
>

Anbefaling for en generel hændelsesbesked

Version 1.0

Indhold

>

Indledning	3
Anbefalet brug	3
Baggrund	3
Proces	3
Beskrivelse af indhold	3
Kapitel 1: Mål og principper	4
Hvad er en generel hændelsesbesked kort?	4
Mål og principper	4
Kapitel 2. Proces integration med hændelser	5
Anvendelse af hændelser til løs kobling	5
Begreber	5
Anvendelses scenarie	8
Kapitel 3. WSI-SD – Ringsted-projektet	10
Format for hændelsesbesked - EventStructure	10
Kapitel 4. Internationale standarder	11
Web service Notification (WSN)	11
Format for hændelsesbesked	11
Webservices Eventing (WSE)	11
Format for hændelsesbesked	11
Kapitel 5. En generel hændelsesbesked	12
Anvendelse af beskedformatet sammen med	14
WS-Eventing eller WS-Notification	14
Eksempler	14
Referencer	16
Appendiks A – XML Schema for WSN	18
Appendiks B – Foreslået XML Schema til WSE	19
Appendiks C – Forslag fra [Haendelse08]	20

Indledning

Dette dokument er et udkast til en anbefaling for en generel hændelsesbesked.

Anbefalet brug

Beskedformatet defineret i dette dokument kan anvendes til generelt at overføre hændelser på en ensartet måde, med sigte på en overordnet besked interoperabilitet.

Dokumentbaseret messaging og web services har i dag nogen udbredelse i den offentlige sektor, mens proces integration med hændelser stadig er nyt for de fleste med undtagelse af enkelte sektorer og problemområder. Dette kan udtrykkes som, at modenheden inden for dette område endnu ikke er så stor. Inden for Sag & Dokument-området forventes hændelser dog at blive et vigtigt element hen imod realiseringen af reference arkitekturen, hvilket vil være medvirkende til en øget modenhed.

Baggrund

På baggrund af et webservice pilotprojekt for Integration af Sager og Dokumenter (WSI-SD) [WSISD], blev der i foråret af 2008 udarbejdet et høringsudkast til standardisering af en generel hændelsesbesked. Forslaget til standardisering af hændelsesbeskeden blev dog ikke godkendt af OIO-komiteén, der savnede en tydeliggørelse af forslagets relationer til internationale standarder på området. I stedet blev det besluttet, at der skulle gennemføres en åben OIO-proces med inddragelse af relevante nationale og internationale erfaringer, og at dette skulle munde ud i et revideret forslag til standardisering af generel hændelsesbesked, der skal sendes i høring og behandles i OIO-komiteén inden udgangen af 2009.

Proces

Det oprindelige udkast er blevet holdt op imod de internationale standarder, der findes for generelle hændelsesbeskeder. Der blev før sommeren afholdt en halvdags workshop, hvoraf det blev besluttet, at de internationale standarder ikke havde nogen særskilt værdi. Derfor skulle det eksisterende udkast tilpasses til kravene der hidtil er kommet ud af standardiseringen af Sag & Dokument. Der har været søgt efter lignende formater som allerede er i brug, men dette har ikke været tilfældet.

Beskrivelse af indhold

Dette dokument indeholder først et kapitel, som kort beskriver, hvad en generel hændelsesbesked er for en størrelse og de principper, der ligger til grund for valget. Kapitel to har en kort introduktion til integration med hændelser, og fastlægger de grundlæggende begreber samt generelle scenarier. Det tredje kapitel beskriver kort den generelle hændelsesbesked, som var i det oprindelige høringsudkast "Hændelse - Semantik og Skema" [Haendelse08]. Det fjerde kapitel beskriver de dele af relevante internationale standarder, som kan bruges i forhold til en generel hændelsesbesked og kapitel fem beskriver det anbefalede format som også er realiseret i skemaer tilgængelige på digitaliser.dk. Sidst følger der en liste af relevante referencer og i de følgende appendix en kort xml skemabeskrivelse af de internationale kandidater.

Kapitel 1: Mål og principper

Hvad er en generel hændelsesbesked kort?

En generel hændelsesbesked er et beskedformat (message), som kan anvendes ved overførsel af hændelser mellem it-systemer. Det generelle betyder, at det kan bruges bredt for alle hændelser uanset fagområde, sektor eller konkret hændelse. Den generelle hændelsesbesked fungerer primært som en del af infrastrukturen og ligger i grænselaget imellem den specifikke hændelsesbesked (den egentlige forretnings hændelse) og transportlaget (adressering, protokol og wireformat), som vist i figur 1 nedenfor:



Figur 1

Dette gør det muligt for en aftager umiddelbart at modtage en række forskellige hændelser uden at have understøttelse for de specifikke formater, og kan på denne måde simplificere grænsefladen og anvendes til f.eks. en overordnet indholdsbaseet routing.

Mål og principper

En generel hændelse skal opfylde følgende mål:

- Skal kunne bruges generelt for alle hændelser uanset fagområde, sektor eller konkret hændelse
- Skal være let at forstå og anvende/transport
- Skal være uafhængig af udvekslings protokol
- Skal kunne give værdi i mere komplekse tekniske arkitekturer uden at gøre det unødigt komplekst for simple arkitekturer (topologier)
- Skal understøtte at specifikke hændelsesmeddelelser kan være både meget simple eller komplekse
- Det skal være muligt at sikre integritet fra producent til modtager
- Skal kunne anvendes uanset metode til autentifikation eller autorisation.

Omvendt er følgende IKKE et mål:

- Valg af teknisk protokol (OIORASP, OWSA ModelT mv.)
- Valg af abonnements håndtering, oprettelse, vedligeholdelse, herunder abonnementsudtryk
- Valg af syntaks for abonnements udtryk eller emneopdeling
- Valg af event/notifikations standard bredt
- Push versus Pull
- Indholdet af specifikke hændelsesbeskeder samt eventuelt standardisering og generalisering i sektorer og domæner
- Fastlæggelse af taksonomi for hændelsestyper, emner og lignende
- Guide til vurdering af egnede hændelser
- Erstatte de beskedformater der følger med de fleste messaging/transport protokoller som f.eks. JMS, MSMQ, AMQP osv.
- Erstatte eksisterende management protokoller som f.eks. Simple Network Management Protocol (SNMP). Fokus er på forretningshændelser.

Kapitel 2. Proces integration med hændelser

Der findes en mængde litteratur og artikler, som beskriver hændelsesbaseret arkitektur (Event-Driven Architecture) i dybden. Dette er blot en kort beskrivelse. Den meget anvendte bog *Enterprise Integration Patterns* [EIP] beskriver de relevante mønstre, herunder hændelser (events).

Nedenfor beskrives ganske kort, hvorledes integration med hændelser kan skabe en sammenhængende og samtidigt løst koblet arkitektur. Herefter defineres de centrale begreber og sidst gives et grundlæggende scenarie for anvendelse af en generel hændelsesbesked.

Anvendelse af hændelser til løs kobling

Arbejdsprocesser som er forbundne men foregår uafhængigt af hinanden kan med fordel integreres med hændelser. Arbejdsprocesser som typisk har en længere levetid, foregår ofte som en række på hinanden følgende procestrin, hvor afslutning af det ene procestrin starter det næste. Dette kan teknisk realiseres ved at beskrive afslutningen af et procestrin med en hændelsesbesked (der kan betragtes som data meddelelser eller dokumenter) som overføres og starter næste procestrin. Karakteristisk for hændelsesbaseret integration er, at der ikke er nogen central styring af det samlede flow, når først det er designet og implementeret. Det skal naturligvis overvåges for at sikre, at det forløber hensigtsmæssigt og som forventet.

Et eksempel på en sådan integration kan ses i referencearkitekturen for Sag & Dokument [SDREFARK].



Figur 2 Procesintegration med hændelser (fra figur 17 fra [SDREFARK])

Hele referencearkitekturen for Sag & Dokument er i høj grad baseret på integration imellem processerne ved hjælp af hændelser. Der refereres til BPMN's definition (afsnit "9.3 Events" i [BPMN11]) som dækker bredt:

"An Event is something that "happens" during the course of a business process. These Events affect the flow of the Process and usually have a cause or an impact. The term "event" is general enough to cover many things in a business process. The start of an activity, the end of an activity, the change of state of a document, a message that arrives, etc., all could be considered events. However, BPMN has restricted the use of events to include only those types of events that will affect the sequence or timing of activities of a process. BPMN further categorizes Events into three main types: Start, Intermediate, and End."

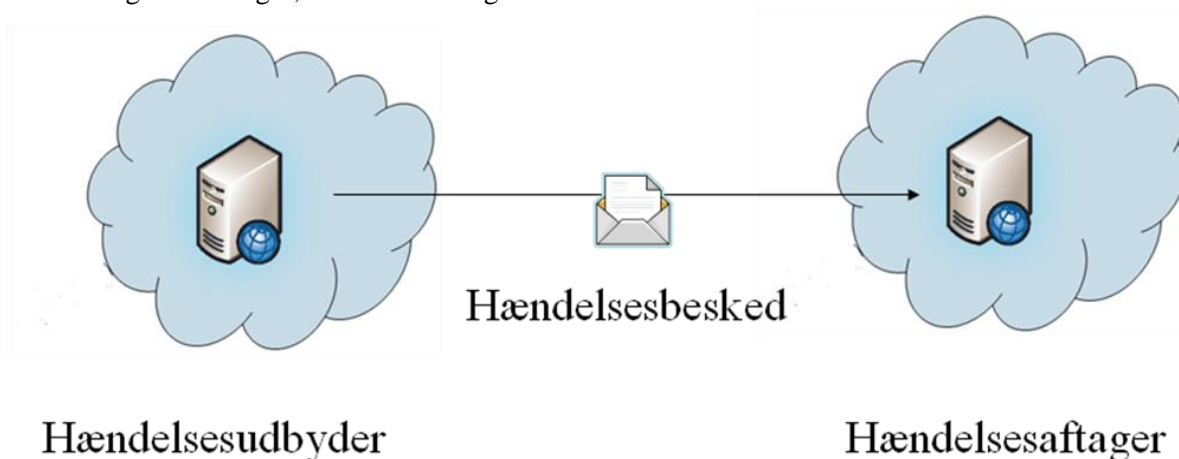
Bemærk: Uanset hvilken arkitekturmodel der anvendes ved integration, skal aftaler eller formelle kontrakter, som regulerer anvendelsen af informationerne, overholdes. Hændelsesbaseret integration kan give nogle tekniske fordele, blandt andet i forhold til løbende at kunne understøtte ændrede arbejdsprocesser. Men selvom systemerne bliver løstkoblede er det stadig systemejernes ansvar at informationerne anvendes til det korrekte formål. Eksempelvis sikring af at der kun sendes korrekte hændelser, at der kun kan laves abonnementer som dækker reelt behov, at følsomme data forbliver fortrolige osv.

Begreber

I dette afsnit er de vigtigste begreber beskrevet med udgangspunkt i terminologien, der er anvendt i Web Service Notification [WSN]. Herved genbruges det arbejde, der er lagt i den terminologi og det kendskab der er til det og da den konceptuelle model passer godt med den som blev brugt i [WSISD]. Web service Eventing har en mere simpel begrebsmodel, som kan mappes ud fra denne. Selve begreberne er på dansk

med udgangspunkt i de begreber, der blev defineret i [WSISD] og med de engelske termer fra Web Service Notification bagefter i parentes.

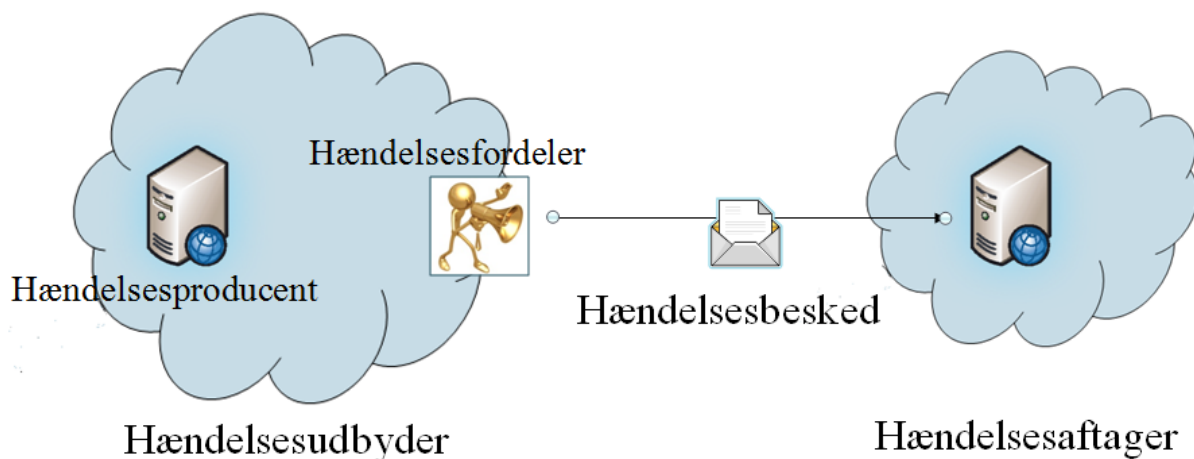
Begreberne dækker over to arkitekturlag. I det øverste og mest forretningsnære arkitekturlag er der blot en afsender og en modtager, som det ses i figur 3.



Figur 3 Simpel udveksling af hændelsesbeskeder

Dette rummer hvad formålet med det hele er, nemlig at få hændelsen overført til modtageren. Dette svarer til *Message Channel* som beskrevet i [EIP].

Et underliggende lag i arkitekturen afspejler et mere applikations og infrastruktur nært niveau, hvor hændelsesudbyderen er funktionelt opdelt i henholdsvis en producent og en hændelsesfordeler (figur 4).



Figur 4 Udvexling af hændelsesbeskeder via en hændelsesfordeler

Ved anvendelse af en hændelsesfordeler afhjælpes producenten den endelige distribution. Kontrakten for hændelsesintegrationen vil stadig være som vist i figur 3, men hændelsesudbyderen har rent teknisk funktionelt opdelt opgaverne internt, uden at dette har nogen betydning for hændelsesaftageren.

Her følger de mest anvendte begreber.

Abonnement (Subscription)

En aftale mellem en hændelsesaftager og en hændelsesudbyder, om levering eller adgang til hændelser ud fra et valgt abonnementsudtryk eller emne.

Abonnementsudtryk (Subscription filter)

Et logisk udtryk, der stiller krav til indholdet. En hændelse kan distribueres på basis af et abonnementsudtryk.

Abonnent (Subscriber)

En abonnent er den som anmoder om et abonnement, og behøver ikke at være den endelige hændelsesaftager.

Emne (Topic)

Et *emne* er det koncept der bruges til at kategorisere hændelser og deres indhold.

Emner bruges som en del af processen, som bestemmer hvilke hændelsesaftagere der skal modtage en hændelsesbesked. Når en hændelsesproducent laver en hændelsesbesked kan denne tilknyttes et emne.

Hændelse (Situation)

En hændelse er en eller anden begivenhed eller logisk tilstandsændring, som er kendt af hændelsesproducenten og at potentiel interesse for andre parter/applikationer.

Hændelsesbesked (Notification)

En hændelsesbesked er en konkret meddelelse, som beskriver en *hændelse*, når dette skal formidles fra et it-system. En hændelsesbesked er et artefakt af selve hændelsen, som indeholder information omkring hændelsen som en entitet ønsker at kommunikere til andre entiteter.

Hændelsesproducent (Publisher)

En som har adgang til at 'fange' hændelser og er i stand til at omsætte dette til hændelsesbeskeder.

Hændelsesudbyder (NotificationProducer)

En der i praksis udbyder *hændelsesbeskeder* til andre, dette kan være knyttet direkte til *hændelsesproducenten* eller foretages af en anden part/applikation kaldet en *hændelsesfordeler*. For *hændelsesaftagerne* har dette ingen praktisk betydning, da de logisk set er engageret med én part og ikke bør kende topologien/applikationsarkitekturen hos den anden part.

Hændelsesfordeler (NotificationBroker)

En hændelsesfordeler kan optræde i to roller. Den første er at fordele *hændelsesbeskeder* for *hændelsesproducenter* og herunder styring af abonnenter og adgang. Den anden rolle er som *hændelsesaftager* som gateway f.eks. af sikkerhedshensyn eller til intern fordeling.

Hændelsesaftager (NotificationConsumer)

Dette er modtageren af hændelsesbeskeder fra en hændelsesudbyder.

Hændelsestype

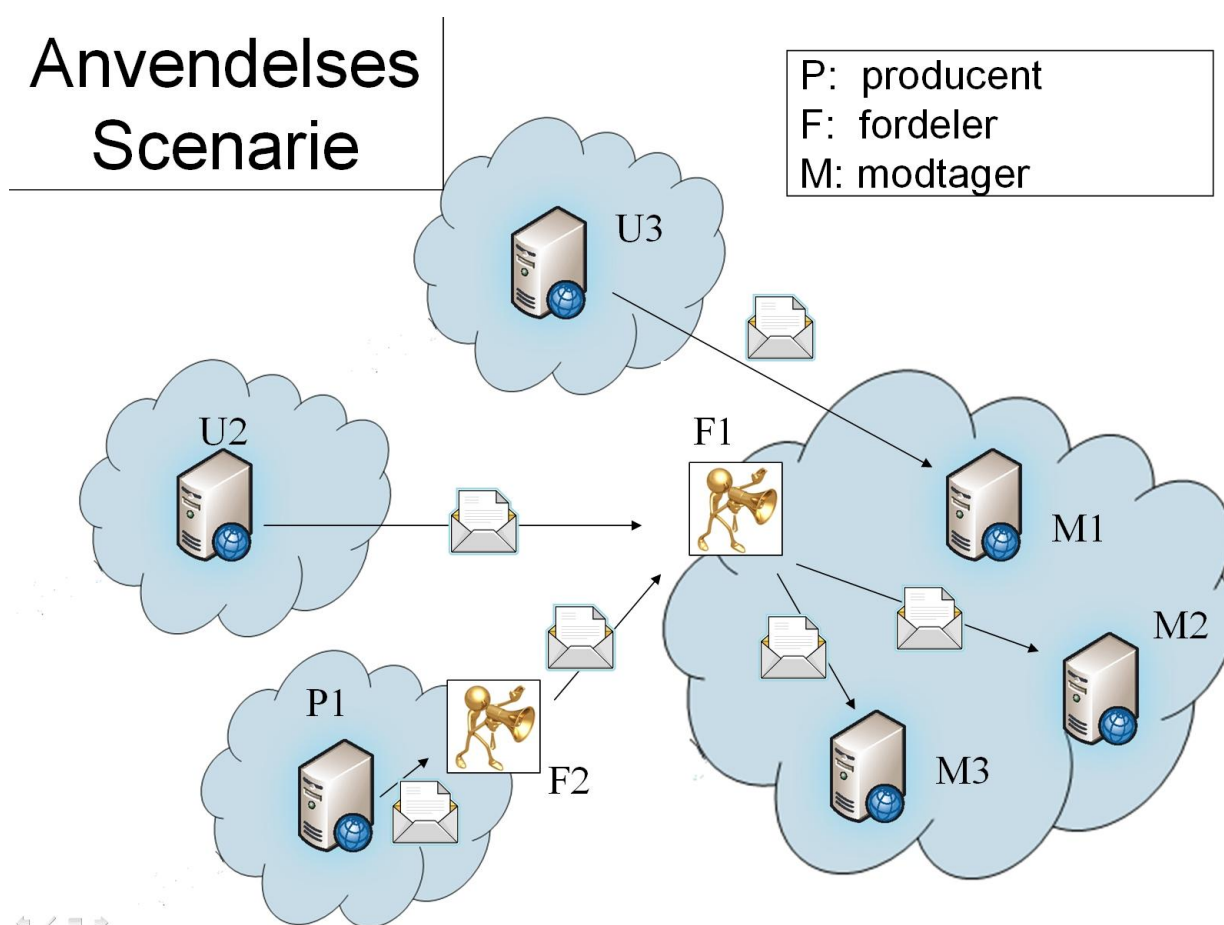
Typen af hændelsen, som er repræsenteret ved en given hændelsesbesked, og definerer implicit indholdet af en hændelse - enten ud fra hvilken specifik hændelse, der er tale om, eller indholdet af denne.

Fejlbesked

Hvis et modtagende system ikke kan behandle en hændelsesbesked på grund af syntaktisk eller semantiske problemer (forretningslogik), bør der sendes en fejlbesked retur sådan, der kan følges op på det (*Error Channel* fra [EIP]).

Anvendelses scenarie

Til at beskrive hvordan en generel hændelse forventes anvendt - og til argumentation for hvilke elementer der indgår og deres anvendelse - er der i figur 4 lavet et scenarie i form af en overordnet arkitekturtegning, som dækker flere muligheder.



Figur 5 Overordnet arkitektur for anvendelsesscenarie

Målet med den generelle hændelse er at skabe muligheden for, at infrastrukturen forholdsvis let kan håndtere hændelsesbeskeder på et logisk niveau – altså ikke primært den fysiske aflevering og adressering. Alt kan sagtens lade sig gøre uden nogen generel hændelsesbesked, men det kan i nogle tilfælde være lidt besværligt og omkosteligt.

Det er i høj grad hændelsesfordelere, som kan drage nytte af den generelle hændelsesbesked idet disse er infrastruktur komponenter, som ikke nødvendigvis anvender/’forstår’ indholdet af de specifikke hændelsesbeskeder. I figur 5 ses et helt generelt anvendelsesscenarie:

- Hændelsesproducenten, P1, detekterer forretningshændelsen og producerer en hændelse. P1 skal ikke kunne håndtere distributionen, herunder abonnementer, eventuel gensendelse af hændelsesbeskeder eller oversigter over de seneste hændelsesbeskeder.

- Hændelsesfordeleren, F2, foretager distributionen af hændelsesbeskeder, baseret på forskellige kriterier, f.eks. regler om afsenderen, hændelsesobjektet, emnet eller den specifikke hændelsestype.
- Hændelsesfordeleren, F1, har defineret abonnementer på vegne af hændelsesaftageren, M2 og M3, og fungerer som en slags applikations proxy, eventuelt med yderligere funktionalitet som beskedtransformation, filtrering, protokol transformation, håndtering af sikkerheds aspekter osv.

En generel hændelsesbesked gør også, at de som definerer de specifikke hændelsesbeskeder har en ramme at placere de generelle informationer i.

Kapitel 3. WSI-SD – Ringsted-projektet

Her er en kort gengivelse af den generelle hændelsesbesked, der blev defineret i WSISD projektet ”Hændelse - Semantik og Skema”, Høringsudkast 1.0 [Haendelse08] (se i øvrigt appendiks C).

Format for hændelsesbesked - EventStructure

Den generelle hændelsesbesked har navnet *EventStructure*, og består af følgende elementer:

1. **EventDetailStructure** – holder specifikke hændelsesbesked (forretningshændelsen). EventDetailStructure indeholder et separat skema, der fastlægger strukturen for den konkrete hændelse, f.eks. skemaet for en systemhændelse eller en returhændelse. Der kan kun inkluderes én detalje-struktur per hændelse.
2. **EventIdentificationStructure** - hændelsesID, bruges til identifikation af en hændelse.
3. **EventRegistrationDateTime** – Registreringstidspunkt: Det tidspunkt hvor en personaktør eller en systemaktør har aktiveret en funktion, der gemmer oplysninger. Forskellig fra forsendelsestidspunkt (kuverter) og virkningstidspunkt (hændelsesindhold).
4. **EventCreatorStructure** - generel information omkring skaber af en hændelse hhv. informationssystem og ligeledes information om skaberens organisation og/eller person.
5. **EventSenderIdentifier** – afsenderID, angiver unikt afsendernavn for den sidste (foregående) afsender. I tilfælde af at en hændelse kommer fra en hændelsesudbyder, har udbyderen udfyldt feltet. Når hændelsen videredistribueres af hændelsesfordeleren overskrives elementet af hændelsesfordeleren, således at det altid afspejler afsenderID’et for sidste (foregående) it-system. Elementet er beregnet til at løse problematikken omkring *echo off*.
6. **EventSubscriptionIdentifier** (valgfri) - abonnementsID, identifikation af det abonnement.

Den specifikke hændelsesbesked ligger indlejret i *EventDetailStructure* og de fleste elementer har et relativt simpelt indhold - bortset fra *EventCreatorStructure*, som har ret stor informations rigdom. Den indeholder både systemet som skabte hændelsen (*hændelsesproducent*) og udførende skaber i form af eks. organisation eller person, systeminformation, returkriterium samt evt. navnet på den udførende aktivitet (optionelt), hvori/hvorved hændelsen er opstået. Skal indeholde oplysninger om det it-system, der har dannet /skabt hændelsen og organisationen eller personen som udfører publiceringen af hændelsen. I forbindelse med kommunale fællesskaber kan udførende organisation være forskellig fra ansvarlig organisation.

Kapitel 4. Internationale standarder

Der er to konkurrerende internationale standarder for understøttelse af *web service* baseret hændelsesdistribution, som er relevante i forhold til en generel hændelsesbesked.

Web service Notification (WSN)

Standard organisationen OASIS har stået for standardiseringen af web service Notification, som har sit udspring i GRID computing. Den aktuelle version er 1.3 fra oktober 2006 [WSN].

Format for hændelsesbesked

Web service Notification giver mulighed for, at hændelsesbeskeder enten sendes direkte (*raw Notification*) eller som indlejret i en generel hændelsesbesked *wsnt:Notify* [WSNXSD1], se Appendix A for se selve skemaet.

Denne indeholder dels nogle metadata (Notification Metadata) og et element til indlejring af den specifikke hændelse. Formatet giver mulighed for at overføre flere hændelsesbeskeder på en gang. De metadata elementer, der er defineret er:

- **SubscriptionReference** – En Web Service Addressing EndpointReference til det abonnement som er associeret med hændelsesbeskeden. (Valgfri)
- **Topic** – Et emneudtryk som beskriver netop ét emne, som skal være det emne som er tilknyttet hændelsesbeskeden. Dette element beskriver det emne som matchede abonnementet, som fik hændelsesudbyderen til at sende hændelsesbeskeden til hændelsesaftageren. (Valgfri)
- **ProducerReference** – En Web Service Addressing EndpointReference til den hændelsesudbyder, som har leveret hændelsesbeskeden. (Valgfri)
- **Message** – (Krævet)

Webservices Eventing (WSE)

Et udkast til specifikationen for Webservices Eventing blev givet til W3C i 2006 med en umiddelbar forventning om, at den ville blive gjort færdig som standard. Der var desuden kortvarigt et arbejde på at få konvergens med Web Service Notification - men dette strandede. Først i 2009 er arbejdet med specifikationen blevet genoptaget under programmet med webservices Ressource Access (WS-RA) [WS-RA] og tidsplanen angiver, at arbejdsgruppen forventer, at det bliver en færdig standard i foråret af 2010. Selvom specifikationen ikke er en formel standard, har den haft nogen anvendelse

Format for hændelsesbesked

I det aktuelle arbejdsudkast [WSE] (24. September 2009) er formatet angivet med en simpel wrapper *wse:Notify*, som kort er vist i appendix B. Dette er en tilføjelse som er kommet med under den aktuelle standardisering, fra et ønske om at have et generelt format kaldet "Event Sink Wrapper" som en del af specifikationen.

Wse:Notify består i praksis i et element, som indeholder den specifikke hændelsesbesked samt muligheden for at angive xml-attributten *actionURI* til angivelse af hvilken hændelsestype der er tale om i indholdet.

Udbredelse og understøttelse

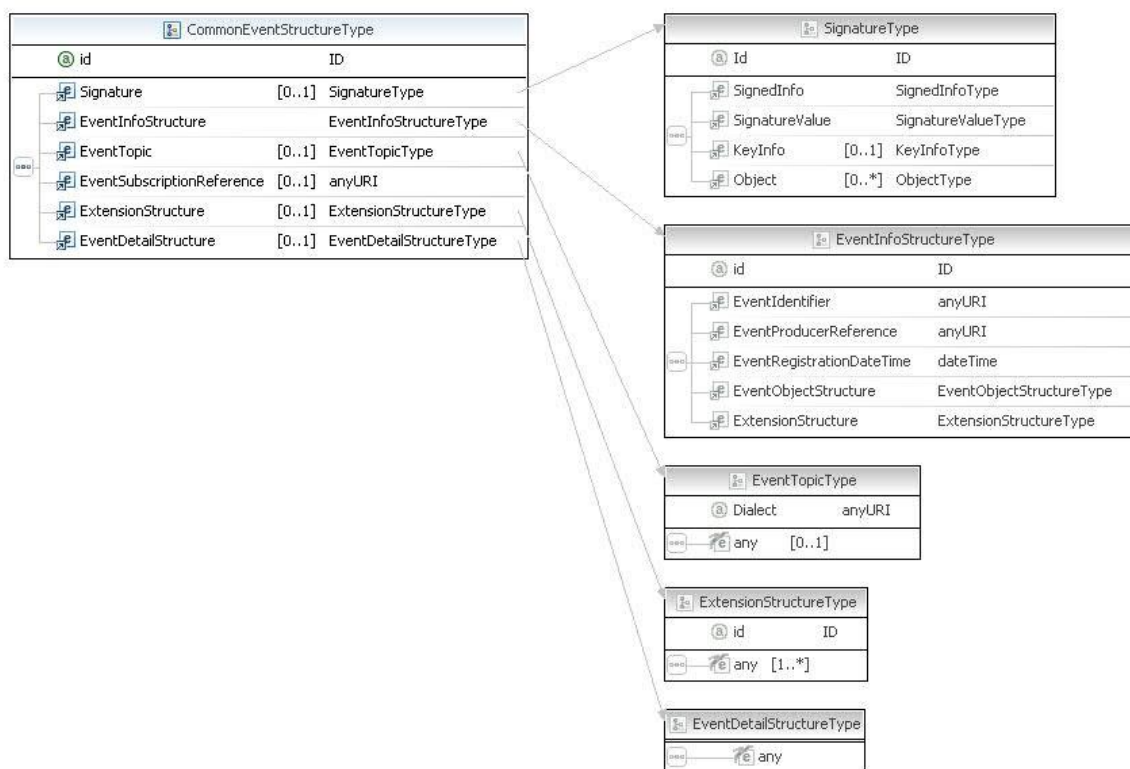
Der findes ikke noget overblik over anvendelsen af disse standarder (for WSE kommende). En umiddelbar internet søgning viser umiddelbart en ret lav grad af implementering i eksisterende produkter og hermed heller ikke nogen erfaringer i forhold til graden af interoperabilitet på disse implementeringer.

Kapitel 5. En generel hændelsesbesked

Strukturen for formatet er opbygget af fire hovedelementer:

1. *Signature* - Mulighed for angivelse af en digital signering som kan bruges til integritetscheck, både ved direkte afsendelse til en modtager eller ved distribution igennem flere hændelsesfordelere.
2. *EventInfoStructure* - Generelle egenskaber for en hændelse bestående af identifikator, registreringstidspunkt, hændelsesproducent, angivelse af hvilket objekt hændelsen omhandler og hvilken handling der er foretaget. Her er der givet mulighed for udvidelser.
3. *EventTopic* og *EventSubscriptionReference* – distributions informationer med mulighed for at angive hvilket emne det er tilknyttet og eventuelt abonnement. Der er desuden plads til udvidelser.
4. *EventDetailStructure* – den specifikke hændelse.

I figur 6 vises en grafisk præsentation.



Figur 6 Grafisk præsentation af format

Her følger en beskrivelse af de enkelte strukturer.

CommonEventStructure		
Signature	Valgfri	Mulighed for angivelse af digital signatur
EventInfoStructure	Krævet	Struktur med generelle attributter, som kan videredistribueres uden ændring, eksempelvis signeret.
EventTopic	Valgfri	Emne som hændelsen er tilknyttet. Dette kan ændres under videredistribution og er derfor

		placeret uden for <i>EventInfoStructure</i> . Kan eksempelvis være en klassifikation.
EventSubscriptionReference	Valgfri	Reference til det abonnement som giver anledning til distribution af hændelsen
ExtensionStructure	Valgfri	Dette er en åben struktur for at give mulighed for at tilføje udvidelser
EventDetailStructure	Valgfri	Kan indeholde en vilkårlig specifik hændelse, og er derfor en åben struktur

EventInfoStructure		
EventIdentifier	Krævet	Identifikator for hændelsen
EventProducerReference	Krævet	Reference til producenten af hændelsen
EventRegistrationDateTime	Krævet	Tidspunkt hvor hændelsen indtraf
EventObjectStructure	Krævet	En struktur som angiver hvilket forretningsobjekt, hændelsen omhandler
ExtensionStructure	Valgfri	Dette er en åben struktur som kan indeholde en vilkårlig specifik hændelse

EventObjectStructure		
ObjectTypeReference	Krævet	Angivelse af hvilket type objekt der er tale om, f.eks. en person, ejendom, sag.
EventObjectReference	Krævet	En reference til forretningsobjektet, dette kan enten være et pseudonym, en ressource eller en anden struktureret identifikator som f.eks. CVR-nummer
actionSchemeReference	valgfri	Giver mulighed for angivelse af udfaldsrum for handlinger som ikke er bundet til <i>@ObjectTypeReference</i>
EventObjectActionCode	Krævet	Angivelse af hvilken handling der er foretaget på forretnings objektet. Det kan være en handling der har ændret tilstanden af et objekt eller f.eks. opdatering af en relation hvor objektet ikke skifter egentlig tilstand

EventTopic		
Dialect	Valgfri	Angive hvilken type/klassifikation af emner der anvendes
Værdi	Krævet	Angive et emne som hændelsen er tilknyttet

Bemærk at repræsentationstermen '*reference*' anvendes på flere elementer og attributter. Anvendelse af dette er for at understrege at definitionen er foretages et andet sted, og at det er denne definition der refereres til. En sådan reference kan altså principielt være (til) et CVR-Nummer, en URN eller URL, hvor evt. yderligere information er tilgængeligt.

Anvendelse af beskedformatet sammen med WS-Eventing eller WS-Notification

I de to webservice standarder WS-Eventing og WS-Notification er der mulighed for enten at bruge et vilkårligt format (raw) eller anvende det beskedformat som de selv definerer (wrapped). Derfor kan den generelle hændelsesbesked beskrevet i dette dokument uden problemer bruges sammen med begge standarder som *raw*.

Eksempler

Her følger et par eksempler (ikke normative), baseret på eksisterende eksempler fra [WSISD].

Først et simpelt eksempel, hvor der blot er de generelle informationer og ikke en specifik hændelse i *EventDetailStructure*:

```
<?xml version="1.0" encoding="UTF-8"?>
<ew:CommonEventStructure
  xsi:schemaLocation="urn:oio:itst.dk:events:20090901 ITST_CommonEventStructure.xsd"
  xmlns:ew="urn:oio:itst.dk:events:20090901" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ew:EventInfoStructure>
    <ew:EventIdentifier>urn:uuid:3a556b60-3f37-11dc-a2a7-0002a5d5c51b</ew:EventIdentifier>
    <ew:EventProducerReference>urn:kmd:system:boligstoette</ew:EventProducerReference>
    <ew:EventRegistrationDateTime>2001-12-17T09:30:47.0Z</ew:EventRegistrationDateTime>
    <ew:EventObjectStructure objectTypeReference="urn:itst:oio:sogd:sag:20091001">
      <ew:EventObjectReference>URN:UUID:D80828C6-F609-42BF-94BA-
C19B15CD744C</ew:EventObjectReference>
      <ew:EventObjectActionCode>opstaaet</ew:EventObjectActionCode>
    </ew:EventObjectStructure>
  </ew:EventInfoStructure>
  <ew:EventSubscriptionReference>urn:kmd:system:boligstoette:88889999</ew:EventSubscriptionReference>
</ew:CommonEventStructure>
```

Det næste eksempel anvender næsten alle elementerne i den generelle hændelsesbesked inklusiv en signatur (som ikke er korrekt):

```
<?xml version="1.0" encoding="UTF-8"?>
<ew:CommonEventStructure
  xsi:schemaLocation="urn:oio:itst.dk:events:20090901 ITST_CommonEventStructure.xsd
http://www.w3.org/2000/09/xmldsig# http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd
http://rep.oio.dk/kl.dk/xml/schemas/2007/07/10/ http://rep.oio.dk/kl.dk/xml/schemas/2007/07/10/KL_SagshaendelseStruktur.xsd"
  xmlns:ew="urn:oio:itst.dk:events:20090901" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ds:Signature>
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
      <ds:Reference URI="# identifier_1">
        <ds:Transforms>
          <ds:Transform
            Algorithm="http://www.w3.org/2000/09/xmldsig#envelopedsignature" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <ds:DigestValue>TCDVSuG6grhyHbzhQFWFzGrxIPE=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>SGVqIHZlcmRlbG==</ds:SignatureValue>
  </ds:Signature>
  <ds:KeyInfo>
    <ds:X509Data>
      <ds:X509Certificate>SGVqIHZlcmRlbG==</ds:X509Certificate>
    </ds:X509Data>
```

```

</ds:KeyInfo>
</ds:Signature>
<ew:EventInfoStructure id="identifier_1">
  <ew:EventIdentifier>URN:UUID:086a1ef2-2897-4d98-99fc-6a348b10aad2</ew:EventIdentifier>
  <ew:EventProducerReference>http://www.ringsted.dk/system/sogd/</ew:EventProducerReference>
  <ew:EventRegistrationDateTime>2007-04-04T09:30:47.0Z</ew:EventRegistrationDateTime>
  <ew:EventObjectStructure objectTypeReference="urn:itst:oio:sogd:sag:20091001">
    <ew:EventObjectReference>URN:UUID:D80828C6-F609-42BF-94BA-
C19B15CD744C</ew:EventObjectReference>
    <ew:EventObjectActionCode
actionSchemeReference="urn:itst:oio:sogd:sag:20091001">opstaaet</ew:EventObjectActionCode>
  </ew:EventObjectStructure>
</ew:EventInfoStructure>
<ew:EventTopic
  dialect="http://www.kl.dk/Fagomrader/Administration-og-
digitalisering2/Informationsstyring/KLEmnesystematik/">urn:uuid:32.06.08</ew:EventTopic>
  <ew:EventSubscriptionReference>urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6</ew:EventSubscriptionReference>
<!-- Eksempel på en udvidelse. Er ud-kommenteret da der ikke er lavet schema for den og processing er strict

  <ew:ExtensionStructure>
    <ext:NotValidAfterDateTime xmlns:ext="urn:unknown:extension">2007-04-
05T09:30:47.0Z</ext:NotValidAfterDateTime>
  </ew:ExtensionStructure>
-->
  <ew:EventDetailStructure>
    <kl:SagshaendelseStruktur
      xmlns:cpr="http://rep.oio.dk/cpr.dk/xml/schemas/core/2005/03/18/"
      xmlns:cvr="http://rep.oio.dk/cvr.dk/xml/schemas/2005/03/22/"
      xmlns:itst2="http://rep.oio.dk/itst.dk/xml/schemas/2006/01/17/"
      xmlns:itst="http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/"
      xmlns:kl="http://rep.oio.dk/kl.dk/xml/schemas/2007/07/10/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <kl:SagshaendelseTypeStruktur>
        <kl:SagstilstandshaendelseStruktur>
          <kl:SagstilstandshaendelseSagspartStruktur
            primaerPartIndikator="true" uofficielIndikator="false">
              <kl:SagstilstandshaendelseSagspartIdentifikationStruktur>

                <cpr:PersonCivilRegistrationIdentifier>2512484916</cpr:PersonCivilRegistrationIdentifier>
                </kl:SagstilstandshaendelseSagspartIdentifikationStruktur>
                <kl:SagstilstandshaendelseSagspartRolleKode>PRI
                </kl:SagstilstandshaendelseSagspartRolleKode>
              </kl:SagstilstandshaendelseSagspartStruktur>
              <kl:SagstilstandshaendelseVirkningDatoTid>2007-11-01T00:00:00.0Z
              </kl:SagstilstandshaendelseVirkningDatoTid>
              <kl:SagstilstandKode>opstaaet</kl:SagstilstandKode>
            </kl:SagstilstandshaendelseStruktur>
          </kl:SagshaendelseTypeStruktur>
        <itst:EventCreatorOrganisationStructure>
          <itst:EventCreatorOrganisationTypeStructure>
            <cpr:AuthorityCode>0329</cpr:AuthorityCode>
          </itst:EventCreatorOrganisationTypeStructure>
          <cvr:CVRnumberIdentifier>18957981</cvr:CVRnumberIdentifier>
        </itst:EventCreatorOrganisationStructure>
        <kl:SagsidentifikationStruktur>
          <itst:UniformResourceName>URN:UUID:63ea3b6b-cd26-43e5-a97a-
884c8b558d94</itst:UniformResourceName>
        </kl:SagsidentifikationStruktur>
        <kl:KLEnummerKode>16.23.02</kl:KLEnummerKode>
        <kl:HandlingsfacetKode>G01</kl:HandlingsfacetKode>
      </kl:SagshaendelseStruktur>
    </ew:EventDetailStructure>

```

Referencer

Her er en liste af de kilder, der er refereret i dokumentet

- [WSISD] Web Service Integration af Sag & Dokument, WSISD
<http://www.itst.dk/wsisd>
- [ES] http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/itst_eventstructure.xsd
- [WSRA] Web Services Resource Access Working Group
<http://www.w3.org/2002/ws/ra/>
- [WSN] Web Service Notification (WSN), OASIS Standard oktober 2006
<http://www.oasis-open.org/specs/index.php#wsn>
- [WSNXSD1] XML Schema for Web Services Base Notification 1.3
<http://docs.oasis-open.org/wsn/b-2.xsd>
- [WST] Web Services Topics 1.3 (WS-Topics)
http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.htm
- [WSE] Web Services Eventing (WS-Eventing), W3C Working Draft 24 September 2009
<http://www.w3.org/TR/2009/WD-ws-eventing-20090924/>
- [Haendelse08] Hændelse Semantik og Skema, Høringsudkast 1.0
] <http://www.itst.dk/wsisd/haendelse/haendelse.pdf>
- [SDREFAR] ”Referencearkitektur for sags- og dokumentområdet”
K] <http://digitaliser.dk/resource/230679>
- [BPMN11] Business Process Modeling Notation, V1.1, Januar 2008
<http://digitaliser.dk/resource/323177>
- [WSEXSD] Web Services Eventing (WS-Eventing), W3C Working Draft 24 September 2009, Appendix B XML Schema
<http://www.w3.org/TR/2009/WD-ws-eventing-20090924/#Schema>
- [SOGDGEN] Generelle egenskaber for services på sags- og dokumentområdet
[http://digitaliser.dk/resource/444163/artefact/Generelle+egenskaber+for+services+p% c3% a 5+sags--og+dokumentomr%c3% a5det+--+OIO-Godkendt.pdf](http://digitaliser.dk/resource/444163/artefact/Generelle+egenskaber+for+services+p%c3%a5+sags--og+dokumentomr%c3%a5det+--+OIO-Godkendt.pdf)
- [WSBAN] Web Services Base Notification 1.3 (WS-BaseNotification)
http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.htm
- [WSBRN] Web Services Brokered Notification 1.3 (WS-BrokeredNotification)
http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.htm
- Øvrige relevante refencer
- [WSA] Web Services Addressing 1.0 – Core
<http://www.w3.org/TR/ws-addr-core/>
Web Services Addressing 1.0 - SOAP Binding
<http://www.w3.org/TR/ws-addr-soap/>

Web Services Addressing 1.0 - WSDL Binding

<http://www.w3.org/TR/ws-addr-wsdl/>

Hændelsesorienteret arkitektur

[WSISD-EDA]

http://www.itst.dk/arkitektur-og-standarder/Standardisering/datastandardisering/tvergaende-sags-og-dokumenthaendtering/projekt-ws-i-sd/ws-is-d-dokumentoversigt/loesning/haendelsesorienteret_arkitektur.pdf

[WSISD-AH]

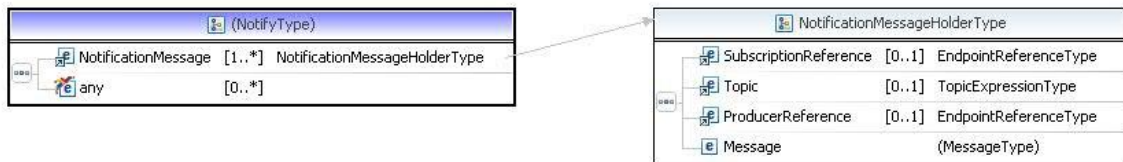
Abonnementshåndtering

<http://www.itst.dk/wsisd/forretningsregler/abonnementshaandtering.pdf>

Appendiks A – XML Schema for WSN

Dette er beskrevet i [WSBAN] og [WSNXSD1]

<http://docs.oasis-open.org/wsn/b-2.xsd>



```
<!-- ===== Notification Metadata ===== -->
<xsd:element name="SubscriptionReference"
  type="wsa:EndpointReferenceType" />
<xsd:element name="Topic"
  type="wsnt:TopicExpressionType" />
<xsd:element name="ProducerReference"
  type="wsa:EndpointReferenceType" />

<!-- ===== Message Helper Types ===== -->
<xsd:complexType name="NotificationMessageHolderType" >
  <xsd:sequence>
    <xsd:element ref="wsnt:SubscriptionReference"
      minOccurs="0" maxOccurs="1" />
    <xsd:element ref="wsnt:Topic"
      minOccurs="0" maxOccurs="1" />
    <xsd:element ref="wsnt:ProducerReference"
      minOccurs="0" maxOccurs="1" />
    <xsd:element name="Message">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:any namespace="##any" processContents="lax"
            minOccurs="1" maxOccurs="1"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="NotificationMessage"
  type="wsnt:NotificationMessageHolderType"/>

<!-- ===== Message Types for NotificationConsumer ===== -->
<xsd:element name="Notify" >
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="wsnt:NotificationMessage"
        minOccurs="1" maxOccurs="unbounded" />
      <xsd:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Appendiks B – Foreslået XML Schema til WSE

Dette kan ses i [WSEXSD] som aktuelt er <http://www.w3.org/TR/2009/WD-ws-eventing-20090924/#Schema>.

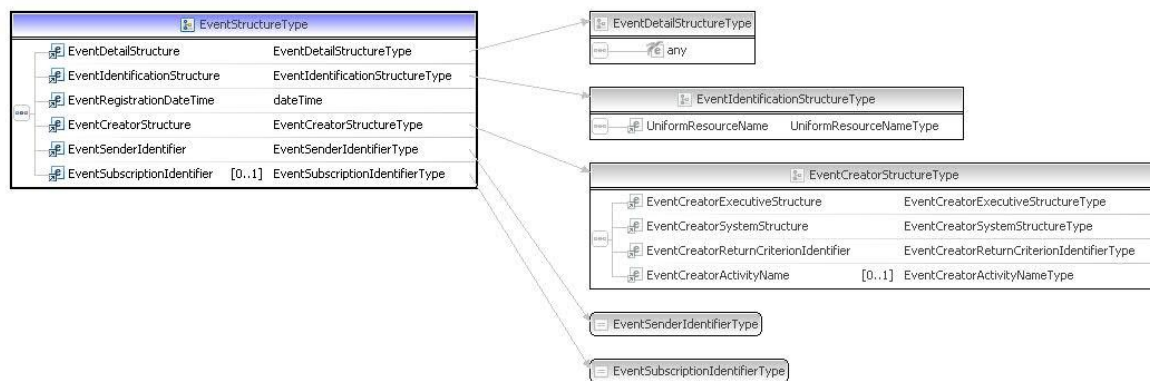
```
<xs:schema
  targetNamespace="http://www.w3.org/2009/09/ws-evt"
  xmlns:tns="http://www.w3.org/2009/09/ws-evt"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  blockDefault="#all">

  <!-- Wrapped Events -->
  <xs:complexType name="EventType"
    mixed="true">
    <xs:sequence>
      <xs:any namespace="##any"
        processContents="lax"
        minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="actionURI"
      type="xs:anyURI"
      use="optional" />
    <xs:anyAttribute namespace="##other"
      processContents="lax" />
  </xs:complexType>
  <xs:element name="Notify" type="tns:EventType" />
</xs:schema>
```

Appendiks C – Forslag fra [Haendelse08]

Dette er dels beskrevet i [Haendelse08] og selve skemaet [ES]. Det kan hentes i det der tidligere var ISBens repository.

http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/itst_eventstructure.xsd



```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:itst="http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/"
  targetNamespace="http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/"
  elementFormDefault="qualified">
  <include
    schemaLocation="http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/ITST_EventDetailStructure.xsd"/>
  <include
    schemaLocation="http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/ITST_EventIdentificationStructure.xsd"/>
  <include
    schemaLocation="http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/ITST_EventRegistrationDateTime.xsd"/>
  <include
    schemaLocation="http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/ITST_EventCreatorStructure.xsd"/>
  <include
    schemaLocation="http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/ITST_EventSenderIdentifier.xsd"/>
  <include
    schemaLocation="http://rep.oio.dk/itst.dk/xml/schemas/2007/07/06/ITST_EventSubscriptionIdentifier.xsd"/>
  <element name="EventStructure" type="itst:EventStructureType"/>
  <complexType name="EventStructureType">
    <sequence>
      <element ref="itst:EventDetailStructure"/>
      <element ref="itst:EventIdentificationStructure"/>
      <element ref="itst:EventRegistrationDateTime"/>
      <element ref="itst:EventCreatorStructure"/>
      <element ref="itst:EventSenderIdentifier"/>
      <element ref="itst:EventSubscriptionIdentifier"
  
```

minOccurs="0"/>

```
        </sequence>
    </complexType>
</schema>
```