at samle og levere alle parts (person, virksomhed, forening m.m.) data i vedligeholdte og delt version. Delt er såvel internt i kommunen mellem dennes it-systemer såvel som det er eksternt for kommunen mellem de forskellige kommuner.</p> <p>Partsdata er som udgangspunkt fælles for alle kommuner og alle systemer dog således, at kommuneindividuelle parter er "private" til kommunen, og at personoplysninger generelt skal beskyttes meget tæt og ikke udstilles generelt.</p>

<b>Individuelle "ejer" af parter</b>	<p>Den enkelte part har en og kun en dedikeret "ejer", der er ansvarlig for vedligehold. Det kan være en "ejer" i form af autoritative kilder som CPR, CVR osv. Eller det kan være den enkelte kommune, der selv vedligeholde egne kommuneindividuelle parter. Det er muligt at vise dele af partsoplysningerne kan vedligeholdes af andre end partens' primære ejer. Dette vil dog afhænge af den endelige afgrænsning af funktionalitet og indhold af partstjenesten.</p> <p>Selvom der er individuelle "ejere", kan alle se alle parter (dog kan kommuneindividuelle kun ses af "kommunens systemer"), såfremt part indeholder mulighed for supplerende oplysninger ud over "ejerens" generelle oplysninger vil disse være "private" i forhold til den kommune, hvorfra de er tilføjet.<sup>1</sup></p>
<b>Vedligeholdes direkte via udstillede snitflader</b>	<p>Da partstjenesten er et centralt autoritative repository, hvor alle data skal findes, er det nødvendigt at alle data vedligeholdes direkte i det centrale repository via de udstillede snitflader:</p> <ul style="list-style-type: none"> <li>• Administrationsklient</li> <li>• Webservices (SOA)</li> <li>• Bulk import</li> </ul>
<b>Ansvarligt" repository</b>	<p>Ansaret for opdateringen er enten "ejers" eller tjenesten selv.</p> <p>En "ejer" kan vælge selv at stå for opdateringen via de udstillede snitflade. Eller også kan en "ejer" vælge at give besked (via Beskedfordeleren) om, at en part er opdateret hos den autoritative ejer. Herefter er det partstjenestens ansvar at opdatere sine data enten ved at bruge beskedens oplysninger eller ved at hente opdateret data fra den autoritative ejer.</p> <p>Partstjenesten har således tre metoder til opdatering, hvor initiativet reelt er hos den autoritative ejer:</p> <ul style="list-style-type: none"> <li>• "Ejer" opdaterer via udstillede snitflader</li> <li>• Partstjenesten opdaterer selv direkte via modtaget besked</li> <li>• Partstjenesten opdaterer selv direkte via opslag hos de/den autoritativ(e) ejer(e) efter at have fået besked om ændring via Beskedfordeleren: <ul style="list-style-type: none"> <li>○ Kan ske øjeblikkeligt når besked modtages</li> <li>○ Sker som batch; beskeder markerer hvilke parter, der skal opdateres</li> </ul> </li> </ul> <p>Men der findes en fjerde variant, der handler om, at partstjenesten selv kan være at igangsætte en opdatering alene baseret på, at data er blevet gammel, dvs.:</p> <ul style="list-style-type: none"> <li>• Partstjenesten opdaterer selv direkte via opslag hos den/de autoritativ(e) ejer(e) ud fra en trigger, der siger, at data er forældet (triggeren kan sættes individuelt pr. partstype eller pr. part)</li> </ul>
<b>Udsender beskeder via Beskedfordeler</b>	<p>Når data opdateres udsender organisationstjenesten beskeder via Beskedfordeleren. Der udsendes to forskellige typer beskeder:</p> <ol style="list-style-type: none"> <li>1. Part er direkte blevet ændret</li> <li>2. Part er indirekte ændret ved bitemporalt periodeskifte</li> </ol>

Tabel 6 Partstjenestens egne forretningsbehov

<sup>1</sup> Der er et udestående omkring part, da denne ikke er blevet behandlet lige så dybdegående forud for nærværende dokument. Part var ikke omfattet af Silverbullet's foranalyse til KOB Projektet.

Disse behov tegner et klart og ensartet billede af en partstjeneste, der er tilgængelig med yderst begrænset funktionalitet i én central implementering (med mulighed for logiske instanser pr. kommune). Specielt hele vedligeholdelsesfunktionaliteten taler for én central forretningstjeneste, da der ellers vil komme et unødigt højt overhead på antal opdateringer og kald op mod de autoritative kilder.

### Fagsystemernes forretningsbehov for partstjenesten

Forretningsbehov	Uddybning
<b>Filtrering/udvælgelse af relevante data (dataindhold)</b>	Relevans kriterium defineret af fagsystem på baggrund af data i partens metadata. Dvs. i forespørgslen angiver fagsystemet hvilke data der ønskes. Fagsystemet vil ikke stå for filtreringen.
<b>Se både generelle og kommuneindividuelle data</b>	<p>Det skal være muligt ved ét opslag at hente både de generelle og kommuneindividuelle partsoplysninger.</p> <p>Angivelsen af om der både ønskes generelle og kommuneindividuelle partsoplysninger skal kunne angives af fagsystemet i forbindelse med forespørgslen.</p>
<b>Hente/læse data</b>	<p>Der er forskellige behov, der skal indfries i forbindelse med hent/læs af data fra partstjenesten.</p> <ul style="list-style-type: none"> <li>• Data skal være tilgængelig "hele tiden"</li> <li>• Svartiderne skal være "korte"</li> </ul> <p>Disse behov afstedkommer disse relaterede behov:</p> <ul style="list-style-type: none"> <li>• Replikadata skal findes lokalt, dvs. enten i fagsystemet eller i en lokal instans af partskomponenten <ul style="list-style-type: none"> <li>○ Caching i lokal instans af partskomponenten for at opnå opetid og gode svartider uden at kræve replikadata</li> <li>○ Evt. med initiel-caching ved opstart af lokal instans. Skal være konfigurerbar fra fagsystem</li> </ul> </li> <li>• Live opslag til centralt repository muligt, hvis svartider og opetid er acceptabelt, kan fx være ved sjældne anvendte parter.</li> </ul>
<b>Asynkrone data</b>	<p>Et fagsystem kan vælge at gemme data i fagsystemet, hvis det er kritiske data, der SKAL være tilstedet for at fagsystemet kan anvendes.</p> <p>Dette kan også kaldes, at fagsystemet har asynkrone data i forhold til det centrale og autoritative repository.</p> <p>De asynkrone data vedligeholdes automatisk i fagsystemet så snart der er forbindelse mellem fagsystem og partstjenesten.</p>
<b>Have besked ved ændringer (via Beskedfordeler)</b>	<p>Når der sker ændringer i parter, skal fagsystemet have en besked, uanset om det er asynkrone eller live data, da ændringer i parter kan afstedkomme en handling i fagsystemet enten automatisk, semiautomatisk eller manuelt.</p> <p>Der ønskes besked om</p> <ol style="list-style-type: none"> <li>1. Direkte ændring af data på part uanset om det er trådt i kraft eller ej</li> <li>2. Indirekte ændring af data på part, når disse "træder i kraft" ved bitemporal periodeskifte.</li> </ol>

**Vil helst ikke have ansvar for integration** Integrationen fra fagsystem til partstjenesten skal helst ikke "belaste" fagsystemet, men være generelt tilgængelig for fagsystemet og/eller andre systemer hos samme leverandør

---

**Table 7 Fagsystemernes forretningsbehov for partstjenesten**

Hvor partstjenestens egne forretningsbehov peger på én central komponenten, er fagsystemets krav mere diffuse. Der er stadig behov for ét centralt og autoritativt repository, men der er behov for mere for at sikre opetid og svartider. Det kan være i form af en klient/plug'in eller tilsvarende som leveres sammen med det centrale repository, eller også kan det være direkte understøttelse i fagsystemet.

## Love og regler

Uanset hvordan man vender og drejer det, skal støttesystemerne respektere gældende lovgivning og regler og øvrige regulativer, der sigter mod den offentlige forvaltning.

Nedenstående liste er ikke udtømmende, da den alene beror på kendte aspekter og krav. Skal listen være udtømmende, skal dette adresseres uafhængigt<sup>2</sup> af denne funktionsbeskrivelse.

### Persondataloven<sup>3</sup>

Persondataloven indeholder en masse reguleringer, der skal sikre at personhenførbare data ikke misbruges eller falder i forkertes hænder, og det gælder bredt, og dermed også langt ud for de kommunale systemer og processer.

Det centrale for KOB projektet er Persondatalovens kapitel 4, §§ 5-14 og kapitel 11, §§ 41-42. Begge disse kapitler handler om behandlingen af data i systemerne eller i den generelle sagsbehandling i kommunerne. Persondataloven beskriver en række begreber, der kan bruges til at forstå ansvar:

Begreb	Persondatalovens definition <sup>4</sup>
<b>Personoplysninger</b>	Enhver form for information om en identificeret eller identificerbar fysisk person (den registrerede).
<b>Register</b>	Enhver struktureret samling af personoplysninger, der er tilgængelige efter bestemte kriterier, hvad enten denne samling er placeret centralt, decentralt eller er fordelt på et funktionsbestemt eller geografisk grundlag
<b>Dataansvarlig</b>	Den fysiske eller juridiske person, offentlige myndighed, institution eller ethvert andet organ, der alene eller sammen med andre afgør, til hvilket formål og med hvilke hjælpemidler der må foretages behandling af oplysninger.
<b>Databehandleren</b>	Den fysiske eller juridiske person, offentlige myndighed, institution eller ethvert andet organ, der behandler oplysninger på den dataansvarliges vegne.
<b>Behandling</b>	Enhver operation eller række af operationer med eller uden brug af elektronisk databehandling, som oplysninger gøres til genstand for.

Tabel 8 Begrebsdefinitioner fra Persondatalovens § 3.

Disse begreber (Tabel 8) hjælper lidt hen ad vejen til forståelsen af, hvad et system (register) er og hvem der har ansvar for hvad heri. Men ansvaret peger på bestemte personer og juridiske enheder ... dvs. alt andet end et system.

Ansvaret for overholdelse af persondatalovens bestemmelser påhviler således den sagsbehandlende kommune og dennes medarbejdere. Set fra KOB projektets synspunkt er dette komplicerende, da den enkelte kommune kan have individuelle principper, som støttesystemerne skal kunne respektere, men

<sup>2</sup> Dette bør udredes af jurister, da dette er grundlæggende for alle støttesystemer og fælles offentlige repositories.

<sup>3</sup> Persondataloven eller mere korrekt [lov om behandling af personoplysninger \(LOV nr. 429 31/05/2000\)](#) er den centrale spiller, når det gælder personhenførbare data, enten om dette er direkte eller indirekte.

<sup>4</sup> Jf. Persondatalovens Kapitel 2, Definitioner, "§3. I denne lov forstås ved ...."

hvad værre er, siger Persondataloven kun noget om person til system integrationer, og intet om system til system integrationer.

### Sikkerhedsbestemmelserne

Blandt andet til adressering af system til system integration er der i forlængelse af Persondataloven udarbejdet en række bestemmelser, der skal sikre at Persondataloven overholdes, disse bestemmelser er samlet i ” *Bekendtgørelse om sikkerhedsforanstaltninger til beskyttelse af personoplysninger, som behandles for den offentlige forvaltning*”<sup>5</sup>.

I bekendtgørelsen er det kapitel 2 (§§ 5 – 14) Generelle sikkerhedsbestemmelser, der er interessant, og helt konkret § 14, der er specielt interessant:

*”... § 14. Der må kun etableres eksterne kommunikationsforbindelser, hvis der træffes særlige foranstaltninger for at sikre, at uvedkommende ikke gennem disse forbindelser kan få adgang til personoplysninger...”*

Dette er med til at regulere principper og metoder for integration mellem støttesystem og fagsystem.

### Mindre betydende regler og forskrifter

#### Forvaltningsloven<sup>6</sup>

Forvaltningsloven er ikke umiddelbart dækkende for KOB projektet og dets afledte støttesystemer, da forvaltningslovens krav til aktindsigt ikke rammer støttesystemet. Dog kan KOB projektet indirekte blive del af en aktindsigt, såfremt der skal redegøres for adgange til omtalte akter. Den indirekte vej vil fx være via et rollebaseret fagsystem, der har brug for Organisation til at præcisere, hvem der havde adgangsmuligheder hvornår.

---

<sup>5</sup> Bekendtgørelse om sikkerhedsforanstaltninger til beskyttelse af personoplysninger, som behandles for den offentlige forvaltning, [BEK nr. 528, 16/06/2000](#) – også kaldet sikkerhedsbestemmelserne – ”... gælder for behandling af personoplysninger, som foretages for den offentlige forvaltning helt eller delvis ved hjælp af elektronisk databehandling.” (§ 1).

<sup>6</sup> [Forvaltningsloven](#) er den regulerende lov omkring den offentlige forvaltning agerende ved sagsbehandlingen.



## Løsningsarkitektur

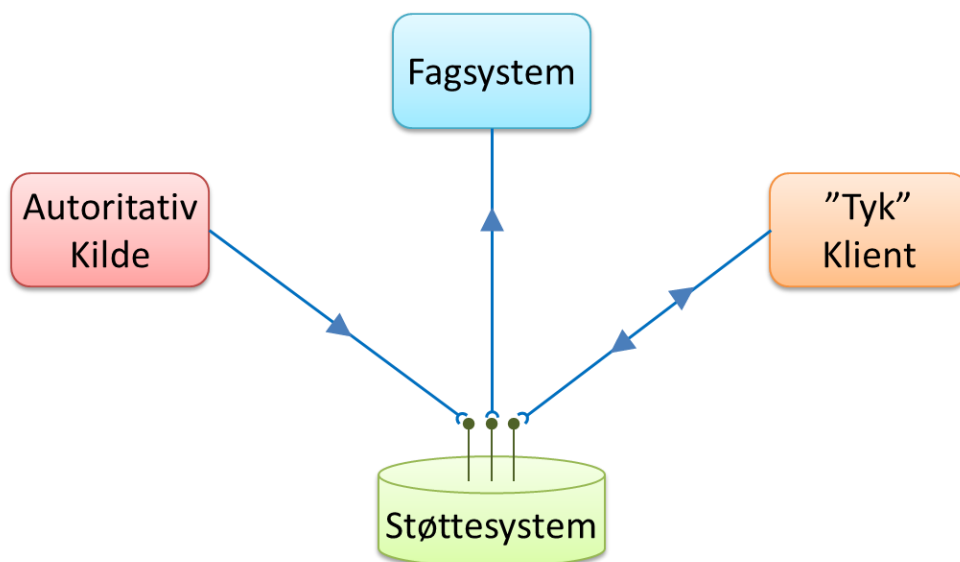
Forretningsbehovene har ført frem til følgende erkendelser:

- Der skal "altid" være adgang til centrale data, dvs. "høj" opetid og korte svartider
- Kernerdata i forretningstjenesterne skal findes i ét centralt og autoritativt repository (pr. forretningstjeneste)
- Fagsystemer skal kunne fastholde egen kopier af ønskede data uagtet at de er opdateret i deres respektive, centrale repository
- Fagsystemet vil ikke have ansvaret for integrationen, men skal have kontrol med data

Samlet set er dette grundlaget for den proces, der fører frem til Den anbefalede arkitektur.

## Principper

Grundprincippet for en arkitektur, der kan bære et af de tre støttesystemer i et komplekst systemmiljø, er illustreret herunder (Figur 1):



Figur 1 Det grundlæggende arkitekturprincip

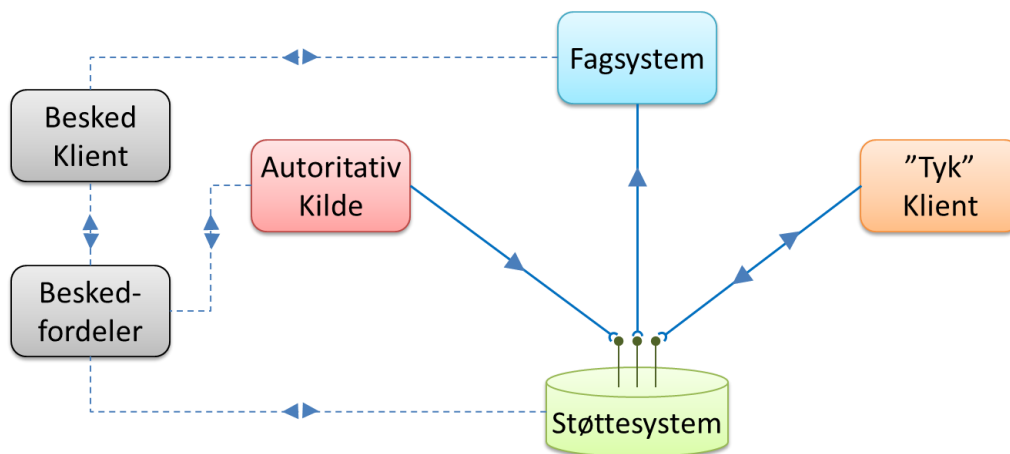
Illustrationen forsøger at beskrive disse grundlæggende karakteristika:

- Støttesystemet er passivt, dvs. data læses og skrives efter initiativ fra de respektive fagsystemer eller fra de autoritative kilder
- Støttesystemet er meget simpelt, dvs. primært opfattet som et repository uden "unødvendig" funktionalitet
- Der findes udvidelsesmuligheder til støttesystemet via "tykke" klienter, der indeholder al relevant funktionalitet hørende til støttesystemet. Den principielle sondring mellem repository og "tyk" klient skal findes i behovet for maksimal stabilitet og leverancekvalitet på det centrale repository.
- Den tykke klient kan både læse og skrive, og dermed anvendes som udvidelse for både fagsystem og autoritativ kilde
- Et system kan optræde i begge roller, dvs. som både fagsystem og autoritativ kilde, men naturen af hver af de to roller gør, at de optræde som forskellige systemer

## Robust princip

Det simple princip har dog visse mangler, da alt initiativ ligger uden for støttesystemet, der ellers skal optræde som autoritativ kilde for de fagsystemer, der skal bruge støttesystemet. Og der er ikke meget autoritativt over at være passiv.

Derfor er det vigtigt for de enkelte systemer at kunne kommunikere om vigtige ændringer og behov for handling. Dvs. støttesystemet skal kunne udvise initiativ, der kan få enten fagsystem eller autoritativ kilde til at handle. Dette opnås ved at anvendes beskeder (hændelser) til kommunikation mellem de enkelte systemer således som det er skitseret herunder (Figur 2).

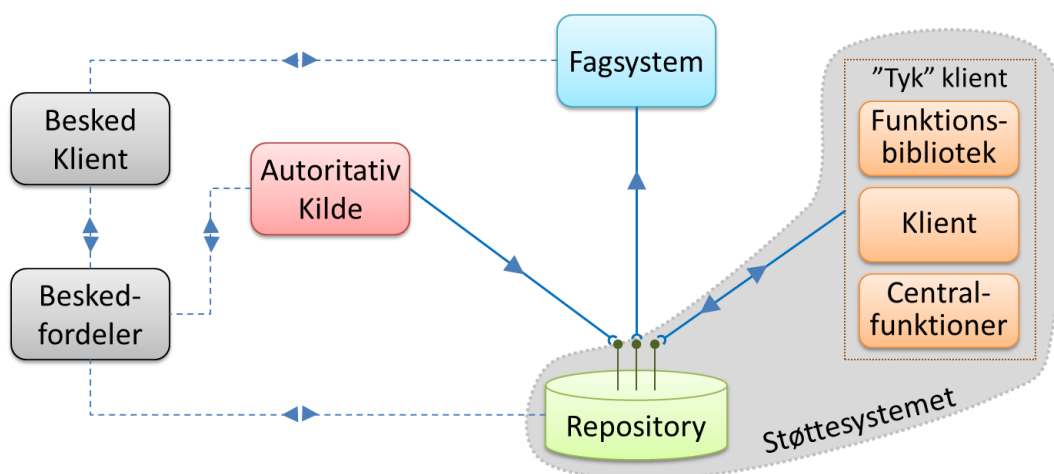


Figur 2 Det mere robuste arkitekturprincip

Figur 2 beskriver et mere robust setup, der giver alle systemet adgang til både at udvise initiativ og passivitet mens, at støttesystemet stadig kan "tillade" sig at være passiv og udstille services/snitflader til de systemer, der skal skrive og/eller læse data i støttesystemet.

## Den tykke klient

Den tykke klient er mere end blot én klient. Den tykke klient er den funktionelle udvidelse af støttesystemets repository, dvs. en mere korrekt illustration kunne være:



Figur 3 Klientprincippet

Støttesystemet er således blevet opdelt i repository og ”tyk” klient, hvor sidstnævnte er delt i tre dele. Den følgende tabel vil detaljere de 3 klientdele nærmere:

Klientdele	Beskrivelse
<b>Klient</b>	<p>Klienten er en ægte klient, der leveres som en konkret applikation til installation på beskrevet software og med få konfigurationsmuligheder. Klienten er en metode til integration via en lokal (i forhold til fagsystemet) klient med relevant funktionalitet, der giver tryk og sikker adgang til støttesystemets repository.</p> <p>Klienten rummer de samme kald, som findes i funktionsbiblioteket, og klienten er substitut for samme, hvor der ikke ønskes tæt integration. Klienten er leveret som del af støttesystemet, men det ”lever” hos fagsystemet og hører driftsmæssigt sammen med fagsystemet.</p>
<b>Funktionsbibliotek</b>	<p>Funktionsbiblioteket holder den samme funktionalitet som klienten<sup>7</sup>, men i stedet for at være en egentlig applikation er funktionsbiblioteket alene et API til at bruges og direkte integrere i fagsystemerne.</p> <p>Funktionsbiblioteket rummer de fleste muligheder og bør anvendes, hvor fagsystemet ønsker maksimal kontrol og fleksibilitet (og hvor de ikke vil nøjes med de generelt udstillede services på støttesystemets repository). Funktionsbiblioteket er leveret som det af støttesystemet, men det anvendes under fagsystemets ansvar og driftsmodeller.</p>
<b>Centralfunktioner</b>	<p>De centrale funktioner adskiller sig fra klienten og funktionsbiblioteket ved ikke at være tilgængelig for et fagsystem, men alene er knyttet til repositoryet. De centrale funktioner er kernefunktionerne for støttesystemets repository, og de hører tæt sammen med disse. De er dog rent arkitektonisk placeret udenfor repositoryet for at illustrere, at de udstillede services er direkte på repositoryet og ikke via de centrale funktioner. De centrale funktioner er leveret som del af fagsystemet og de anvendes under ansvar og samme driftsmodel som støttesystemet.</p>

Tabel 9 De tre dele af støttesystemsklienten

Som nævnt er hensigten med denne opdeling mellem repository og de tre støttesystemsklienter, at etablere en robust struktur med maksimal fleksibilitet og masser af muligheder.

Som Tabel 9 viser, er et central aspekt ved klientprincippet (Figur 3) er ansvarsfordelingen. Dette er uddybet i følgende tabel (Tabel 10).

	Støttesystem	Fagsystem	
<b>Klientdel</b>	<b>Klient</b>	Udvikling og konfigurationsbeskrivelse	Drift ud fra konfigurationsbeskrivelse
	<b>Funktionsbibliotek</b>	Udvikling og API-vejledning	Implementering og drift
	<b>Centralfunktioner</b>	Udvikling og drift	N/A

Tabel 10 Ansvarsfordeling mellem Støtte- og fagsystem

Ansvarsfordelingen for driften er essentiel da den hænger tæt sammen med ansvar for adgangskontrol og logning af dataadgang iht. aktuelle og gældende regler (se evt. kapitel: *Love og regler*).

<sup>7</sup> Helt konkret er der forskel mellem klient og funktionsbibliotek, men funktionalitetsmæssigt er de helt ens dog således, at klienten kan rumme midlertidig lager (caching) og anden applikationsnære funktionalitet, der ikke findes i funktionsbiblioteket.

Ved at have én sammenhængende funktionalitet fordelt på de tre klientdele koblet på centrale repository er det muligt at have den robuste arkitektur sammen med den nødvendige funktionalitet og fleksibilitet, og det fører frem til den anbefalede arkitektur.

## Den anbefalede arkitektur

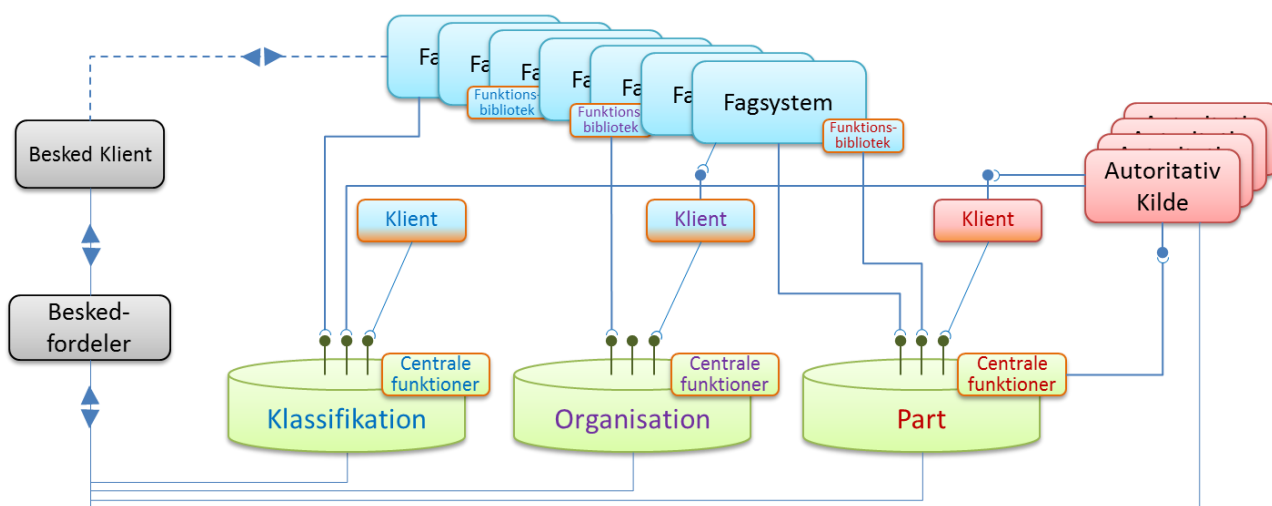
De centrale behov er kort beskrevet:

- Adgang til fælles og autoritative data
- Stor opetid og korte svartider
- Kvalitet og sikkerhed
- Flexibilitet og individualitet

Og disse behov er reflekteret i den anbefalede arkitektur, der er modelleret oven på den de principper, der er gennemgået i forrige afsnit (Principper), og som forsøger at etablere en arkitektur, der er fleksibel og robust i forhold til driftsmiljøer og leverandøruafhængig.

Det er klart at der ikke kan forventes transparens i forhold til ændret leverandør eller driftsmiljø, så arkitekturen handler ikke over transitionen, men mangfoldigheden.

### ”Farveladen”



Figur 4 Anbefalet arkitektur (farveladen)

Ovenstående figur (Figur 4) viser den anbefalede arkitektur, og for at kunne vise den med alle ”parametre” er der brugt en hel del farver, deraf er tilnavnet farveladen tildelt. For at prøve at gøre figuren lidt mere forståelig, er her en læsevejledning:

- Støttesystemerne er beskrevet med farvet tekst
  - Klassifikation er blå
  - Organisation er lilla
  - Part er rød
- Klienterne er markeret med orange rammer for at indikere klienter
- Klienterne er markeret med fyldfarve alt efter hvem, der har driftsansvaret og – hvor persisteret data er omfattet af persondataloven – hvor der har dataansvaret (se afsnit Persondataloven)
  - Støttesystemet er ansvarlig, når farven er lysegrøn

- Fagsystemet er ansvarlig, når farven er **lyseblå**
- En autoritativ kilder er ansvarlig, når farven er **lyserød**
- Hvor der er en **orange** farve nederst (på klienterne) er ansvaret stadig relativt til resten af farven, men støttesystemet har del af ansvaret for korrekt konfiguration
- Beskedklienterne er bevist lavet **grå**, da de ikke er omfattet af nærværende dokumentation, men de er samtidigt medtaget, da de indgår i integrationsmønstrene med støttesystemerne.

## Arkitekturen

Arkitekturen som den kan læses af farveladen (Figur 4)<sup>8</sup> kan sammenfattes således:

Støttesystemet består af et repository og tre funktionalitetsklienter. Leverandøren af støttesystemet er ansvarlig for udviklingen af de 4 dele af støttesystemet i overensstemmelse med nedenstående beskrivelser (Tabel 11):

Repository	
Repository	<p>Centralt passivt register over støttesystemets kerne data. Det enkelte repository er ansvarlig for at overholde de generelle forretningsregler og sikre konsistente data for at kunne garantere datakvalitet, autoritativ rolle og robust data adgang.</p> <p>Det er også i repositoryet at ansvaret for opetid og svartider er placeret.</p> <p>Som udgangspunkt er det enkelte repository tilstrækkeligt i sig selv til at levere funktionalitet til at kaldende systemer kan skrive og læse data. Der er ikke brug for den funktionalitet, der er findes i klienterne. De er alene at betragte som udvidelser.</p>
Klienter	
Centrale funktioner	<p>De centrale funktioner er med til at udvide grundfunktionaliteten i det enkle repository. De centrale funktioner er rettet mod vedligeholdelsen af data i repositoryet og generel funktionalitet, der kan udføres uden at et fagsystem der involveret. Dette kan være hent af data fra en autoritativ kilde, levering til fællesoffentlige registre (borger.dk, Danmarks Statistik osv.)</p> <p>De centrale funktioner anvender data fra ét repository, og der er således ikke tale om fagsystemslignende funktionalitet, men alene en adskillelse af den grundlæggende støttesystems funktionalitet i et repository og så de centrale funktionaliteter.</p> <p>De centrale funktioner driftes <b>altid</b> sammen med det korresponderende repository (dvs. under samme SLA<sup>9</sup>).</p>
Funktionsbibliotek	<p>Funktionsbiblioteket er som navnet siger et funktionsbibliotek, der kan udvide mulighederne for det enkelte støttesystem og gøre det lettere for de fagsystemerne at anvende støttesystemet og dets muligheder. Funktionsbiblioteket leveres til leverandøren af fagsystemet, og det er herefter leverandørens valg om funktionsbiblioteket skal implementeres sammen med fagsystemet.</p> <p>Funktionsbiblioteket driftes som ethvert funktionsbibliotek sammen med det system, hvor det er implementeret – her fagsystemet – og det er fagsystemets leverandør, der er ansvarlig for kvalitet og korrekt anvendelse af funktionalitet.</p>

<sup>8</sup> Og nok også rummende en del større baggrundsviden end blot det Figur 4 viser.

<sup>9</sup> SLA = Service Level Agreement = Leverings- og servicebetingelser på dansk.

Klient<sup>10</sup>

Klienten er en pakke bestående af den samme funktionalitet som i funktionsbiblioteket (udstillet som services) og en række ekstra funktioner, der fx kan være lokale replika eller lokale services, der kan kombinerer mere end ét kald til støttesystemets repository.

Klienten er ikke en lokal kopi af det centrale repository, da den ikke (nødvendigvis):

- indeholder alle data
- indeholder opdaterede data (= er ikke garanteret autoritativ)
- tillader lokal opdatering (er ikke autoritativ)
- har de centrale funktioner med

Klienten er tænkt som en enkel måde for fagsystemer at funktionsberige deres brug af støttesystemet, hvis fagsystemet ikke kan eller vil bruge funktionsbiblioteket. Klienten skal stadig driftes sammen med fagsystemet, da dataansvaret (primært ved lokal replika) ligger hos fagsystemet. Leverandøren af støttesystemet er ansvarlig for at levere dækkende konfigurationsbeskrivelser.

Tabel 11 Støttesystemets 4 dele

Støttesystemets funktionalitet i den anbefalede arkitektur og som beskrevet herover sigter mod at give fagsystemet så ideelle muligheder for at løse sine egne opgaver og "blot" trække på støttesystemet, når det giver mening.

For at fagsystemet skal kunne trække på støttesystemet er der behov for en række integrationsmønstre. I den anbefalede arkitektur (farveladen - Figur 4) er der illustreret en række integrationsmønstre, der vil blive beskrevet nærmere i følgende tabel.

### Direkte

#### Direkte kald til udstillet service

Den mest simple integration er et direkte kald fra fagsystem til støttesystem. Støttesystemet udstiller en række services<sup>11</sup> som fagsystemet kan kalde. Hvert kald er synkront og der gives svar tilbage ved kaldets afslutning.

Støttesystemet leverer en beskrivelse af alle services med angivelse af hvorledes de kaldes, og hvilke svar de kan returnere, samt mulige fejlmeldinger.

Fagsystemet er ansvarlig for at kalde services korrekt og for at kunne håndtere de mulige svar, om de er reelle svar eller fejlmeldinger. Fagsystemet er også ansvarlig for at hele transaktionen gennemføres og styres korrekt således, at data committet af støttesystemet også markeres som så i fagsystemet, samt at data, der ikke blev committet i støttesystemet, bliver forsøgt gendelt senere (afhængig af fejlmeldingen).

Der skal ske en autorisation via certifikater for alle kald af de udstillede services.

Når fagsystemet anvender funktionsbiblioteket betragtes integrationsmønstret stadig som direkte kald til udstillede services, også selvom der kan være anden form for adgangvalidering end ved det almindelige direkte kald.

<sup>10</sup> Ud fra beskrivelsen af klienten er der visse ligheder med den lokale instans af service platformen, og det kan give god mening at disse ligner hinanden og at de koordineres således at der ikke skal driftes unødigt mange applikationer, platforme eller klienter. Dette er dog uden for scope af dette dokument.

<sup>11</sup> Sandsynligvis webservices efter principper som REST, OWSA Model T etc.

Proxy	
Via klient	<p>Hvis et fagsystem ikke ønsker at skulle kalde den centrale service for støttesystemet kan fagsystemet kalde en lokal placeret klient (for støttesystemet). Den lokale klient kan tilbyde de samme services som ved det direkte kald, men da kaldet sker lokalt til fagsystemet, er der simple principper for autorisation<sup>12</sup>.</p> <p>Klienten varetager den videre integration til støttesystemets repository og er således bærende for sikkerhed og autorisation.</p> <p>Klienten tager servicekaldet fra fagsystemet og behandler det iht. de indbyggede regler, der fx kan være at alle kald viderestilles til støttesystemet, eller at der holdes et lokalt replika, som giver mulighed for opslag direkte i klienten selv, hvis støttesystemet ikke svarer inden for konfigureret tid. Klienten kan kun persistere data fra støttesystemet som allerede er committet. Dvs. al skrivning til støttesystemet via klienten viderestilles igennem til støttesystemet og kun der kan data committes.</p> <p>Som helhed er kaldet synkront, og svaret giver ved kaldets afslutning. Kan kaldet ikke afsluttes før det timer ud, giver klienten en fejlmelding herom.</p> <p>Fagsystemet er ansvarlig for at kalde klienten korrekt og for at kunne håndtere de mulige svar, om de er reelle svar eller fejlmeldinger. Fagsystemet er også ansvarlig for at hele transaktionen gennemføres og styres korrekt således, at data committet af klienten (på vegne af støttesystemet) også markeres som så i fagsystemet, samt at data, der ikke blev committet af klienten, bliver forsøgt gendsendt senere (afhængig af fejlmeldingen).</p>
Via serviceplatformen	<p>Dette integrationsmønster er alene medtaget for at give overblik, da den i realiteten er dækket af klienten mht. funktionalitet og principper. Dog adskiller integrationen via serviceplatform sig fra klienten ved at klienten er leveret af støttesystemet og dermed er 100 % kompatibel med denne og under kontrol af samme, mens serviceplatformen er leveret uafhængig af støttesystemet med andre regler for bl.a. autorisation, fejlmeldinger og evt. persistering.</p> <p>Set fra fagsystemet er der dog tale om en lokal instantiering af støttesystemet med et givent sæt af service kald, -svar og fejlmeldinger, præcist som for klienten, men dog forskelligt fra denne. Serviceplatformen kan kun persistere data fra støttesystemet som allerede er committet. Dvs. al skrivning til støttesystemet via serviceplatformen viderestilles igennem til støttesystemet og kun der kan data committes.</p> <p>Kaldet til serviceplatformen er som udgangspunkt synkront hele vejen, men da serviceplatformen leveres via andet projekt, der dette ikke nødvendigvis den eneste mulighed. Der henvises til serviceplatformen for yderligere oplysninger.</p> <p>Fagsystemet er ansvarlig for at kalde serviceplatformen korrekt og for at kunne håndtere de mulige svar, om de er reelle svar eller fejlmeldinger. Fagsystemet er også ansvarlig for at hele transaktionen gennemføres og styres korrekt således, at data committet af serviceplatformen (på vegne af støttesystemet) også markeres som så i fagsystemet, samt at data, der ikke blev committet af serviceplatformen, bliver forsøgt gendsendt senere (afhængig af fejlmeldingen).</p>

<sup>12</sup> Termen lokal er her valgt i betydningen, at klienten driftes i samme driftsmiljø som fagsystemet og at det er samme driftsoperatør, der er ansvarlig for begges drift. Begge skal være placeret i samme – beskyttede – netværk, hvorved klienten "automatisk" ved at fagsystemet er det, som det giver sig ud for at være. Dvs. integrationen mellem fagsystem og klient kan ske uden brug af certifikat eller tilsvarende (men det er stadig en mulighed).

Indirekte	
Beskedfordeling	<p>Hvor fagsystemet ikke har brug for svar med det samme, eller slet ikke har brug for svar, men blot vil give besked til andre omkring noget er sket eller skal ske, kan fagsystemet vælge at anvendes en løsere integration, eller indirekte integration, ved at bruge beskeder og beskedfordeling.</p> <p>I beskedfordelings integrationsmønsteret afsender fagsystemet en besked ved at etablere en synkron forbindelse til beskedklienten<sup>13</sup>, der svarer retur med ok, hvis den "overtager" ansvar for beskeden, og med en konkret fejlmelding, hvis kaldet ikke er vellykket.</p> <p>Når beskedklienten har "overtaget" ansvaret for beskeden, er fagsystemets kald afsluttet, og fagsystemet kan markere informationen for committet, også selvom den rent fysisk endnu ikke er registreret i støttesystemet. Beskedfordeleren bærer herefter ansvaret for, at støttesystemet i sidste ende modtager beskeden, mens støttesystemet selv er ansvarlig for at reagere op beskeden og der udfører det endelige commit.</p> <p>Svagheden i dette integrationsmønster er, at data ikke altid er fuldt opdateret i støttesystemet, og at en opdatering kan sendes fra et fagsystem uden at det efterfølgende kan committes i støttesystemet. Sidstnævnte svaghed kan delvist håndteres ved at støttesystemet leverer et funktionsbibliotek, som også kan levere (validerede) beskeder til beskedklienten.</p> <p>Styrken i dette integrationsmønster er dog flerfoldigt (under forudsætning af, at fagsystemet alligevel skulle understøtte beskedfordeling):</p> <ul style="list-style-type: none"> <li>• Fagsystemet kan committe selvom støttesystem er nede</li> <li>• Information kan deles af flere systemer ved kun én integration (specielt hvor der findes mere end én instans af samme støttesystem)</li> <li>• Fagsystemet skal ikke selv forespørge på nye/ændrede data, men kan abonnere på beskeder herom (dvs. støttesystem kan gå fra at være passiv til at være aktiv)</li> <li>• Løskoblingen sikrer fleksibilitet i forhold til forskellige driftsmiljøer og SLA'er</li> </ul> <p>Brugen af beskedfordeling stiller krav til fagsystemet, da fagsystemet skal være sit ansvar bevist. Fagsystemet er det eneste sted at data med 100 % sikkerhed findes og er vedligeholdt.</p>

Tabel 12 Integrationsmønstre

Der er således 4 forskellige integrationsmønstre beskrevet i Tabel 12. De er forskellige og også gensidigt udelukkende. Dvs. et fagsystem kan ikke både bruge direkte og indirekte integrationsmønstre på de samme data, da det vil medføre inkonsistente data<sup>14</sup>. Som farveladen (Figur 4) viser, kan et fagsystem sagtens gøre brug af alle integrationsmønstre samtidigt, så længe det er på forskellige data. Det er således op til det enkelte fagsystem at vælge integrationsmønster på hvilke data.

<sup>13</sup> Dette dokument omhandler ikke beskedfordeling (men det er dog en del af KOB projektet), der i stedet er beskrevet i et tilsvarende dokument alene for beskedfordeleren, hvortil der henvises for yderligere oplysninger om beskedklient og beskedfordeler.

<sup>14</sup> Fagsystemet kan dog reelt gøre som de vil, men de ejer selv ansvaret for data kvalitet og konsistens, og dermed hvor autoritativ støttesystemet kan være



## Driftsmodel

Den anbefalede arkitektur (Figur 4) rummer et kraftigt hint om den rette driftsmodel, men det er ikke muligt at aflæse hele svaret for den rette driftsmodel, da det ikke alene baserer sig på logik og arkitektur, men også er knyttet til politik og lokale (kommuneindividuelle principper).

Som udgangspunkt viser den anbefalede arkitektur (Figur 4) at:

- Der findes ét centralt repository pr. støttesystem
- Der findes en række klienter (services, funktioner etc.), der beriger det centrale repository
- Det centrale repository behøver ikke ligge samme sted som fagsystemet eller vice versa
- Flere fagsystemer kan bruge samme repository

Men heraf kan der ikke udledes, om ”centralt” betyder, at der kun findes ét støttesystem i Danmark, at der kun findes ét støttesystem pr. kommune eller om der findes flere støttesystemer, og at den enkelte af disse kan bruges af flere fagsystemer!

Ud fra forretningsbehovene alene er det ikke muligt at anbefale én bestemt driftsmodel, da forskellige principper har forskellige fordele og ulemper. Og det vurderes, at KOB projektet og arkitekturen omkring integrationsmønstre ikke er moden endnu til at beslutte en endelig driftsmodel.

## Forventede dataindhold i de centrale repositories

Indholdsmæssige forventes de centrale repositories at kunne repræsentere det forretningsobjekt, deres respektive forretningstjeneste er bragt til verden for at fastholde og udstille. For at kunne løfte denne opgave med at repræsentere forretningsobjektet, skal det sikres at det enkelte forretningsobjekt er korrekt og fyldestgørende beskrevet til at kunne rumme de nødvendige og relevante fællesdata til brug via forretningstjenesternes integrationer.

## Snitfladebeskrivelserne

Snitfladebeskrivelser bygger på de OIO<sup>15</sup> godkendte snitfladebeskrivelser for de enkelte forretningsobjekter (gælder ikke for part). Snitfladebeskrivelser er retningsgivende for de servicebaserede integrationer, mens bulk integration optimeres for kvantitet og effektivitet og ikke nødvendigvis i forhold til OIO snitfladebeskrivelsens ord. Dog er det afgørende at de enkelte forretningstjenester kan behandle data ens uanset valg af integration.

## OIO Klassifikation

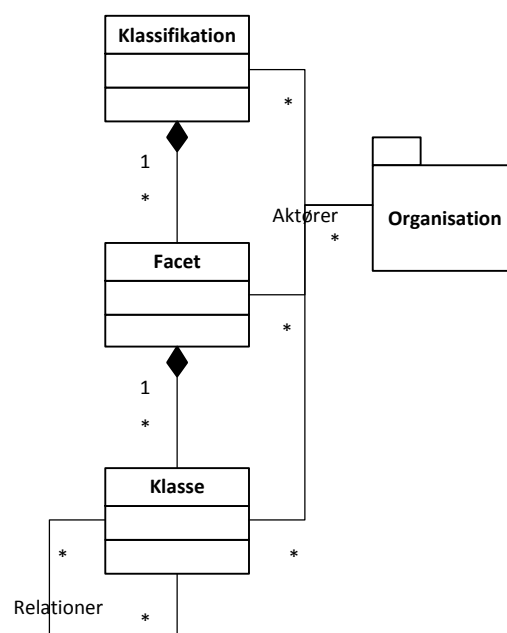
OIO klassifikation består af mange forskellige snitfladebeskrivelser for de forskellige handlinger, der kan udføres på klassifikationsobjektet via servicekald. Disse gennemgås ikke her i dette dokument, men ligger til grund for den simple informationsmodel i Figur 8.

Klassifikationen er opbygget af tre grundelementer som vist i Figur 8 her til højre.:

<b>Klassifikation</b>	Det øverste niveau, der beskriver og definerer de forskellige klassifikationer, som fx KLE, FORM, roller og stillingsbetegnelser.
<b>Facet</b>	En klassifikation defineres ved én eller flere facetter, der er indbyrdes uafhængige således, at enhver kombination af facetternes klasser er legal. Fx består KLE af 2 facetter; Emnenumre og handlingsfacetter
<b>Klasse</b>	Klassen er den egentlige systematiske og semantiske inddeling af den enkelte facet, og hver klasse repræsenterer en unik og beskrivende værdi, og det kan fx for KLE's emnenumre være 23.05.01 eller for handlingsfacetten være G01. Selve semantikken på klassen defineres pr. facet.  De enkelte klasser kan relateres til hinanden for at danne mønstre og hierarkier.

Tabel 13 Beskrivelse af klassifikations 3 grundelementer

Klassifikationen har integrationer til organisation for at kunne styre ejerskabet og rollerne; ansvarlig og redaktør. Ud over integrationerne til organisation, er der ingen integration fra klassifikation, og klassifikation er derfor at opfatte som den mest centrale af de 5 centrale forretningstjenester i rammearkitekturen<sup>16</sup>.



Figur 5 Grundstruktur Klassifikation

<sup>15</sup> Alle OIO standarder er hentet på [digitaliser.dk](http://digitaliser.dk).

<sup>16</sup> De 5 centrale forretningstjenester i rammearkitekturen er Klassifikation, organisation, part, dokument og sag

## OIO Organisation

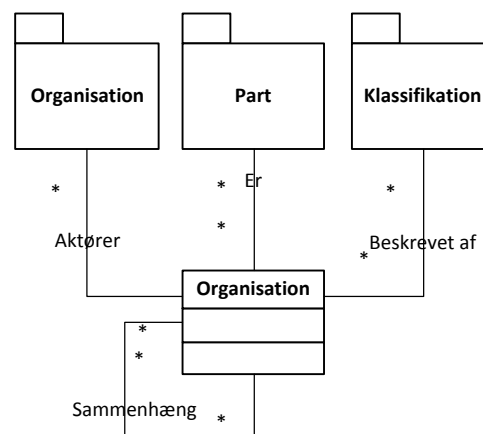
Hvor klassifikation er meget ordnet i sin opbygning, er organisation meget mere rodet, da det, organisation skal rumme, er alt andet end ensartet opbygget. Organisation, brugere, it-systemer, interessefællesskaber osv. er opbygget efter behov og lyst og dermed ikke efter regler og systematikker.

Organisation er reelt kun opbygget af ét eneste grundelement som vist i Figur 6:

Organisation	Organisation er det rummelige grundelement, der holder alle de organisatoriske data, mellem hvilke der kan etableres sammenhænge på kryds og tværs.  Det samlede sæt af data og sammenhænge beskriver de enkelte organisationer i deres helhed.
--------------	---

Tabel 14 Beskrivelse af organisations ene grundelement

Organisation har integrationer til klassifikation, organisation og part. Klassifikation anvendes til at beskrive opgaver og kontekster, part anvendes til at beskrive brugere (hvem de er) og organisation bruger sig selv til opslag af aktører.



Figur 6 Grundstruktur Organisation

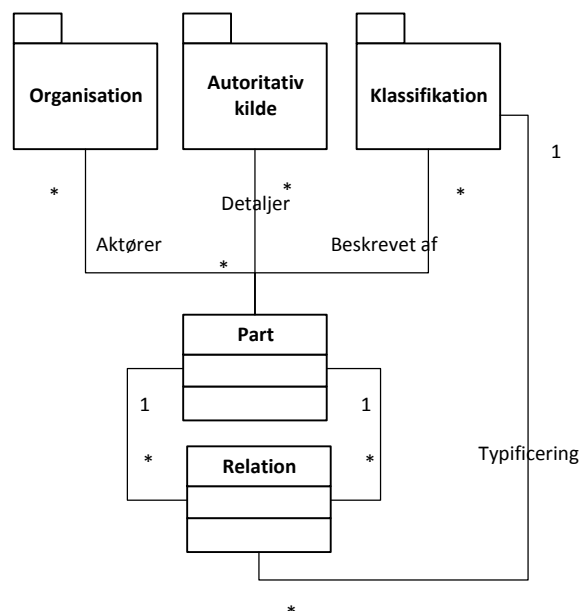
Desuden kan der være enkle andre integrationer til fx et adresse-register, disse integrationer er pt. ikke specificeret. Desuden er det muligt at part også vil rumme adresser.

## Part

Klassifikation og organisation er yderst velbeskrevet, hvilket man desværre ikke kan sige om part. Der findes pt. ingen konkrete oplysninger på en parts-snitflader på digitaliser.dk og dermed findes heller ikke en standard, der kan læses op ad.

Der har været tiltag til en OIO standard for part-snitfladen for person, og den har været i offentlig høring, men resultatet blev, at udkastet til standarden blev trukket tilbage igen<sup>17</sup>, da en standard for person bør defineres af en større kreds som del af det kommende arbejde i forbindelse med den fælles offentlige digitaliseringsstrategi 2011 – 2015<sup>18</sup>.

Inspireret af parts definitioner fra Referencearkitekturen for ESDH, Rammearkitekturen for ydelsessystemer ver. 0.6 og det fremlagte udkast til standard for "person", kan der udstilles den viste grundstruktur for Part (Figur 7 her til højre), der viser, at part kan beskrives ved 2 grundelementer:



Figur 7 Grundstruktur Part

<sup>17</sup> Både høring og resultat kan ses under [høringsportalen på borger.dk](http://borger.dk/horingsportalen).

<sup>18</sup> Digitaliseringsstrategien 2011 – 2015 kan findes her: <http://www.itst.dk/politik-og-strategi/arkiv/digitaliseringsstrategien> eller direkte her som [pdf-fil](#).

Part	Part er det rummelige grundelement, der holder alle relevante data om de enkelte parter.  Med relevans hentydes her til, at part ikke vil indeholde alle data om de enkelte parter, men alene de data, der er nødvendige for at den enkelte part kan identificeres og beskrives, Det vil fx være CPR-nummer, navn og adresse på en given person. Øvrige informationer hentes via opslag i relevante autoritative kilder.
Relation	Parter har som natur, at de kan være relateret til hinanden som forældre til børn, ejer af ejendom og ledelse af virksomhed. Dette rummes direkte i part, da disse relationer betragtes som væsentlige og derfor altid bør være til stede. Relationerne kan enten trækkes fra de respektive autoritative kilder, eller de kan vedligeholdes i part selv.

**Tabel 15** Beskrivelse af parts 2 grundelementer

Filosofien bag den simple part skal ses i forhold til, at parter kan være meget forskelligt og samtidigt kan ses meget rig på informationer, hvorved et komplet partsregister vil være abnormt og ekstremt kompliceret uden reel værdi.

Den konkrete part beskrives med færrest mulige oplysninger (fleste fælles oplysninger) således at part entydigt kan identificeres og beskrives, mens resten af detaljerne hentes i respektive autoritative kilder. Part holder sin egen historik, mens de respektive autoritative kilder forventes at holde de "ægte" bitemporale egenskaber for den enkelte part.

Part integrerer med organisation til styring af aktører (ansvarlig og redaktør), klassifikation til typificering af både part og relationer og autoritative kilder for træk af nødvendige data og relationer.

Detaljer om de enkelte parter forventes hentet af fagsystemerne selv.

## Informationsmodel

Informationsmodellen for de tre støttesystemer er ikke fuldkommen og må "kun" ses som første skridt i den endelige proces med at afklare konkret indhold og konkrete snitflader.

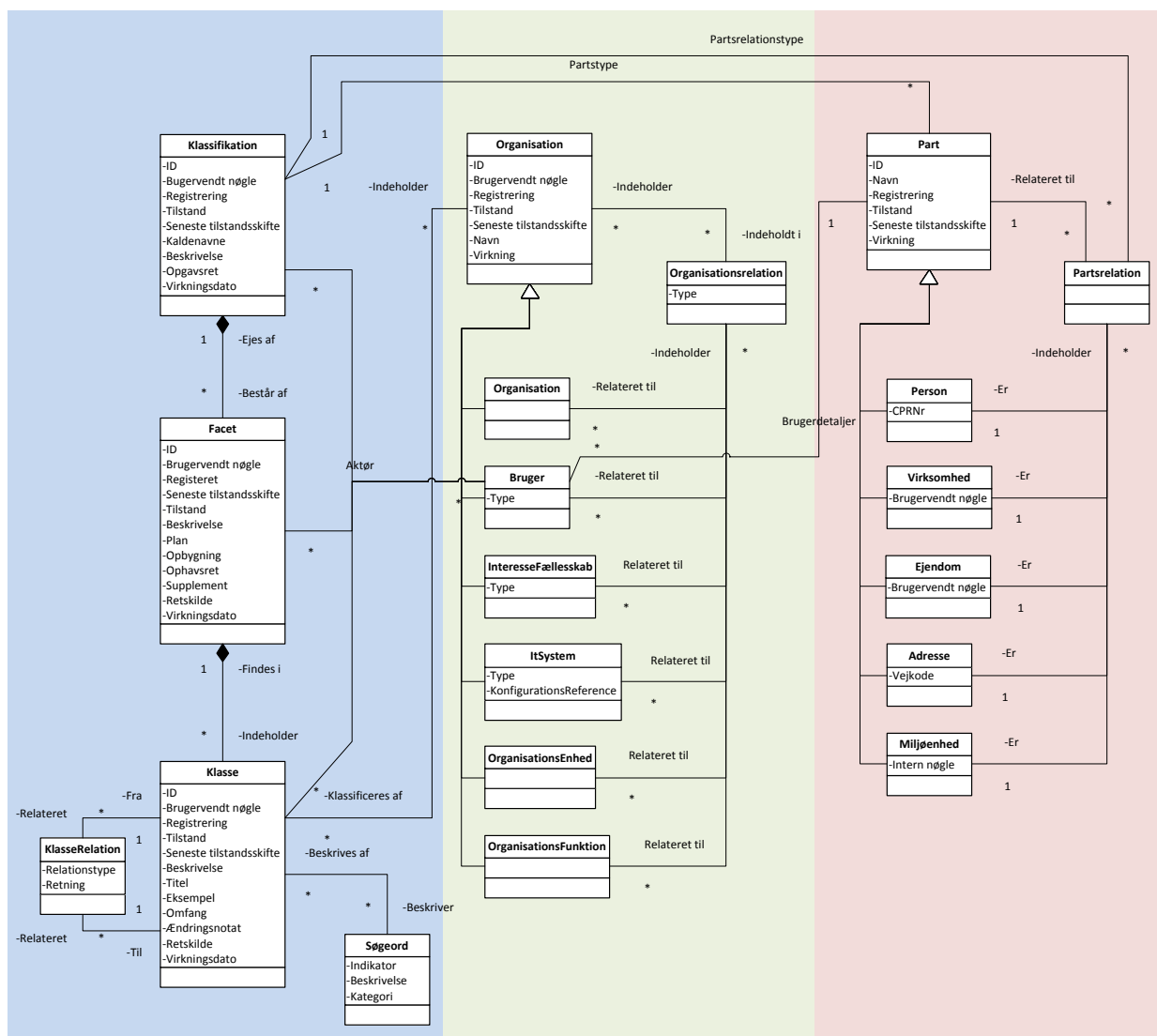
For Klassifikation og Organisation er der tale om fuld compliance med OIO snitflade standarderne, mens Part er tillempet med fokus på relevans for fagsystemerne (altså støttesystem tankegangen).

Selvom informationsmodellen er tegnet samlet for de tre støttesystemer, er der tale om tre forskellige støttesystemer og repositories, der anvender hinanden, men derudover er helt uafhængig. Dette er illustreret på Figur 8 med farver for at undgå tro om leverance af ét fuldt system.

Der er dog ikke indtegnet autoritative kilder på informationsmodellen, da de ikke beriger modellen signifikant, men til alle tre og specielt til Part er der eksterne autoritative kilder, der skal respekteres og anvendes, hvor det giver mening.

Helt overordnet viser informationsmodellen følgende karakteristika for de tre støttesystemer:

- Klassifikation er som udgangspunkt identisk med OIO standarden og dermed også med den i Figur 5 vist grundstruktur
- Organisation er som udgangspunkt identisk med OIO standarden, mens den adskiller sig lidt fra den i Figur 6 viste grundstruktur, da den forventes realiseret med en række specialiseringer, der relateres gennem Organisationsrelation
- Part realiseres i store træk som et fælles register for en række specialiserede data, og med tværgående relationer registreret i Partsrelation



Figur 8 Simple informationsmodel Klassifikation (blåt), Organisation (grønt) og Part (rødt)

Informationsmodellen er meget enkel og kan anvendes til at belyse de generelle karakteristika for de tre støttesystemers repositories. Og heraf ser man (dette er ikke godkendte konklusioner endnu<sup>19</sup>):

<sup>19</sup> Med dette menes blot, at projektet KOB ikke har fastlagt dette som den endelige strategi, men konklusionerne er konsistente med oplægget til løsningsarkitektur i nærværende dokument (*Den anbefalede arkitektur*).

- Støttesystemerne har til hensigt at støtte fagsystemerne, ikke at begrænse disse
- Der er ingen regler, dvs. der er frit slag til at fylde vilkårlige data ind i de tre repositories, så der kan oprettes:
  - Præcist de ønskede klassifikationer og semantikker
  - Enhver organisation, medarbejder, interessegruppe som kommunen måttet ønske
  - Alle mulige og umulige parter til at indgå i de kommunale systemer, fælles eller ej
- Friheden i data, stiller krav om leverandørerne af klassifikationer, organisationer og parter, leverer korrekte data og at disse er valideret (i forhold til relevante regler) inden disse lægges i støtte-systemernes repositories
- Der er kun plads til gyldige data, dvs. ingen simulering og ingen midlertidige data, dog understøttes bitemporalitet, så rettelser kan lægges ind i med fremtidig ikrafttrædelse og dermed understøtte at alle data oprettes før de samlet træder i kraft

Af disse krav, og ikke helt så direkte som ved punkterne herover, kan man udlede, at støttesystemerne primært skal være passive, og således blot stille sin funktionalitet og sine data til rådighed, som en autoritativt repository.

Støttesystemerne er som udgangspunkt også passive i forhold til de autoritative kilder således at forstå, at støttesystemerne kun opdateres, når dette sker efter en "henvendelse"<sup>20</sup> fra de autoritative kilder<sup>21</sup>.

---

<sup>20</sup> Henvendelse kan være både en hændelse, men også et konkret datasæt til opdatering.

<sup>21</sup> Støttesystemet har som udgangspunkt ingen "hente"-automatik eller –synkronisering indbygget, da disse forventes leveret udenfor støttesystemet, og helst som del af den autoritative kilde.

## Oversigt over tabeller & figurer

### Tabeller

TABEL 1 ORDLISTE OG BEGREBSDEFINITIONER.....	4
TABEL 2 KLASSEKATIONSTJENESTENS EGNE FORRETNINGSBEHOV.....	6
TABEL 3 FAGSYSTEMERNES FORRETNINGSBEHOV FOR KLASSEKATIONSTJENESTEN .....	8
TABEL 4 ORGANISATIONSTJENESTENS EGNE FORRETNINGSBEHOV .....	9
TABEL 5 FAGSYSTEMERNES FORRETNINGSBEHOV FOR ORGANISATIONSTJENESTEN .....	10
TABEL 6 PARTSTJENESTENS EGNE FORRETNINGSBEHOV.....	11
TABEL 7 FAGSYSTEMERNES FORRETNINGSBEHOV FOR PARTSTJENESTEN .....	13
TABEL 8 BEGREBSDEFINITIONER FRA PERSONDATA LOVENS § 3. ....	14
TABEL 9 DE TRE DELE AF STØTTESYSTEMSKLIENEN .....	18
TABEL 10 ANSVARFORDDELING MELLEM STØTTE- OG FAGSYSTEM.....	18
TABEL 11 STØTTESYSTEMETS 4 DELE .....	21
TABEL 12 INTEGRATIONSMØNSTRE .....	23
TABEL 13 BESKRIVELSE AF KLASSEKATIONS 3 GRUNDELEMENTER .....	25
TABEL 14 BESKRIVELSE AF ORGANISATIONENS 3 GRUNDELEMENTER .....	26
TABEL 15 BESKRIVELSE AF PARTS 2 GRUNDELEMENTER .....	27

### Figurer

FIGUR 1 DET GRUNDLÆGGENDE ARKITEKTURPRINCIP .....	16
FIGUR 2 DET MERE ROBUSTE ARKITEKTURPRINCIP .....	17
FIGUR 3 KLIENTPRINCIPPET .....	17
FIGUR 4 ANBEFALET ARKITEKTUR (FARVELADEN).....	19
FIGUR 5 GRUNDSTRUKTUR KLASSEKATION .....	25
FIGUR 6 GRUNDSTRUKTUR ORGANISATION.....	26
FIGUR 7 GRUNDSTRUKTUR PART .....	26
FIGUR 8 SIMPEL INFORMATIONSMODEL KLASSEKATION (BLÅT), ORGANISATION (GRØNT) OG PART (RØDT).....	28