



Direktoratet for
e-helse

HISD 1171:2017

Veiledning til riktig implementasjon og bruk av ebXML som rammeverk for meldingsutveksling



Publikasjonens tittel:

HISD 1171:2017

Veiledning til riktig implementasjon og bruk av ebXML som rammeverk for meldingsutveksling

Rapportnummer

HISD 1171:2017

Utgitt:

02/2017

Utgitt av:

Direktoratet for e-helse

Kontakt:

postmottak@ehelse.no

Postadresse:

Postboks 6737 St. Olavs plass, 0130 OSLO

Besøksadresse:

Verkstedveien 1, 0277 Oslo

Tlf.: 21 49 50 70

Publikasjonen kan lastes ned på:

www.ehelse.no

Innhold

1	Innledning	5
1.1	Formål.....	5
1.2	Omfang	5
1.3	Dokumenthistorie	5
2	Referanser	6
3	Termer og Definisjoner	7
3.1	Begrepsdefinisjoner	7
3.2	Forkortelser	9
4	ebXML.....	10
4.1	Generelt	10
5	ebXML/SOAP konvolutten	13
5.1	Generelt	13
5.2	SOAP:Header	14
5.2.1	eb:MessageHeader	15
5.2.2	eb:AckRequested	19
5.2.3	eb:Acknowledgment	19
5.2.4	eb:ErrorList	21
5.2.5	ds:Signature.....	22
5.3	SOAP:Body	23
5.3.1	eb:Manifest	24
6	Payload.....	25
7	Valideringsprosessen	27
8	Transportkvittering og feilmelding	28
8.1	Generelt	28
8.2	Transportkvittering (ebXML Acknowledgment)	28
8.3	Feilmelding (ebXML Error Signal)	29
8.4	Warning (ebXML Error Signal med highestSeverity="Warning")	31
9	Scenarier	32
9.1	Transportkvittering og positiv applikasjonskvittering	33
9.1.1	Enkel konversasjon med fagmelding og applikasjonskvittering	33
9.1.2	Stor konversasjon med flere fagmeldinger og applikasjonskvitteringer	34
9.2	Transportkvittering og negativ applikasjonskvittering	36

9.3	Transportkvittering, men ikke applikasjonskvittering.....	37
9.4	Fagmelding ikke mottatt – Transportkvittering ikke sendt.....	38
9.5	Fagmelding mottatt, men transportkvittering ikke sendt	40
9.6	Fagmelding mottatt, transportkvittering er sendt men ikke mottatt	41
9.7	Mottatt feilmelding - Feil i konvolutt.....	43
10	Eksempler	45
10.1	Signert ebXML-konvolutt.....	45
10.2	Transportkvittering (ebXML Acknowledgment)	46
10.3	Feilmelding (ebXML Error med highestSeverity "Error")	47
10.4	Warning (ebXML Error med highestSeverity "Warning")	48

1 Innledning

1.1 Formål

Formålet med veiledningen er å hjelpe virksomheter og leverandører til korrekt og enhetlig forståelse, implementering og bruk av ebXML-rammeverket [2]. Dette vil bidra til å skape lik forståelse av innholdet i ebXML-meldingene. Riktig implementasjon og bruk av ebXML-rammeverket skal bidra til økt tillit til meldingsutvekslingen og en mer robust elektronisk kommunikasjon i helse- og omsorgstjenesten.

Fra sektoren og leverandørene er det gitt uttrykk for at ebXML-rammeverket, selv etter en oppdatering i 2011, ikke er entydig nok, hvilket gir rom for ulik tolkning og påfølgende risiko for feil og usikkerhet i meldingsutvekslingen. Formålet med veiledningen er derfor å redusere tolkningsrommet, slik at aktørene implementerer sin meldingsutveksling på lik måte.

1.2 Omfang

Veiledningen utfyller og presiserer innholdet i "*Rammeverk for elektronisk meldingsutveksling i helsevesenet basert på ebXML (HIS 1037:2011)*" [2].

Veiledningen har tatt utgangspunkt i utfordringer avdekket gjennom "Program for Felles infrastruktur" (FIA), spesielt prosjektet som omhandlet "Lokal Meldingsforvaltning". Veiledningen bygger også på tidligere kjente problemstillinger rundt elektronisk meldingsutveksling og bruk av ebXML som rammeverk for meldingsutveksling i helsesektoren. Merk at veiledningen ikke representerer en fullstendig implementasjonsveiledning for alle aspekter knyttet til ebXML-rammeverket og må leses sammen med den normative dokumentasjonen [2].

Denne veiledningen beskriver ikke andre krav enn de som er beskrevet i den normative dokumentasjonen for ebXML-rammeverket [2]. Veiledningen gir imidlertid en mer utfyllende beskrivelse av hvordan rammeverket skal brukes og tolkes korrekt, enn det som er beskrevet i de normative dokumentene. Veiledningen beskriver også hva som er korrekt bruk av rammeverket for tilfeller som er mangelfullt beskrevet i den normative dokumentasjonen.

1.3 Dokumenthistorie

Dato	Detaljer
14.2.2017	Dokument versjon 1.0 publisert.
14.05.2018	Rettet referanser

2 Referanser

Følgende dokumenter er nødvendige for anvendelsen av denne veiledningen.

- [1] OASIS og UN/CEFACT, «ebXML Message Service Specification. Versjon 2.0», 2002, URL: <https://ehelse.no/ebxml-message-service-specification-v20>
- [2] Direktoratet for e-helse, «HIS 1037:2011 Rammeverk for elektronisk meldingsutveksling i helsevesenet basert på ebXML», 2011, URL: <http://ehelse.no/his1037-2011>
- [3] Direktoratet for e-helse, «HISD 1172:2016 ebXML validering: Funksjonelle krav og valideringsregler», 2017, URL: <http://ehelse.no/hisd1172-2017>
- [4] Direktoratet for e-helse, «HIS 1153:2016 Standard for tjenestebasert adressering», 2016, Del 1 Generelle krav, URL: <http://ehelse.no/his1153-1-2016>
Del 2 Identifikatorer ved elektronisk samhandling, URL: <http://ehelse.no/his1153-2-2016>
Del 3 Adressering, URL: <http://ehelse.no/his1153-3-2017>
- [5] Direktoratet for e-helse, «Norm for informasjonssikkerhet Helse- og omsorgstjenesten (Normen)», 2016, URL: <http://ehelse.no/normen>
- [6] World Wide Web Consortium, «XMLDSIG - Joint W3C/IETF XML-Signature Syntax and Processing specification», 2002, URL: <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>
- [7] World Wide Web Consortium, «W3C-Draft-Simple Object Access Protocol (SOAP) v1.1», Don Box, DevelopMentor; David Ehnebuske, IBM; Gopal Kakivaya, Andrew Layman, Henrik Frystyk Nielsen, Satish Thatte, Microsoft; Noah Mendelsohn, Lotus Development Corp.; Dave Winer, UserLand Software, Inc.; W3C Note, 2000, URL: <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [8] World Wide Web Consortium, «SOAP Messages with Attachments», John J. Barton, Hewlett Packard Labs; Satish Thatte and Henrik Frystyk Nielsen, Microsoft, 2000, URL: <http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211>

3 Termer og Definisjoner

3.1 Begrepsdefinisjoner

Definisjoner av viktige begreper brukt i denne veiledningen:

Begrep	Definisjon
ebXML-melding (Melding)	MIME-entitet som inneholder en SOAP-melding med vedlegg [1]. <i>Merk:</i> SOAP meldingen benytter ebXML extensions til SOAP protokollen derav navnet ebXML-melding. ebMS-spesifikasjonen [1] omtaler ebXML-melding som "message package".
ebXML-konvolutt	SOAP Envelope med ebXML-utvidelser. <i>Merk:</i> ebXML-konvolutten inneholder opplysninger om avsender, mottaker, informasjon om forretningsprosessen konvolutten er en del av og eventuell referanse til vedlegget som inneholder Payload.
Meldingstjener	Programvarekomponent som håndterer meldingsformidling i tråd med ebMS-spesifikasjonen [1]. <i>Merk:</i> En meldingstjener kan være sammensatt av flere programvarekomponenter i et større system for meldingsformidling og kommunikasjon med eksterne og interne kommunikasjonsparter.
Fagsystem	Informasjonssystem som mottar og/eller sender fagmeldinger. <i>Merk:</i> Slike fagsystem vil ofte være et EPJ-system.
Payload	Forretningsdokument transportert som et vedlegg i en ebXML-melding. <i>Merk:</i> Forretningsdokumentet ligger i en separat MIME del og er i ebXML-meldingen alltid kryptert og signert.
Forretningsdokument	Selvstendig dokument som inneholder forretningsinformasjon beregnet på sluttbrukeren eller sluttbrukerens fagsystem. <i>Merk:</i> Et forretningsdokument er i dette dokumentet alltid enten av typen fagmelding (for eksempel en Henvising) eller applikasjonskvittering.
Meldingsstandard	Standard som beskriver informasjonen som utveksles mellom aktører.

	<p><i>Merk:</i> 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	Forretningsdokument som inneholder informasjon relatert til helsehjelp eller administrasjon av helsehjelp.
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><i>Merk:</i> 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>
Transportkvittering	<p>ebXML-konvolutt som benyttes som en ebXML Acknowledgment.</p> <p><i>Merk:</i> En transportkvittering forteller avsender av ebXML-meldingen at mottakers meldingstjener har tatt i mot 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><i>Merk1:</i> En feilmelding varsler i hovedsak avsender av ebXML-meldingen at denne er avvist av mottakers meldingstjener.</p> <p><i>Merk2:</i> 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>
Tjenestebasert adressering	Adresseringsmetode hvor det adresseres til og fra kommunikasjonsparter som representerer tjenester.

3.2 Forkortelser

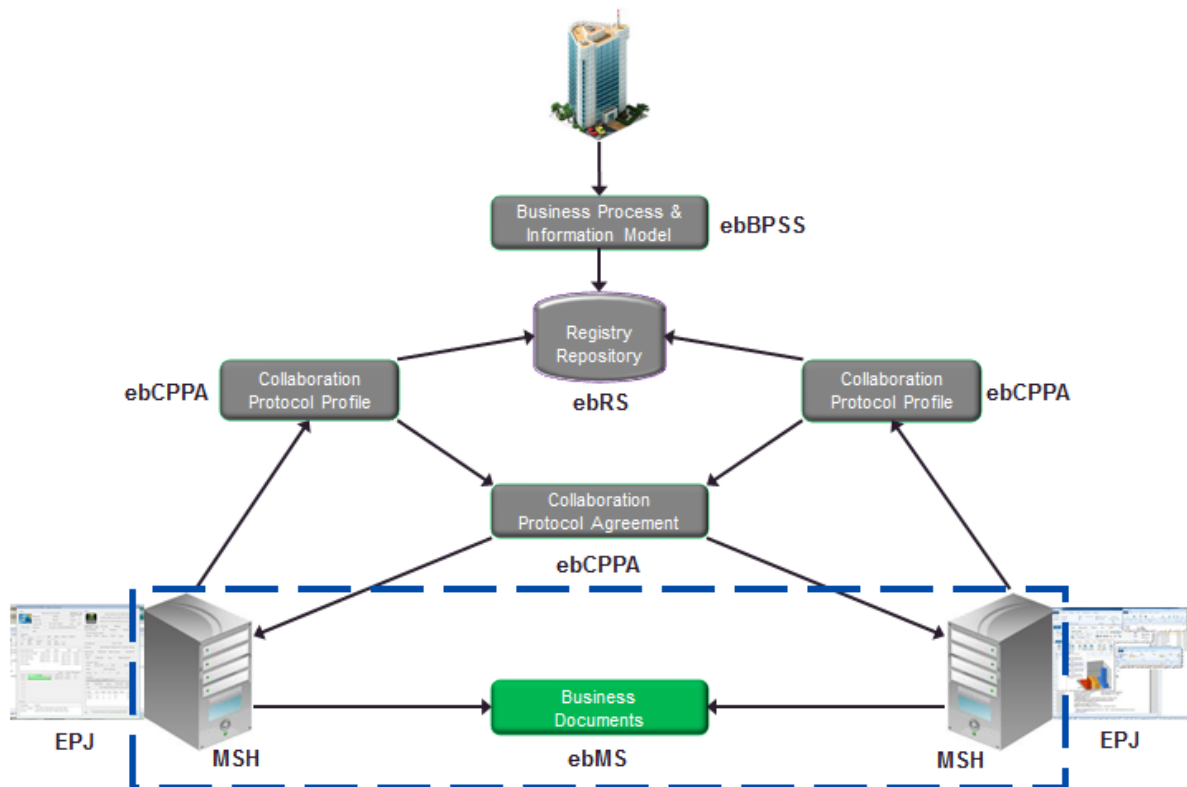
Begrep	Definisjon
ebXML	<p>electronic business eXtensible Markup Language.</p> <p>Internasjonal standard etablert i regi av UN/CEFACT og OASIS med formål å tilrettelegge for en åpen XML-basert infrastruktur til utveksling av elektronisk forretningsinformasjon.</p>
ebMS	<p>ebXML Messaging Services.</p> <p>Standard innen ebXML som beskriver meldingshåndtering og sikkerhet i forbindelse med utveksling av elektronisk forretningsinformasjon.</p>
ebCPPA	<p>ebXML Collaboration Protocol Profile and Agreement.</p> <p>Standard innen ebXML som beskriver format for definisjon og utveksling av kommunikasjonsparametere som danner grunnlag for utveksling av elektronisk forretningsinformasjon.</p>
ebBPSS	<p>ebXML Business Process Specification Schema.</p> <p>Standard innen ebXML som beskriver forretningsprosesser som benyttes i utveksling av elektronisk forretningsinformasjon.</p>
ebRS	<p>ebXML Registry Services Specification.</p> <p>Standard innen ebXML for å beskrive Registry/Repository, som er et sett av tjenester som benyttes for å lagre, søke på og dele informasjon som benyttes i og er viktig for den elektroniske samhandlingen, som for eksempel kommunikasjonsparametere og -avtaler, og annen informasjon relatert til samhandlingen (bl.a. prosesser og meldingsformater).</p>

4 ebXML

4.1 Generelt

ebXML er et rammeverk for meldingsutveksling definert av OASIS. ebXML er et omfattende rammeverk og består av en gruppe XML-baserte standarder med formål å tilrettelegge for en åpen XML-basert infrastruktur til utveksling av elektronisk forretningsinformasjon. ebXML er fastsatt som standard av ISO og består av flere deler, deriblant ebMS (ebXML Messaging Service specification). ebMS er en åpen standard for meldingsutveksling, som igjen er en utvidelse av standarden SOAP [7].

Den delen som foreløpig er tatt i bruk i den norske helsesektoren er det laveste funksjonalitetsnivået, ebXML Messaging Service (ebMS) [1].



Figur 1 viser de enkelte delene av ebXML-rammeverket (ebMS, ebCPPA, ebBPSS, ebRS)

PKI benyttes som sikkerhetsmekanisme for å beskytte meldinger basert på ebXML som sendes over helsenettet. Dette innebærer at alle virksomheter som skal utveksle informasjon må sikre sin meldingsutveksling ved å kryptere og signere meldingene som utveksles med virksomhetssertifikat. Sertifikatene skal være gyldige på tidspunktet for kryptering og signering og være utstedt av en godkjent sertifikatutsteder.

ebMS er en nøytral kommunikasjonsprotokoll som utvider SOAP-standardens tjenester for sikkerhet og pålitelighet som er nødvendige for å utveksle meldinger på en trygg måte. Standarden definerer både et meldingsformat (en konvolutt for andre meldinger), og tekniske prosesser for programvare som utveksler ebXML-meldinger.

Den norske profilen HIS 1037:2011 [2] av ebMS-spesifikasjonen [1] er obligatorisk for sikker elektronisk kommunikasjon i helsesektoren¹.

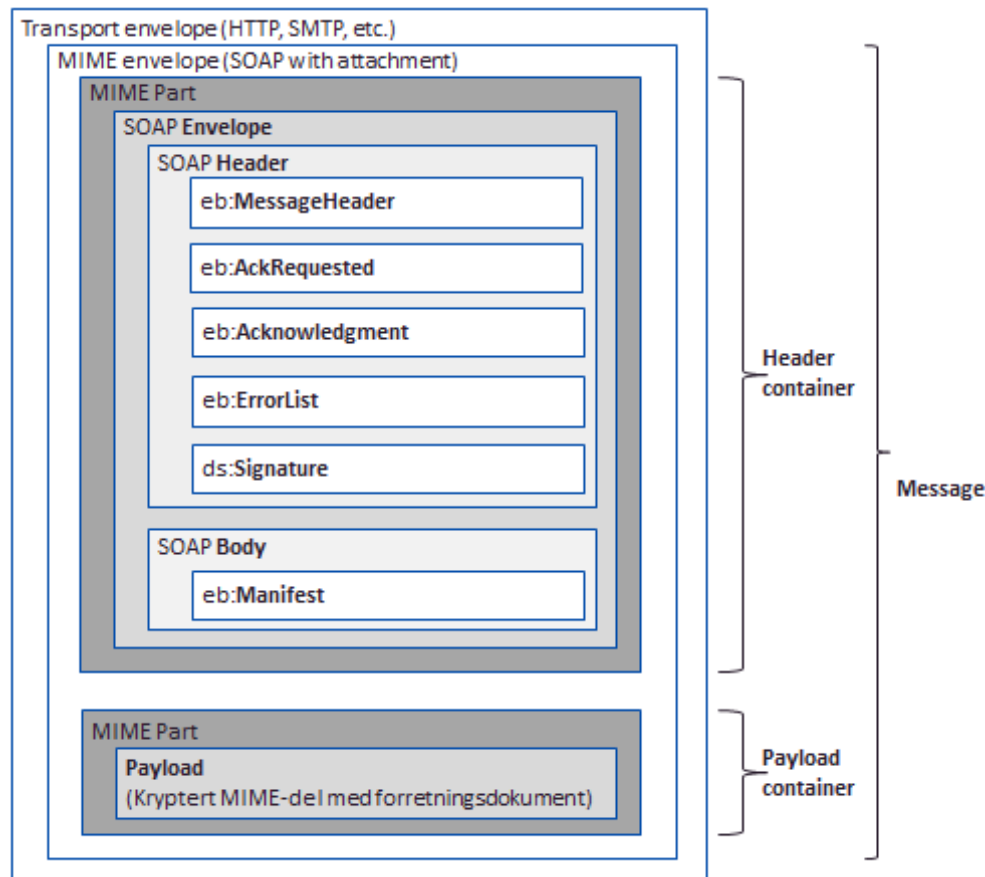
ebMS-spesifikasjonen definerer:

- et meldingsformat (ebXML-melding) som er en konvolutt for å transportere forretningsdokument
- tekniske prosesser for programvare som utveksler ebXML-meldinger, slik som funksjonalitet for å sende meldinger på nytt

En ebXML-melding består av to deler:

- 1 SOAP:Envelope
Inneholder informasjon om meldingsutvekslingen
- 2 Payload
Payload inneholder selve forretningsdokumentet som utveksles

¹ <https://ehelse.no/standarder-kodeverk-og-referansekatalog/referansekatalogen/del-4-elektronisk-samhandling>



Figur 2 viser hvordan en ebXML-melding er bygget opp med Header, Body og Payload

I henhold til HIS 1037:2011 [2] og ebMS 2.0 [1] skal avsenders meldingstjener:

- Påse at overføringen adresseres til riktig mottaker/meldingstjener
- Benytte gyldige sertifikater for å kryptere payload og påføre signatur på konvolutten (Virksomhetssertifikat)
- Påse at innholdet i ebXML-konvolutten er korrekt og i henhold til krav
- Benytte parametere som er spesifisert for ebXML-meldingen
- Håndtere eventuelle feilmeldinger for ebXML-meldingen eller eventuelle funksjoner som ikke støttes av mottaker

På samme måte skal mottakers meldingstjener:

- Verifisere at overføringen har kommet til riktig mottaker/meldingstjener
- Verifisere sikkerhetsinformasjon, det vil si at sertifikatet som er benyttet og signatur på konvolutten er gyldig (Virksomhetssertifikat)
- Sjekke at innholdet i ebXML-konvolutten er i henhold til krav
- Behandle ebXML-meldingen i henhold til de parametere som er spesifisert i meldingen

- Rapportere eventuelle feil i datainnholdet i ebXML-meldingen og eventuelle funksjoner som ikke støttes av mottaker tilbake til avsender

5 ebXML/SOAP konvolutten

5.1 Generelt

ebXML-konvolutten (SOAP:Envelope) slik den er beskrevet i ebXML-rammeverket [2] skal inneholde informasjon om:

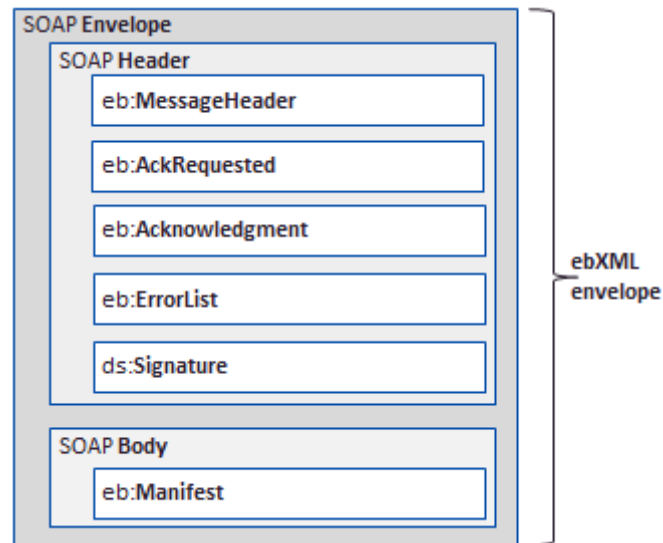
- gir en entydig identifikasjon av og adressering til kommunikasjonspartene (avsender og mottaker)
- identifiserer selve forretningsdokumentet som utveksles (sykmelding, resept, etc.)

ebXML-konvolutten skal videre sikre mulighetene til:

- å overføre transportkvitteringer på mottak av meldinger på meldingstjener-nivå
- å rapportere, ved hjelp av en feilmelding, eventuelle feil som oppdages ved behandling av mottatt XML-melding og ved dekrypteringen av innholdet i mottatt Payload

Hvis det utveksles et forretningsdokument, SKAL ebXML-konvolutten bygges opp av to deler:

- *SOAP:Header*
inneholder informasjon som skal sikre meldingsutvekslingen mellom avsender og mottaker
- *SOAP:Body*
inneholder referanse til Payload, det vil si referanse til det forretningsdokumentet som utveksles



Figur 3 viser innholdet i en ebXML-konvolutt

Utteksles det en transportkvittering (ebXML Acknowledgment) eller en feilmelding (ebXML Error Signal) skal ebXML-konvoluttet bestå av en utfylt *SOAP:Header*, mens *SOAP:Body* vil være tom.

ebXML-meldingen skal inneholde en global unik identifikator (*eb:MessageId*) og den skal inneholde informasjon om avsender (*eb:From*) og mottaker (*eb:To*). Meldingen skal signeres med virksomhetssertifikatet som tilhører avsender og som er ment for signering.

Det er også viktig å merke seg at det er forretningsdokumentet og *ikke* ebXML-konvoluttet som skal krypteres.

Hvis en melding ikke kan signeres eller ikke er i henhold til gjeldende regler, anses den ikke som gyldig og skal følgelig *ikke* sendes. Det er viktig at meldingstjeneren (MSH) varsler om slike avvik og at det er etablert gode varslingsrutiner for dette.

5.2 SOAP:Header

Elementet *SOAP:Header* inneholder informasjon som skal bidra til å sikre at meldingsutvekslingen mellom avsender og mottaker skjer på en korrekt måte.

SOAP:Header elementet består av følgende underelementer:

- *eb:MessageHeader* (skal alltid være med)
- *eb:AckRequested* (benyttes kun når det er utveksling av et forretningsdokument)
- *eb:Acknowledgment* (benyttes kun hvis meldingen er en transportkvittering (ebXML Acknowledgment))
- *eb:ErrorList* (benyttes kun hvis meldingen er en feilmelding (ebXML Error Signal))
- *ds:Signature* (skal alltid være med)

5.2.1 eb:MessageHeader

Elementet *eb:MessageHeader* inneholder informasjon om avsender og mottaker av meldingen, og parametere som viser hvilket forretningsdokument som utveksles.

Attributtet *SOAP:mustUnderstand* er obligatorisk og skal ha en fast verdi *soap:mustUnderstand="1"* (true). Attributtet gir beskjed til mottaker av forsendelsen at denne må være i stand til å prosessere innholdet i *eb:MessageHeader*.

Merk at det ikke skal sendes transportkvittering på en mottatt transportkvittering.

Eksempel på utfylte verdier i *eb:MessageHeader*:

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

5.2.1.1 eb:From og eb:To

Elementet *eb:From* er obligatorisk og benyttes til å angi id-en til den kommunikasjonsparten som er avsender av meldingen, slik det kreves i henhold til HIS 1153 [4]. Elementet består av to underliggende elementer: *eb:PartyId* og *eb:Role*.

Eksempel på utfylte verdier i *eb:From*:

```
<eb:From>
  <eb:PartyId eb:type="HER">56704</eb:PartyId>
  <eb:Role>Sykmelder</eb:Role>
</eb:From>
```

Elementet *eb:To* er obligatorisk og benyttes til å angi id-en til den kommunikasjonsparten som er mottaker av meldingen, slik det kreves i henhold til HIS 1153 [4]. Også dette elementet består av to underliggende elementer: *eb:PartyId* og *eb:Role*.

Eksempel på utfylte verdier i *eb:To*:

```
<eb:To>
  <eb:PartyId eb:type="HER">79768</eb:PartyId>
  <eb:Role>Saksbehandler</eb:Role>
</eb:To>
```

Det skal ALLTID oppgis både avsender og mottaker av en melding.

5.2.1.1.1 eb:PartyId

Elementet *eb:PartyId* ligger under elementene *eb:From* og *eb:To* og er obligatorisk. Elementet skal inneholde gjeldende id som representerer den kommunikasjonsparten som er avsender (*eb:From*) eller mottaker (*eb:To*) av meldingen.

Det er kun lovlig å identifisere én avsender og én mottaker i ebXML-konvolutt [1]. Dette betyr at hvis det oppgis flere id-er, skal alle id-ene tilhøre samme kommunikasjonspart. Oppgis det flere id-er skal det kun oppgis en av hver type.

For å sikre korrekt adressering av meldinger skal adresseringsprinsippene som er beskrevet i standard for tjenestebasert adressering [4] benyttes. Det er de faktiske kommunikasjonspartene i virksomhetene som samhandler, som skal angis som avsender og mottaker i ebXML-meldingene. Det er altså kommunikasjonspartenes HER-id som skal benyttes i adresseringen av meldingen. Det er avsenders og mottakers HER-id slik den er registrert i Adresseregisteret som skal legges i elementet *eb:From/eb:PartyId* (for avsender) eller *eb:To/eb:PartyId* (for mottaker).

5.2.1.1.2 eb:Role

Elementet *eb:Role* er obligatorisk hvis meldingen er en ebXML-konvolutt for utveksling av et forretningsdokument. Elementet benyttes til å angi avsenders (*eb:From*) eller mottakers (*eb:To*) aktive rolle. De enkelte aktørene (avsender og mottaker) skal kun inneha én rolle (Role) i en forretningsprosess, og den aktive rollen knyttes direkte til samhandlingsprosessen. Avsender og mottaker må defineres med hver sin aktive rolle for å være en "gyldig" aktør i en prosess.

Verdien som legges i dette elementet er knyttet direkte til de verdiene som legges i elementene *eb:Service* (kapittel 5.2.1.4) og *eb:Action* (kapittel 5.2.1.5). Hvilke verdier som skal benyttes i *eb:Service*, *eb:Action* og *eb:Role* skal være dokumentert i den enkelte meldingsstandard eller profil av meldingsstandard.

Hvis meldingen er en transportkvittering eller en feilmelding skal *eb:Role* ikke benyttes.

5.2.1.2 eb:CPAId

Elementet *eb:CPAId* er obligatorisk i enhver meldingsutveksling og refererer til en CPA som viser de felles mulighetene to kommunikasjonsparter har til å utføre elektronisk samhandling med hverandre.

Hvis meldingen er en transportkvittering eller en feilmelding skal *eb:CPAId* (*eb:MessageHeader/eb:CPAId*) være identisk med *eb:CPAId* i meldingen som denne meldingen er et tilsvar på.

Krav til utforming av *eb:CPAId* er beskrevet i HIS 1037:2011 [2], kapittel 6.2.1. Merk at i Norsk Helsenett sitt API for utveksling av kommunikasjonsparametere vil UUID benyttes som format for CPAId.

Eksempel på forskjellig utfylt verdi i *eb:CPAId*:

```
<eb:CPAId>56704_79768</eb:CPAId>
```

```
<eb:CPAId>nav:89252</eb:CPAId>
```

```
<eb:CPAId>EA52C1B4-F543-411F-A15A-F1D6BAE9E1F4</eb:CPAId>
```


5.2.1.3 eb:ConversationId

Elementet *eb:ConversationId* benyttes for å angi en unik identifikasjon for et sett meldinger som utgjør en konversasjon som beskrevet i HIS 1037:2011 [2], kapittel 4.1.

Hvis meldingen er en transportkvittering eller feilmelding, skal *eb:ConversationId* (*eb:MessageHeader/eb:ConversationId*) være identisk med *eb:ConversationId* i meldingen som denne meldingen er et tilsvarende på.

Eksempel på utfylt verdi i *eb:ConversationId*:

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

5.2.1.4 eb:Service

En forretningsprosess består av en eller flere komponenter (Service) som skal kunne etableres uavhengig av hverandre. Benevnelsen på den aktuelle komponenten skal legges i elementet *eb:Service*. Elementet er obligatorisk og verdien som legges i dette elementet er knyttet direkte til de verdiene som legges i elementene *eb:Role* (kapittel 5.2.1.1.2) og *eb:Action* (kapittel 5.2.1.5).

Innholdet i elementene *eb:Role*, *eb:Service* og *eb:Action* viser hva som utveksles og kan blant annet benyttes av mottaker til å formidle meldingen fra meldingstjeneren til riktig fagsystem.

Hvilke verdier som skal benyttes i *eb:Service* skal være dokumentert i den enkelte meldingsstandard eller profil av meldingsstandarden.

Benyttes meldingen som en transportkvittering eller en feilmelding, skal verdien "urn:oasis:names:tc:ebxml-msg:service" settes i elementet *eb:Service*.

Eksempel på forskjellig utfylte verdier i *eb:Service*:

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

```
<eb:Service>urn:oasis:names:tc:ebxml-msg:service</eb:Service>
```

5.2.1.5 eb:Action

Hver enkelt forretningsprosess består av et utvalg forretningstransaksjoner som viser den faktiske utvekslingen av forretningsdokumentene innen forretningsprosessen. Den konkrete handlingen som skjer i en forretningstransaksjon kalles, i ebXML-rammeverket [2], for en Action. Benevnelsen av hendelsen skal legges i elementet *eb:Action*.

Hvilke verdier som skal benyttes i *eb:Action* skal være dokumentert i den enkelte meldingsstandard eller profil av meldingsstandard.

Elementet *eb:Action* er obligatorisk og verdien som legges i dette elementet er knyttet direkte til de verdiene som legges i elementene *eb:Role* (kapittel 5.2.1.1.2) og *eb:Service* (kapittel 5.2.1.4).

Innholdet i elementene *eb:Role*, *eb:Service* og *eb:Action* viser hva som utveksles og kan blant annet benyttes av mottaker til å formidle meldingen fra meldingstjeneren til riktig fagsystem.

Benyttes meldingen som en transportkvittering, skal verdien "Acknowledgment" settes i elementet *eb:Action*.

Benyttes meldingen som en feilmelding, skal verdien "MessageError" settes i elementet *eb:Action*.

Eksempel på forskjellig utfylte verdier i *eb:Action*:

```
<eb:Action>HENVISNING</eb:Action>
```

```
<eb:Action>Registrering</eb:Action>
```

```
<eb:Action>Acknowledgment</eb:Action>
```

```
<eb:Action>MessageError</eb:Action>
```

5.2.1.6 eb:MessageData

Elementet *eb:MessageData* benyttes for å identifisere en melding. Elementet består av flere underelementer som til sammen gir en unik identifikasjon av meldingen som sendes.

Eksempel på utfylte verdier i *eb:MessageData* for en feilmelding:

```
<eb:MessageData>
  <eb:MessageId>a741e05d-f220-4f89-a92d-8cd111ee0749</eb:MessageId>
  <eb:Timestamp>2016-05-25T10:34:42Z</eb:Timestamp>
  <eb:RefToMessageId>7ea0d643-f089-4c73-841f-53621f3e560b</eb:RefToMessageId>
</eb:MessageData>
```

Merk at elementet *RefToMessageId* ikke skal brukes dersom meldingen er en transportkvittering, se kapittel 5.2.1.6.3.

5.2.1.6.1 eb:MessageId

Elementet *eb:MessageId* er obligatorisk og inneholder en global unik identifikasjon av meldingen.

5.2.1.6.2 eb:Timestamp

Elementet *eb:Timestamp* er obligatorisk og inneholder tidspunktet for når meldingen ble generert. Tidspunktet skal angis i formatet *dateTime* ("CCYY-MM-DDThh:mm:ssZ").

5.2.1.6.3 eb:RefToMessageId

Elementet *eb:RefToMessageId* inneholder referanse til en tidligere meldingstransaksjon som meldingen er et tilsvar på eller som er en del av en pågående konversasjon (meldingsflyt).

Innholdet i dette elementet skal hentes fra *eb:MessageId* (*eb:MessageHeader/eb:MessageData/eb:MessageId*) fra forrige melding i konversasjonen.

Hvis meldingen er en transportkvittering skal ikke dette elementet, men elementet *eb:RefToMessageId* under *eb:Acknowledgment* benyttes i stedet (kapittel 5.2.3.2).

Er meldingen en feilmelding skal *eb:RefToMessageId* inneholde verdien fra *eb:MessageId* i den meldingen det er avdekket feil i.

Merk at elementet *ikke* skal benyttes hvis meldingen er første melding i en konversasjon.

5.2.1.7 eb:DuplicateElimination

Elementet *eb:DuplicateElimination* er obligatorisk når meldingen ikke er en transportkvittering eller en feilmelding.

Elementet betyr at en meldingstjener som mottar en melding som er identisk med en tidligere mottatt melding (dvs. duplikat), så skal meldingstjeneren kvittere med den samme transportkvittering eller feilmelding den sendte på den tidligere mottatte meldingen.

Det er også viktig at forretningsdokumentet som ble mottatt som duplikat ikke sendes videre til og behandles av fagsystemet.

Elementet skal ikke benyttes hvis meldingen er en transportkvittering eller en feilmelding.

Eksempel på *eb:DuplicateElimination*:

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

5.2.2 eb:AckRequested

For å sikre pålitelighet i meldingsutvekslingen skal elementet *eb:AckRequested* benyttes for å gi beskjed til mottaker av forsendelsen at denne skal sende en transportkvittering eller en feilmelding i retur for mottatt melding. Transportkvitteringen eller feilmeldingen vil kun være en kvittering for meldingen fra mottakers meldingstjener, men gir avsender av meldingen en kvittering på at forsendelsen er tatt i mot av mottaker, og enten akseptert eller avvist.

Merk at *eb:AckRequested* ikke skal brukes hvis meldingen er en transportkvittering eller en feilmelding.

Attributtet *soap:mustUnderstand* er obligatorisk og skal ha en fast verdi *soap:mustUnderstand="1"* (true).

Attributtet *eb:signed* er også obligatorisk og brukes for å indikere at avsender krever en signert transportkvittering i retur. Dette angis ved å sette *eb:signed="true"*.

Eksempel på utfylte verdier i *eb:AckRequested*:

```
<eb:AckRequested SOAP:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH"  
SOAP:mustUnderstand="1" eb:signed="true" eb:version="2.0"/>
```

5.2.3 eb:Acknowledgment

Elementet *eb:Acknowledgment* benyttes når en melding er mottatt og godkjent av mottakers meldingstjener. Meldingen benyttes i denne sammenheng som en transportkvittering.

Når meldingen benyttes som en transportkvittering, skal verdien "urn:oasis:names:tc:ebxml-msg:service" settes i *eb:MessageHeader/eb:Service* elementet (kapittel 5.2.1.4) og verdien "Acknowledgment" settes i *eb:MessageHeader/eb:Action* elementet (kapittel 5.2.1.5).

Merk at dette elementet *ikke* benyttes hvis den mottatte meldingen er avvist av mottakers meldingstjener, det vil si at elementene *eb:Acknowledgment* og *eb:ErrorList* (kapittel 5.2.4) ikke skal forekomme samtidig i samme melding.

Se kapittel 8.2 for mer informasjon.

Eksempel på utfylte verdier i *eb:Acknowledgment*.

```
<eb:Acknowledgment SOAP:mustUnderstand="1" eb:version="2.0">
  <eb:Timestamp>2016-05-31T10:07:55Z</eb:Timestamp>
  <eb:RefToMessageId>a741e05d-f220-4f89-a92d-8cd111ee0749</eb:RefToMessageId>
  <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/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>S9J4o3r7f1L80rDDkqa7tKuEcqQ=</ds:DigestValue>
  </ds:Reference>
  <ds:Reference URI="cid:693c35d8-0841-4cc4-b7a3-b58d6b98d087">
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>XxPiWXUWZAWYT7G2XajkOL4bxHU=</ds:DigestValue>
  </ds:Reference>
</eb:Acknowledgment>
```

5.2.3.1 eb:Timestamp

Elementet *eb:Timestamp* er obligatorisk og inneholder tidspunktet for når meldingen som transportkvitteringen er en kvittering på, ble akseptert av mottaker. Tidspunktet skal angis i formatet *dateTime* ("CCYY-MM-DDThh:mm:ssZ").

5.2.3.2 eb:RefToMessageId

Elementet *eb:RefToMessageId* er obligatorisk og inneholder referanse til en tidligere meldingstransaksjon som denne meldingen er en kvittering på. Innholdet i dette elementet skal hentes fra *eb:MessageId* (*eb:MessageHeader/eb:MessageData/eb:MessageId*) fra forrige melding i konversasjonen. Dette betyr at *eb:RefToMessageId* i transportkvitteringen skal inneholde samme verdi som *eb:MessageId* fra opprinnelig ebXML-melding.

5.2.3.3 ds:Reference²

Elementet *ds:Reference* er obligatorisk og skal ha samme innhold som *ds:Reference* under *ds:Signature* i meldingen som denne transportkvitteringen er en kvittering på. Elementet indikerer at den opprinnelige meldingen har kommet korrekt frem til riktig mottaker og at mottaker har sjekket signaturen.

5.2.4 eb:ErrorList

Elementet *eb:ErrorList* benyttes når en melding er mottatt og det er funnet feil i meldingen. ebXML-meldingen benyttes i denne sammenheng som en feilmelding.

Når en melding med feil avvises hos mottaker skal *eb:ErrorList* brukes og *highestSeverity* skal settes til «Error», se kapittel 8.3

Når en melding inneholder feil, men blir likevel behandlet videre skal også *eb:ErrorList* benyttes, se kapittel 8.4. I slike tilfeller skal *highestSeverity* settes til «Warning». Mottaker av en *eb:ErrorList* med *highestSeverity=Warning* skal oppfatte meldingen på linje med en transportkvittering, dvs. at meldingen ble godtatt hos mottaker.

En skal benytte en mest mulig korrekt feilkode i attributtet *eb:errorCode* under elementet *eb:Error*. Elementet *eb:Description* bør fylles ut med mest mulig konkret beskrivelse av feilen.

Når *eb:ErrorList* benyttes skal *alltid* verdien "urn:oasis:names:tc:ebxml-msg:service" settes i *eb:MessageHeader/eb:Service* elementet (kapittel 5.2.1.4) og verdien "MessageError" settes i *eb:MessageHeader/eb:Action* elementet (kapittel 5.2.1.5).

Se kapittel 8.3 for mer informasjon.

Eksempel på utfylte verdier i *eb:ErrorList* (Error):

² Merk at skjemaet fra Oasis (http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd) inneholder en tvetydighetsfeil:

ds:Reference under *eb:Acknowledgment*-elementet er etterfulgt av en generell åpning for utvidelser ved hjelp av et *xs:any* med *##other* som namespacekrav.

Et element plassert her kan valideres som en *ds:Reference* eller noe helt annet. Det er et poeng at *ds:Reference*-feltene valideres da de er i bruk grunnet at det er stilt krav om bruk av signert kvittering [1]. Valgt løsning er bruk av et alternativt skjema der den generelle åpningen lukkes (les: fjernes helt) [1]. Det er mulighet for at ingen bruker denne utvidelsesmuligheten i norsk helsesektor, slik at *ds:Reference* dermed vil valideres entydig uten annen konsekvens.

```
<eb:ErrorList SOAP:mustUnderstand="1" eb:highestSeverity="Error" eb:version="2.0">
  <eb:Error eb:errorCode="SecurityFailure" eb:location="xpointer(/MessageHeader/Signature)"
    eb:severity="Error">
    <eb:Description xml:lang="en">Message a741e05d-f220-4f89-a92d-8cd111ee0749 has
    one or more invalid signature elements: null</eb:Description>
  </eb:Error>
</eb:ErrorList>
```

Eksempel på utfylte verdier i *eb:ErrorList* (Warning):

```
<eb:ErrorList SOAP:mustUnderstand="1" eb:highestSeverity="Warning" eb:version="2.0">
  <eb:Error eb:errorCode="ValueNotRecognized"
    eb:location="xpointer(/)" eb:severity="Warning">
    <eb:Description xml:lang="en">Incorrect encoding (ISO-8859-1) used in message
    2aaf1196-c3fe-472b-8a91-ad89ef7822dd. Correct encoding is UTF-
    8</eb:Description>
  </eb:Error>
</eb:ErrorList>
```

5.2.5 ds:Signature

Elementet *ds:Signature* er obligatorisk i ebXML-meldingen. På bakgrunn av dette skal meldingen *ALLTID* signeres digitalt med et virksomhetssertifikat som er gyldig på signeringstidspunktet, er utstedt av en godkjent sertifikatutsteder og i overensstemmelse med XMLDSIG [6]. Sertifikatet skal tilhøre kommunikasjonsparten som er angitt som avsender av meldingen. Signering av ebXML-meldingen utføres av meldingstjeneren. Det skal kun benyttes sertifikat med bruksområde signatur (vanligvis beskrevet med key usage: "non-repudiation"³).

Signaturen skal omfatte alle Payloads som følger med i ebXML-meldingen, og hver enkelt Payload som ligger i meldingen skal *alltid* identifiseres med et eget *ds:Reference* element.

Innholdet i *ds:Reference* skal være samme verdi som finnes i elementet *eb:Reference* i *SOAP:Body*, det vil si Payload-delen av ebXML-meldingen.

ds:Reference elementet skal *alltid* inneholde tre *ds:Transform* elementer, som oppgitt i ebMS 2.0 [1].

Eksempel på utfylte verdier i *ds:Transform*:

³ I henhold til Kravspesifikasjon for PKI i offentlig sektor, kapittel 4.1.1 Generelle krav, alle sertifikattyper.


```
<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"/>
```

Eksempel på utfylte verdier i *ds:Signature*:

```
<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/2000/09/xmldsig#rsa-sha1"/>
    <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="h ENV="http://schemas.xmlsoap.org/soap/envelope/">not(ancestor-or-self::node()[@SOAP-
ENV:act ENV:actor="urn:oasis:names:tc:ebxml-msg:actor:nextMSH"] | ancestor-or-self::node()[@SOAP-
ENV:act 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/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>S9J4o3r7f1L80rDDkqa7tKuEcqQ=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="cid:693c35d8-0841-4cc4-b7a3-b58d6b98d087">
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>XxPiWXUWZAWYT7G2XajkOL4bxHU=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
      oHxcJZxzk/GVesh+...y0rMgrh/eZHdKxxA==
    </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>
          MIIFBDDCA+...sf6bLOzM5ZdRqSUBxnEXw==
        </ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
```

5.3 SOAP:Body

SOAP:Body inneholder informasjon om forretningsdokumentet som ligger i Payload (kapittel 0).

Hvis meldingen er en transportkvittering eller feilmelding skal *SOAP:Body* være tomt.

5.3.1 eb:Manifest

I elementet *eb:Manifest* ligger adressen til MIME vedlegget (CID) som inneholder Payloaden, det vil si referansen til forretningsdokumentet som utveksles (*eb:Reference*), og skal peke på riktig CID i Payload (kapittel 0).

Elementet *eb:Manifest* skal kun være med dersom det sendes en Payload. Elementet skal altså ikke være med dersom meldingen er en transportkvittering (kapittel 8.2) eller en feilmelding (kapittel 8.3).

Eksempel på utfylte verdier i *eb:Manifest*:

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


6 Payload

Payload inneholder selve forretningsdokumentet som utveksles. Payloaden skal være identisk med det som er referert i elementet *eb:Manifest* under *u* (kapittel 5.3.1).

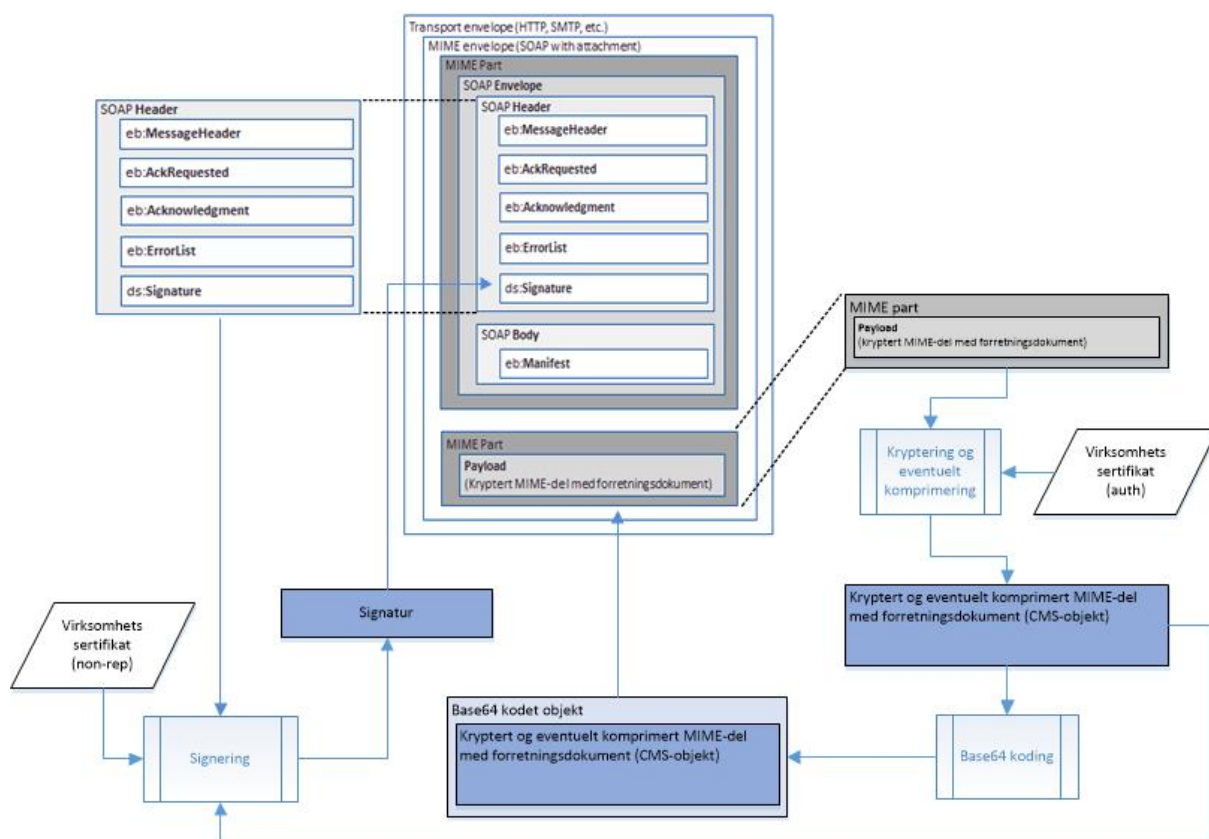
Norm for informasjonssikkerhet Helse- og omsorgstjenesten (Normen) [5] sier at overføring av helse- og personopplysninger skal skje i samsvar med prosedyrer som ivaretar kravene til blant annet konfidensialitet og integritet. PKI (Public Key Infrastructure) benyttes for å sikre meldingene som sendes over helsenettet.

Payload skal, med noen få unntak, krypteres, og det er mottakende virksomhets offentlige krypteringssertifikat som skal benyttes, det vil si det sertifikatet som har key usage: "key encipherment", "key agreement" og/eller "data encipherment" ⁴ skal benyttes. Payload skal alltid krypteres, så fremt ikke annet er angitt i den enkelte meldingsstandard. Krypteringen av Payload skjer i prosessen før man konstruerer ebXML-meldingen.

Ved sending av store fagmeldinger kan man i en meldingsstandard kreve at fagmeldingen komprimeres for å redusere størrelsen på Payload med tanke på overføringen av meldingen. Det er i så fall den enkelte meldingsstandard som setter krav til hvilken komprimeringsmetode som skal benyttes. Komprimering av Payload skjer i prosessen før man konstruerer ebXML-meldingen.

For oppbygging av Payload, se [1], og HIS 1037:2011 [2] kapittel 5.5.

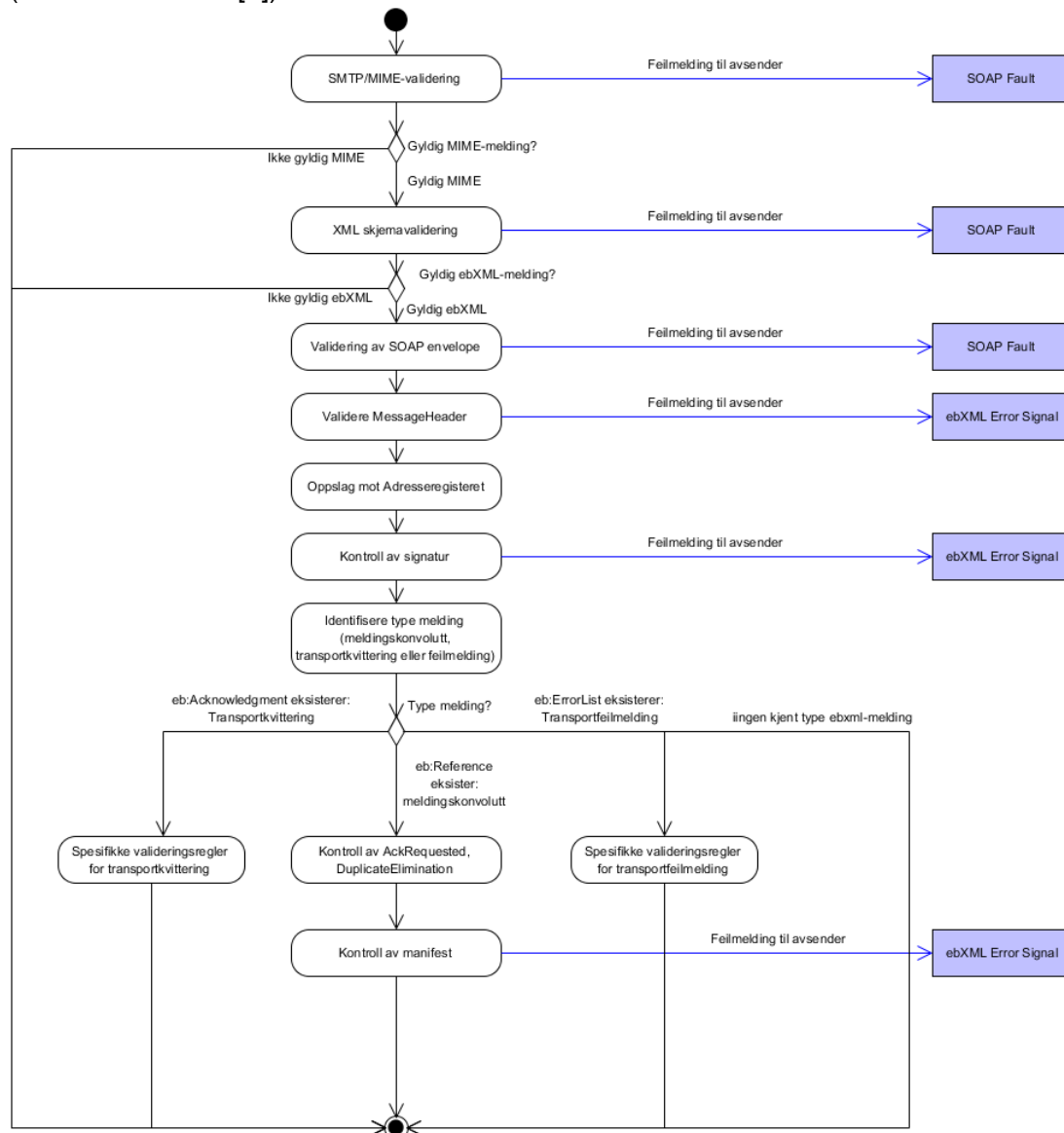
⁴ I henhold til Kravspesifikasjon for PKI i offentlig sektor, kapittel 4.1.1 Generelle krav, alle sertifikattyper.



Figur 4 viser signering og kryptering av ebXML-konvolutt og payload

7 Valideringsprosessen

Med bakgrunn i HIS 1037:2011 [2] og ebMS 2.0 [1] er det beskrevet et sett med valideringsregler som anbefales brukt for både mottak og sending av ebXML-meldinger (HISD 1172:2016 [3]).



Figur 5 Valideringsprosessen

Merk at enkelte av valideringsreglene som er beskrevet i dokumentet [3] kun gjelder for transportkvittering eller feilmelding, mens andre regler er spesifikke for ordinære ebXML-meldinger det vil si meldinger som benyttes til utveksling av forretningsdokumenter.

8 Transportkvittering og feilmelding

8.1 Generelt

I henhold til HIS 1037:2011 [2] skal det sendes transportkvittering (ebXML Acknowledgment) eller feilmelding (ebXML Error Signal) for en mottatt ebXML-melding som inneholder en Payload (forretningsdokument). Transportkvittering eller feilmelding skal sendes umiddelbart etter at ebXML meldingstjener har mottatt forsendelsen og validert denne i henhold til reglene spesifisert i HIS 1037:2011 [2] kapittel 4.3, pkt. 4. ebMS 2.0s funksjoner for pålitelig meldingsutveksling skal benyttes (ebMS 2.0 [1], kapittel 6 "Reliable Messaging Module").

Pålitelig meldingsutveksling gjennomføres ved at avsenders meldingstjener ber om en transportkvittering på mottak (Acknowledgment Request) i den meldingen som sendes (kapittel 5.2.2). Mottakers meldingstjener skal svare med en transportkvittering, som inneholder et *eb:Acknowledgment* element og en status, eller en feilmelding som inneholder et *eb:ErrorList* element og en feilstatus, samt referanse til den opprinnelige meldingsutvekslingen. Hvis avsender ikke mottar en slik transportkvittering eller feilmelding innen en gitt tidsperiode, skal meldingen sendes på nytt et gitt antall ganger, som beskrevet i HIS 1037:2011 [2], kapittel 5.2.

Tidsperioden for resending er angitt i elementet *RetryInterval* under elementet *DocExchange* i CPA eller spesifisert i den enkelte meldingsstandard. Standard verdi i CPA og ebXML generelt er 12 timer. Imidlertid kan enkelte meldingsstandarder benytte andre *RetryInterval*. For eksempel benytter eResept et *RetryInterval* på 1-4 timer avhengig av hvilken eResept-melding som sendes.

Før en melding sendes på nytt er det imidlertid viktig at en sjekker mailserver⁵ om den aktuelle transportkvittering som savnes, allerede er mottatt men av en eller annen grunn ikke er videreformidlet til meldingstjeneren.

Merk at det aldri skal kvitteres fra meldingstjener for en mottatt transportkvittering eller feilmelding.

8.2 Transportkvittering (ebXML Acknowledgment)

En melding som er tatt feilfritt i mot i mottakers meldingstjener skal kvitteres for ved at det sendes en transportkvittering (ebXML Acknowledgment) i retur for den mottatte meldingen, uavhengig av om det oppdages feil ved Payload eller behandlingen av forretningsdokumentet i mottakers systemer. Dette betyr at transportkvitteringen er en

⁵ Mailtransfer agent som definert i RFC 2821

bekreftelse til avsenders meldingstjener på at meldingen har kommet frem og funnet i orden i henhold til reglene for ebXML-meldinger.

Når en melding er mottatt og godkjent av mottakers meldingstjener, benyttes elementet *eb:Acknowledgment* til å gi avsender av meldingen informasjon om at meldingen har kommet korrekt frem til riktig mottaker. Elementet *eb:RefToMessageId* skal inneholde samme verdi som oppgitt i *eb:MessageId* (*eb:MessageHeader/eb:MessageData/eb:MessageId*) fra den meldingen som ble akseptert og som det kvitteres for. Elementet *eb:Timestamp* benyttes til å angi tidspunktet for når transportkvitteringen ble generert.

Elementet *eb:MessageHeader/eb:Service* skal ha verdien "urn:oasis:names:tc:ebxml-msg:service" (kapittel 5.2.1.4) og i elementet *eb:MessageHeader/eb:Action* skal verdien "Acknowledgment" settes (kapittel 5.2.1.5).

Merk at *eb:Acknowledgment* elementet *ikke* benyttes hvis den mottatte meldingen er avvist av mottakers meldingstjener, det vil si at elementet *eb:Acknowledgment* (kapittel 5.2.3) ikke skal benyttes i en feilmelding (kapittel 8.3).

Transportkvittering skal kun sendes én gang per mottatt melding. Eventuelle avvik etter at det er sendt transportkvittering skal varsles til avsender ved at det sendes applikasjonskvittering, godkjent avviksmelding eller gjennom vanlig avvikshåndtering.

8.3 Feilmelding (ebXML Error Signal)

Hvis det avdekkes feil i en mottatt melding slik at meldingen ikke kan behandles skal *ikke* forretningsdokumentet prosesseres videre, dvs. overføres til mottakers fagsystem. Meldingstjeneren skal i disse tilfeller generere en feilmelding (ebXML Error Signal) med *highestSeverity="Error"* og sende denne i retur til avsenders meldingstjener som en beskjed om at overføringen ble avvist og angi årsaken til avvisningen.

Når en melding er mottatt av mottakers meldingstjener og av ulike årsaker er avvist benyttes elementet *eb:ErrorList* til å gi avsender av meldingen som ble avvist informasjon om hvilken melding som er avvist og årsaken til at meldingen ble avvist.

Det er viktig at man benytter en så korrekt feilkode som mulig i attributtet *eb:errorCode* i elementet *eb:Error*. Elementet *eb:Description* bør fylles ut med mest mulig konkret beskrivelse av årsaken til at meldingen ble avvist.

Følgende feilkoder⁶ kan benyttes:

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

Hvordan feilkodene benyttes er nærmere beskrevet i [3].

Grunnet at *eb:ErrorList* ikke har et eget indre element som refererer til *eb:MessageId* (*eb:MessageHeader/eb:MessageData/eb:MessageId*) i den melding som er avvist, benyttes innholdet i elementet *eb:MessageHeader/eb:MessageData/eb:RefToMessageId* til å identifisere hvilken melding avvissningen gjelder.

Ved *highestSeverity="Error"*, skal verdien "urn:oasis:names:tc:ebxml-msg:service" settes i *eb:MessageHeader/eb:Service* elementet (kapittel 5.2.1.4) og verdien "MessageError" settes i *eb:MessageHeader/eb:Action* elementet (kapittel 5.2.1.5).

Merk at *eb:ErrorList* elementet *ikke* benyttes hvis den mottatte meldingen er godkjent av mottakers meldingstjener. Elementet *eb:ErrorList* (kapittel 5.2.4) skal altså ikke benyttes i en transportkvittering (kapittel 8.2).

En melding som er besvart med feilmelding er avvist av mottakers meldingstjener og skal *ikke* videreformidles til og behandles i mottakers fagsystem.

Feilmelding skal *kun* sendes én gang per mottatt melding.

⁶ I henhold til 1037:2011 [2], kapittel 7.2.4 og ebMS 2.0 [1], kapittel 4.2.3.4.1

8.4 Warning (ebXML Error Signal med **highestSeverity="Warning"**)

Hvis det oppdages feil i den mottatte meldingen, men mottakers meldingstjener allikevel velger å prosessere den, skal det sendes en feilmelding (ebXML Error Signal) med *highestSeverity="Warning"* (advarsel) i retur. I og med at feilmeldingen har status "Warning" skal meldingen oppfattes som om den var en transportkvittering av mottakers meldingstjener og følgelig ikke medføre avvisning (kapittel 5.2.4).

Følgende hendelser kan gi *highestSeverity="Warning"*:

- Annen encoding enn UTF-8, for eksempel ISO8859-1.

Annen versjon av XML enn versjon 1.0, helt ukjent encoding, samt forskjell i encoding og karaktersett skal gi feil og følgelig medføre avvisning av den mottatte meldingen.

Ved *highestSeverity="Warning"*, skal verdien "urn:oasis:names:tc:ebxml-msg:service" settes i *eb:MessageHeader/eb:Service* elementet (kapittel 5.2.1.4) og verdien "MessageError" settes i *eb:MessageHeader/eb:Action* elementet (kapittel 5.2.1.5).

Feilmelding skal kun sendes én gang per mottatte melding. Eventuelle avvik etter at det er sendt feilmelding med *highestSeverity="Warning"* må varsles til avsender på andre måter.

9 Scenarier

Kapitlene nedenfor viser enkelte scenarier for meldingsutveksling som vil kunne oppstå i en normal driftssituasjon.

Scenario 9.1 - 9.6:

Generelt vil kvitteringen fra mottakers meldingstjener skje ved at det genereres en ebXML-melding (transportkvittering) som kun består av en ebXML-konvolutt med referanse til den mottatte ebXML-meldingen foruten et element som angir at dette er en bekreftelse. En slik kvittering vil ikke inneholde et forretningsdokument i form av en Payload, da selve forretningsdokumentet i dette tilfellet kun er konvolutt.

Den opprinnelig mottatte meldingen identifiseres ved at man oppgir referansen til det MIME-elementet som inneholdt konvolutt.

Det skal ikke kvitteres for en mottatt transportkvittering fra meldingstjener.

Scenario 9.7:

Feilrapporteringen fra mottakers meldingstjener vil skje ved at det genereres en ebXML-melding (feilmelding) som kun består av en ebXML-konvolutt med referanse til den mottatte ebXML-meldingen og en oversikt eller liste som angir hvilke feil som er oppdaget. En slik kvittering vil ikke inneholde et forretningsdokument i form av en Payload.

Den opprinnelig mottatte meldingen identifiseres ved at man oppgir referansen til det MIME-elementet som inneholdt konvolutt.

Det skal ikke kvitteres for en mottatt feilmelding fra meldingstjener.

Når feilen er funnet og rettet, kan ny korrekt ebXML-melding genereres og sendes.

Følgende forkortelser benyttes scenariene i figur 6 - 13:

P = Payload, forretningsdokument transportert som et vedlegg i en ebXML-melding.

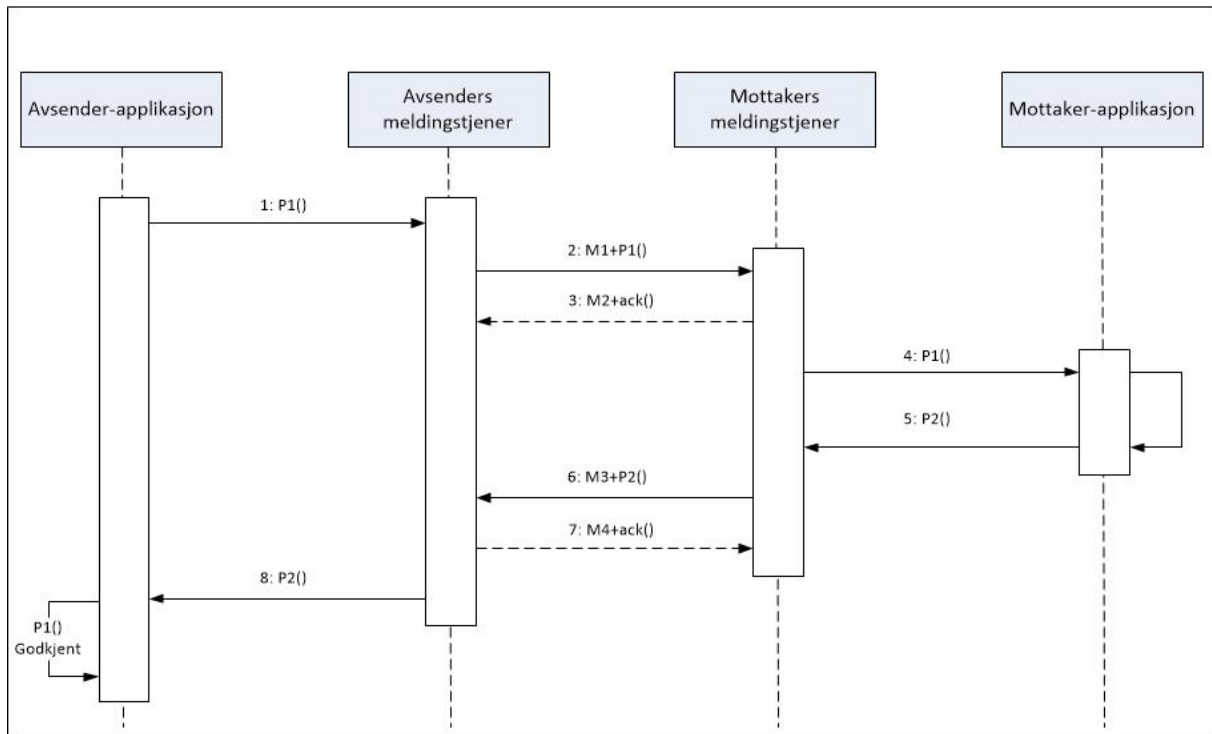
M = ebXML-konvolutt, SOAP Envelope med ebXML-utvidelser

ack = acknowledgment/transportkvittering, ebXML-konvolutt som benyttes som en ebXML Acknowledgment

err = feilmelding, ebXML-konvolutt som benyttes som en ebXML Error Signal

9.1 Transportkvittering og positiv applikasjonskvittering

9.1.1 Enkel konversasjon med fagmelding og applikasjonskvittering



Figur 6 viser en konversasjon hvor det er sendt både transportkvittering og positiv applikasjonskvittering

I scenario 9.1.1 sender "Avsender-applikasjon" fagmelding (P1) som i "Avsenders meldingstjener" pakkes inn i en ebXML-konvolutt (M1+P1) som sendes til "Mottakers meldingstjener".

"Mottakers meldingstjener" validerer M1+P1, finner alt ok og svarer med en transportkvittering (M2+ack). Fagmeldingen P1 pakkes deretter ut av ebXML-konvolutten (M1+P1) og sendes til "Mottaker-applikasjon".

Meldingen P1 valideres av "Mottaker-applikasjon", som finner alt ok og genererer en positiv applikasjonskvittering (P2). P2 sendes deretter til "Mottakers meldingstjener" som pakker meldingen inn i en ebXML-konvolutt (M3+P2) og sender denne til "Avsenders meldingstjener".

"Avsenders meldingstjener" validerer M3+P2, finner alt ok og svarer med en transportkvittering (M4+ack). Applikasjonskvitteringen P2 pakkes deretter ut av ebXML-konvolutten (M3+P2) og sendes til "Avsender-applikasjon".

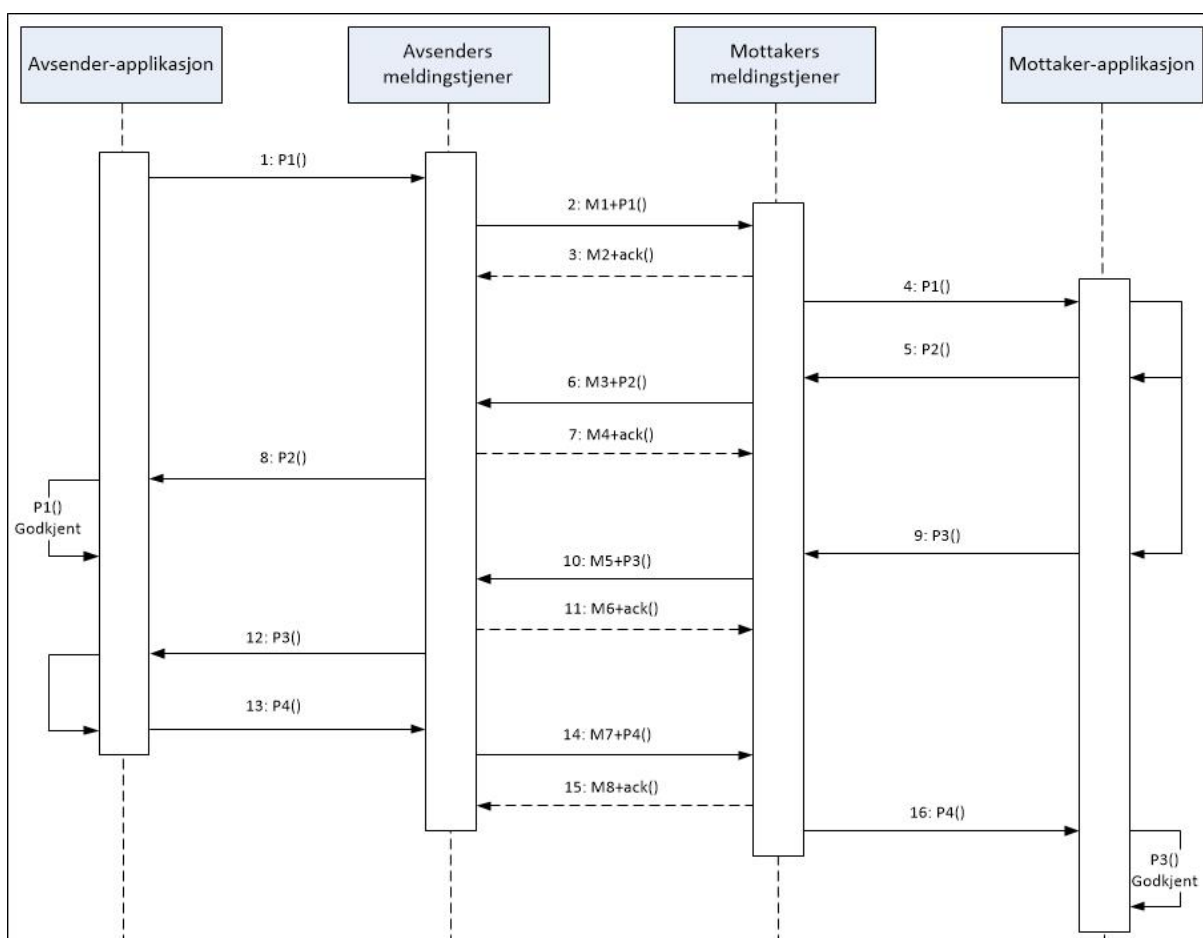
Siden meldingen er en positiv applikasjonskvittering oppdaterer "Avsender-applikasjon" meldingen P1 med status=OK.

NB!

Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M2+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M1+P1.

I melding M4+ack skal verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M3+P2.

9.1.2 Stor konversasjon med flere fagmeldinger og applikasjonskvitteringer



Figur 7 viser en konversasjon hvor det er sendt både transportkvittering og positiv applikasjonskvittering

I scenario 9.1.2 sender "Avsender-applikasjon" fagmelding (P1) som i "Avsenders meldingstjener" pakkes inn i en ebXML-konvolutt (M1+P1) som sendes til "Mottakers meldingstjener".

"Mottakers meldingstjener" validerer M1+P1, finner alt ok og svarer med en transportkvittering (M2+ack). Fagmeldingen P1 pakkes deretter ut av ebXML-konvolutten (M1+P1) og sendes til "Mottaker-applikasjon".

Meldingen P1 valideres av "Mottaker-applikasjon", som finner alt ok og genererer en positiv applikasjonskvittering (P2). P2 sendes deretter til "Mottakers meldingstjener" som pakker meldingen inn i en ebXML-konvolutt (M3+P2) og sender denne til "Avsenders meldingstjener".

"Avsenders meldingstjener" validerer M3+P2, finner alt ok og svarer med en transportkvittering (M4+ack). Applikasjonskvitteringen P2 pakkes deretter ut av ebXML-konvolutten (M3+P2) og sendes til "Avsender-applikasjon".

Siden meldingen er en positiv applikasjonskvittering oppdaterer "Avsender-applikasjon" meldingen P1 med status=OK.

Som en del av den samme forretningsprosessen skal "Mottaker-applikasjon" behandle meldingen P1 og svare tilbake med en svarmelding (P3). Meldingen P3 sendes til "Mottakers meldingstjener" som pakker meldingen inn i en ebXML-konvolutt (M5+P3) og sender denne til "Avsenders meldingstjener".

"Avsenders meldingstjener" tar imot og validerer M5+P3, finner alt ok og svarer med en transportkvittering (M6+ack). Fagmeldingen P3 pakkes deretter ut av ebXML-konvolutten (M5+P3) og sendes til "Avsender-applikasjon".

"Avsender-applikasjon" validerer P3, finner alt ok, genererer en positiv applikasjonskvittering (P4) og sender denne til "Avsenders meldingstjener". P4 pakkes deretter inn i en ebXML-konvolutt (M7+P4) av "Avsenders meldingstjener" og sender denne til "Mottakers meldingstjener".

"Mottakers meldingstjener" tar imot og validerer M7+P4, finner alt ok og svarer med en transportkvittering (M8+ack). Applikasjonskvitteringen P4 pakkes deretter ut av ebXML-konvolutten (M7+P4) og sendes til "Mottaker-applikasjon".

Siden meldingen er en positiv applikasjonskvittering oppdaterer "Mottaker-applikasjon" meldingen P3 med status=OK.

NB!

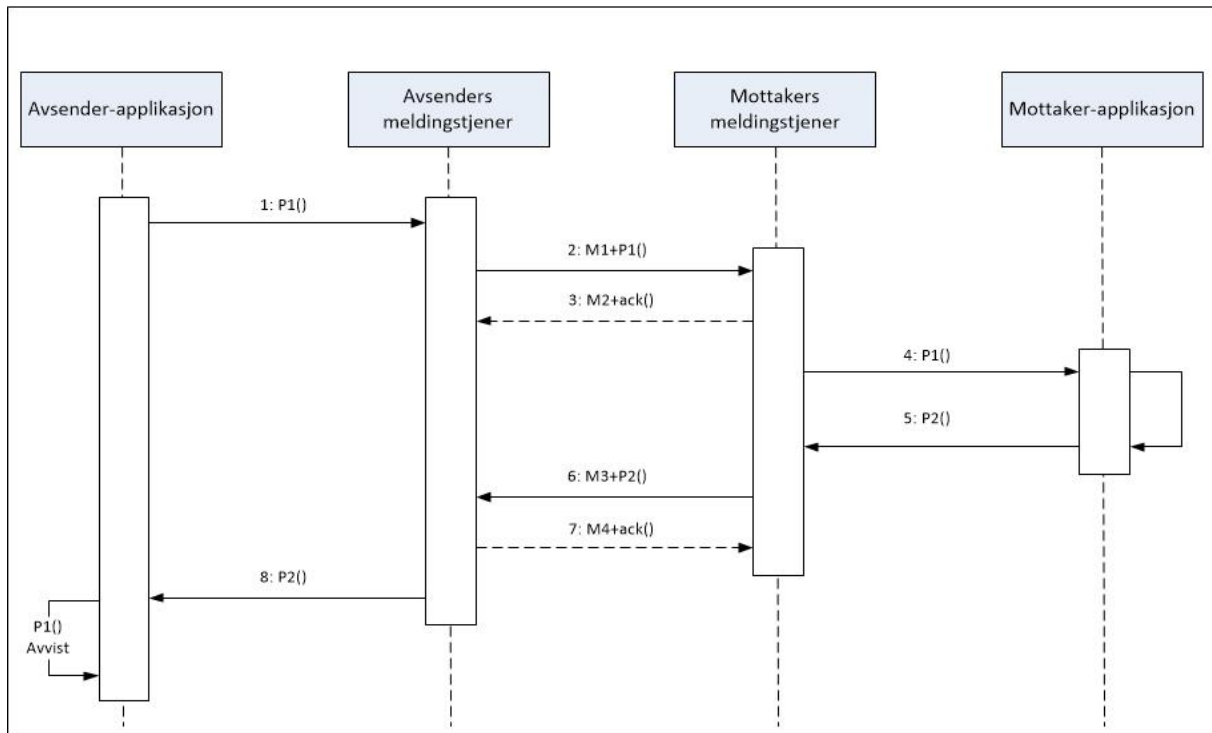
Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M2+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M1+P1.

Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M4+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i meldingen M3+P2

Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M6+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M5+P3.

Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M8+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M7+P4.

9.2 Transportkvittering og negativ applikasjonskvittering



Figur 8 viser en konversasjon hvor det er sendt transportkvittering og negativ applikasjonskvittering

I scenario 9.2 sender "Avsender-applikasjon" fagmelding (P1) som i "Avsenders meldingstjener" pakkes inn i en ebXML-konvolutt (M1+P1) som sendes til "Mottakers meldingstjener".

"Mottakers meldingstjener" validerer M1+P1, finner alt ok og svarer med en transportkvittering (M2+ack).

Fagmeldingen P1 pakkes deretter ut av ebXML-konvolutten (M1+P1) og sendes til "Mottaker-applikasjon", som validerer meldingen og finner at P1 inneholder en feil som gjør at meldingen ikke kan behandles. På bakgrunn av dette genereres det en negativ applikasjonskvittering (P2) som sendes til "Mottakers meldingstjener". P2 pakkes deretter inn i en ebXML-konvolutt (M3+P2) og sendes til "Avsenders meldingstjener".

I "Avsenders meldingstjener" valideres M3+P2. Når man finner alt ok svares det med en transportkvittering (M4+ack), som deretter pakker P2 ut av ebXML-konvolutten (M3+P2) og sender meldingen videre til "Avsender-applikasjon".

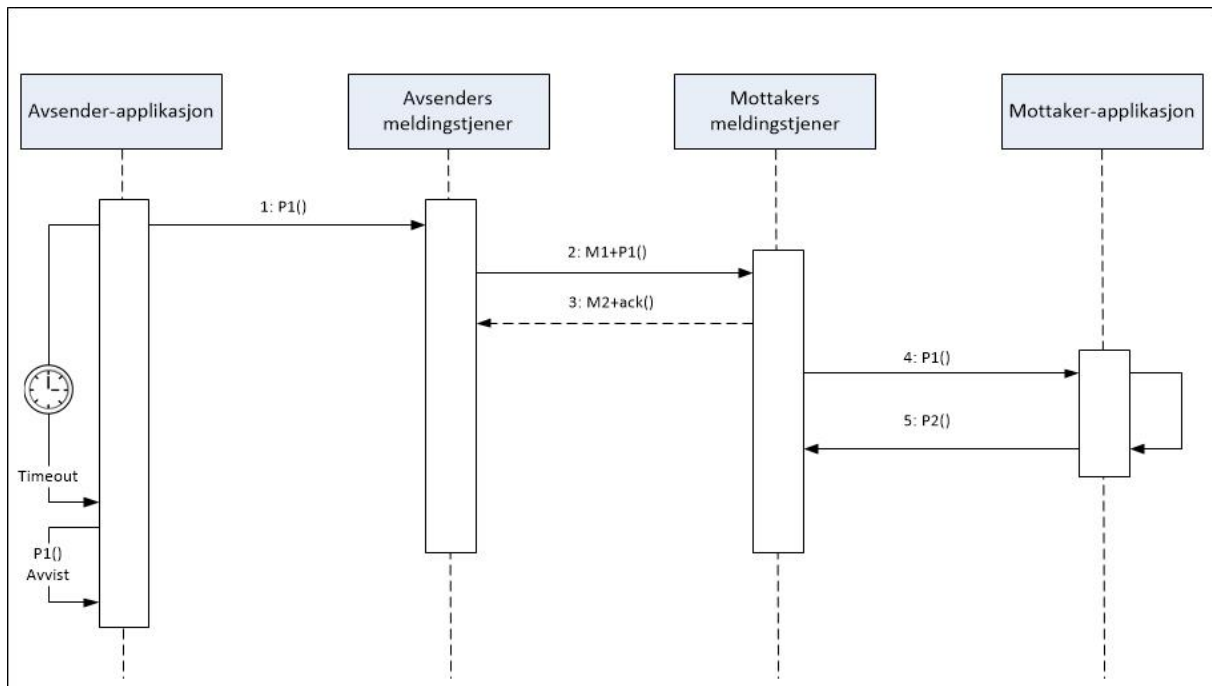
Siden meldingen er en negativ applikasjonskvittering, oppdaterer "Avsender-applikasjon" meldingen P1 med status=Avvist.

NB!

Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M2+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M1+P1.

I melding M4+ack skal verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M3+P2.

9.3 Transportkvittering, men ikke applikasjonskvittering



Figur 9 viser en konversasjon hvor det er sendt en transportkvittering men ikke applikasjonskvittering

I scenario 9.3 sender "Avsender-applikasjon" fagmelding (P1) som i "Avsenders meldingstjener" pakkes inn i en ebXML-konvolutt (M1+P1) som sendes til "Mottakers meldingstjener".

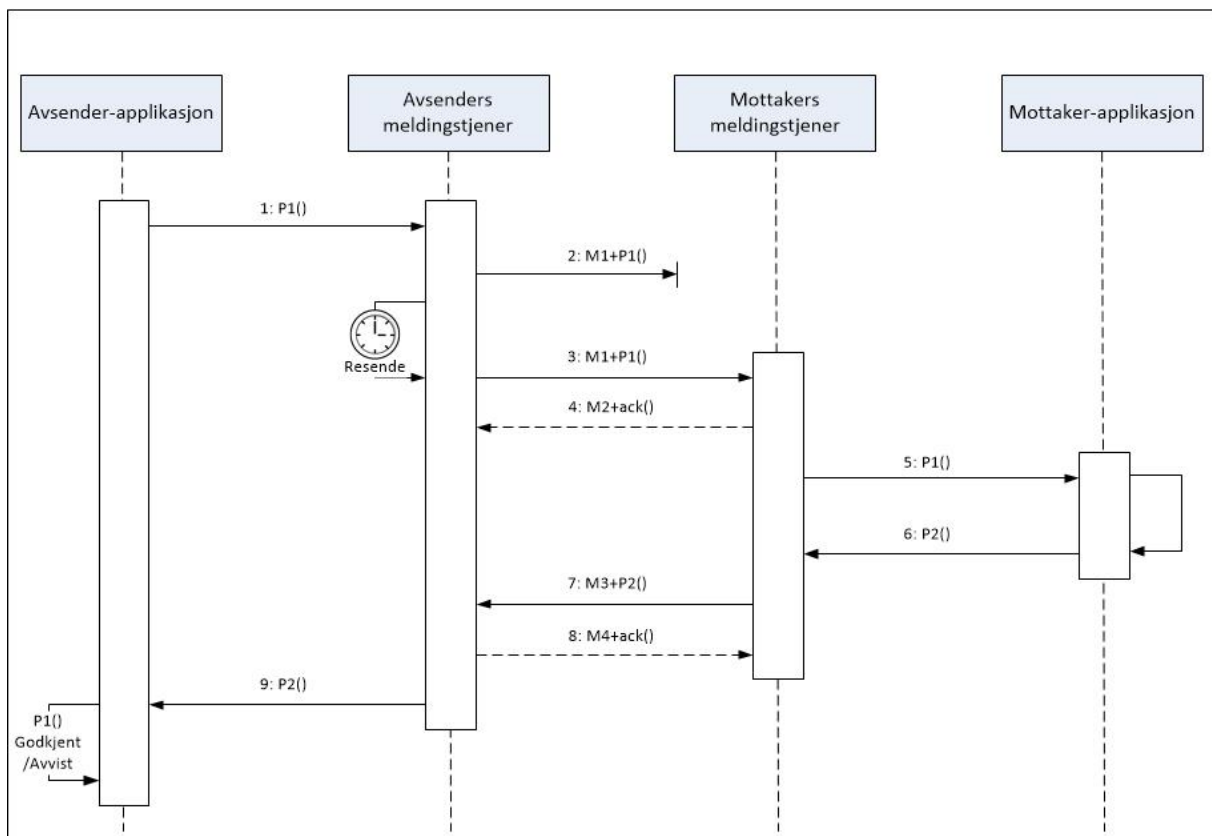
"Mottakers meldingstjener" validerer M1+P1, finner alt ok og svarer med en transportkvittering (M2+ack). Fagmeldingen P1 pakkes deretter ut av ebXML-konvolutt (M1+P1) og sendes til "Mottaker-applikasjon".

"Mottaker-applikasjon" validerer P1, finner alt ok, genererer en positiv applikasjonskvittering (P2) og sender denne til "Mottakers meldingstjener".

Av ukjent årsak blir ikke P2 pakket inn i en ebXML-konvolutt og sendt til "Avsenders meldingstjener", slik at "Avsender-applikasjon" etter en angitt maks-tid (timeout) anser meldingen P1 som tapt og oppdaterer meldingen med status=Avvist.

Merk: Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M2+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M1+P1.

9.4 Fagmelding ikke mottatt – Transportkvittering ikke sendt



Figur 10 viser en konversasjon hvor fagmelding ikke er mottatt og det følgelig ikke er sendt transportkvittering

I scenario 9.4 sender "Avsender-applikasjon" fagmelding (P1) som i "Avsenders meldingstjener" pakkes inn i en ebXML-konvolutt (M1+P1) som sendes til "Mottakers meldingstjener".

"Mottakers meldingstjener" mottar aldri M1+P1 og vil av den grunn aldri kunne kvittere på mottaket.

På grunn av at "Avsenders meldingstjener" ikke mottar kvittering på M1+P1, sendes meldingen M1+P1 på nytt til "Mottakers meldingstjener". Denne gang er sendingen suksessfull.

"Mottakers meldingstjener" validerer M1+P1, finner alt ok og svarer med en transportkvittering (M2+ack). Fagmeldingen P1 pakkes deretter ut av ebXML-konvolutt (M1+P1) og sendes til "Mottaker-applikasjon". Meldingen P1 blir validert i "Mottaker-

applikasjon" og genererer en applikasjonskvittering (P2) og sender denne til "Mottakers meldingstjener". Meldingen P2 kan enten være en positiv eller negativ applikasjonskvittering.

"Mottakers meldingstjener" pakker P2 inn i en ebXML-konvolutt (M3+P2) og sender denne til "Avsenders meldingstjener".

I "Avsenders meldingstjener" valideres M3+P2 og man finner alt ok og svarer med en transportkvittering (M4+ack). Applikasjonskvitteringen P2 pakkes deretter ut av ebXML-konvolutt (M3+P2) og sendes til "Avsender-applikasjon".

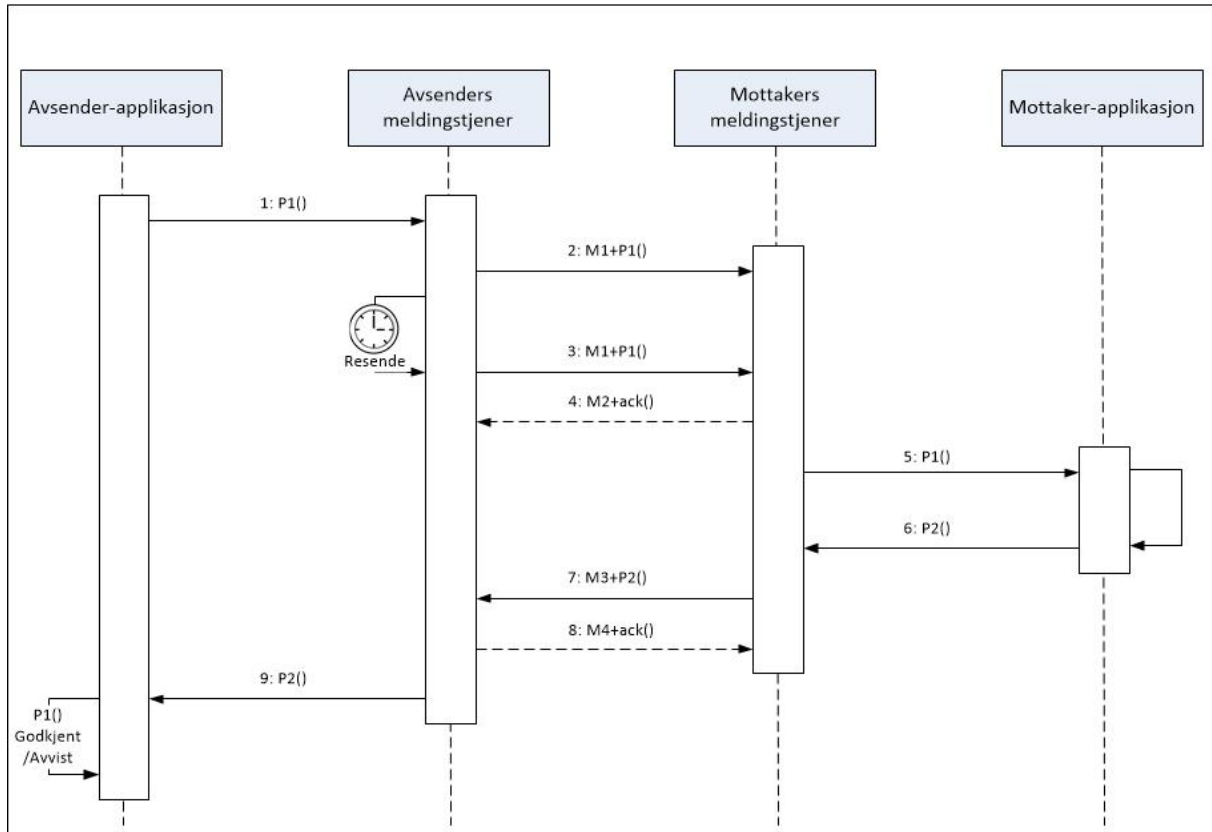
På bakgrunn av innholdet i meldingen P2 oppdaterer "Avsender-applikasjon" meldingen P1 med status=OK eller status=Avvist.

NB!

Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M2+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M1+P1.

I melding M4+ack skal verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M3+P2.

9.5 Fagmelding mottatt, men transportkvittering ikke sendt



Figur 11 viser en konversasjon hvor en fagmelding er mottatt men hvor det ikke er sendt transportkvittering

I scenario 9.5 sender "Avsender-applikasjon" fagmelding (P1) som i "Avsenders meldingstjener" pakkes inn i en ebXML-konvolutt (M1+P1) som sendes til "Mottakers meldingstjener".

"Mottakers meldingstjener" mottar M1+P1, men kvitterer ikke på mottaket.

På grunn av at "Avsenders meldingstjener" ikke mottar kvittering på M1+P1, sendes meldingen M1+P1 på nytt til "Mottakers meldingstjener".

"Mottakers meldingstjener" tar imot og validerer M1+P1, finner alt ok og svarer med en transportkvittering (M2+ack). Fagmeldingen P1 pakkes deretter ut av ebXML-konvolutten (M1+P1) og sendes til "Mottaker-applikasjon".

"Mottaker-applikasjon" validerer P1, genererer en applikasjonskvittering (P2) og sender denne til "Mottakers meldingstjener". Meldingen P2 pakkes deretter i en ebXML-konvolutt (M3+P2) av "Mottakers meldingstjener" og sender denne til "Avsenders meldingstjener". Meldingen P2 kan enten være en positiv eller negativ applikasjonskvittering.

"Avsenders meldingstjener" validerer M3+P2, finner alt ok og svarer med en transportkvittering (M4+ack). Applikasjonskvitteringen P2 pakkes deretter ut av ebXML-konvolutten (M3+P2) og sendes til "Avsender-applikasjon".

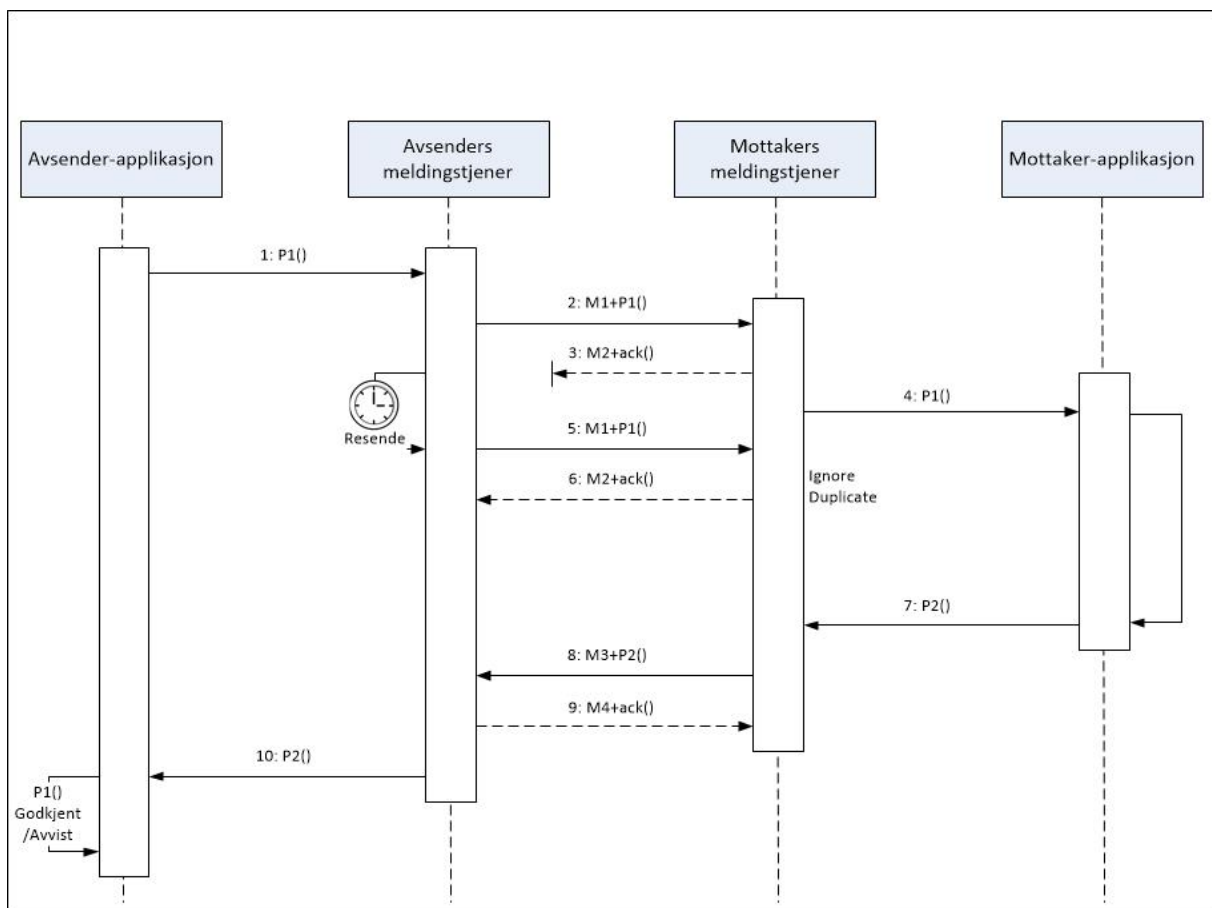
På bakgrunn av innholdet i meldingen P2 oppdaterer "Avsender-applikasjon" meldingen P1 med status=OK eller status=Avvist.

NB!

Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M2+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M1+P1.

I melding M4+ack skal verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M3+P2.

9.6 Fagmelding mottatt, transportkvittering er sendt men ikke mottatt



Figur 12 viser en konversasjon hvor en fagmelding er mottatt og transportkvittering er sendt men ikke mottatt

I scenario 9.6 sender "Avsender-applikasjon" en fagmelding (P1) til "Avsenders meldingstjener" som pakker P1 inn i en ebXML-konvolutt (M1+P1) og sender denne til "Mottakers meldingstjener".

"Mottakers meldingstjener" validerer M1+P1, finner alt ok og svarer med en transportkvittering (M2+ack). Fagmeldingen P1 pakkes deretter ut av ebXML-konvolutt (M1) og sendes til "Mottaker-applikasjon".

Av ukjent grunn mottar ikke "Avsenders meldingstjener" transportkvitteringen (M2+ack), og sender meldingen M1+P1 på nytt til "Mottakers meldingstjener".

"Mottakers meldingstjener" validerer M1+P1, finner alt ok men oppfatter det som duplikat og re-sender den positive transportkvitteringen (M2+ack) som ble sendt tidligere. Det er viktig at meldingen (P1) som ble mottatt som duplikat ikke sendes til "Mottaker-applikasjon".

"Mottaker-applikasjon" validerer P1, genererer en applikasjonskvittering (P2) og sender denne til "Mottakers meldingstjener" som pakker P2 inn i en ebXML-konvolutt (M3+P2) og sender denne til "Avsenders meldingstjener". Meldingen P2 kan enten være en positiv eller negativ applikasjonskvittering.

"Avsenders meldingstjener" validerer M3+P2, finner alt ok og svarer med en transportkvittering (M4+ack). Applikasjonskvitteringen P2 pakkes deretter ut av ebXML-konvolutt (M3+P2) og sendes til "Avsender-applikasjon".

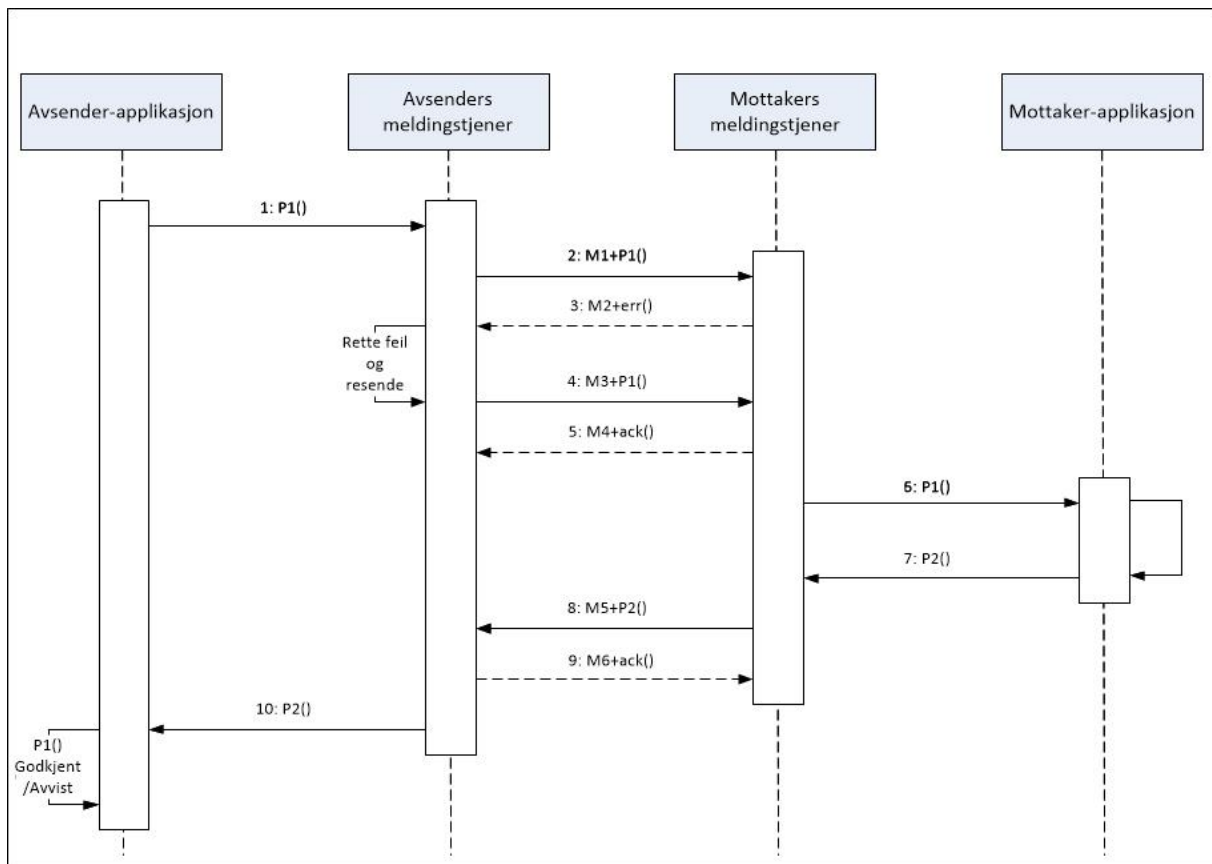
På bakgrunn av innholdet i meldingen P2 oppdaterer "Avsender-applikasjon" meldingen P1 med status=OK eller status=Avvist.

NB!

Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M2+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M1+P1.

I melding M4+ack skal verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M3+P1.

9.7 Mottatt feilmelding - Feil i konvolutt



Figur 13 viser en konversasjon hvor det er mottatt en feilmelding

I scenario 9.7 sender "Avsender-applikasjon" fagmelding (P1) som i "Avsenders meldingstjener" pakkes inn i en ebXML-konvolutt (M1+P1) som sendes til "Mottakers meldingstjener".

"Mottakers meldingstjener" finner feil i meldingen M1+P1 og svarer med en negativ transportmelding (M2+err) som sendes til "Avsenders meldingstjener".

"Avsenders meldingstjener" retter feilen, pakker fagmeldingen P1 inn i en ny og korrekt ebXML-konvolutt M3+P1 og sender den til "Mottakers meldingstjener".

"Mottakers meldingstjener" validerer M3+P1, finner alt ok og svarer med en transportkvittering (M4+ack). Fagmeldingen P1 pakkes deretter ut av ebXML-konvolutten (M3+P1) og sendes til "Mottaker-applikasjon".

"Mottaker-applikasjon" validerer P1, genererer en applikasjonskvittering (P2) og sender denne til "Mottakers meldingstjener".

"Mottakers meldingstjener" pakker P2 inn i en ebXML-konvolutt (M5+P2) og sender denne til "Avsenders meldingstjener". Meldingen P2 kan enten være en positiv eller negativ applikasjonskvittering.

"Avsenders meldingstjener" validerer M5+P2, finner alt ok og svarer med en transportkvittering (M6+ack). Applikasjonskvitteringen P2 pakkes deretter ut av ebXML-konvolutten (M5+P2) og sendes til "Avsender-applikasjon".

På bakgrunn av innholdet i meldingen P2 oppdaterer "Avsender-applikasjon" meldingen P1 med status=OK eller status=Avvist.

NB!

Verdien som legges i *eb:MessageHeader/eb:MessageData/eb:RefToMessageId* i meldingen M2+err skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M1+P1.

Verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* i meldingen M4+ack skal hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M3+P1.

I melding M6+ack skal verdien som legges i *eb:Acknowledgment/eb:RefToMessageId* hentes fra *eb:MessageHeader/eb:MessageData/eb:MessageId* i melding M5+P2.

10 Eksempler

10.1 Signert ebXML-konvolutt

```
<?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>EPIKRISesender</eb:Role>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="HER">91101</eb:PartyId>
        <eb:Role>EPIKRISereceiver</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-EPIKRIS</eb:Service>
      <eb:Action>EPIKRIS</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="false" 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/2000/09/xmldsig#rsa-sha1"/>
        <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/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>S9J4o3r7f1L80rDDkqa7tKuEcqQ=</ds:DigestValue>
        </ds:Reference>
        <ds:Reference URI="cid:693c35d8-0841-4cc4-b7a3-b58d6b98d087">
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>XxPiWxUWZAWYT7G2XajkOL4bxHU=</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>
        oHxcJZzk/GVesh+...y0rMgrh/eZHdKxxA==
      </ds:SignatureValue>
    </ds:Signature>
  </SOAP:Header>
  <SOAP:Body/>
</SOAP:Envelope>
```

```

<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>

```

10.2 Transportkvittering (ebXML Acknowledgment)

```

<?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:RefToMessageId>a741e05d-f220-4f89-a92d-8cd111ee0749</eb:RefToMessageId>
      <eb:From>
        <eb:PartyId eb:type="HER">91101</eb:PartyId>
      </eb:From>
    </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/2000/09/xmldsig#rsa-sha1"/>
        <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:Transforms>
        </ds:Reference>
      </ds:SignedInfo>
    </ds:Signature>
  </SOAP:Header>
  <SOAP:Body>
    <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:RefToMessageId>a741e05d-f220-4f89-a92d-8cd111ee0749</eb:RefToMessageId>
      <eb:From>
        <eb:PartyId eb:type="HER">91101</eb:PartyId>
      </eb:From>
    </eb:Acknowledgment>
  </SOAP:Body>
</SOAP:Envelope>

```

```
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>UWZAWYXxPiWXT7G2XL4bxHUajkO=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="cid:693c35d8-0841-4cc4-b7a3-b58d6b98d087">
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>XxPiWXUWZAWYT7G2XajkOL4bxHU=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
6oUuKNm2hohg1K...eZHdKx==
</ds:SignatureValue>
<ds:KeyInfo>
<ds:X509Data>
<ds:X509Certificate>
IIBCgKCAQEAx1Y...LOzM5ZdRqSUBxn==
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
</SOAP:Header>
<SOAP:Body/>
</SOAP:Envelope>
```

10.3 Feilmelding (ebXML Error med highestSeverity "Error")

```
<?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>MessageError</eb:Action>
<eb:MessageData>
<eb:MessageId>c9595e2c-65b8-46ca-9405-6dca95ab61e9</eb:MessageId>
<eb:Timestamp>2016-05-10T16:00:34Z</eb:Timestamp>
<eb:RefToMessageId>a741e05d-f220-4f89-a92d-8cd111ee0749</eb:RefToMessageId>
</eb:MessageData>
</eb:MessageHeader>
<eb:ErrorList SOAP:mustUnderstand="1" eb:highestSeverity="Error" eb:version="2.0">
<eb:Error eb:errorCode="SecurityFailure" eb:location="xpointer(/MessageHeader/Signature)"
eb:severity="Error">
<eb:Description xml:lang="en">Message a741e05d-f220-4f89-a92d-8cd111ee0749 has
one or more invalid signature elements: null</eb:Description>
</eb:Error>
</eb:ErrorList>
<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/2000/09/xmldsig#rsa-sha1"/>
<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/2000/09/xmldsig#sha1"/>
  <ds:DigestValue>UWZAWYXxPiWXT7G2XL4bxHUajkO=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="cid:693c35d8-0841-4cc4-b7a3-b58d6b98d087">
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <ds:DigestValue>XxPiWXUWZAWYT7G2XajkOL4bxHU=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
  6oUuKNm2hohg1K...eZHdKx==
</ds:SignatureValue>
<ds:KeyInfo>
  <ds:X509Data>
    <ds:X509Certificate>
      IIBCgKCAQEAx1Y...LOzM5ZdRqSUbxn==
    </ds:X509Certificate>
  </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
</SOAP:Header>
<SOAP:Body/>
</SOAP:Envelope>
```

10.4 Warning (ebXML Error med highestSeverity "Warning")

Sendt ebXML-melding med feil encoding:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<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>EPIKRISesender</eb:Role>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="HER">91101</eb:PartyId>
        <eb:Role>EPIKRISereceiver</eb:Role>
      </eb:To>
    </eb:MessageHeader>
  </SOAP:Header>
  <SOAP:Body/>
</SOAP:Envelope>
```



```
<eb:CPAId>90998_91101</eb:CPAId>
<eb:ConversationId>d5942eaf-807f-4c3c-a24d-ae6ee81a54d9</eb:ConversationId>
<eb:Service eb:type="string">S-EPIKRISE</eb:Service>
<eb:Action>EPIKRISE</eb:Action>
<eb:MessageData>
  <eb:MessageId>2aaf1196-c3fe-472b-8a91-ad89ef7822dd</eb:MessageId>
  <eb:Timestamp>2016-05-25T10:34:42Z</eb:Timestamp>
  . . . . .
```


Warning:

```
<?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>MessageError</eb:Action>
      <eb:MessageData>
        <eb:MessageId>c9595e2c-65b8-46ca-9405-6dca95ab61e9</eb:MessageId>
        <eb:Timestamp>2016-05-10T16:00:34Z</eb:Timestamp>
        <eb:RefToMessageId>2aaf1196-c3fe-472b-8a91-ad89ef7822dd</eb:RefToMessageId>
      </eb:MessageData>
    </eb:MessageHeader>
    <eb:ErrorList SOAP:mustUnderstand="1" eb:highestSeverity="Warning" eb:version="2.0">
      <eb:Error eb:errorCode=" ValueNotRecognized" eb:location="xpointer(/)" eb:severity="Warning">
        <eb:Description xml:lang="en">Incorrect encoding (ISO 8859-1) used in message
        2aaf1196-c3fe-472b-8a91-ad89ef7822dd. Correct encoding is UTF-8</eb:Description>
      </eb:Error>
    </eb:ErrorList>
    <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/2000/09/xmldsig#rsa-sha1"/>
        <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/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>UWZAWYXxPiWXT7G2XL4bxHUajkO=</ds:DigestValue>
```

```

        </ds:Reference>
        <ds:Reference URI="cid:693c35d8-0841-4cc4-b7a3-b58d6b98d087">
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>XxPiWXUWZAWYT7G2XajkOL4bxHU=</ds:DigestValue>
        </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
        6oUuKNm2hoghg1K.../eZHdKx==
    </ds:SignatureValue>
    <ds:KeyInfo>
        <ds:X509Data>
            <ds:X509Certificate>
                IIBCgKCAQEAx1Y...LOzM5ZdRqSUBxn==
            </ds:X509Certificate>
        </ds:X509Data>
    </ds:KeyInfo>
</ds:Signature>
</SOAP:Header>
<SOAP:Body/>
</SOAP:Envelope>

```

 Direktoratet for e-helse

Besøksadresse

Verkstedveien 1
0277 Oslo

Postadresse

Postboks 6737
St. Olavs plass
0130 OSLO