



FESD Grænseflade til CMS-løsninger

Høring

Dette udkast til forslag til FESD-standard er i offentlig høring i perioden fra 12. juli 2007 til 22. august 2007

IT- og Telestyrelsen København den 12. juli 2007

FESD-standardisering

Grænseflade til CMS-løsninger. Snitflade. Version 0.8

Kolofon:

FESD-standardisering. Grænseflade til CMS-løsninger. Snitflade. Version 0.8

Denne standard kan frit anvendes af alle. Citeres der fra standarden i andre publikationer til offentligheden, skal der angives korrekt kildehenvisning.

Forslag til FESD-standarder udarbejdes af IT- og Telestyrelsen, IT-Arkitektur kontoret, FESD-standardiseringsgruppen i samarbejde med de tre FESD-leverandører Software Innovation A/S, Accenture I/S og CSC Danmark A/S.

Kontaktperson i FESD-standardisering:

Projektleder Palle Aagaard, Mailadresse paa@itst.dk

Telefon 33 37 92 67 (direkte)

Accenture A/S

Arne Jacobsens Allé 15

2300 København S

Telefon: 72 28 80 00

Web-adresse: <http://www.accenture-fesd.dk/>

CSC Danmark A/S

Retortvej 8

1780 København V

Telefon: 36 14 40 00

Web-adresse: <http://www.fesd-alliancen.dk/>

Software Innovation A/S

Nærum Hovedgade 10

DK-2850 Nærum

Telefon: 45 58 88 88

Web-adresse: <http://www.softwareinnovation.dk/>

Ministeriet for Videnskab, Teknologi og Udvikling

IT- og Telestyrelsen

IT-Arkitektur kontoret

National IT and Telecom Agency

Ministry of Science, Technology and Innovation

Holsteinsgade 63

DK-2100 København Ø

Telf. +45 35 45 00 00

Fax. +45 35 45 00 10

<http://www.itst.dk>

itst@itst.dk

Indholdsfortegnelse

1	FORORD	5
1.1	Teknisk forord.....	5
2	DEL A – BUSINESS CASE.....	7
2.1	Indledning og afgrænsning	7
2.2	Formål.....	7
2.3	Reference til andre standarder	7
2.4	Målgruppe	8
2.5	Anvendelse	8
3	DEL B – BESKRIVELSE AF FORRETNINGSARKITEKTUR	9
3.1	Indledning.....	9
3.2	Data	10
3.2.1	Dataobjekter	10
3.2.2	Fortrolighed	10
3.2.3	Rettighed.....	10
3.2.4	Udpeget til publicering	10
3.3	Forretningskrav	10
3.3.1	Aktører i brugsscenarier.....	10
3.3.2	Funktionelle krav	10
3.3.3	Publicering	12
3.3.4	Opdatering af data.....	13
3.3.5	Afpublicering	13
4	DEL C – BESKRIVELSE AF DATAMODEL OG XML-SKEMAER.....	16
4.1	Publiceringsstatus	16
4.1.1	publishingStatus	16
4.1.2	publishingStatusCode	16
4.2	Webservice grænseflader	17
4.2.1	getContentTree.....	17
4.2.2	createContentPosition	18
4.2.3	updateContentPosition	18
4.2.4	removeContentPosition.....	19
4.2.5	createCMDData	19
4.2.6	updateCMDData	19
4.2.7	removeCMDData.....	20
4.2.8	getCMContentList	20
4.2.9	getCMContentList	21
4.2.10	getCMDData.....	21

4.2.11	Samlet oversigt over webservices	23
5.	BILAG A	25
Anvendte typer i FESD-modellerne		25
integer		25
boolean.....		25
Identifikationstyper.....		25
char		26
date, dateTime og time.....		27
string		27
float		27

1 Forord

Den offentlige sektors IT-systemer på statsligt, kommunalt og regionalt niveau skal kunne spille sikkert og effektivt sammen. Derfor arbejdes der målrettet på at få gennemført fælles standarder for elektronisk sags- og dokumenthåndtering - den såkaldte FESD-standard. Målet med standardiseringsarbejdet er at fremme digital forvaltning i den offentlige sektor, og midlet er at sikre, at de forskellige elektroniske sags- og dokumenthåndteringssystemer (ESDH) får en fælles kernefunktionalitet, og at det samtidig sikres, at denne kerne videreudvikles ensartet. En fælles kernefunktionalitet skal sikre:

- at der kan foretages sagsbehandling på tværs af flere organisationer
- at myndigheder, der arbejder med åbne sager, kan lægges sammen
- at der kan flyttes opgaver mellem forskellige myndigheder

I forlængelse af FESD-projektkonkurrencen, som havde sin afslutning primo 2004, og hvor der blev fundet tre FESD-leverandører, blev det i forbindelse med kontraktforhandlingerne besluttet at starte en standardiseringsproces – den såkaldte FESD-standardisering.

For at sikre interoperabiliteten, både til andre systemer, men også så tredjepart kan udvikle moduler til systemet, blev det anset for afgørende, at der udvikles en fælles offentlig datamodel samt andre standarder på ESDH-området.

Koordinering af FESD-standardiseringen er efterfølgende lagt i IT- og Telestyrelsen (ITST). Den konkrete udarbejdelse af forslag/udkast til standarder foregår i et samarbejde mellem de tre FESD-leverandører og en FESD-standardiseringsgruppe i ITST.

Arbejdet med forslag/udkast til standarder tager udgangspunkt i Noark 4's datamodel og databeskrivelser samt leverandørernes løsninger. Standarderne kan afvige fra Noark 4 på de områder, hvor det er nødvendigt for at understøtte dansk forvaltningspraksis, eller hvor parterne i FESD-standardiseringen kan opnå enighed om en afvigelse.

Udkast/forslag sendes herefter i offentlig høring i ca. 1 måned. FESD-standardiseringsgruppen tilretter og færdiggør på baggrund af høringen de endelige "Forslag til standarder".

Standardforslagene forelægges herefter OIO-Datastandardiseringskomiteen til godkendelse.

Efter den samlede godkendelse bliver standarderne således offentliggjort og indgår i IT- og Telestyrelsens "OIO-Katalog", som indeholder en oversigt over godkendte og anbefalede standarder til digital forvaltning i det offentlige.

I standarden kan forekomme brug af særligt ordvalg. Følgende termer anvendes konsekvent i den følgende betydning:

- **"skal"/"obligatorisk"**: betyder, at den nævnte metode/element/mulighed/etc. skal benyttes eller skal forefindes – dvs. må ikke udelades.
- **"må ikke"**: betyder, at den nævnte metode/element/mulighed/etc. ikke må forefindes eller må ikke benyttes.
- **"bør"/"anbefalet"**: betyder, at det i høj grad anbefales, at den nævnte metode/element/mulighed/etc. benyttes eller forefindes. Der skal være tungtvejende grunde til at udelade.
- **"kan"/"optionel"**: betyder, at den nævnte metode/element/mulighed/etc. er en valgmulighed og derfor valgfri at medtage.

1.1 Teknisk forord

Grundlaget for FESD-datamodellen er blevet udarbejdet på en periode på mere end 2 år, hvor der er udarbejdet standarder for de forskellige delområder. Arbejdsmetoder, terminologi og anvendelse af datatyper har ændret sig i denne periode – FESD-standardiseringsgruppen har f.eks. indført en konsekvent brug af UML-notation i

de senere standarder. Konsolideringen af datamodellen har derfor forudsat, at der blev defineret en fælles modelleringsmetode og et sæt af primitive datatyper, der var kompatibelt med alle del-datamodellerne. De primitive datatyper, som modellen er opbygget af, fremgår af bilag A.

2 DEL A – Business Case

2.1 Indledning og afgrænsning

Styregruppen for FESD-projektet har besluttet at der skal gennemføres et standardiseringsarbejde indenfor området ESDH – CMS integration. Dette arbejde skal omfatte udarbejdelse af en business case, forretningsarkitektur og datamodel.

ESDH-systemer bruges til sagsbehandling, dvs. skabe, opsamle, journalisere, distribuere, og arkivere sagsdata indenfor den sagsbehandlende organisation. Med andre ord, håndtering af en sags livscyklus. Da sager ofte vil omfatte følsomme informationer, er det et væsentligt kendetegn ved ESDH-systemer at brugerne tildeles et og samme veldefinerede rettigheds-domæne.

I modsætning til dette fokuserer CMS på at organisere og præsentere tekster og billeder mm.. Dette er typisk fra en organisations interne domæne til det offentlige domæne (dvs. world wide web).

Det er oplagt, at der er væsentlige dele af det data, der opsamles i ESDH, der vil være relevant at formidle via CMS. Således forventes integration mellem ESDH og CMS at være et hyppigt forekommende ønske.

Hvilken funktionalitet der ønskes i en sådan integration er derimod mindre klart. Der er et utal af mulige udformninger; men af retslige, organisatoriske, sikkerhedsmæssige og økonomiske grunde er det relevant at skelne mellem tre forskellige behovsområder:

1. publicering af data, der er af offentlig interesse i henhold til Lov om offentlighed i forvaltningen
2. publicering af data, der ikke automatisk giver aktindsigt eller data der er konfidentielt,
3. overførelse af data fra det offentlige net (www) til ESDH (udefra ind).

Nærværende standard retter sig kun mod punkt 1.

Da der i regi af det fælles offentlige er konkrete overvejelser om at iværksætte en analyse med henblik på at lave et fælles, koordineret systemsetup til at give borgere overblik over deres ESDH-sager (svarende til punkt 2) holdes dette ude af nærværende specifikation.

Det vurderes ligeledes at punkt 3 skal holdes ude for nærværende specifikation, dels fordi der er kraftige afhængigheder til punkt 2, og dels fordi det vurderes at markedets modenhed endnu ikke kan underbygge en business case, der tilskriver et sådant arbejde.

2.2 Formål

Med udarbejdelsen af en standard for området sigtes mod at:

- afdække de sikkerhedsmæssige hensyn, der skal tages for en sådan integration
- eksemplificere mulige systemarkitekturer til en sådan integration
- tilvejebringe standard snitflader fra ESDH, der er særligt rettede mod CMS

Arbejdet skal samlet set gøre det lettere, hurtigere og billigere at lave integrationsprojekter efterfølgende.

2.3 Reference til andre standarder

[1] FESD Datamodel for sager og dokumenter (<http://bib.oio.dk/udgiver/stat/vtu/itogtele/ita/fesd/fesd-datamodel-for-sager-og-dokumenter/>).

[2] FESD Generisk integrationsmodel Fase 1 (<http://bib.oio.dk/udgiver/stat/vtu/itogtele/ita/fesd/fesd-standardisering-generisk-integrationsmodel-grensesnit>)

2.4 Målgruppe

Målgruppen for denne standardisering er software- og konsulentvirksomheder, der gennemfører integration fra CMS til ESDH, samt ESDH-udviklingsvirksomheder, der skal eller ønsker at møde standarderne for FESD-projektet.

2.5 Anvendelse

Standarden fastsætter minimumskrav til snitflader mellem ESDH og CMS. Dette sikrer at ESDH-leverandørene har forberedt et vist omfang af funktionalitet, som implementeringspartnere kan gøre brug af i forbindelse med konkrete integrationsprojekter.

Det er derimod ikke hensigten med standarden at sætte grænser for integrationen mellem ESDH og CMS på andre områder, end dem der direkte omtales i standarden. Fx kan det tænkes at lave en anden og mere omfattende integration til et CMS der driver et intranet, der er placeret i et betroet domæne.

3 Del B – Beskrivelse af forretningsarkitektur

3.1 Indledning

Integrationen mellem de to systemer – ESDH og CMS – består af nogle snitflader, i form af webservices, som systemerne hver især kan eller skal tilbyde eller benytte.

For ikke at stille krav, der kan medføre at systemerne påvirker hinandens performance negativt, stilles der ingen krav om synkronisitet mellem systemerne. Men det er heller ikke et krav at kommunikationen skal være asynkron.

I udformningen af standarden er der blevet lagt vægt på at standarden ikke skal indskrænke mulighederne for forskellige systemarkitekturer unødigt.

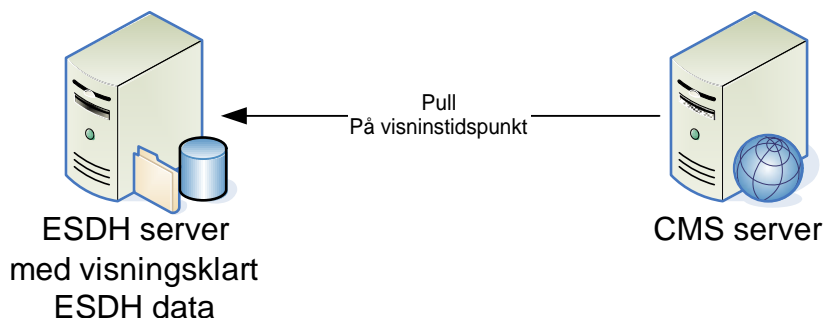
Således er der taget hensyn til alle tre af følgende systemarkitekturer:

1. Løsninger, hvor CMS-systemet tager initiativ til at hente data fra ESDH
2. Løsninger, hvor ESDH tager initiativ til at sende data til CMS
3. Løsninger, hvor de to systemer ikke kan tale direkte sammen, men kommunikerer via en mellemmand.

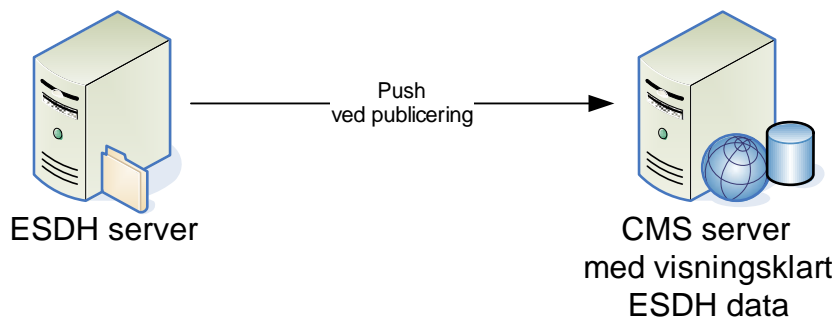
Eftersom data (indenfor rammerne af denne standard) fødes i ESDH men præsenteres i CMS, så kan første løsningsmodel kaldes en *Pull*-model, anden løsningsmodel kaldes en *Push*-model, mens den tredje model bruger både Push og Pull.

Nedenfor er de tre modeller illustreret i tre simple tegninger.

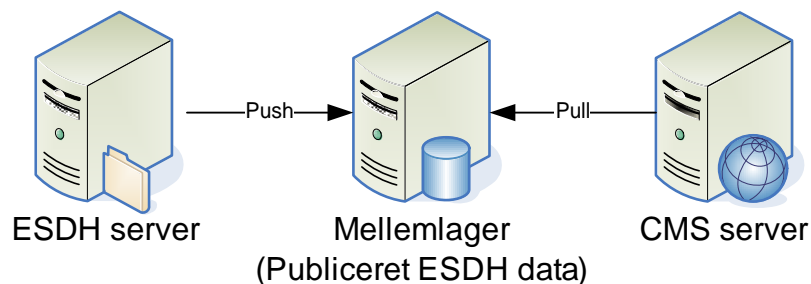
Model 1:



Model 2:



Model 3:



Uanset hvilken model der vælges i en given integration, er det et krav, at det er ESDH-systemet, der bestemmer hvilket udvalg af data, der skal kunne vises i CMS. Efter ESDH-systemet har 'frigjort' data som publiceret, er det op til CMS at bestemme over præsentationen af data.

3.2 Data

En afgørende egenskab ved ESDH-CMS integrationer er at sikre, at det kun er data, der opfylder visse kriterier, der må publiceres. Nedenfor gennemgås de fire kriterier, der afgør om et givent dataelement må publiceres.

3.2.1 Dataobjekter

De dataobjekter, det kan komme på tale at publicere, fremgår af standarden 'FESD Datamodel for sager og dokumenter'[1].

Det vil sige, at det skal som minimum være muligt at publicere sager og dokumenter fra ESDH til CMS, men integrationen ikke behøver at være begrænset til disse dataobjekter.

I tilfældet sag vil der indgå en række underobjekter (dokumenter, journalposter mv.). Det er vigtigt at disse ikke automatisk sættes til publicering når sagen er publiceret. ESDH-systemet skal i dette tilfælde sikre, at brugeren præsenteres for en oversigt over underelementer på sagen, som kunne tænkes at være relevante til publicering, og brugeren kan fra denne oversigt til- eller fravælge publicering af de enkelte underelementer.

3.2.2 Fortrolighed

For at data kan komme på tale til publicering skal det være åbent for offentligheden, det vil sige, at det der automatisk gives aktindsigt i ikke er konfidentielt.

3.2.3 Rettighed

Et element kan naturligvis kun publiceres af en bruger, der har rettigheder til at publicere.

3.2.4 Udpeget til publicering

Brugeren skal have valgt at publicere dataet. Det kan ikke komme på tale at publicere data, der ikke eksplicit er blevet sat til publicering.

3.3 Forretningskrav

3.3.1 Aktører i brugsscenarier

Med henblik på begrebsafklaring følger her en kort beskrivelse af de roller, der refereres til i det følgende.

Sagsbehandler	Bruger af ESDH-systemet med almindelige brugerrettigheder (typisk ikke administrator). Det kan være stor forskel fra organisation til organisation hvor mange rettigheder sagsbehandlere har, herunder om de har ret til publicering direkte til CMS.
Leder (ressortansvarligt)	En rolle som kan til- eller fravælges i integrationen er en leder, der skal godkende publiceringen. I et setup med inddragelse af leder-rollen kan sagsbehandleren kun markere et element til publicering, mens publicering kun kan finde sted hvis lederen accepterer det.
Webmaster	Den ansvarlige for indholdsstrukturen på web'en. Ved organisering af konkrete implementeringer bør det afklares hvem der har ret til – hhv. ansvar for – dataudvalg og struktur på webstedet
Slutbruger	Den bruger der i sidste ende læser data fra CMS'et

3.3.2 Funktionelle krav

Standarden stiller seks funktionelle krav:

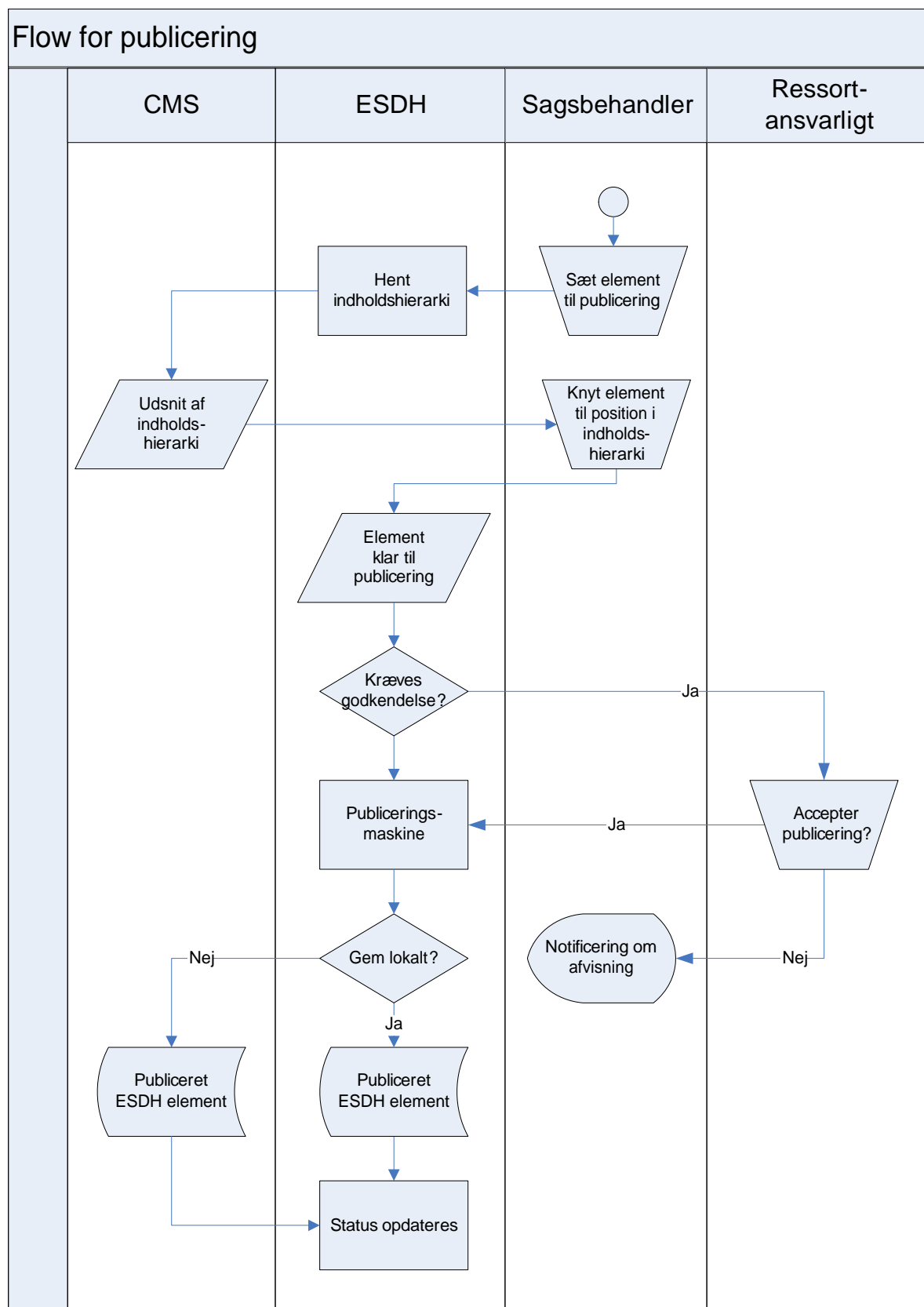
- Sagsbehandler (evt. ressortansvarlig) skal kunne publicere ESDH-data
- Sagsbehandler (evt. ressortansvarlig) skal kunne opdatere publiceret ESDH-data
- Sagsbehandler (evt. ressortansvarlig) skal kunne afpublicere publiceret ESDH-data
- Sagsbehandler (og ressortansvarlig) skal kunne se ESDH-systemets publiceringsstatus (men ikke nødvendigvis CMS-systemets)
- ESDH-systemet skal logge ændringer i ESDH-systemets publiceringsstatus
- Slutbruger skal kunne læse publiceret ESDH-data i CMS

I afsnit 3.1 er opstillet tre forskellige modeller for integration mellem ESDH og CMS. Det fremgår implicit at selve ESDH-dataet ikke nødvendigvis flyttes mellem systemerne på publiceringstidspunktet; det kan også ske på læsningstidspunktet.

Standarden understøtter således både at data kan gemmes lokalt (i ESDH-systemet) eller i et fremmed system (mellemlager eller CMS).

I de næste afsnit uddybes kravene til publicering, opdatering og afpublicering. Grænsefladerne til understøttelse af dette fremgår af afsnit 4.

3.3.3 Publicering



Use case navn	Publicering fra ESDH til CMS
Formål	ESDH-bruger kan publicere et element eller et sæt elementer fra ESDH til CMS
Forventet resultat	Elementet er klar til brug for CMS Elementet er indplaceret i CMS' indholdshierarki Publiceringsstatus er sat i ESDH
Aktører	Sagsbehandler, Ressortansvarlig, ESDH system, evt. Mellemlager, CMS
Beskrivelse:	
Sæt element til publicering	Sagsbehandleren har markeret et element i ESDH-systemet og vælger 'Publicér'. Ved implementering af brugergrænsefladen til dette, skal ESDH-leverandøren være opmærksom på, at der ikke må herske tvivl om hvilket element, der vælges til publicering. Det kan fx ske hvis brugeren har åbnet en sag og derefter højremusseklikker på en anden sag og vælger 'Publicér' på lokalmenuen. Se også 3.2.
Hent indholdshierarki	ESDH-systemet kalder en webservice der returnerer en XML nodeliste, med CMS' indholdshierarki. Format for kald er defineret i afsnit 4.2.1
Udsnit af indholdshierarki	Webservicen returnerer en XML nodeliste, der repræsenterer et udsnit af CMS' indholdshierarki under hensyntagen til den aktuelle brugers (ESDH-bruger eller ESDH-system-bruger) rettigheder. Implementeringspartneren kan vælge at inkludere et element i indholdshierarkiet, der fungerer som et 'medie-bibliotek', således at det ikke indgår direkte i CMS' navigérbare indholdshierarki, men kun bruges til at linke fra andre tekster eller i forbindelse med søgning.
Knyt element til position i indholdshierarki	ESDH-brugeren udpeger et eller flere steder i indholdshierarkiet hvor ESDH-elementet skal indsættes. Afhængig af implementeringen kan dette være som sideordnet (sibling) eller under-(child-) element.
Element klar til publicering	Publiceringsstatus sættes til 'Afventer publicering'.
Kræves godkendelse?	ESDH-leverandøren kan vælge at indsætte et led i publiceringsprocessen hvor en ressortansvarlig skal acceptere at det givne element publiceres. Tilsvarende kan ESDH-leverandøren vælge at notificere sagsbehandleren om hvilken beslutning, der er truffet.
Publiceringsmaskine	Såfremt ESDH-elementet skal publiceres, skal ESDH-systemet enten gemme dataet lokalt (kan implementeres som et flag, der angiver at elementet er publiceret) eller kalde en webservice med ESDH-elementet samt metadata om dette. Format for dette kald er defineret i afsnit 4.2.5. ESDH-leverandøren kan vælge at formatere, transformere eller konvertere data inden det sendes til mellemlageret.
Publiceret ESDH element	Når ESDH-elementet er publiceret, indgår det i CMS' forretningslogik. ESDH-systemet kan alene bestemme om et element er publiceret eller ikke publiceret.
Status opdateres	Publiceringsstatus sættes til 'Publiceret'

3.3.4 Opdatering af data

Opdatering af data skal ske som følge af en aktiv handling fra ESDH-brugeren. ESDH-systemet bør prompte ESDH-brugeren hvis der gemmes ændringer i et element, der er publiceret og bør i denne forbindelse give mulighed for at opdatere elementet i mellemlageret.

3.3.5 Afpublicering

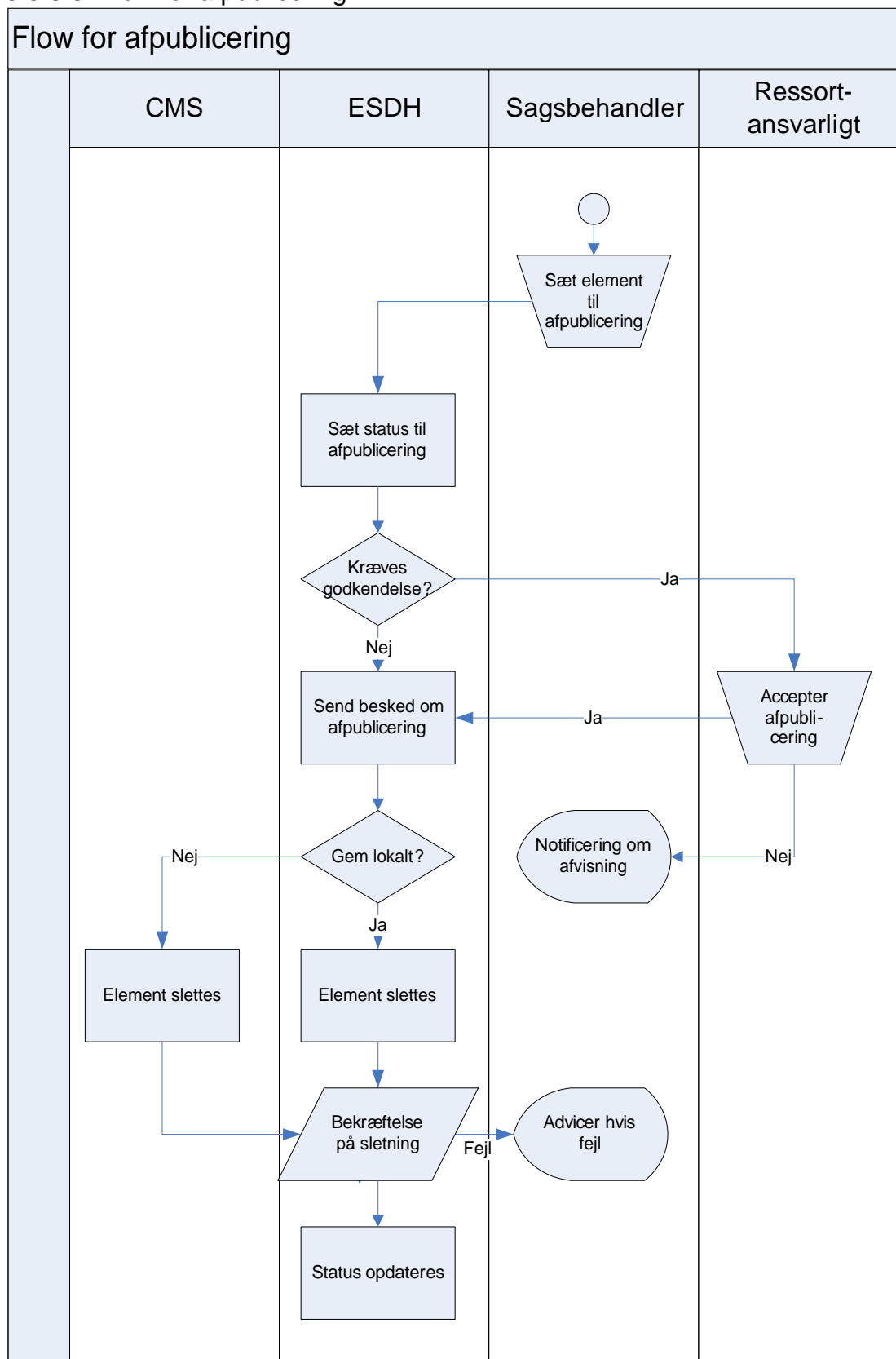
3.3.5.1 Eksplicit

Brugeren vælger eksplicit at afpublicere elementet.

3.3.5.2 Implicit

Elementet afpubliceres som underelement i en afpubliceret mængde.

3.3.5.3 Flow for afpublicering



Use case navn	Afpublicering af ESDH element fra CMS
Formål	ESDH-bruger kan afpublicere et element eller et sæt elementer således at det ikke længere er tilgængeligt for slutbrugeren.
Forventet resultat	Elementet fjernes fra CMS, mellemlager eller lokalt lager og kan derefter ikke vises i CMS Publiceringsstatus er sat i ESDH til 'Ikke publiceret'.
Aktører	Sagsbehandler, Ressortansvarlig, ESDH system, evt. Mellemlager, evt. CMS
Beskrivelse:	
Sæt element til afpublicering	Sagsbehandleren vælger at iværksætte afpubliceringsproceduren. ESDH-systemet bør gøre det klart for brugeren, at dette ikke i sig selv betyder, at elementet faktisk er blevet afpubliceret. Publiceringsstatus sættes til 'Afventer afpublicering'.
Accepter afpublicering	ESDH-leverandøren kan vælge at inkludere et led i afpubliceringsproceduren, hvor den ressortansvarlige skal acceptere afpubliceringen. Såfremt ressortansvarlige ikke vælger at acceptere afpubliceringen, rettes publiceringsstatus, og systemet kan returnere en notifikation til sagsbehandleren.
Send besked om afpublicering	Hvis data er gemt lokalt, slettes det (eller flag sættes, der angiver at elementet ikke er publiceret). Såfremt data ikke er gemt lokalt, skal ESDH-systemet – så længe status er sat til afpublicering – forsøge at kalde en webservice fra mellemlageret/CMS, der skal få dette system til at fjerne ESDH-elementet hos. Format for dette kald er defineret i afsnit 4.2.7. Det anbefales at ESDH-leverandøren inkluderer en facilitet, hvorefter ESDH-systemet forsøger afpublicering et antal gange, og – ved gentagne forsøg med negativt udfald - advarer sagsbehandler (og/eller ressortansvarlig) om det negative udfald.
Element slettes	Mellemlageret/CMS skal slette elementet – ikke bare markere det som slettet eller lignende. Dog må det gerne (bør) gemme en log-information om at elementet er slettet.
Bekræftelse på sletning	Det skal sikres, at ESDH-brugere kan få information om den reelle publiceringsstatus, både når det lykkes og når det ikke lykkes at gennemføre afpubliceringen.

4 Del C – Beskrivelse af datamodel og XML-skemaer

4.1 Publiceringsstatus

For at kunne opbevare publiceringsstatus for de publicerede elementer samt elementer, der har været publiceret, skal FESD-datamodellen[1] udvides med to tabeller. En tabel til opbevaring af status, og en støttetabel med udfaldsrummet af status.

4.1.1 publishingStatus

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
elementID	itemID	UUID	[1]	Id på det indholdselement som posten er status for.
brugerID	userIdentifier	String	[1]	Tekststreng der identificerer den bruger, der har bevirket elementets nuværende status.
publiceringsStatusFor- kortelse	StatusCode	Code	[1]	Fremmednøgle. Til tabellen publishingSta- tusCode
senesteAendringDa- toTid	lastChangeDateTime	DateTime	[1]	Dato og tid for hvornår publiceringsstatus senest er sat.

4.1.2 publishingStatusCode

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
publiceringsStatusFor- kortelse	StatusCode	Code	[1]	Fremmednøgle. Til tabellen CaseFileSta- tusCode
publiceringsStatusBe- tegnelse	PublishingStatusNa- me	Name	[1]	Publiseringsstatus betegnelse i klar tekst.
publiceringAnmod- ningsTidspunkt	publishRequestStart	DateTime	[0..1]	Dato og tid for hvornår indholdselementet ønskes publiceret.
afpubliceringAnmod- ningsTidspunkt	publishRequestEnd	DateTime	[0..1]	Dato og tid for hvornår indholdselementet ønskes afpubliceret.
publiceringFaktisk- Tidspunkt	publishActualStart	DateTime	[0..1]	Dato og tid for hvornår indholdselementet faktisk er blevet publiceret.
afpubliceringFaktisk- Tidspunkt	publishActualEnd	DateTime	[0..1]	Dato og tid for hvornår indholdselementet faktisk er blevet afpubliceret.

Udfaldsrummet for publiceringsstatus skal, jf. forretningskravene ovenfor, som minimum omfatte

- Ikke publiceret
- Afventer publicering
- Publiceret
- Afventer afpublicering
- Afpubliceret

4.2 Webservice grænseflader

4.2.1 getContentTree

Implementeres i CMS.

ESDH anmoder CMS om at få en kopi af det indholdshierarki, som eksporteret data kan indplaceres i.

4.2.1.1 Input

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
brugerID	userIdentifier		[1]	<p>Tekststreng der identificerer brugeren i det fremmede system.</p> <p>Dette kan være en generisk bruger, således at alle ESDH-brugere optræder som samme bruger i mellemlagret; eller det kan være ESDH-brugeren også er oprettet i mellemlagret, således at deres brugerID kan bruges i dette.</p>

4.2.1.2 Output

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
DannetTidspunkt	ListGeneratedDate- Time	DateTime	[1]	Tidspunkt for hvornår nodelisten er dannet.
DannetSystem	ListGeneratorName	String	[0..1]	Navn på systemet, der har produceret nodelisten
MenuListe	MenuItemCollection	XML ele- ment	[1]	Liste af menupunkter

4.2.1.2.1 MenuList-parameter

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
menuPunktID	MenuItemStructure	XML ele- ment	[0*]	Et menupunkt – container for attributter og elementer der udgør egenskaberne for menupunktet.
Id	MenuItemIdentifier	Uuid	[1]	<p>CMS' id på menupunktet. Det er dette id, der er den unikke og utvetydige identifikation af det punkt i CMS' indholdshierarki, hvorunder ESDH-elementet indsættes som sideordnet (sibling) eller under-(child-) element.</p> <p>Det vil typisk være et Uuid, men det læses som string, da der ikke kan være sikkerhed for at det er et Uuid.</p>
AccepterUnderpunkter	MenuItemAcceptAt- tachmentIndicator	Boolean	[1]	<p>For at kunne præsentere ESDH-brugeren for et sammenhængende indholdshierarki, som det er muligt at navigere rundt i, kan det være nødvendigt at inkludere elementer i nodelisten, som CMS ikke accepterer at der hættes ESDH-elementer på.</p> <p>Såfremt denne attribut er af værdien Sand (1), kan der vedhættes ESDH-elementer på dette MenuItem; ellers ikke.</p>

MenuTekst	MenuItemText	String	[1]	Dette er den tekst, som ESDH-brugeren præsenteres for til navigering i indholdstræet. Det er ikke noget krav at den er præcis som den tilsvarende tekst i CMS, men det skal naturligvis være muligt for ESDH-brugeren entydigt at kunne identificere positionen ud fra teksten og dens relative placeringen i indholdshierarkiet (nodelisten).
MenuListe	MenuItemCollection	XML element	[0..1]	Eventuelle undermenupunkter

4.2.2 createContentPosition

Kan implementeres i CMS. ESDH skal kunne konfigureres til at benytte dette.

ESDH sender oplysning til CMS om hvilket data der sættes ind i indholdshierarkiet med oplysning om hvor det indsættes (men uden data).

4.2.2.1 Input

Attribut Dan	Attribut Eng	Type	Kardinalitet	Beskrivelse
brugerID	userIdentifier		[1]	Tekststreng der identificerer brugeren i det fremmede system.
indholdsplacering	contentPosition		[0*]	Sæt af hhv. indholdselementer og deres indplaceringer

4.2.2.1.1 contentPosition-parameter

Attribut Dan	Attribut Eng	Type	Kardinalitet	Beskrivelse
elementID	ContentitemIdentifier	UUID	[1]	Id på et indholdselement
menuPunktID	menuItemIdentifier	UUID	[1*]	Et antal id'er på de placeringer som indholdselementet skal tilknyttes.

4.2.2.2 Output

–

4.2.3 updateContentPosition

Kan implementeres i CMS. ESDH skal kunne konfigureres til at benytte dette.

ESDH sender oplysning til CMS om ændrede indsætningspunkter for allerede publicerede indholdselementer.

4.2.3.1 Input

Attribut Dan	Attribut Eng	Type	Kardinalitet	Beskrivelse
brugerID	userIdentifier		[1]	Tekststreng der identificerer brugeren i det fremmede system.
Indholdsplacering	contentPosition		[0*]	Sæt af hhv. indholdselementer og deres indplaceringer. Se contentPosition-parameter ovenfor.

4.2.3.2 Output

—

4.2.4 removeContentPosition

Kan implementeres i CMS. ESDH skal kunne konfigureres til at benytte dette.

ESDH sender oplysning til CMS om at et indholdselement er blevet afpubliceret, at og oplysninger om dets indsætningspunkter derfor skal slettes.

4.2.4.1 Input

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
brugerID	userIdentifier		[1]	Tekststreng der identificerer br ugeren i det fremmede system.
Indholdsplacering	contentPosition		[0*]	Sæt af hhv. indholdselementer og deres indplaceringer. Se contentPosition-parameter ovenfor.

4.2.4.2 Output

—

4.2.5 createCMDData

Kan implementeres i CMS. ESDH skal kunne konfigureres til at benytte dette.

Modtager pakke med data, der skal hæftes på indholdshierarkiet med oplysning om hvor på træet det skal hæftes på mm.

4.2.5.1 Input

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
brugerID	userIdentifier		[1]	Tekststreng der identificerer br ugeren i det fremmede system.
dokumentOmslag	documentWrapper		[0..1]	Container-element til et dokument (se documentWrapper-parameter nedenfor)
sagsOmslag	caseWrapper		[0..1]	Container-element til et sag (se caseWrapper-parameter nedenfor)

4.2.5.2 Output

—

4.2.6 updateCMDData

Kan implementeres i CMS. ESDH skal kunne konfigureres til at benytte dette.

Modtager opdaterede oplysninger om et allerede publiceret indholdselement.

4.2.6.1 Input

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
brugerID	userIdentifier		[1]	Tekststreng der identificerer br ugeren i det fremmede system.

dokumentOmslag	documentWrapper		[0..1]	Container-element til et dokument (se documentWrapper-parameter nedenfor)
sagsOmslag	caseWrapper		[0..1]	Container-element til et sag (se caseWrapper-parameter nedenfor)

4.2.6.2 Output

–

4.2.7 removeCMDData

Kan implementeres i CMS. ESDH skal kunne konfigureres til at benytte dette.

Anmoder om at få fjernet et indholdselement fra CMS eller mellemlager.

4.2.7.1 Input

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
brugerID	userIdentifier		[1]	Tekststreng der identificerer br ugeren i det fremmede system.
elementID	ContentitemIdentifier	UUId	[1]	Id på et indholdselement

4.2.7.2 Output

–

4.2.8 getCMContentList

Bemærk – forskellig fra nedenstående, idet denne implementeres i CMS. Kaldes fra ESDH.

Returnerer en liste over alle de ESDH-objekter, der optræder i CMS. Dette med henblik på sikkerhedstjek på synkroniteten mellem CMS og ESDH-systemets oplysninger om publiceringsstatus.

4.2.8.1 Input

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
brugerID	userIdentifier		[1]	Tekststreng der identificerer brugeren i det fremmede system.

4.2.8.2 Output

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
indholdsplacering	contentPosition		[0*]	Sæt af hhv. indholdselementer og deres indplaceringer

4.2.8.2.1 contentPosition-parameter

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
elementID	ContentitemIdentifier	UUId	[1]	Id på et indholdselement
menuPunktID	menuitemIdentifier	UUId	[1*]	Et antal id'er på de placeringer som indholdselementet er tilknyttet.

4.2.9 getCMContentList

Bemærk – forskellig fra ovenstående, idet denne implementeres i ESDH.

CMS kan kalde denne webservice for at få oplyst hvilke indholdselementer, der er publiceret. Såfremt der er uoverensstemmelser, kan CMS enten slette overskydende poster lokalt eller hente manglende eller ikke-opdaterede indholdselementer ved brug af servicen readCMDData.

4.2.9.1 Input

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
brugerID	userIdentifier		[1]	Tekststreng der identificerer br ugeren i det fremmede system.

4.2.9.2 Output

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
indholdsplacering	contentPosition		[0*]	Sæt af hhv. indholdselementer og deres indplaceringer

4.2.9.2.1 contentPosition-parameter

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
elementID	ContentitemIdentifier	UUId	[1]	Id på et indholdselement
menuPunktID	menuitemIdentifier	UUId	[1*]	Et antal id'er på de placeringer som indholdselementet er tilknyttet.

4.2.10 getCMDData

Implementeres i ESDH. Kaldes fra CMS.

Udleverer dokument eller sag fra ESDH til CMS.

4.2.10.1 Input

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
brugerID	userIdentifier		[1]	Tekststreng der identificerer br ugeren i det fremmede system.
elementID	itemIdentifier		[1]	UniquelIdentifier til indholdselementet

4.2.10.2 Output

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
dokumentOmslag	documentWrapper		[0..1]	Container-element til et dokument (se documentWrapper-parameter nedenfor)
sagsOmslag	caseWrapper		[0..1]	Container-element til et sag (se caseWrapper-parameter nedenfor)

4.2.10.2.1 documentWrapper-parameter

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
elementID	ContentitemIdentifier	UUID	[1]	Id på et indholdselement
dokumentTitel	documentTitleText	text char(255)	[0..1]	Dokumentets titel
dokumentIndhold	documentContentData		[0..1]	Dokumentindholdet i binært format.
dokumentDato	documentCreateDate	Date	[0..1]	Dokumentdato
dokumentVersionID	documentVersionIdentifier	Integer(5)	[0..1]	Dokumentets versionnummer indenfor dokumentet
dokumentStatusBetegnelse	DocumentStatusText	String	[0..1]	Dokumentstatus i klar tekst.
dokumentFilNavn	DocumentFileName	String	[0..1]	
filType	fileTypeCode	Char(10)	[0..1]	Angiver standardformat filtyper eksempelvis DOC for Word. StorageFormat.defaultFileType

4.2.10.2.2 caseWrapper-parameter

Attribut Dan	Attribut Eng	Type	Kardinali- tet	Beskrivelse
elementID	ContentitemIdentifier	UUID	[1]	Id på et indholdselement
sagDato	caseFilecreationDate	Date	[1]	Oprettelsesdato for sagen
kassationsTekst	disposalName	Name char(70)	[0..1]	Angiver hvad der skal ske når kassationsdatoen er nået Feltet Name fra tabellen CaseFileDisposal
sagKassationsDato	caseFileDisposalDate	Date	[0..1]	Kassationdato
sagSenestOpdateretAf	caseFileLatestEditorReference	Name char(70)	[1]	Navn på person, der senest har registreret sagen. Udfyldes om muligt (er pt. ikke implementeret i FESDs datamodel).
sagSenesteJournalPost	caseFileLatestRecordDate	Date	[0..1]	Dato for seneste registrerede journalpost.
senestOpdateretDato	latestUpdateDate	Date	[1]	Dato for seneste opdatering.
sagAnsvarligSagsbehandler	caseFileManagerReference	Name char(70)	[0..1]	Feltet udfyldes med navn på ansvarlig sagsbehandler. Hentes fra Bruger.brugerNavn
sagTidligereSagNr	caseFileNumberHistoricIdentifier	Char(40)	[0..1]	Frit tidligere sagsnr. Eventuelt sammenstillinger af andre felter. Udfyldes med sagNr, hvis sagen har været omjournaliseret. Tidligere sagsnummer. Fri tekst.
sagNr	caseFilenumberIdentifier	Char(40)	[1]	Frit sagsnummer.
sagPapir	caseFilePaperStorageIndicator	Indicator Char(1)	[0..1]	Angiver om sagen arkiveres på papirform eller elektronisk.
sagSletningsfrist	caseFileRetentionTimeYear	Integer(2)	[0..1]	Antal år sagen skal bevares.
sagLoebenummer	caseFileSequenceNumber	SequenceNumber	[1]	Indeholder sagsens sekvensnummer eventuelt indenfor året. Anvendes sammen med sagsens oprettelsesår til at danne journalnummeret. Entydigt maskingenereret løbe-

		Integer(6)		nummer. Nulstilles evt. hvert år.
sagStatusBetegnelse	caseFileStatusName	Name char(70)	[0..1]	Sagens status i fritext
sagTitelAlternativ	caseFileTitleAlternativeText	Text char(255)	[0..1]	Hvis sagstitlen er uofficielle kan dette felt indeholde en alternative official sagstitel, til postlister ect. Feltet indeholder en alternativ titel i klar tekst
sagTitel	caseFileTitleText	text char(255)	[1]	Sagens titel
sagsTypeTekst	caseFileTypeName	Name char(70)	[0..1]	Feltet Name fra tabellen CaseFileType
sagAar	caseFileYear	Year	[1]	Sagens oprettelsesår
dokumentOmslag	documentWrapper		[0..1]	Container-element til et dokument (se documentWrapper-parameter ovenfor)

4.2.11 Samlet oversigt over webservices

For at lette overblikket over de webservices der er beskrevet ovenfor, følger her en samlet oversigt over disse. Endvidere er tilføjet kolonnen 'Egnet til', hvor der med reference til de tre modeller, der er beskrevet i afsnit 3.1, er anført hvilken af disse modeller, webservicen er rettet mod.

4.2.11.1 Implementeres i CMS

Navn	Obligatorisk for CMS	Obligatorisk for ESDH	Egnet til
readContentTree	Ja	Ja	model 1 model 2 model 3
createContentPosition	Nej	Ja	model 1 model 3
updateContentPosition	Nej	Ja	model 1 model 3
removeContentPosition	Nej	Ja	model 1 model 3
createCMDData	Nej	Ja	model 1 model 2
updateCMDData	Nej	Ja	model 1 model 2
removeCMDData	Nej	Ja	model 1 model 2
getCMContentList	Ja	Ja	model 1

			model 2 model 3
--	--	--	--------------------

4.2.11.2 Implementeres i ESDH

Navn	Obligatorisk for CMS	Obligatorisk for ESDH	Egnet til
getCMContentList	Nej	Ja	model 1 model 2 model 3
getCMDData	Nej	Ja	model 1 model 3

Bemærk vedrørende denne høring:

OIOXML-skemaer udarbejdes efter afsluttet høring, idet høringen kan påvirke udvalget af datalementer (attributter) og dermed også de tilhørende OIOXML-skemaer.

5. Bilag A

Anvendte typer i FESD-modellerne

FESD-modellerne er udarbejdet i UML med det formål at beskrive den logiske informationsarkitektur i en FESD-løsning.

UML-modellen er opbygget af et antal klasser, der igen er opbygget af relationer til andre klasser og primitive typer, så man kan opfatte UML-modellen som opbygget af byggesten, hvoraf den mindste er de primitive datatyper.

I UML-modellen beskriver de primitive datatyper det logiske domæne, som datatypen kan antage, men ikke hvordan den fysiske repræsentation af data skal være.

En gennemgang af de FESD-modeller, der på nuværende tidspunkt foreligger, viser, at vi har anvendt nedenstående primitive datatyper. I skemaerne ses for hver primitiv datatype, der anvendes i FESD-UML-modellen, den tilsvarende datatype for hhv. SQL og XSD.

integer

Benyttes til angivelse af heltal.

Der kan benyttes en angivelse af max længde – hvis intet er angivet, vil domænet være i intervallet mellem - 2.147.483.648 og 2.147.483.647. Anvendte længder i den nuværende model: 2, 3, 4, 6, 8, 10.

Benyttes alene til naturlige tal (positive heltal).

Benyttes også til at danne identifikator med udfaldsrum 1 – 2000000000.

Nedenstående skema viser eksempler på heltalstyper og deres repræsentation i hhv. SQL og XSD

UML Type	Beskrivelse	SQL datatype	XSD datatype
Integer	Heltal i rummet mellem - 2.147.483.648 og 2.147.483.647, begge tal inklusive. Den maksimale længde angives i modellen (*Hvordan ?*)	NUMBER(Maksimal længde)	xsd:int
NonNegativeInteger	Naturlige tal Heltal i rummet mellem 0 og 2.147.483.647, begge tal inclusive	NUMBER(Maksimal længde)	xsd:int

boolean

Er en grundtype i UML.

UML Type	Beskrivelse	SQL datatype	XSD datatype
Boolean	Udfaldsrummet er binært true / false (eller rigtigt / forkert).	BOOLEAN	xsd:boolean

Identifikationstyper

UML Type	Beskrivelse	SQL datatype	XSD datatype
URI	Enhver mulig lovlig URI. Der kan evt. anvendes en max-længde.	STRING	xsd:anyURI
HttpURI	Lovlig http(s)-adresse	STRING	xsd:anyURI

			Evt. restrictions der afgrænser til http.
MailToURI	Smtip-adresse på en mailmodtager.	STRING	xsd:anyURI Evt. restrictions der afgrænser til mailadresser.
UUID	Identifikator på objekt	UUID	xsd:anyURI Evt. restrictions der afgrænser til UUID.
UserIdentifier	Det er en identifikation af bruger		
SystemIdentifier	Det er en identifikation af systemer		
BinaryObject	Et binary large object - også kaldet "blob" - er en samling af binære data, der opbevares som en separat entitet i et database management system. Blobs er typisk billeder, lyd og andre medieobjekter, selvom binær eksekverbar kode til tider også opbevares som en "blob". Database support for "blob" er ikke universal.	Binary Large Object (BLOB)	

char

Benyttes til karakterstreng af varierende længde, men med en defineret maksimal længde. Anvendte længder: 1, 2, 3, 4, 15, 16, 20, 34, 40, 50, 60, 70, 110, 120, 255.

Benyttes også nogle steder til at angive en boolskværdi.

Bør ændres så typen alene bruges til tekststreng:

UML Type	Beskrivelse	SQL datatype	XSD datatype
Char	Den minimale og maksimale længde er angivet i modellen.	VARCHAR(maksimal længde)	xsd:string Med restriction på den maximale længde
ShortText	Anvendes til felter der indeholder en kort beskrivende tekst	VARCHAR(70)	xsd:string Med restriction på den maximale længde
Text	Anvendes til en beskrivende tekst med en fast længde.	VARCHAR(255)	xsd:string Med restriction på den maximale længde
Code	Anvendes til at beskrive en kode, der er nøgle/fremmednøgle i en opslags-tabel.	VARCHAR(2)	xsd:string Med restriction på den maximale længde
Name	Anvendes til felter der indeholder navne, der kan opfattes som brugervendte nøgler.	VARCHAR(70)	xsd:string Med restriction på den maximale længde
ReferenceText	Anvendes til at felter der indeholder tekster, der af brugerne	VARCHAR(40)	xsd:string

	opfattes som fremmednøgler.		Med restriction på den maximale længde
--	-----------------------------	--	--

date, dateTime og time.

Typen 'Date' bruges til at angive dato.

Typen 'DateTime' bruges til 'Dato og tidspunkt', også i betydningen 'tidsstempel'.

Typen 'Time' bruges til 'klokkeslæt'.

UML Type	Beskrivelse	SQL datatype	XSD datatype
Year	Årstal	VARCHAR (4)	xsd:gYear
Date	Dato	DATE	xsd:date
DateTime	Dato og tidspunkt	DATETIME	xsd:dateTime
Time	Tidspunkt uden datoangivelse	TIME	xsd:time

string

Er en grundlæggende type i UML.

UML Type	Beskrivelse	SQL datatype	XSD datatype
String	Tekststreng af vilkårlig længde	STRING	xsd:string

float

Defineres som sådan:

UML Type	Beskrivelse	SQL datatype	XSD datatype
Float	Decimaltal hvor længde og præcision begrænses til en samlet størrelse på 16-bit.	DECIMAL(X,Y)	xsd:float