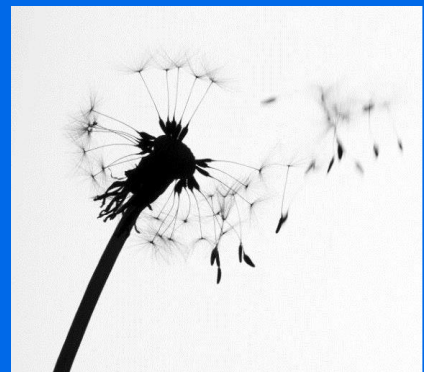




Direktoratet for
e-helse

Rammeverk for elektronisk meldingsutveksling i helsevesenet: Basert på ebXML



HIS 1037:2011, oppdatert 09/2021

Tittel:

Rammeverk for elektronisk
meldingsutveksling i helsevesenet:
Basert på ebXML

Rapport nr.:

HIS 1037:2011, oppdatert 09/2021

Utgitt av:

Direktoratet for e-helse

Utgitt:

06/2011

Kontakt:

postmottak@ehelse.no

Publikasjonen kan lastes ned fra:

www.ehelse.no

Innhold

Endringshistorikk	4
1 Definisjoner av viktige begreper.....	5
2 Innledning	7
2.1 Om ebXML.....	7
2.2 Målsetning.....	8
3 Krav og forutsetninger.....	8
3.1 Funksjonelle krav	8
3.2 Begrensninger.....	8
3.3 Funksjoner som er ivaretatt	9
4 ebXML meldingstjeneste.....	9
4.1 Bruk av ebXMLs meldingstjeneste	11
4.2 Bruk av ebXML-konvolutt	12
4.3 Overordnet prosessbeskrivelse	14
5 Funksjonell arkitektur	16
5.1 Pålitelig meldingsutveksling.....	16
5.2 Bruk av Service, Action og Role	17
5.3 Signering og kryptering av ebXML-konvolutt og payload	19
5.4 Bruk av samhandlingsavtaler	20
6 Datainnholdet i meldingskonvolutt	21
6.1 SOAP:Envelope	21
6.2 SOAP:Header	22
6.3 SOAP:Body.....	37
7 Transportkwittering og feilmelding	40
7.1 SOAP:Envelope	40
7.2 SOAP:Header	40
8 Eksempler.....	47
8.1 Signert og kryptert melding.....	47
8.2 Transportkwittering	49
9 Utvekslingsscenarier.....	51
9.1 Overføring av forretningsdokument (P1) med feil i konvolutt	52
9.2 Overføring av forretningsmelding (P ₁) med bekreftelse fra MSH og applikasjonskwittering	53
10 Referanser	54

Endringshistorikk

Dette dokumentet er oppdatert. Dette betyr at endringer som er gjort siden standarden ble fastsatt er innarbeidet i dokumentet. Endringene er beskrevet i tabellen under.

Det originale dokumentet er tilgjengelig fra [Sarepta teknisk arkiv](#) under [ebXML Rammeverk](#). Her finnes også et eget dokument som viser endringene i mer detalj.

Dato	Endring	Type
15.09.2021	<ul style="list-style-type: none">SHA-256 innføres høsten/vinteren 2021. Se informasjon om innføringsperioden hos Norsk helsenetOppdatering av krypterings- og signeringsalgoritme til SHA-256 – lagt til tekst som gjelder for overgangsfasen der både SHA-256 og SHA-1 kan benyttes	Addendum
15.12.2020	<ul style="list-style-type: none">Manifest er obligatorisk når meldingen har MIME-vedlegg. Rettet feil forekomst for eb:Manifest i tabellene i kap. 4.2, 6.3 og 6.3.1.	Errata
15.03.2020	<ul style="list-style-type: none">Rettet feil i eksempel på Service, Action og Role for forretningsprosessen Epikrise, kap. 5.2.4	Presisering
15.12.2019	<ul style="list-style-type: none">Oppdatert kap. 1 OrdforklaringerPresisert at det kun skal være en id av hver type i To og From, kap. 6.2.1Lagt til beskrivelse av hvilket kodeverk som skal benyttes for type id i elementet PartyId	Presisering
15.12.2019	<ul style="list-style-type: none">Signaturalgoritmen er i noen tilfeller feilaktig beskrevet som dsa-sha1. Den skal være rsa-sha1Organisasjonsnummer skal ikke benyttes i stedet for HER-idEn transportkvittering skal ikke inneholde RefToMessageld i MessageData-elementetAcknowledgment-elementet skal inneholde Reference, dvs. transform-elementerByttet ut eksempler i kap. 9.1 og 9.2	Errata
15.12.2019	<ul style="list-style-type: none">Messageld tillates å følge spesifikasjonen i ebMS 2.0 i tillegg til dagens praksis som er at den skal være en UUID	Addendum
06.03.2017	<ul style="list-style-type: none">Innført bruk av faste verdier for Service og Action hvis ErrorList benyttes (Feilmelding), uavhengig av om det er Error eller Warning	Errata

Dato	Endring	Type
	<ul style="list-style-type: none"> Korrigert referanse til tidligere meldingsutveksling for feilmelding og transportkwittering i kap. 6.2.1 slik at det samsvarer med tabell i kap. 4.2 Korrigert eksempel på transportkwittering i kap. 8.2 	
20.01.2017	<ul style="list-style-type: none"> Språklige forbedringer 	Presisering
03.11.2016	<ul style="list-style-type: none"> Rettet acknowledgement til US-engelsk: acknowledgement Rettet eb:TimeStamp dato- og tidsformat 	Presisering
20.11.2015	<ul style="list-style-type: none"> Innført obligatorisk bruk av HER-id i meldinger som benytter ebXML-rammeverket 	Errata

1 Definisjoner av viktige begreper

Begrep	Definisjon
ebXML-melding (Melding)	<p>MIME-entitet som inneholder en SOAP-melding med vedlegg [EBMS].</p> <p>Merk: SOAP meldingen benytter ebXML extensions til SOAP protokollen derav navnet ebXML-melding. ebMS-spesifikasjonen [EBMS] omtaler ebXML-melding som "message package".</p>
ebXML-konvolutt	<p>SOAP Envelope med ebXML-utvidelser.</p> <p>Merk: ebXML-konvolutt inneholder opplysninger om avsender, mottaker, informasjon om forretningsprosessen konvolutt er en del av og eventuell referanse til vedlegget som inneholder Payload.</p>
Meldingstjener (MSH)	<p>Programvarekomponent som håndterer meldingsformidling i tråd med ebMS-spesifikasjonen [EBMS].</p> <p>Merk: En meldingstjener kan være sammensatt av flere programvarekomponenter i et større system for meldingsformidling og kommunikasjon med eksterne og interne kommunikasjonsparter.</p>

Begrep	Definisjon
Fagsystem	<p>Informasjonssystem som mottar og/eller sender fagmeldinger.</p> <p>Merk: Slike fagsystem vil ofte være EPJ-system.</p>
Payload	<p>Forretningsdokument transportert som et vedlegg i en ebXML-melding.</p> <p>Merk: Forretningsdokumentet ligger i en separat MIME del og er i ebXML-meldingen alltid kryptert og signert.</p>
Forretningsdokument	<p>Selvstendig dokument som inneholder forretningsinformasjon beregnet på sluttbrukeren eller sluttbrukerens fagsystem.</p>
Meldingsstandard	<p>Standard som beskriver informasjonen som utveksles mellom aktører.</p> <p>Merk: En melding kan beskrives ved hjelp av en syntaksuavhengig meldingsbeskrivelse (informasjonsmodell) og/eller en syntaksspesifikk meldingsbeskrivelse.</p> <p>Eksempel: Epikrise, Sykmelding, Standard for elektronisk kommunikasjon med pleie- og omsorgstjenesten m.fl.</p>
Fagmelding	<p>Forretningsdokument som inneholder informasjon relatert til helsehjelp eller administrasjon av helsehjelp.</p>
Applikasjonskvittering	<p>Forretningsdokument som er en kvitteringsmelding som angir om en spesifikk instans av en fagmelding er mottatt og kan behandles av det mottakende fagsystemet, eller i motsatt fall angir hva som har gått feil i forbindelse med mottaket.</p> <p>Merk: Applikasjonskvittering angir om en spesifikk instans av en fagmelding er mottatt og kan behandles av det mottakende fagsystemet, eller i motsatt fall angir hva som har gått feil i forbindelse med mottaket.</p>

Begrep	Definisjon
Transportkvittering	<p>ebXML-konvolutt som benyttes som en ebXML Acknowledgment.</p> <p>Merk:</p> <p>En transportkvittering forteller avsender av ebXML-meldingen at mottakers meldingstjener har tatt imot forsendelsen, at signaturen er sjekket ok og at fagmeldingen kan leveres til mottakers fagsystem.</p>
Feilmelding	<p>ebXML-konvolutt som benyttes som en ebXML Error Signal.</p> <p>Merk:</p> <p>En feilmelding varsler i hovedsak avsender av ebXML-meldingen at denne er avvist av mottakers meldingstjener.</p> <p>ebXML Error Signal kan også benyttes for å varsle avsender av ebXML-meldingen om feil i innhold i ebXML-konvolutten, men at meldingen likevel kan behandles av mottakers meldingstjener (Warning).</p>

2 Innledning

2.1 Om ebXML

Denne rapporten beskriver en meldingstjeneste, inklusive en meldingskonvolutt i XML-syntaks [XML], basert på ebXML [EBXML] meldingstjeneste, versjon 2.0 [EBMS], som skal benyttes for å utveksle elektronisk informasjon, både standardiserte meldinger som ofte er definert som XML-meldinger og andre dokumenter som skal utveksles elektronisk mellom parter i helsevesenet. Mellomlaget er bygget på MIME og SOAP [SOAP].

ebXML-rammeverket benytter seg av en åpen standard for meldingsutveksling, ebXML Messaging Service specification (ebMS), som igjen er en utvidelse av den ledende standarden for Web-services, SOAP. ebXML utvider SOAP-standarden med tjenester for sikkerhet og pålitelighet som er nødvendige for å utveksle meldinger på en trygg måte.

Rammeverket definerer både et XML-format for en konvolutt (som kan benyttes til å pakke inn andre meldinger), og tekniske prosesser for programvare som utveksler ebXML-meldinger (som funksjonalitet for å sende meldinger på nytt). Den delen som er tatt i bruk i helsevesenet er det laveste funksjonalitetsnivået, ebXML Messaging Service (meldingshåndtering).

Meldingskonvolutten skal inneholde informasjon for å kunne:

- gi en entydig identifikasjon av og adressering til kommunikasjonspartene (avsender og mottaker) og
- identifisere selve forretningstransaksjonen (sykmelding, resept, etc.)

Meldingskonvolutten skal videre sikre mulighetene til å:

- overføre transportkvitteringer på mottak av meldinger på MSH-nivå

- rapportere eventuelle feil som oppdages ved behandling av mottatt meldingskonvolutt og ved dekrypteringen av innholdet i mottatte meldingskonvolutt

Det er et absolutt krav at selve forretningstransaksjonen skal være frikoblet fra innholdet i meldingskonvolutt, slik at det skal være mulig å kryptere bare deler av innholdet i meldingskonvolutt. Informasjon om forretningstransaksjonen skal ligge i meldingskonvolutt. Meldingskonvolutt skal kunne benyttes sammen med alle typer informasjon som skal utveksles asynkront eller synkront, så som strukturerte meldinger, bilder, lydopptak, videosekvenser, pdf-dokumenter og andre ustrukturerte dokumenter.

2.2 Målsetning

Denne rapporten dokumenterer hvilke deler av ebXML som skal benyttes i helse- og omsorgstjenesten i Norge, og hvordan den skal benyttes for å sikre standardisert og entydig bruk. Rapporten forholder seg til ebMS 2.0-standarden i ebXML, og spesifiserer hvilke element som er valgfrie og hvilke som er påkrevd for å sikre interoperabilitet mellom ulike implementasjoner. Det er også dokumentert hvilke kodeverk som skal benyttes og krav til innhold i de ulike informasjonselementene.

3 Krav og forutsetninger

3.1 Funksjonelle krav

Konvolutt skal dekke følgende funksjonelle krav, som også er i samsvar med ebXML sine krav:

- Elektroniske dokumenter skal kunne utveksles mellom parter, pakket inn i en meldingskonvolutt.
- En meldingskonvolutt kan inneholde et eller flere selvstendige dokumenter (meldinger), forutsatt at alle dokumentene er fra én avsender til én mottaker og har samme verdi på Service, Action og Role for alle dokumentene.
- En meldingskonvolutt kan inneholde et dokument med relaterte vedlegg (for eksempel en patologirekvisisjon og et relatert bilde). Hvis meldingskonvolutt inneholder flere dokumenter, kan meldingen ikke ha relaterte vedlegg/dokumenter i samme meldingsutveksling.
- Entydig avsender- og mottakerinformasjon skal ligge i meldingskonvolutt.
- Meldingskonvolutt med innhold skal kunne utveksles via flere ulike nettverksprotokoller.
- Meldingskonvolutt skal ha en global unik identifikator.
- Når meldingskonvolutt inneholder en transportkittering eller en transport feilrapport skal den inneholde identifikasjon av den meldingsutvekslingen som den er et svar på (id til konvolutt som den er et svar på).
- Meldingskonvolutt skal inneholde informasjon om når den er generert.

3.2 Begrensninger

Ikke alle elementer som inngår i ebMS er obligatoriske.

Følgende elementer av ebMS er ikke påkrevd, men kan tas i bruk etter avtale mellom partene:

- Messages Status service – meldinger for å verifisere status for en melding eller status for MSH (ping/pong-meldinger)
- Message Order – mekanismer for å sikre at meldinger mottas i riktig rekkefølge
- Multi-hop – mekanismer for å overføre meldinger via flere MSH-er

3.3 Funksjoner som er ivaretatt

Følgende funksjoner er dekket i foreliggende versjon:

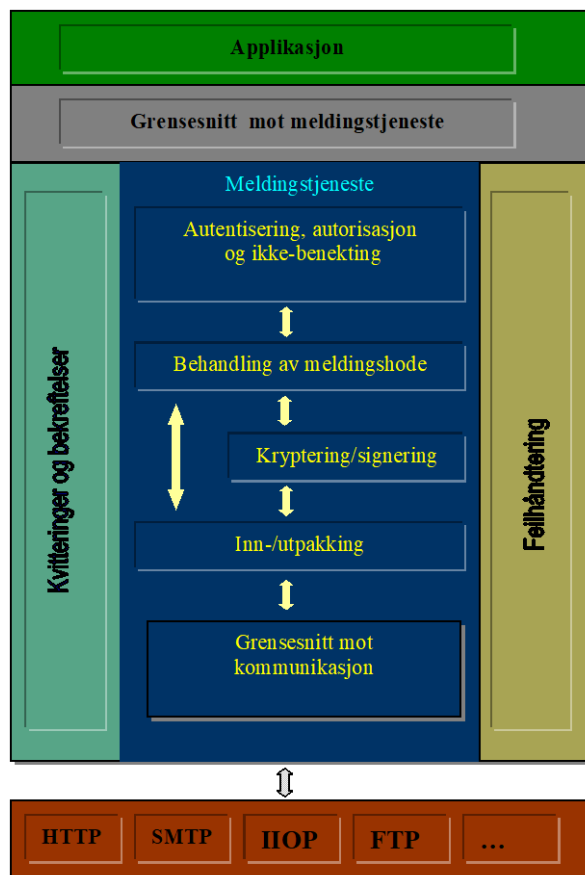
- En meldingskonvolutt for å bære informasjon om forretningsdokumenter (for eksempel en sykmelding eller en applikasjonskvittering)
- Mekanismer for hvordan eventuelle feil i forbindelse med mottak av en meldingskonvolutt rapporteres, det vil si en MSH feilmelding (ebXML Error)
- Mekanismer for å rapportere mottak av en meldingskonvolutt, det vil si en MSH-kvittering (ebXML Acknowledgment / transportkvittering)
- Signering av en meldingskonvolutt og innholdet, inklusive MSH feilmelding (ebXML Error / feilmelding), MSH kvittering eller selve forretningsdokumentet
- Støtte for pålitelig meldingsutveksling med å automatisk sende meldinger på nytt med mindre man har fått kvittering for at meldingen er mottatt, ev. varsle dersom meldingen ikke kommer frem til mottaker
- Behovet for kryptering løses ved at forretningsdokumentet krypteres vha. PKCS #7 (som for S/MIME-protokollen) før det legges inn i meldingskonvolutten

4 ebXML meldingstjeneste

Meldingstjenesten i ebXML beskriver en tjeneste for å utveksle elektroniske dokumenter mellom partnere på en standardisert, sikker og pålitelig måte uavhengig av selve kommunikasjonssystemet.

En ebXML meldingstjeneste (ebXML MSH) kan konseptuelt brytes ned i tre hovedfunksjoner (se figur 1):

1. Grensesnitt mellom applikasjon (avsendende eller mottakende applikasjon) og meldingstjenesten
2. Selve meldingstjenesten, som inneholder funksjoner for:
 - a. Autentisering, autorisasjon og ikke-benektning
 - b. Behandling av elementer i meldingshodet
 - c. Kryptering og signering
 - d. Inn- og utpakking i en helhet egnet for overføring, iht. valgt transportprotokoll
 - e. Feilhåndtering, inklusive rapportering av feil
 - f. Håndtering av kvitteringer
3. Grensesnitt mot underliggende kommunikasjonstjeneste



Figur 1 Arkitektur for en ebXML meldingstjeneste.

En ebXML meldingstjeneste kan støtte både satsvise (enveis) overføringer og dialogorienterte (spørsmål/svar) overføringer.

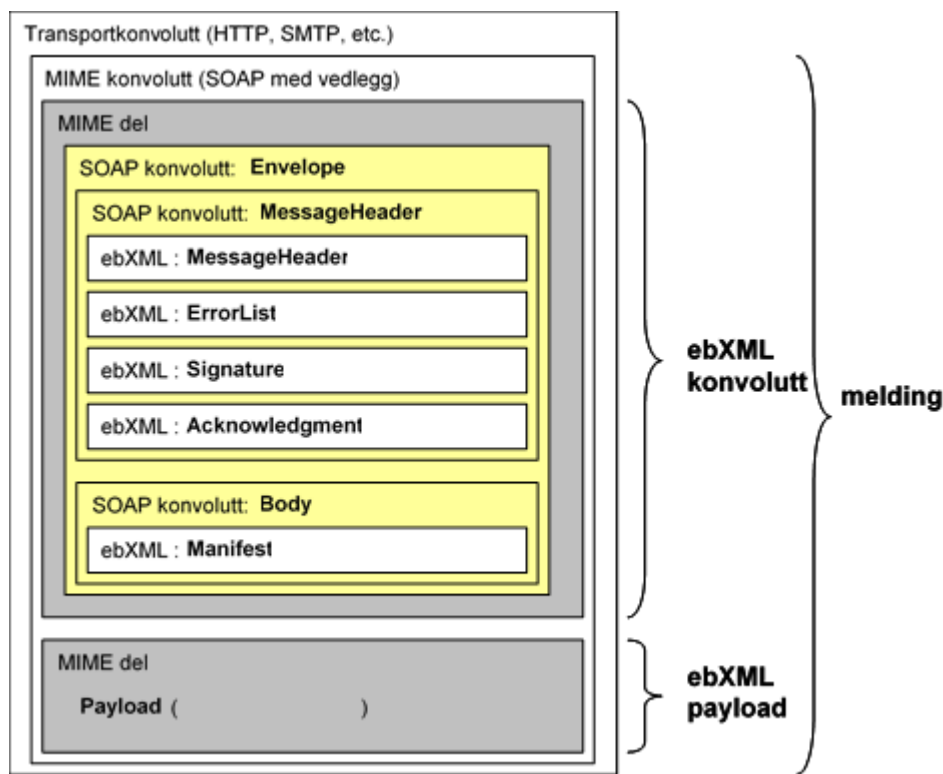
Behandlingen av meldinger i meldingstjenesten, og følgelig også de parametere som skal settes i meldingskonvolutt, reguleres av samhandlingsavtaler som er inngått mellom partene i en utveksling. Samhandlingsavtalen kan også være i form av nasjonale retningslinjer, som dette dokumentet. En samhandlingsavtale beskriver de regler som gjelder for behandling av en melding hos hver av partene. I henhold til ebXML-rammeverket kan slike samhandlingsavtaler etableres på en dynamisk måte i det selve meldingsutvekslingen skal foregå eller være fast definert i et selvstendig dokument som partene er enige om.

Strukturen i en melding som er satt opp i henhold til kravene i ebXML meldingstjenesten er vist i [EBMS].

En meldingsoverføring i henhold til ebXML består av en **transportkonvolutt** som må settes opp i henhold til spesifikasjonene som gjelder for den transporttjenesten som benyttes, og selve **meldingskonvolutt** (MIME-konvolutt). Meldingsoverføringen er uavhengig av den transporttjenesten som benyttes og omslutter det hele informasjonsinnholdet som skal overføres. Meldingskonvolutt realiseres i form av en multipart MIME-melding hvor ebXML-konvolutt (meldingskonvolutt) overføres i en MIME-del (MIME body part) og selve forretningsdokumentet (payload) i en eller flere andre MIME-del(er).

Meldingskonvolutt skal inneholde en **ebXML-konvolutt** og kan inneholde en **ebXML payload**. ebXML-konvolutt skal være satt opp i henhold til SOAP 1.1 [SOAP] og skal

overføres i en egen MIME bodypart. ebXML payload, som inneholder selve forretningsdokumentet/-dokumentene, skal overføres i en egen MIME bodypart.



Figur 2 Strukturen i en meldingsoverføring med ebXML

Det benyttes en rekke utvidelser til SOAP (SOAP extentions) for å overføre de parameterne som er nødvendig for å styre behandlingen i ebXML meldingstjenesten. Meldingskonvolutten skal inneholde et **MessageHeader**-element og kan inneholde et **Manifest** for å gi referanser til forretningsdokumentene i meldingskonvolutten. **MessageHeader**-elementet skal være pakket inn i en SOAP envelope. **Manifest**-elementet skal være pakket inn i et SOAP body-element.

En meldingskonvolutt kan brukes for å:

- gi informasjon om dokumenter som overføres mellom to parter. Dette kan være et forretningsdokument i form av en standardisert XML-melding, for eksempel en sykmelding, en henvisning eller en applikasjonskvittering
- rapportere om eventuelle feil som er oppdaget i en mottatt meldingskonvolutt
- gi en kvittering på at en melding er mottatt av MSH tjenesten

Meldingskonvolutten kan signeres direkte. Forretningsdokument signeres separat, f.eks. integrert i dokumentet (XML, PDF) eller eksternt som vedlegg. Se kapittel 8.

4.1 Bruk av ebXMLs meldingstjeneste

Denne spesifikasjonen beskriver et subsett av ebXML sin meldingstjeneste (Message Service Specification Version 2.0).

En del elementer som er frivillige i ebXML er ikke inkludert. Dette er i første rekke elementer knyttet til funksjoner for kryptering.

Meldingskonvolutt skal benyttes for å identifisere avsender og mottaker, selve meldingsutvekslingen og tidspunkt for denne, forretningsprosessen og type innhold i denne samt eventuelt identifisere entydig hvilken konversasjon denne meldingsutvekslingen tilhører.

Et **forretningsdokument** (payload) er et selvstendig dokument/brev/svar. Eksempler på forretningsdokumenter er et laboratoriesvar, en laboratorierekvisisjon, en epikrise, en journal etc. Et forretningsdokument kan ha flere vedlegg som oversendes i samme meldingskonvolutt. En rekvisisjon kan for eksempel bestå av selve rekvisisjonen (tilsvarende dagens papirskjema) med referanse til et bilde der bildet er lagt ved som et selvstendig vedlegg. Rekvisisjonen vil da bestå av to deler (bodyparts), én for selve rekvisisjonen (XML-instansmelding) og én for bildet (for eksempel en JPEG-fil).

En **konversasjon** på transportnivået (ebXML) vil bestå av selve forretningsdokumentet og tilhørende transportkittering eller feilmelding.

4.2 Bruk av ebXML-konvolutt

En forekomst av en meldingskonvolutt komponeres av et sett med informasjonselementer. Hvilke elementer som skal benyttes vil variere avhengig av den funksjon den aktuelle meldingsutvekslingen skal ivareta. Nedenstående tabell gir en fremstilling av de viktigste elementene og den sammenheng disse benyttes i. Det må påpekes at ikke alle elementene er gjengitt i denne tabellen.

Følgende notasjon benyttes for å angi krav til tilstedeværelse for det enkelte elementet:

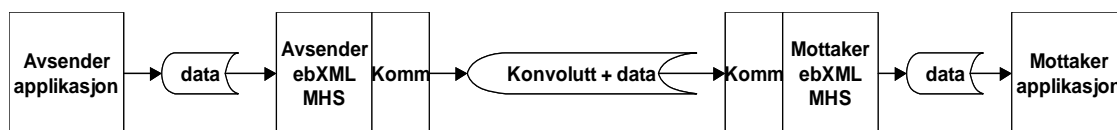
- R (Required); innebærer at elementet alltid skal benyttes (kardinalitet 1 eller 1..n). Dersom elementet ikke forekommer i en mottatt melding skal dette rapporteres som en feil.
- O (Optional); innebærer at elementet kan benyttes (kardinalitet 0..1 eller 0..n). Det er med andre ord valgfritt om elementet benyttes eller ikke.
- N (Not used); innebærer at elementet ikke skal benyttes. Dersom elementet forekommer i en mottatt melding skal dette rapporteres som en feil.

ebXML element	Funksjonelt innhold	Overføring av forretningsdokument	MSH feilmelding	MSH kvittering
MIME part	MIME-referanse til ebXML- konvolutten	R	R	R
SOAP:Envelope	Innkapsling	R	R	R
SOAP:Header	Innkapsling	R	R	R
eb:MessageHeader	Innkapsling	R	R	R
eb:From	Avsender	R	R	R
eb:To	Mottaker	R	R	R
eb:CPAId	Unik identifikasjon av samhandlingsavtale	R	R	R
eb:ConversationId	Unik identifikasjon av konversasjon	R	R	R
eb:Service	Angivelse av tjeneste	R	R	R
eb:Action	Angivelse av prosess innenfor en tjeneste	R	R	R
eb:MessageData	Innkapsling	R	R	R
eb:MessageId	Unik identifikasjon av meldingsutvekslingen i en meldingskonversasjon	R	R	R
eb:TimeStamp	Når meldingskonvolutten ble generert	R	R	R
eb:RefToMessageId	Referanse til tidligere meldingsutveksling (=eb:MessageId i mottatt melding)	N	R	N
eb:TimeToLive	Meldingsutvekslingens gyldighetsperiode	O	N	N
eb:DuplicateElimination	Angir at avsender ber mottaker om å gjennomføre duplikateliminerings	O	N	N
eb:Description	Tekstlig beskrivelse av meldingskonvoluttens innhold	O	O	O
eb:AckRequested	Angir krav om kvittering fra MSH	O	N	N
eb:Acknowledgment	Innkapsling av kvittering fra MSH	N	N	R
eb:TimeStamp	Når kvitteringen ble generert	N	N	R
eb:RefToMessageId	Referanse til den meldingskonvolutten det kvitteres for (=eb:MessageId i mottatt melding)	N	N	R
eb:ErrorList	Innkapsling av feilmeldinger	N	R	N
eb:Error	Feilmelding	N	R	N

	ds:Signature	Innkapsling av sikkerhetsinformasjon	R	R	R		
		ds:SignedInfo	Informasjon om konvoluttsignaturen	R	R	R	
			ds:Reference	Referanse til et objekt som signeres (=MIME referanse)	R	R	R
		ds:DigestValue	Signatur for objektet	R	R	R	
		ds:SignatureValue	Signatur for konvoluttet	R	R	R	
		ds:KeyInfo	Informasjon om nøkler bruk for signering av konvolutt	O	O	O	
	SOAP:Body	Innkapsling av informasjon om innhold i meldingen	R	R	R		
		eb:Manifest	Innkapsling	R	N	N	
			eb:Reference	Referanse til payload (elektronisk dokument) (=MIME-referanse)	R	N	N
			eb:Schema	Identifikasjon av den XSD som gjelder for denne payload	O	N	N
			eb:Description	Tekstlig beskrivelse av payload	O	N	N
MIME part	MIME referanse til payload (forretningsdokumentet)	R	N	N			
	Payload	Selve forretningsdokumentet	R	N	N		

4.3 Overordnet prosessbeskrivelse

Meldingstjenesten i ebXML (ebXML MSH) baserer seg på en tradisjonell lagdelt arkitektur som vist i figur 3.



Figur 3 Funksjonell arkitektur

Arkitekturen baserer seg på følgende saksgang:

1. Avsenders applikasjon klargjør informasjonen som skal overføres til mottaker, det vil si selve forretningsdokumentet (engelsk: payload) og leverer dette sammen med parametere som styrer overføringen (blant annet hvem som er mottaker) til avsenders meldingstjeneste (dette bør fortrinnsvis ligge inne i selve forretningsdokumentet for standardiserte meldinger).
2. Avsenders meldingstjeneste genererer meldingskonvoluttet og inkluderer forretningsdokumentet før overføringen starter. Dette steget vil også omfatte kryptering av forretningsdokumentet og signering av meldingskonvoluttet med meldingstjenestens nøkkel (virksomhetssertifikat).

3. Avsenders meldingstjeneste starter så selve overføringen via en av de aktuelle kommunikasjonsprotokollene (HTTP, SMTP, etc. – Komm i figuren) for å overføre meldingskonvoluttet inkludert forretningsdokumentet til mottaker. Mottakers kommunikasjonstjeneste mottar overføringen og leverer meldingskonvoluttet inkludert forretningsinnholdet til mottakers meldingstjeneste. Dette steget vil også omfatte dekryptering av forretningsdokumentet.
4. Mottakers meldingstjeneste verifiserer virksomhetssertifikatet og sjekker datainnholdet i meldingskonvoluttet og behandler meldingskonvoluttet og forretningsdokumentet i henhold til de parametere som er spesifisert i meldingskonvoluttet. Hvis alt er OK overføres selve forretningsdokumentet til mottakers applikasjon.

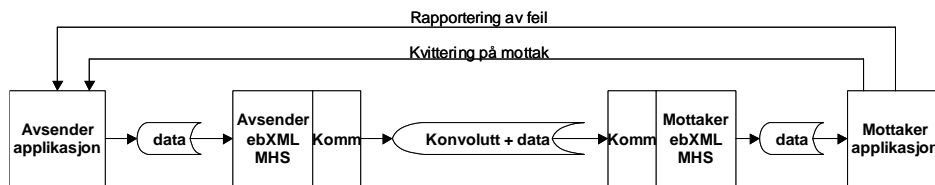
Behandlingen i mottakers meldingstjeneste skal omfatte:

- Verifikasjon av at overføringen har kommet til riktig mottaker/meldingstjeneste
 - Verifikasjon av sikkerhetsinformasjon, det vil si verifikasjon av signatur på konvoluttet (virksomhetssertifikat)
 - Generere og sende kvittering (acknowledgment) tilbake til avsenders meldingstjeneste på at overføringen har kommet frem og dekrypteringen har gått bra
 - Rapportering av eventuelle feil i datainnholdet i meldingskonvoluttet, eventuelle feil ved dekrypteringen eller rapportere om funksjoner som ikke støttes tilbake til avsender
5. Mottakers applikasjon behandler selve forretningsdokumentet i henhold til gjeldende regelverk for aktuell forretningsprosess. Denne behandlingen kan omfatte:
 - Verifikasjon av sikkerhetsinformasjon, det vil si verifikasjon av signatur på selve forretningsdokumentet
 - Gi kvittering på at forretningsdokumentet har kommet frem til mottakers applikasjon tilbake til avsenders applikasjon
 - Verifisere datainnholdet i forretningsdokumentet og rapportere eventuelle feil tilbake til avsenders applikasjon

Sett fra meldingstjenesten vil meldinger som inneholder applikasjonskvitteringer eller andre feilrapporter fra applikasjon i utgangspunktet bli behandlet som et hvilket som helst annet forretningsdokument.



Figur 4 Kvittering og feilrapportering fra meldingstjenesten (MSH Acknowledgment og Error).



Figur 5 Kvittering og feilrapportering fra applikasjon (Applikasjonskvittering).

5 Funksjonell arkitektur

Under følger en trinnvis beskrivelse for opprettelsen av en ebXML-konvolutt med payload som skal sendes over SMTP.

1. Opprett ID (CID) til payload.
Eksempel: `cid:8b97336d-f999-40a1-ab6d-8f2b37bd58dc`
2. Opprett meldingskonvolutt etter krav beskrevet i dette dokumentet, inkl. avsender- og mottakerinformasjon, referanser osv.
3. Kryptér forretningsdokumentet (ev. dokumentene) og generer signatur iht. kapittel 5.3.
4. Opprett SMTP/MIME-melding ved å legge inn meldingskonvolutt og payload som MIME-attachments. Payload skal ha korrekt CID og være kodet som base64.

5.1 Pålitelig meldingsutveksling

Ved meldingsutveksling med dette rammeverket skal ebMS 2.0s funksjoner for pålitelig meldingsutveksling benyttes (ref. ebMS 2.0 – kapittel 6 "Reliable Messaging Module").

Pålitelig meldingsutveksling gjennomføres ved at avsenders MSH ber om en transportkvittering på mottak (Acknowledgment Request). Mottakers MSH svarer med en transportkvittering (Acknowledgment Message), en meldingskonvolutt som inneholder et Acknowledgment-element med referanse til den opprinnelige meldingsutvekslingen. Hvis avsender ikke mottar en slik transportkvittering innen en gitt tidsperiode skal meldingen sendes på nytt et gitt antall ganger.

Parametrene for ventetid før resending og antall resendinger kan avtales i en CPA mellom partene, eller inngå som anbefalinger/krav for den enkelte fagmelding. Hvis det ikke er beskrevet verdier gis følgende parametere som en anbefaling:

- DuplicateElimination skal være tilstede i header i ebXML-meldingen
- Retries – antall forsøk på å sende meldingen på nytt bør være 5
- RetryInterval – tiden mellom hver resending bør være 12 timer

Hvis avsenders MSH ikke har mottatt transportkvittering innen tiden angitt av RetryInterval etter å ha forsøkt å resendere det antall ganger angitt av Retries, må MSH varsle det lokale systemet som er opphavet til forretningsdokumentet og/eller systemadministrator om situasjonen. Hvordan dette løses, er implementasjonsavhengig og kan variere fra virksomhet til virksomhet.

5.2 Bruk av Service, Action og Role

For å kunne beskrive hvilken forretningsprosess sendingen dreier seg om, hvilket ledd i prosessen man er i og hva slags rolle de enkelte partene har, benyttes det koder for **Service**, **Action** og **Role**.

En forretningsprosess består av en eller flere tjenester (**Service**) som skal kunne etableres uavhengig av hverandre.

Hver enkelt tjeneste består av et utvalg av forretningstransaksjoner som viser den reelle utvekslingen av forretningsdokumentene innen tjenesten. Den konkrete handlingen som skjer med en forretningstransaksjon kalles en **Action**. Et eksempel på en handling i "Sykmelding"-prosessen kan være "Registrere" (registrere en sykmelding) og "Bekreft" (bekrefte at en sykmelding er mottatt og kontrollert ok).

De enkelte aktørene kan inneha flere ulike roller (**Role**) i en forretningsprosess. Den aktive rollen knyttes direkte til en aktuell samhandling. Både avsender og mottaker må være definert med hver sine aktive roller for å være en "gyldig" aktør i en prosess. Flere aktører kan knyttes opp mot en og samme rolle i en prosess. F.eks. kan både Lege og Fysioterapeut inneha rollen som Behandler i "Sykmelding"-prosessen selv om de er forskjellige aktører, mens f.eks. Lege og NAV vil ha rollene som henholdsvis "Behandler" og "Saksbehandler" i den samme prosessen. Denne metoden vil vanligvis også benytte CPP/CPA.

Service, Action og Role er beskrevet i HITS 1209:2018 [SERV] for blant annet basismeldinger, PLO-meldinger og dialogmeldinger.

5.2.1 Service

Service defineres i elementet **Service** i ebMS. Service kan også spesifiseres under ServiceBinding i ebCPA hvis CPP/CPA benyttes.

Verdier for **Service** skal beskrives for den enkelte forretningsprosess/meldingsutveksling.

Eksempler på verdier er:

Reseptoppgjør	Reseptoppgjør (eResept)
Trekkopplysning	Trekkgrunnlag (Vederlagstrekk)
S-EPIKRISE	Epikrise

5.2.2 Action

Action defineres i elementet **Action** i ebMS. **Action** defineres i elementet **ThisPartyActionBinding** under *CanSend* og/eller *CanReceive* i ebCPA hvis CPP/CPA benyttes.

Verdier for **Action** skal beskrives innenfor den enkelte forretningsprosess/meldingsutveksling.

Eksempler på verdier er

Opgjørskrav	ERM18 – Reseptoppgjør (eResept)
HentPasientliste	Fastlege – Spørring på en pasients fastlege (TOKT)
Innmelding	Trekkopplysning – Innmelding av trekkgrunnlag (Vederlagstrekk)

5.2.3 Role

Role defineres i elementet **Role** i ebMS. **Role** defineres under *CollaborationRole* i ebCPPA når CPP/CPA benyttes.

Verdier for **Role** skal beskrives innenfor den enkelte forretningsprosess/meldingsutveksling.

Eksempler på verdier er:

Utleverer	Aktøren Apotek sin rolle i tjenesten Reseptoppgjør (eResept)
Rekvirent	Aktøren Lege sin rolle som Rekvirent i tjenesten eResept (eResept)
Fordringshaver	Aktøren Kommune sin rolle i tjenesten Trekkoppgjør (Vederlagstrekk)
Fastlege	Aktøren Lege sin rolle i tjenesten PasientlisteForespørsel (TOKT)
Sykmelder	Aktøren Lege sin rolle i tjenesten Sykmelde (Sykmelding)
EPIKRISSEsender	Avsender av epikrise i tjenesten S-EPIKRISSE
EPIKRISSEreceiver	Mottaker av epikrise i tjenesten S-EPIKRISSE

5.2.4 Eksempler på bruk av Service, Action, Role i forretningsprosesser

Eksempel på Service, Action og Role for forretningsprosessen Reseptoppgjør:

Service er OppgjorsKontroll:

Action	From (Sender)		To (Receiver)		Meldingstype
	From	Role	To	Role	
Oppgjorskrav	Apotek	Utleverer	NAV/HELFO	KontrollUtbetaler	ERM18.xsd
Svarmelding	NAV/HELFO	KontrollUtbetaler	Apotek	Utleverer	APPREC.xsd
Oppgjorsresultat	NAV/HELFO	KontrollUtbetaler	Apotek	Utleverer	ERM22.xsd
Svarmelding	Apotek	Utleverer	NAV/HELFO	KontrollUtbetaler	APPREC.xsd
Utbetaling	NAV/HELFO	KontrollUtbetaler	Apotek	Utleverer	ERM23.xsd
Svarmelding	Apotek	Utleverer	NAV/HELFO	KontrollUtbetaler	APPREC.xsd

Eksempel på Service, Action og Role for forretningsprosessen Epikrise:

Service	Action	Role (Sender)	Role (Receiver)	Meldingstype
S-EPIKRISSE	EPIKRISSE	EPIKRISSESender	EPIKRISSEReceiver	Epikrise
	APPREC	EPIKRISSEReceiver	EPIKRISSESender	Applikasjonskvittering

5.3 Signering og kryptering av ebXML-konvolutt og payload

Både ebXML-meldingen med forretningsdokumentet og ebXML-kvitteringen (transportkvittering) skal signeres digitalt og det er sertifikatet med *KeyUsage NonRepudiation* som skal benyttes.

Sikring av konfidensialitet ved bruk av ebXML over SMTP-protokollen skal foregå ved at det aktuelle forretningsdokumentet krypteres som et CMS-objekt med mime-headers:

```
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
```

Meldingskonvolutt skal signeres med virksomhetssertifikatet som tilhører avsender. For hvert dokument (hver payload) i meldingskonvolutt må det legges inn et ds:Reference-element med URI for dokumentet. URI skal være den samme som den xlink:href som legges inn i Manifest/Reference-elementet i meldingskonvoluttens SOAP-body.

Reference/Digest for det enkelte vedlegg skal beregnes før vedlegget kodes med base-64.

Framgangsmåte for kryptering og signering på MSH-nivå blir som følger:

1. Forretningsdokumentet krypteres som et CMS-objekt med mottakers virksomhetssertifikat.
2. Det beregnes en hash-verdi for objektet i tråd med XML-Dsig og valgt hash-algoritme. Hash-algoritmen skal være sha256. I en overgangsperiode kan sha1 fortsatt benyttes. Se informasjon om [innføringsperioden hos Norsk helsenett](#).

- a. sha256 skal identifiseres med
<DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

- b. sha1 skal identifiseres med
<DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

3. ds:SignedInfo bygges opp med CanonicalizationMethod-element og SignatureMethod-element. CanonicalizationMethod skal være:
<CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
4. ds:SignedInfo bygges opp av Reference-elementer for ebXML-konvolutt og for CMS-objektet. Reference-elementet for konvolutt må ha en URI med verdi="" og tre Transform-elementer:
<Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>

```

<Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-
19991116">
  <XPath xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">not (ancestor-
or-self::node() [@SOAP-ENV:actor="urn:oasis:names:tc:ebxml-
msg:actor:nextMSH"] | ancestor-or-self::node() [@SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"])
  </XPath>
</Transform>
<Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>

```

5. I Reference-elementet for CMS-objektet legges hash-verdien (DigestValue) som er beregnet i trinn 2. I Reference-elementet for konvolutten legges hash-verdien som beregnes etter gjennomføringen av transformasjonen og kanonikaliseringen beskrevet i trinn 3 og 4.
6. Signaturverdien beregnes basert på algoritmene i ds:SignedInfo-elementet. Signaturalgoritmen skal være rsa-sha256. I en overgangsperiode kan rsa-sha1 fortsatt benyttes. Se informasjon om [innføringsperioden hos Norsk helsenett](#).
 - a. rsa-sha256 skal identifiseres med

```

<SignatureMethod
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>

```
 - b. rsa-sha1 skal identifiseres med

```

<SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>

```
7. Signatur-elementet konstrueres med SignedInfo, KeyInfo og SignatureValue og legges inn i ebXML-headeren. Signeringsnøkkelen legges ved i KeyInfo.
8. CMS-objektet kodes med base64 og legges inn i MIME-strukturen med riktig cid og Content-Type som spesifisert ovenfor og med "Content-Encoding: base64".

5.4 Bruk av samhandlingsavtaler

I denne versjonen av spesifikasjonen er det ikke lagt opp til en dynamisk etablering av samhandlingsavtaler (Collaboration Protocol Agreement - CPA) på grunnlag av partenes samhandlingsprofiler (Collaboration Protocol Profile - CPP).

CPA kan benyttes av de aktører som har høyfrekvent meldingsutveksling eller som har behov for forhåndsgodkjenning av samhandlingsparter.

Det forutsettes derfor at de eksakte regler som gjelder for en meldingsutveksling er definert i et selvstendig dokument som partene er enig om. Innholdet i samhandlingsavtalen vil regulere hvilke av de valgfrie elementene i konvolutten som skal benyttes og det eksakte verdiinnholdet for disse elementene.

Samhandlingsavtalen kan være et dokument med nasjonale retningslinjer som vedlikeholdes sentralt, og samhandlingsavtalen er at alle parter forholder seg til denne. I de tilfeller hvor det ikke benyttes samhandlingsavtaler søker dette dokumentet å spesifisere tilstrekkelig nødvendige elementer til å sikre at kommunikasjon kan gjennomføres på tross av dette. CPAId settes da på en forhåndsdefinert måte beskrevet i dette dokumentet, jf. kapittel 6.2.1 eb:MessageHeader.

6 Datainnholdet i meldingskonvolutt

Dette kapitlet beskriver struktur og innhold til meldingskonvolutt, som er en elektronisk konvolutt i XML-format.

Meldingskonvolutt baserer seg på struktur fra SOAP versjon 1.1 (W3C Note 08 May 2000), SOAP Messages with Attachments (W3C Note 11 December 2000) samt komponenter fra OASIS ebXML Message Service Specification version 2.0 (1 April 2002). Basis for det hele er "XML Schema Recommendation 2 May 2001".

Versjonene av SOAP foreligger bare som notat, men anses som stabilt.

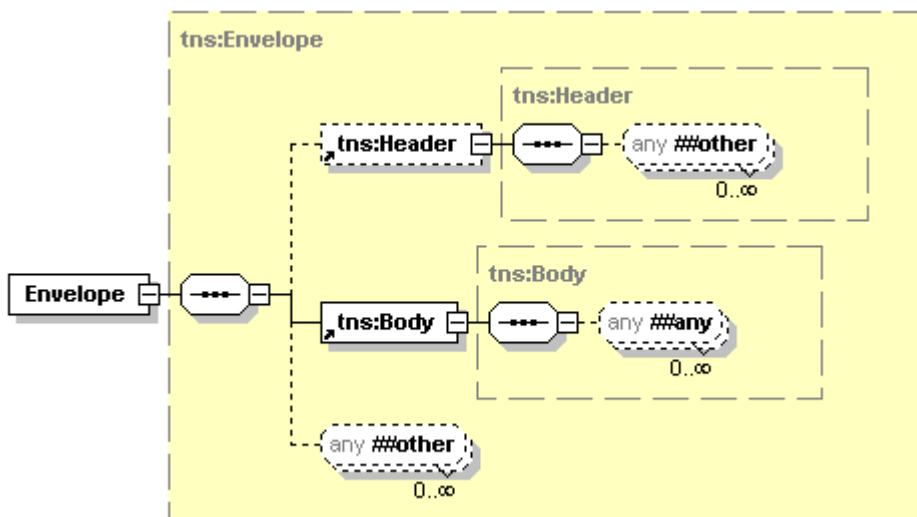
I SOAP er meldingsinnholdet spesifisert til å være inneholdt i SOAP-konvolutt. ebXML bryter dette prinsippet i og med at en baserer seg på spesifikasjonen for vedlegg med SOAP. Dermed vil meldingsinnholdet komme etter konvolutt innkapslet i en MIME konvolutt. Meldingskonvolutt her kan betraktes som et separat dokument i XML-format med informasjon om avsender, mottaker, samhandlingsprosess for en eller flere forretningsdokumenter som følger etter meldingskonvolutt. Disse forretningsdokumentene er dermed frigjort fra XML-formatet.

Alle elementer i meldingskonvolutt er beskrevet i en hierarkisk struktur som gjenspeiler seg i tittel på delkapitlene. Innholdsfortegnelsen gir derfor et godt bilde av strukturen i meldingskonvolutt.

I beskrivelsen finnes det grafiske oversikter over en del grunnleggende elementer. Disse viser ikke attributtene i XML-elementene.

XML Schema for meldingskonvolutt finnes på http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd.

6.1 SOAP:Envelope



Figur 6 Hoveddelene til SOAP-konvolutt.

Dette er rot-elementet i meldingskonvolutt.

I spesifiseringen til SOAP er kun "SOAP:Body" et påkrevd (required) element. ebXML har et stort fokus på informasjon om parter rundt en sending og har "SOAP:Header" som et påkrevd element. I denne meldingskonvolutt er "SOAP:Header" påkrevd.

I spesifiseringen til SOAP er "SOAP:Body" innkapsling av selve meldingsinnholdet (payload). ebXML benytter MIME konvolutt hvor selve meldingsinnholdet (payload) er et MIME-vedlegg. Innholdet i Body virker dermed noe overflødig, men er påkrevd og benyttes til å gi informasjon om innholdet i payload.

Det er ikke definert noen attributter utover standard XML-attributter, og elementet fungerer kun som innkapsling av meldingen.

Type	Rot node - container node
Tekstnode	Ingen
Attributter	Ingen
Lovlige ytre elementer	Ingen
Indre elementer	SOAP:Header (1) SOAP:Body (1)

6.2 SOAP:Header

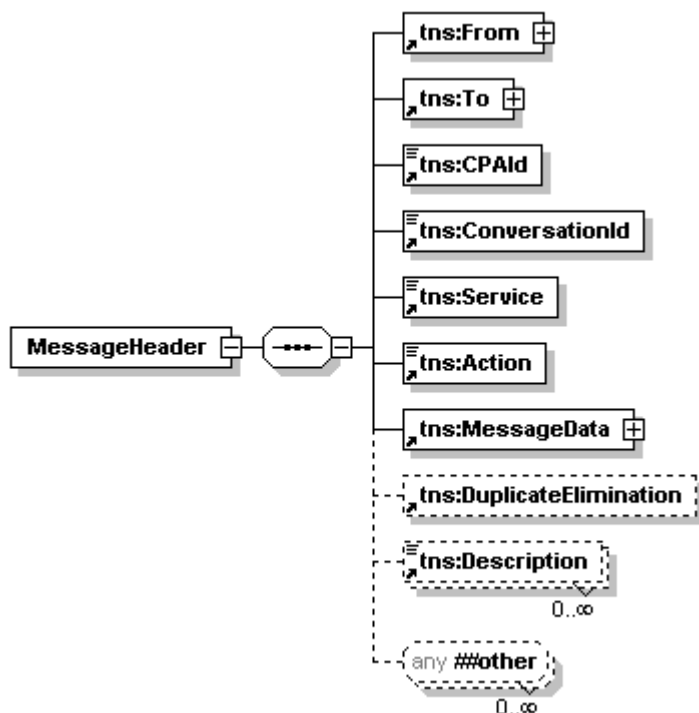
Dette elementet er en innkapsling av elementer som inneholder informasjon om utvekslingen og ikke selve meldingsinnholdet. Elementet eb:MessageHeader er påkrevd.

Elementet eb:AckRequested benyttes når en transportkvitteing for mottatt meldingsutveksling er ønsket. Denne transportkvitteingen vil kun være en kvitteing for konvolutten fra MSH.

Ønskes en kvitteing fra applikasjonen (applikasjonskvitteing), må dette defineres i en samhandlingsprosess, eller i form av krav i payload som blir overført til applikasjonen.

Type	Container node
Tekstnode	Ingen
Attributter	Ingen
Indre elementer	eb:MessageHeader (1) eb:AckRequested (0..1) ds:Signature (1)

6.2.1 eb:MessageHeader



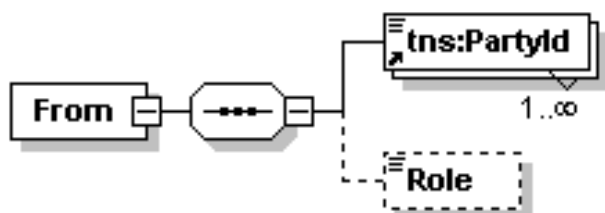
Figur 7 Grafisk oversikt over eb:MessageHeader

Dette elementet er innkapsling av en rekke elementer som inneholder informasjon om parter og krav rundt sendingen. Attributtene eb:version og soap:mustUnderstand er påkrevd. Attributtet soap:mustUnderstand skal settes til "true".

Type	Container node med attributt
Tekstnode	Ingen
Attributter	<p>eb:version</p> <p>Type String</p> <p>Innhold Fast verdi = 2.0</p> <p>Bruk Påkrevd. Definerer versjon av ebXML Message Services</p> <p>soap:mustUnderstand</p> <p>Type Boolsk</p> <p>Innhold (true 1) (false 0)</p> <p>Bruk Påkrevd. Definerer krav til at mottaker kan prosessere innholdet innkapslet av MessageHeader. Om mottaker ikke kan utføre prosesseringen må det resultere i en feilmelding</p>
Indre elementer	<p>eb:From (1)</p> <p>eb:To (1)</p> <p>eb:CPAId (1)</p> <p>eb:ConversationId (1)</p> <p>eb:Service (1)</p>

	eb:Action (1) eb:MessageData (1) eb:DuplicateElimination (0..1) eb:Description (0..1)
--	--

eb:From/eb:To



Figur 8 Grafisk oversikt over eb:From. eb:To er lik.

Beskrivelse av elementene under eb:From og eb:To er like.

Innkapsling av informasjon som identifiserer meldingsutvekslingens avsender/mottaker. Det kan kun være én avsender/mottaker i en meldingskonvolutt. Med denne begrensningen vil det ikke være mulig for en MSH å samle flere avsendere/mottakere i en konvolutt.

Elementet PartyId er påkrevd. Avsender/mottaker identifiseres med HER-id.

Type	Container node
Tekstnode	Ingen
Attributter	Ingen
Indre elementer	eb:PartyId (1..*) eb:Role (0..1)

eb:PartyId

De forskjellige typer Id som kan benyttes, er definert i attributtet "type".

Type	Tekstnode med attributt
Tekstnode	Type non-empty-string Innhold Identifikator for avsender
Attributter	eb:type Type non-empty-string Innhold Kode for type identifikator, hentet fra kodeverk 9051 Bruk Påkrevd. Definisjon av type identifikator som ligger i tekstnode
Indre elementer	Ingen

Spesifisering:

HER-id skal benyttes til å identifisere avsender/mottaker. Andre identifikatorer kan benyttes i tillegg. Oppgis det flere id-er skal det kun oppgis en av hver type.

Eksempel:

Meldingen skal sendes til Sykepleietjeneste i Trondheim kommune (HER-id=88219)

```
<eb:PartyId eb:type="HER">88219</eb:PartyId>
```

Obligatorisk identifikator:

Kode	Betydning	Kommentar
HER	HER-id	ID-nummer fra Adresseregisteret (AR)

eb:Role

Definerer hvilken rolle avsender har.

Type	Tekstnode med attributt	
Tekstnode	Type Innhold	non-empty-string / URI Identifikator for rolle avsender. Se kapittel 5.2.3 for gyldige koder.
Attributter	Ingen	
Indre elementer	Ingen	

Spesifisering:

Koder for Role skal spesifiseres for den enkelte tjenesten.

Eksempler:

```
<eb:Role>EPIKRISereceiver</eb:Role>  
<eb:Role>Rekvirent</eb:Role>
```

eb:CPAId

Unik identifisering av en protokoll for samhandling. Protokoll her innebærer et sett med regler som partene må forholde seg til. Meldingstransaksjonen identifisert ved denne meldingskonvolutten må være i overensstemmelse med denne protokollen. Dette elementet er påkrevd.

Siden CPAId angis på høyeste nivå i meldingskonvolutten må alle forretningsdokumentene og applikasjonskvitteringer som overføres i en meldingstransaksjon være underlagt den samme samhandlingsavtalen.

Innholdet i elementet bør være en URI slik det er anbefalt av OASIS. Innholdet defineres av den som oppretter CPA og må leses av den som mottar CPA og brukes i meldingsutvekslingen.

En dokumentert CPA er ikke en forutsetning for meldingsutveksling, og hvis samhandlingspartene ikke har definert en CPA på forhånd skal CPAId utformes på en forhåndsdefinert måte som beskrevet nedenfor.

Type	Container node
Tekstnode	Type non-empty-string / URI Innhold Identifikator for forretningsprosess
Attributter	Ingen
Indre elementer	Ingen

Spesifisering:

For kommunikasjonsprosesser hvor det er avtalt CPA mellom partene, skal CPAId spesifiseres på følgende måte:

```
CPAId:== <parten som lager CPA>:<valgfridd undernivå>:<ID som er unik per aktør og undernivå>
```

For kommunikasjonsprosesser hvor det ikke er avtalt CPA mellom partene, skal CPAId spesifiseres på følgende måte:

```
CPAId:== <laveste_HER-id>'_'<høyeste_HER-id>'
```

Eksempler på CPAId med URI:

```
<eb:CPAId>nav:test:XX99</eb:CPAId>
```

```
<eb:CPAId>rf:nnnnn</eb:CPAId> (CPA generert av RF for bruk i produksjons-miljøet)
```

```
<eb:CPAId>slv:nnnnn</eb:CPAId> (CPA generert av SLV for bruk i produksjons-miljøet)
```

```
<eb:CPAId>nav:nnnnn</eb:CPAId> (CPA generert av NAV for bruk i produksjons-miljøet)
```

```
<eb:CPAId>nav:test:nnnnn</eb:CPAId> (CPA generert av NAV for bruk i test-miljøet)
```

Eksempel på CPAId i uten bruk av CPA:

```
<eb:CPAId>79768_80225</eb:CPAId>
```

eb:ConversationId

Dette elementet skal angis for å gi en unik identifikasjon for et sett meldingstransaksjoner som utgjør en konkret konversasjon. Identifikator skal være unik for alle deltagende parter. Identifikator genereres av initiativtaker til konversasjonen, og skal ha samme verdi for alle meldingstransaksjoner som inngår i konversasjonen.

Type	Container node
Tekstnode	Type non-empty-string / URI Innhold Unik identifikator for konversasjon
Attributter	Ingen
Indre elementer	Ingen

Spesifisering:

ConversationID skal være en UUID.

Eksempel:

```
<eb:ConversationId>774a6ed1-76dd-4f89-9239-6ed3c1bd7caf</eb:ConversationId>
```

eb:Service

Koder for Service beskriver hvilken tjeneste det er snakk om.

Det er også mulig å benytte en URI.

Type	Tekstnode med attributt
Tekstnode	Type non-empty-string / URI Innhold Kode for tjeneste. Se kapittel 5.2.1 for eksempler på koder.
Attributter	Eb:type Type non-empty-string Innhold Kode for Service. Bruk Påkrevd identifikasjon av kodesett ved bruk av kode for å definere en tjeneste. Er normalt "kithService". Ikke påkrevd ved bruk av URI som tjenestedefinisjon.
Indre elementer	Ingen

Eksempel:

```
<eb:Service eb:type="kithService">S-EPIKRISE</eb:Service>
```

eb:Action

Identifiserer den reelle utvekslingen av et forretningsdokument innen en tjeneste, dvs. den konkrete hendelsen som skjer med forretningsdokumentet, hvilken prosess som skal behandle meldingen. Action er et påkrevd element og må ha en unik verdi innen den tjenesten den representerer. Definerings av krav til tjeneste må gjøres eksplisitt med en presis definisjon av krav til prosessen for hver kode.

MSH som mottar meldingstransaksjonen vil da kunne identifisere hvordan mottatt forretningsdokument skal håndteres videre, og om det er en kvittering eller feilmelding på applikasjonsnivå, og foreta nødvendig prosessering ut fra denne informasjonen. MSH som mottar meldingstransaksjonen må være i stand til å distribuere forretningsdokumentet til korrekt fagapplikasjon for videre behandling i avtalt format.

Type	Tekstnode
Tekstnode	Type non-empty-string Innhold Kode for Action.
Attributter	Ingen
Indre elementer	Ingen

Eksempel:

```
<eb:Action>EPIKRISE</eb:Action>
```

eb:MessageData

Innkapsling av elementer som gir mulighet for en unik identifikasjon av en XML-melding eller et annet elektronisk forretningsdokument og tidspunkt for generering av denne/dette.

eb:MessageData er et påkrevd element som det skal refereres til for når meldingskonvolutten ble generert.

Type	Container node
Tekstnode	Ingen
Attributter	Ingen
Indre elementer	eb:MessageId (1) eb:TimeStamp (1) eb:RefToMessageId (0..1) Benyttes kun for feilmelding. eb:TimeToLive (0..1)

Eksempel på innhold i en transportkvittering til en mottatt melding:

```
<eb:MessageData>
  <eb:MessageId>955536cb-fb9a-49d9-848d-
2baca7b116e1</eb:MessageId>
  <eb:TimeStamp>2006-07-11T12:45:32</eb:TimeStamp>
  <eb:TimeToLive>2006-07-11T14:45:32</eb:TimeToLive>
</eb:MessageData>
```

eb:MessageId

Skal inneholde en global unik identifikator for hver meldingstransaksjon. Hvilket applikasjonslag som skal generere denne er avhengig av bruken. I valgt arkitektur er det naturlig at MSH genererer denne identifikatoren.

Type	Tekstnode
Tekstnode	Type non-empty-string Innhold Global unik identifikator for meldingen
Attributter	Ingen
Indre elementer	Ingen

Spesifisering:

MessageId skal ifølge ebMS 2.0 være iht. RFC-2822.

MessageId kan være en UUID i nasjonal bruk siden dette har vært en anbefalt og innført praksis og en UUID er globalt unik.

Eksempler:

```
<eb:MessageId>70cf2dad-382b-4a7c-b562-6c7b26bc675f</eb:MessageId>
<eb:MessageId>20191010-171922-92495@ebxml4.nav.no</eb:MessageId>
```

eb:TimeStamp

Obligatorisk felt som benyttes for å angi tidspunkt for når meldingskonvolutten ble generert. Tidspunktet skal oppgis iht. ISO 8601 (format).

Merk:

ebXML forutsetter at tidspunkt oppgis i UTC. Hvis lokal tid benyttes er det viktig å ta med tidssone. Dette er motsatt metodikk som benyttes i meldinger der norsk tid forutsettes som basis hvis tidssone ikke er angitt.

Type	Tekstnode
Tekstnode	Type dateTime, CCYY-MM-DDThh:mm:ss Innhold Tidspunkt for generering av konvolutt
Attributter	Ingen
Indre elementer	Ingen

Eksempel:

```
<eb:Timestamp>2006-07-11T12:45:32+02:00</eb:Timestamp>
```

eb:RefToMessageld

Gir referanse til en tidligere meldingstransaksjon som denne meldingen er et svar på eller en del av en pågående konversasjon. Innholdet i dette elementet er fra eb:MessageId-elementet i meldingstransaksjonen den er et tilsvarende på.

Benyttes ikke hvis dette er første meldingstransaksjon i en konversasjon.

eb:RefToMessageId er et påkrevd element når meldingskonvolutt benyttes til feilmelding.

Merk:

I en transportkvittering ligger RefToMessageId i Acknowledgment-elementet.

Type	Tekstnode
Tekstnode	Type non-empty-string Innhold Referanse til tidligere melding
Attributter	Ingen
Indre elementer	Ingen

Eksempler:

```
<eb:RefToMessageId>ef8a2c75-b181-467b-8ebe-0f90f36d465b</eb:RefToMessageId>
```

```
<eb:RefToMessageId>20191010-171922-92495@ebxml4.nav.no</eb:RefToMessageId>
```

eb:DuplicateElimination

Indikerer at avsender ber mottaker om å utføre duplikateliminerings. Mottaker skal da kontrollere om en meldingstransaksjon med samme MessageId er mottatt tidligere. Hvis samme meldingstransaksjon er mottatt tidligere, skal payload ikke sendes videre til applikasjonslaget. Hvis meldingstransaksjonen har en AckRequested skal det gis en ny transportkvittering for meldingstransaksjonen, hvis ikke skal meldingstransaksjonen ignoreres.

Type	Tomt element
-------------	--------------

Tekstnode	Ingen
Attributter	Ingen
Indre elementer	Ingen

Spesifisering

Duplicate Elimination skal benyttes.

Eksempel

```
<eb:DuplicateElimination/>
```

eb:Description

Mulig for en tekstlig beskrivelse av meldingstransaksjonen, hensikt etc. Det er en tekst som er beregnet for lesing av mennesker.

Ikke påkrevd. Benyttes kun hvis bruk er dokumentert i forretningsprosessen. Det stilles ikke krav til mottaker om å ha rutiner for å benytte denne informasjonen til praktisk overvåkning av meldingstrafikken.

Type	Tekstnode med attributt	
Tekstnode	Type Innhold	non-empty-string Tekstlig beskrivelse
Attributter	xml:lang Type Innhold Bruk	non-empty-string to karakterer Kode to bokstaver. Default ="no" Påkrevd. Definerer språk i beskrivelsen.
Indre elementer	Ingen	

Eksempel:

```
<eb:Description xml:lang="no">Legeerklæring</eb:Description>
```

6.2.2 eb:AckRequested

Benyttes når det er krav om å sende en transportkvittering for mottatt meldingstransaksjon.

Transportkvittering skal alltid sendes i henhold til Nasjonal referansearkitektur for meldings- og dokumentutveksling [NSAM]. Avvik fra dette må være eksplisitt definert i tilhørende forretningsprosess og samhandlingsavtale og være begrunnet.

Foreløpig benyttes attributtet eb:signed utelukkende for å indikere om transportkvitteringen må være signert. Innhold i dette attributtet skal være "true" eller "1" (signert kvittering).

Type	Tomt element med attributt
Tekstnode	Ingen
Attributter	eb:version

	Type String Innhold Fast verdi = 2.0 Bruk Påkrevd. Definerer versjon av ebXML Message Services soap:mustUnderstand Type boolsk Innhold (true 1) (false 0) Bruk Påkrevd. Definerer krav til at mottaker kan prosessere innholdet innkapslet av AckRequested. Om mottaker ikke kan utføre prosesseringen må det resultere i en feilmelding soap:actor Type non-empty-string Innhold anyURI Bruk Ikke påkrevd. SOAP attributt med URI hvor rollen som skal ha kvittering er definert. eb:signed Type Boolsk Innhold (true 1) (false 0) Bruk Påkrevd. Skal være (true 1). Definerer krav til mottaker om at transportkvitteringen skal være signert
Indre elementer	Ingen

Spesifisering:

Avsender skal be om transportkvittering på mottatt melding. Avsender skal be om at transportkvitteringen er signert.

Eksempel:

```
<eb:AckRequested eb:version="2.0" soap:mustUnderstand="1" eb:signed="1"/>
```

6.2.3 ds:Signature

Signering av konvolutten utført av MSH.

Meldingskonvolutten skal signeres digitalt i overensstemmelse med [XMLDSIG]. I denne sammenhengen gir det ikke mening med mer enn en signatur per konvolutt, selv om ebXML-standarden gir mulighet for flere signaturer.

Det skal bare benyttes sertifikat som kun skal brukes til signering. Disse sertifikatene er ofte merket med "non-rep" (non-repudiation). Sertifikater benyttet til kryptering er ofte merket med "auth".

Et **ds:Signature** objekt som innkapsler ett **ds:SignedInfo** objekt skal alltid benyttes til signering av ebXML-konvolutten. Signaturen skal omfatte alle payloads som følger med i konvolutten. Det benyttes ett **ds:Reference**-element for å identifisere hvert enkelt payload som følger med konvolutten. Se eksemplet i kapittel 8 for et utfyllt signatur-element.

Type	Container node
Tekstnode	Ingen

Attributter	Ingen benyttes
Indre elementer	ds:SignedInfo (1) ds:SignatureValue (1) ds:KeyInfo (1) ds:Object (0)

ds:SignedInfo

Strukturen ds:SignedInfo inneholder kanoniseringsalgoritmen, signaturalgoritmen og referanse for hver payload i meldingskonvolutten.

Type	Container node
Tekstnode	Ingen
Attributter	ds:Id Type ID Innhold Unik identifikasjon av elementet Bruk Benyttes ikke
Indre elementer	ds:CanonicalizationMethod (1) ds:SignatureMethod (1) ds:Reference (1..*)

ds:CanonicalizationMethod

Elementet identifiserer kanoniseringsalgoritmen som skal brukes på objektet som blir identifisert i SignedInfo før beregning av signatur foretas. Kanoniseringen av XML-dokumenter sikrer at utgangspunktet for beregningen av signatur er likt under signering og validering. Anbefalt verdi er:

```
<CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
```

Type	Container node med attributt
Tekstnode	Ingen
Attributter	ds:Algorithm Type anyURI Innhold Unik identifikasjon av kanoniseringsalgoritmen Bruk Påkrevd
Indre elementer	any (0..*)

ds:SignatureMethod

Elementet beskriver metoden som brukes for å signere og validere et objekt. Algoritmen identifiserer alle kryptografiske funksjoner som er involvert i signeringen. Elementet er påkrevd og skal ha en Algorithm attributt fylt ut. SignatureMethod Algorithm skal ha følgende innhold:

```
<SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
```


I en overgangsperiode kan rsa-sha1 fortsatt benyttes. Se informasjon om [innføringsperioden hos Norsk helsenett](#).

```
<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
```

Type	Container node med attributt
Tekstnode	Ingen
Attributter	ds:Algorithm Type anyURI Innhold Unik identifikasjon av kanoniseringsalgoritmen Bruk Påkrevd
Indre elementer	HMACOutputLength (0..*) Other (0..*)

ds:Reference

Signaturen skal inneholde ett ds:Reference-element for hver payload i meldingskonvolutten. I tillegg skal det være ett Reference objekt for SOAP-konvolutten som identifiseres med URI-verdi "".

Reference-elementet for SOAP-konvolutten skal inneholde følgende to ds:Transform-elementer:

```
<Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
```

```
<Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
```

Hvert payload-element skal ha et Reference-element med URI-verdi som identifiserer payloaden. URI-en skal være Content-ID for payloaden, dvs.: cid:ref-to-msg. Payload-referanser vil normalt ikke inngå, da de i denne sammenheng typisk er krypterte CMS-objekt.

Type	Container node
Tekstnode	Ingen
Attributter	ds:Id Type ID Innhold Unik identifikasjon av elementet Bruk Benyttes ikke ds:URI Type anyURI Innhold URI til et dataelement i meldingskonvolutten (en for hver payload) Bruk Benyttes ikke ds:Type Type anyURI Innhold Bruk Benyttes ikke
Indre elementer	ds:Transforms (0..1) (påkrevd ved signering av SOAP konvolutt, benyttes ikke for payload-elementer)

	ds:DigestMethod (1) ds:DigestValue (1)
--	---

ds:Transforms

Inneholder en liste over Transform-elementer som beskriver hvordan signerende MSH fant data-objektet som ble input til ds:DigestMethod. Utdata fra siste transformasjon blir input til ds:DigestMethod-elementet. Påkrevd for SOAP-konvolutten, benyttes normalt ikke for payload-elementer.

Type	Container node
Tekstnode	Ingen
Attributter	Ingen
Indre elementer	ds:Transform (1..*)

ds:Transform

Type	Container node med attributt
Tekstnode	Ingen
Attributter	ds:Algorithm Type anyURI Innhold Navn på algoritme Bruk Påkrevd
Indre elementer	ds:any (0..*) XPath (0..1)

ds:DigestMethod

Påkrevd element som identifiserer en metode i form av en algoritme for å prosessere det identifiserte objektet og produsere en kode som identifiserer objektet entydig. Input kan være resultatet fra ds:Transform eller objektet selv.

Resultatet av prosessen er en verdi som lagres i ds:DigestValue.

DigestMethod Algorithm skal ha følgende innhold:

```
<DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
```

I en overgangsperiode kan sha1 fortsatt benyttes. Se informasjon om [innføringsperioden hos Norsk helsenett](#).

```
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
```

Type	Container node med attributt
Tekstnode	Ingen
Attributter	ds:Algorithm Type anyURI

	Innhold Bruk	Navn på algoritme Påkrevd
Indre elementer	ds:any (0..*)	

ds:DigestValue

Resultatet av prosesseringen av objektet med algoritmen angitt i ds:DigestMethod. Verdien benyttes for å validere payload-objekt/SOAP-konvolutt angitt i ds:Reference.

Type	Tekstnode uten attributt
Tekstnode	ds:DigestValueType Type base64Binary Innhold Kode etter prosessering Bruk Påkrevd
Attributter	Ingen
Indre elementer	Ingen

ds:SignatureValue

Inneholder den faktiske verdien av den digitale signaturen. Kalkulert i henhold til informasjonen i SignedInfo-elementet.

Type	Tekstnode med attributt
Tekstnode	Type: base64Binary Innhold: Kode etter prosessering
Attributter	ds:Id Type ID Innhold Unik identifikasjon av elementet Bruk Benyttes ikke
Indre elementer	ds:SignedInfo (1) ds:SignatureValue (1) ds:KeyInfo (1) ds:Object (0)

ds:KeyInfo

Et element som gjør det mulig å sende den informasjon som er nødvendig for at mottaker skal kunne få tak i nøkkel for validering av signaturen. Benyttes her for å identifisere det offentlige sertifikatet til avsenderen.

Bare ds:X509-elementet benyttes her.

Type	Container node med attributt
Tekstnode	Ingen
Attributter	ds:Id Type ID Innhold Unik identifikasjon av elementet Bruk Valgfritt

Indre elementer	ds:KeyName (0..*) ds:KeyValue (0..*) ds:RetrievalMethod (0..*) ds:X509Data (1) ds:PGPData (0..*) ds:SPKIData (0..*) ds:MgmtData (0..*)
------------------------	---

ds:X509Data

Benyttes her for å oversende det offentlige sertifikatet til avsenderen. Bare elementet X509Certificate kreves.

Type	Container node
Tekstnode	Ingen
Attributter	ingen
Indre elementer	ds:X509IssuerSerial (0..*) ds:X509SKI (0..*) ds:X509SubjectName (0..*) ds:X509Certificate (1) base64Binary ds:X509CRL (0..*) any (0..*)

ds:Object

Elementet benyttes ikke.

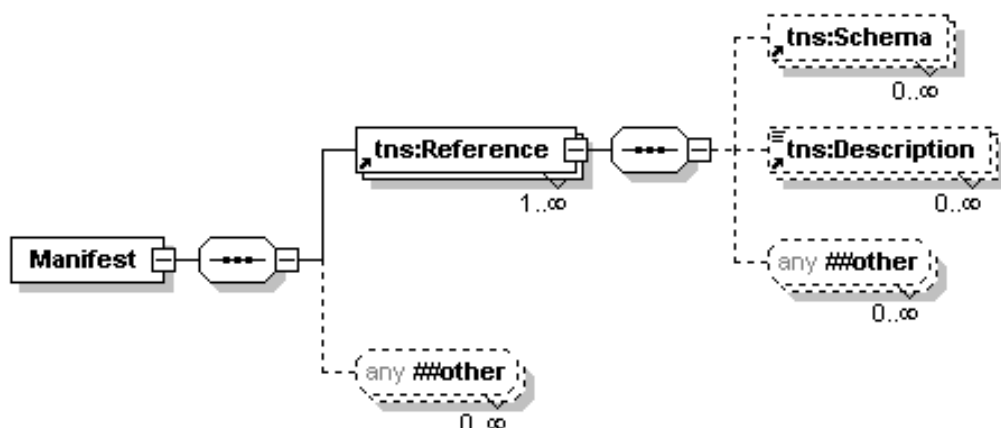
6.3 SOAP:Body

Innkapsling av informasjon om forretningsdokumentene i meldingen. Det er ikke noe payload i body slik det er spesifisert i SOAP. Meldingsdelene (payload) kommer i et eget MIME-vedlegg.

SOAP:Body skal benyttes til å gi referanser til samtlige MIME-vedlegg som følger i denne meldingstransaksjonen. Hvis meldingstransaksjonen ikke inneholder MIME-vedlegg, skal SOAP:Body være et tomt element.

Type	Container node
Tekstnode	Ingen
Attributter	Ingen
Indre elementer	<i>Transportkvittering og feilmelding:</i> Manifest (0) <i>Melding med MIME-vedlegg:</i> Manifest (1)

6.3.1 eb:Manifest



Figur 9 Grafisk oversikt over eb:Manifest. Denne ligger i SOAP:Body og inneholder referanser til MIME-vedleggene der forretningsdokumentene (payload) ligger.

Manifest er innkapsling av elementer som gir informasjon om MIME-vedleggene.

Type	Container node med attributt
Tekstnode	Ingen
Attributter	eb:version Type non-empty-string Innhold Fast verdi = 2.0 Bruk Påkrevd. Definerer versjon av ebXML Message Services
Indre elementer	eb:Reference (1..*)

eb:Reference

Dersom meldingstransaksjonen inneholder payload skal dette elementet benyttes for å gi referanse til payload (det vil si der hvor meldingstransaksjonen overfører et forretningsdokument eller en applikasjonskvittering). Referansen skal inneholde adressen til MIME-vedlegget (CID) som inneholder payloaden og skal være unik i forhold til eb:MessageId.

Elementet skal ikke benyttes dersom meldingstransaksjonen ikke inneholder payload (det vil si der hvor meldingen overfører en transportkvittering eller feilmelding fra MSH-laget).

Type	Container node med attributt
Tekstnode	Ingen
Attributter	<p>xlink:type Type String Innhold Fast verdi = "simple" Bruk Ikke påkrevd.</p> <p>xlink:href Type anyURI Innhold URI Bruk Påkrevd. URI for referanse til payload i MIME konvolutt</p> <p>xlink:role Type anyURI Innhold URI Bruk Ikke påkrevd. URI til informasjon som definerer payload</p>
Indre elementer	eb:Schema (0..*) eb:Description (0..*)

eb:Schema

Om innholdet i meldingen (payload) er av typen XML, kan en vise til hvor XML Schema, DTD eller liknende er tilgjengelig ved bruk av URI.

Denne informasjonen ligger alltid direkte inn i XML-meldingen ved bruk av attributtet `xsi:schemaLocation` for alle nasjonale meldinger. Informasjonen kan også ligge direkte i XML-meldingen med prosessskallet `<?xml-schema type="text/xsd" href="anyURI"?>`. Når informasjonen ligger i XML er det unødvendig å duplisere informasjonen i konvolutt.

Kan benyttes om XML Schema er et MIME-vedlegg i sendingen. Benyttes vanligvis ikke. Bruk skal være dokumentert i tilhørende forretningsprosess.

Type	Tom node med attributt
Tekstnode	Ingen
Attributter	eb:location Type anyURI Innhold URI Bruk Påkrevd. URI for XML Schema. Kan være URI til payload i MIME konvolutt eb:version Type String Innhold Fast verdi = 2.0 Bruk Ikke påkrevd. Definerer versjon av ebXML Message Services
Indre elementer	Ingen

eb:Description

Mulig for en tekstlig beskrivelse av meldingen, hensikt etc.

Benyttes vanligvis ikke. Bruk skal være dokumentert i tilhørende forretningsprosess.

Type	Tekstnode med attributt
Tekstnode	Type non-empty-string Innhold Tekstlig beskrivelse
Attributter	xml:lang Type non-empty-string to karakterer Innhold Kode to bokstaver. Default ="no" Bruk Påkrevd. Definerer språk som er benyttet.
Indre elementer	Ingen

7 Transportkvittering og feilmelding

Dette kapittelet beskriver en transportkvittering og feilmelding til bruk på MSH-nivå. For transportkvittering og feilmelding på MSH-nivå skal det ikke sendes noen payload.

Det benyttes i stor grad elementer fra meldingskonvolутten, derfor er det en del elementer som ikke er beskrevet i dette kapittelet, men det henvises til dokumentasjonen av meldingskonvolутten. Det er definert komponenter i tillegg for transportkvittering og for feilmelding.

7.1 SOAP:Envelope

Dette er rot-elementet i en transportkvittering og alle andre noder i kvitteringen er etterkommere av dette elementet.

I ebXML er det spesifisert at en transportkvitteringsmelding ikke trenger å ha med noen forretningsdokumenter (payload) ved kvittering og feilmelding for meldingskonvolутten. En transportkvittering eller feilmelding for meldingskonvolутten vil dermed ikke ha annet innhold enn selve meldingskonvolутten med kvitterings- eller feilmeldingsinformasjon.

Det er ikke definert noen attributter utover standard XML-attributter, og elementet fungerer kun som innkapsling av meldingen.

7.2 SOAP:Header

Dette elementet er en innkapsling av elementer som inneholder informasjon om utvekslingen og som identifiserer informasjon om melding som kvitteres.

Elementene `eb:MessageHeader`, `eb:Acknowledgment` samt `ds:Signature` er påkrevd i en transportkvittering.

Elementene `eb:MessageHeader`, `eb:ErrorList` samt `ds:Signature` er påkrevd i en feilmelding.

Indre elementer	<i>Ved kvittering:</i> eb:MessageHeader (1) eb:Acknowledgment (1) ds:Signature (1) <i>Ved feilmelding:</i> eb:MessageHeader (1) <code>eb:AckRequested (0..1)</code> eb:ErrorList (1) ds:Signature (1)
------------------------	---

7.2.1 eb:MessageHeader

Dette elementet er innkapsling av en rekke elementer som inneholder informasjon om parter og krav rundt sendingen. Attributtene `eb:version` og `soap:mustUnderstand` er påkrevd. Attributtet `soap:mustUnderstand` skal settes til "true".

Indre elementer	eb:From (1) eb:To (1) eb:CPAId (1)
------------------------	---

	eb:ConversationId (1) eb:Service (1) eb:Action (1) eb:MessageData (1) eb:Description (0..1)
--	---

eb:From

Innkapsling av informasjonselementer som identifiserer part som sendte kvitteringsmeldingen. Det vil alltid være MSH som sender transportkvittering eller feilmelding på mottatt meldingskonvolutt. Verdien i eb:From skal tilsvare verdien i eb:To i tilhørende mottatt meldingskonvolutt. Se kapittel 6 for nærmere spesifisering.

eb:PartyId

Se kapittel 6.2.1 for nærmere spesifisering.

eb:Role

Se kapittel 6.2.1 for nærmere spesifisering.

eb:To

Innkapsling av informasjon om mottaker av transportkvitteringen. Verdien i eb:To skal tilsvare verdiene i eb:From i tilhørende mottatt meldingskonvolutt. En transportkvittering har kun én mottaker.

eb:PartyId

Se kapittel 6.2.1 for nærmere spesifisering.

eb:Role

Se kapittel 6.2.1 for nærmere spesifisering.

eb:CPAId

eb:CPAId i transportkvitteringen må være identisk med eb:CPAId i meldingen som denne meldingen er et tilsvarende på.

eb:ConversationId

eb:ConversationId i transportkvitterings- og feilmeldingen må være identisk med eb:ConversationId i meldingen som denne meldingen er et tilsvarende på.

eb:Service

For transportkvittering skal service være satt til "urn:oasis:names:tc:ebxml-msg:service".

For feilmeldinger skal service settes til "urn:oasis:names:tc:ebxml-msg:service".

eb:Action

For transportkvitteringer skal Action settes til "Acknowledgment".

For feilmeldinger skal Action settes til "MessageError".

eb:MessageData

Eksempel:

```
<eb:MessageData>
  <eb:MessageId>c03e2bcd-ef6a-4507-a10c-
add0a2f17b2b</eb:MessageId>
  <eb:Timestamp>2002-07-11T12:45:32</eb:Timestamp>
  <eb:RefToMessageId>065a91a3-095b-42c7-b4cf-
f936b343cd33</eb:RefToMessageId>
</eb:MessageData>
```

eb:MessageId

Se eksempel over.

eb:TimeStamp

Se eksempel over.

eb:RefToMessageId

Gir referanse til en tidligere meldingskonvolutt som denne feilmeldingen gjelder. Innhold i dette elementet skal være identisk med innholdet i eb:MessageId-elementet i meldingen dette er tilsvar på.

NB! Påkrevd ved feilmelding men benyttes ikke ved transportkvittering. Ved transportkvittering benyttes eb:RefToMessageId definert i eb:Acknowledgment-elementet.

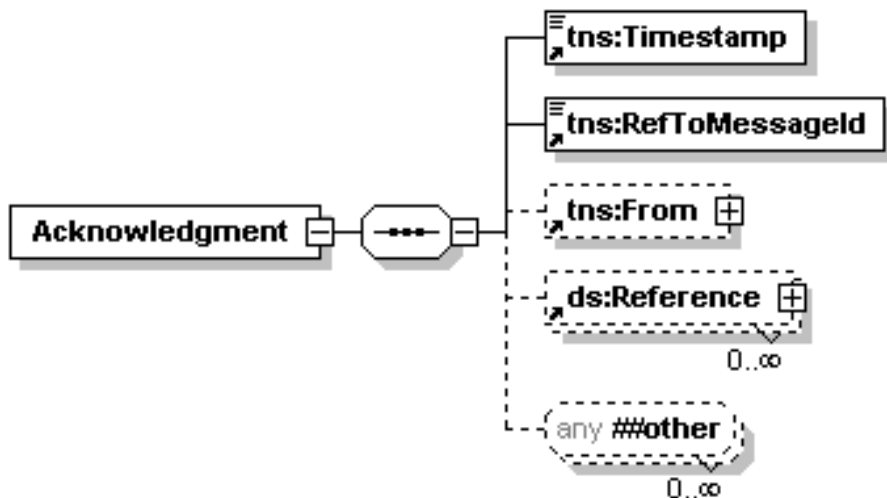
eb:Description

Tekstlig beskrivelse.

7.2.2 eb:AckRequested

Benyttes når det er krav om en transportkvittering for feilmeldingen. Skal ikke brukes i transportkvitteringer, noe som ellers ville medføre en evigvarende løkke.

7.2.3 eb:Acknowledgment



Figur 10 Oversikt over eb:Acknowledgment

Innkapsling av elementer som gir nødvendig informasjon for å identifisere hvilken melding denne MSH transportkvitteringen gjelder. Transportkvitteringen sikrer transaksjonen mellom MSH-tjenestene. Informasjon om avsender blir ikke tatt med i dette elementet.

Type	Container node med attributt
Tekstnode	Ingen
Attributter	<p>eb:version</p> <p>Type string</p> <p>Innhold Fast verdi = 2.0</p> <p>Bruk Påkrevd. Definerer versjon av ebXML Message Services</p> <p>soap:mustUnderstand</p> <p>Type boolsk</p> <p>Innhold (true 1) (false 0)</p> <p>Bruk Påkrevd. Definerer krav til at mottaker kan prosessere innholdet innkapslet av AckRequested. Om mottaker ikke kan utføre prosesseringen må det resultere i en feilmelding</p> <p>soap:actor</p> <p>Type non-empty-string</p> <p>Innhold anyURI</p> <p>Bruk Ikke påkrevd. SOAP attributt med URI hvor rollen som skal ha kvittering er definert. Default opprinnelig sender av melding.</p>
Indre elementer	<p>eb:TimeStamp (1)</p> <p>eb:RefToMessageld (1)</p> <p>eb:Reference (1)</p>

Eksempel:

```
<eb:Acknowledgment SOAP:actor="urn:oasis:names:tc:ebxml-
msg:actor:toPartyMSH" SOAP:mustUnderstand="1" eb:version="2.0">
  <eb:Timestamp>2016-05-31T10:07:55Z</eb:Timestamp>
  <eb:RefToMessageld>a741e05d-f220-4f89-a92d-
8cd111ee0749</eb:RefToMessageld>
  <ds:Reference URI="">
    <ds:Transforms>
      <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmlsig#enveloped-signature"/>
      <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-
19991116">
        <ds:XPath xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">not(ancestor-or-self::node()[@SOAP-
ENV:actor="urn:oasis:names:tc:ebxml-msg:actor:nextMSH"] | ancestor-or-
self::node()[@SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"])</ds:XPath>
        </ds:Transform>
      <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
    </ds:Transforms>
    <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmldsig#sha256"/>
    <ds:DigestValue>S9J4o3r7f1L80rDDkqa7tKuEcqQ=</ds:DigestValue>
  </ds:Reference>
</eb:Acknowledgment>
```

eb:TimeStamp

Påkrevd tidsangivelse for når transportkvitteringen ble generert. Benytter samme sone for tidsangivelse som mottatt meldingstransaksjon.

eb:RefToMessageld

Gir referanse til en tidligere meldingstransaksjon som denne meldingstransaksjonen er kvittering på. Innhold skal være identisk med innholdet i eb:MessageId-elementet i meldingskonvolutten til meldingstransaksjonen som det kvitteres for.

MSH som er mottaker av transportkvitteringen vil dermed ha tilstrekkelig informasjon til å identifisere opprinnelig sendt meldingstransaksjon.

7.2.4 eb:ErrorList

Innkapsling av elementer som skal gi informasjon om feil i meldingstransaksjonen.

Attributtene eb:version, soap:mustUnderstand og eb:highestSeverity er påkrevd.

Type	Container node med attributt
Tekstnode	Ingen
Attributter	<p>eb:version</p> <p>Type String</p> <p>Innhold Fast verdi = 2.0</p> <p>Bruk Påkrevd. Definerer versjon av ebXML Message Services</p> <p>soap:mustUnderstand</p> <p>Type Boolsk</p> <p>Innhold (true 1) (false 0)</p> <p>Bruk Påkrevd. Definerer krav til at mottaker kan prosessere innholdet innkapslet av AckRequested. Om mottaker ikke kan utføre prosesseringen må det resultere i en feilmelding</p> <p>eb:highestSeverity</p> <p>Type NMTOKEN</p> <p>Innhold Warning Error</p> <p>Bruk Påkrevd.</p> <p> "Warning" er ikke-fatal feil. Meldingen kan prosesseres.</p> <p> "Error" er fatal feil. Meldingen kan ikke prosesseres.</p>
Indre elementer	eb:Error (1..*)

Eksempel:

```
<eb:ErrorList eb:version="2.0" soap:mustUnderstand="1"
eb:highestSeverity="Error">
  <eb:Error eb:errorCode="ValueNotRecognized" eb:severity="Error"
eb:location="//eb:From">
    <eb:Description xml:lang="no">Feil
datainnhold</eb:Description>
  </eb:Error>
</eb:ErrorList>
```

eb:Error

Element for spesifisering av feilen i meldingen. Feilen defineres i form av en kode og om ønskelig en tekstlig beskrivelse i et eb:Description-element.

Type	Container node med attributt
Tekstnode	Ingen

Attributter	<p>eb:codeContext Type anyURI Innhold URI Bruk Ikke påkrevd. Definerer navnesett for feilkoder. Default = urn:oasis:names:tc:ebxml- msg:service:errors</p> <p>eb:errorCode Type non-empty-string Innhold feilkode Bruk Påkrevd. Se lovlige verdier under.</p> <p>eb:severity Type NMTOKEN Innhold Warning Error Bruk Påkrevd. "Warning" er en ikke fatal feil. Meldingen kan prosesseres. >Error" er en fatal feil. Meldingen kan ikke prosesseres.</p> <p>eb:location Type non-empty-string Innhold URI for hvor feilen oppsto (URI/XPath) Bruk Ikke påkrevd. Anbefales XPath for å identifisere node i mottatt konvolutt.</p>
Indre elementer	eb:Description (0..1)

Lovlige verdier for errorCode definert av ebXML:

Kode	Kommentar
ValueNotRecognized	Element content or attribute value not recognized
NotSupported	Element or attribute not supported
Inconsistent	Element content or attribute value inconsistent with other elements or attributes
OtherXml	Other error in an element content or attribute value
DeliveryFailure	Message Delivery Failure
TimeToLiveExpired	Message Time To Live Expired
SecurityFailure	Message Security Checks Failed
MimeProblem	URI resolve error
Unknown	Unknown Error

For ytterligere beskrivelse, se [EBMS] kapittel 4.2.3.4.

eb:Description

Tekstlig beskrivelse.

7.2.5 ds:Signature

Digital signatur for hver transportkviktering.

8 Eksempler

8.1 Signert og kryptert melding

Eksempelet viser en signert ebXML-konvolutt med et kryptert vedlegg. Innholdet i signaturelementene og de krypterte dataene er ikke gyldige.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://www.oasis-
open.org/committees/ebxml-msg/schema/envelope.xsd http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd">
```

```
<SOAP:Header>
```

```
<eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">
```

```
<eb:From>
```

```
<eb:PartyId eb:type="HER">90998</eb:PartyId>
```

```
<eb:Role>EPIKRISERsender</eb:Role>
```

```
</eb:From>
```

```
<eb:To>
```

```
<eb:PartyId eb:type="HER">91101</eb:PartyId>
```

```
<eb:Role>EPIKRISERreceiver</eb:Role>
```

```
</eb:To>
```

```
<eb:CPAId>90998_91101</eb:CPAId>
```

```
<eb:ConversationId>d5942eaf-807f-4c3c-a24d-ae6ee81a54d9</eb:ConversationId>
```

```
<eb:Service eb:type="string">S-EPIKRISER</eb:Service>
```

```
<eb:Action>EPIKRISER</eb:Action>
```

```
<eb:MessageData>
```

```
<eb:MessageId>a741e05d-f220-4f89-a92d-8cd111ee0749</eb:MessageId>
```

```
<eb:Timestamp>2016-05-25T10:34:42Z</eb:Timestamp>
```

```
</eb:MessageData>
```

```
<eb:DuplicateElimination/>
```

```

    </eb:MessageHeader>
    <eb:AckRequested SOAP:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH"
    SOAP:mustUnderstand="1" eb:signed="true" eb:version="2.0"/>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-
c14n-20010315"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>
        <ds:Reference URI="">
          <ds:Transforms>
            <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-
19991116">
              <ds:XPath xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">not(ancestor-or-self::node())[ @SOAP-
ENV:actor="urn:oasis:names:tc:ebxml-msg:actor:nextMSH"] | ancestor-or-self::node()[ @SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"]</ds:XPath>
              </ds:Transform>
            <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
          <ds:DigestValue>S9J4o3r7f1L80rDDkqa7tKuEcqQ=</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="cid:693c35d8-0841-4cc4-b7a3-b58d6b98d087">
          <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
          <ds:DigestValue>XxPiWXUWZAWYT7G2XajkOL4bxHU=</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>
        oHxcJZzk/GVesh+...y0rMgrh/eZHdKxxA==
      </ds:SignatureValue>
      <ds:KeyInfo>
        <ds:X509Data>
          <ds:X509Certificate>
            MIIFBDCCA+...sf6bLOzM5ZdRqSUbxnEXw==
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:Signature>

```



```

</SOAP:Header>
<SOAP:Body>
  <eb:Manifest eb:version="2.0">
    <eb:Reference xlink:href="cid:693c35d8-0841-4cc4-b7a3-b58d6b98d087"
xlink:type="simple"/>
  </eb:Manifest>
</SOAP:Body>
</SOAP:Envelope>

```

8.2 Transportkvittering

Dette er et eksempel på en signert transportkvittering på forrige melding. Innholdet i signaturverdiene er ikke gyldige data.

```

<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://www.oasis-
open.org/committees/ebxml-msg/schema/envelope.xsd http://www.oasis-open.org/committees/ebxml-
msg/schema/msg-header-2_0.xsd http://www.oasis-open.org/committees/ebxml-msg/schema/msg-
header-2_0.xsd">
  <SOAP:Header>
    <eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">
      <eb:From>
        <eb:PartyId eb:type="HER">91101</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="HER">90998</eb:PartyId>
      </eb:To>
      <eb:CPAId>90998_91101</eb:CPAId>
      <eb:ConversationId>d5942eaf-807f-4c3c-a24d-ae6ee81a54d9</eb:ConversationId>
      <eb:Service>urn:oasis:names:tc:ebxml-msg:service</eb:Service>
      <eb:Action>Acknowledgment</eb:Action>
      <eb:MessageData>
        <eb:MessageId>324b8888-2179-4466-b25e-92c2507dbfe7</eb:MessageId>
        <eb:Timestamp>2016-05-31T08:07:56Z</eb:Timestamp>
      </eb:MessageData>
    </eb:MessageHeader>
    <eb:Acknowledgment SOAP:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH"
SOAP:mustUnderstand="1" eb:version="2.0">
      <eb:Timestamp>2016-05-31T10:07:55Z</eb:Timestamp>
    </eb:Acknowledgment>
  </SOAP:Header>

```

```

      <eb:RefToMessageId>a741e05d-f220-4f89-a92d-
8cd111ee0749</eb:RefToMessageId>
      <eb:From>
        <eb:PartyId eb:type="HER">91101</eb:PartyId>
      </eb:From>
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
          <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-
19991116">
            <ds:XPath xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">not(ancestor-or-self::node()[@SOAP-
ENV:actor="urn:oasis:names:tc:ebxml-msg:actor:nextMSH"] | ancestor-or-self::node()[@SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"])</ds:XPath>
            </ds:Transform>
          <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <ds:DigestValue>S9J4o3r7f1L80rDDkqa7tKuEcqQ=</ds:DigestValue>
      </ds:Reference>
    </eb:Acknowledgment>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-
c14n-20010315"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>
        <ds:Reference URI="">
          <ds:Transforms>
            <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
            <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-
19991116">
              <ds:XPath xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">not(ancestor-or-self::node()[@SOAP-
ENV:actor="urn:oasis:names:tc:ebxml-msg:actor:nextMSH"] | ancestor-or-self::node()[@SOAP-
ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"])</ds:XPath>
              </ds:Transform>
            <ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
20010315"/>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

```

```

        <ds:DigestValue>UWZAWYXxPiWXT7G2XL4bxHUajkO=</ds:DigestValue>
    </ds:Reference>
    <ds:Reference URI="cid:693c35d8-0841-4cc4-b7a3-b58d6b98d087">
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <ds:DigestValue>XxPiWXUWZAWYT7G2XajkOL4bxHU=</ds:DigestValue>
    </ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
    6oUuKNm2hoghg1K.../eZHdKx==
</ds:SignatureValue>
<ds:KeyInfo>
    <ds:X509Data>
        <ds:X509Certificate>
            IBCgKCAQEAx1Y...LOzM5ZdRqSUbxn==
        </ds:X509Certificate>
    </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
</SOAP:Header>
<SOAP:Body/>
</SOAP:Envelope>

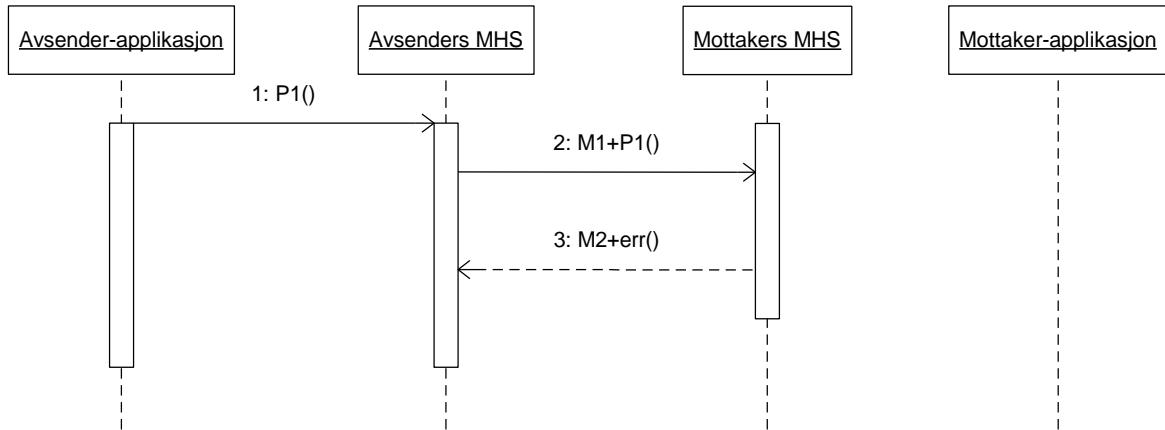
```

9 Utvekslingsscenarier

Med utgangspunkt i ovenstående saksgang kan følgende utvekslingsscenarier forekomme:

Scenario	MSH kvittering	MSH feilrapport	Applikasjonskvittering
Overføring av forretningsdokument (P ₁) med feil datainnhold i konvolutt	Nei	Ja	-
Overføring av forretningsdokument (P ₁) med bekreftelse fra MSH og applikasjonskvittering	Ja	Nei	Ja

9.1 Overføring av forretningsdokument (P₁) med feil i konvolutt



Figur 11 Overføring av forretningsdokument (P₁) med feil i konvolutt.

Figurforklaring:

P = Payload, for eksempel en XML-epikrise

M = Message, SOAP/ebMS-konvolutt

err = error, ebMS feilmelding

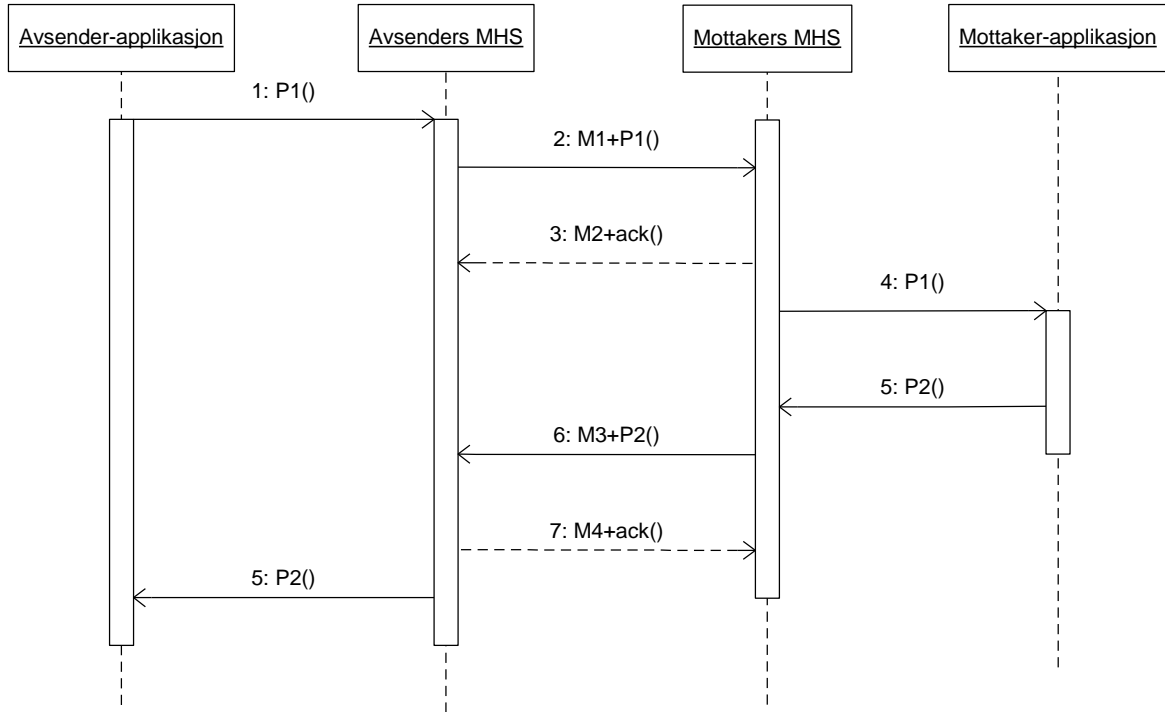
ack = acknowledgment, ebMS kvittering

Feilrapporteringen fra mottakers MSH vil skje ved at det genereres en meldingsutveksling som kun inneholder en meldingskonvolutt (M_{err}). En M_{err} vil ikke inneholde et forretningsdokument i form av en payload. Konvoluttene er selve forretningsdokumentet.

Konvoluttene M_{err} skal inneholde referanse til den mottatte konvolutt M₁ og en liste som angir hvilke feil som er oppdaget. Den opprinnelig mottatte meldingskonvolutt M₁ identifiseres ved at man oppgir referansen til det MIME-elementet som inneholdt konvoluttene. Se kapittel 7.2.1.

Det skal ikke kvitteres for en mottatt feilmelding fra MSH.

9.2 Overføring av forretningsmelding (P_1) med bekreftelse fra MSH og applikasjonskvittering



Figur 12 Overføring av forretningsmelding (P_1) med krav om bekreftelse fra MSH.

Kvitteringen fra mottakers MSH vil skje ved at det genereres en meldingsutveksling som kun inneholder en meldingskonvolutt (M_{ack}). En M_{ack} vil ikke inneholde et forretningsdokument i form av en payload. Konvolutten er selve forretningsdokumentet.

Konvolutten M_{ack} skal inneholde referanse til den mottatte meldingskonvolutten M_1 foruten et element som angir at dette er en bekreftelse. Den opprinnelig mottatte konvolutten M_1 identifiseres ved at man oppgir referansen til det MIME-elementet som inneholdt konvolutten. Se kapittel 7.2.3.

Det skal ikke kvitteres for en mottatt kvittering fra MSH.

10 Referanser

- [EBMS] OASIS og UN/CEFACT: ebXML Message Service Specification. Versjon 2.0.
http://www.oasis-open.org/committees/ebxml-msg/documents/ebMS_v2_0.pdf
- XML Schema for konvolutten:
ebXML SOAP Extension Elements Schema
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
- [EBXML] ebXML: <http://www.ebxml.org/>
- [NSAM] Referansearkitektur for meldings- og dokumentutveksling (2018), Direktoratet for e-helse
- [SERV] Service, Action og Role for basismeldinger, PLO-meldinger og dialogmeldinger (HITS 1209:2018), Direktoratet for e-helse, 2018
- [SOAP] W3C SOAP 1.1
<http://www.w3.org/TR/SOAP/>
XML Schema: <http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd>
- [SOAPAttach] SOAP Messages with Attachments, John J. Barton, Hewlett Packard Labs; Satish Thatte and Henrik Frystyk Nielsen, Microsoft, Published Oct 09 2000 2777
<http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211>
- [XML] Extensible Markup Language (XML) 1.0 (Second Edition)
<http://www.w3.org/TR/2000/REC-xml-20001006>
- [XMLS] XML Schema
Part 0: Primer: <http://www.w3.org/TR/xmlschema-0/>
Part 1: Structures: <http://www.w3.org/TR/xmlschema-1/>
Part 2: Datatypes: <http://www.w3.org/TR/xmlschema-2/>
- [XMLDSIG] Joint W3C/IETF XML-Signature Syntax and Processing specification:
<http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>