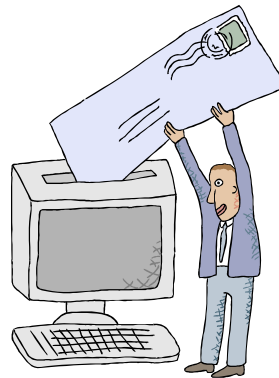


SFTI Teknisk Transportprofil: Bas



Version: 2.0
Status: Standard
Datum: 2005-11-30

Författare:

Anders W. Tell
Martin Forsberg
Sören Lennartsson

Financial Toolsmiths AB
Amnis Consulting
Ooi Data

Ändringshistorik

Not. Ändringar redovisas bara i förhållande till fastställda versioner av transportprofilen, dvs dokumentversioner märkta "Standard".

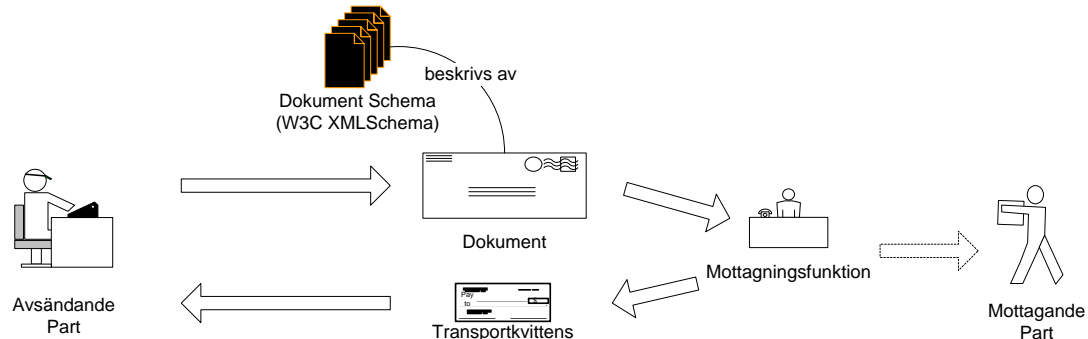
Version	Datum	Beskrivning	Utförd av
1.0.0	2004-06-16	Ursprungligt dokument	Författarna
1.1	2004-11-08	Rättelser av påträffade fel, förtydliganden, etc införda i texten. Ändringshistorik och information om ändringar har förts till. Version anges endast med två siffror.	S Lennartsson
2.0	2005-10-26	Ändring av kvittenshantering. Rättning av exempel. I denna version har mottagningskvitto, enligt BPSS, bytts ut mot transportkvittens enligt ebMS 2.0. Flera dokument kan nu sändas i samma meddelande. Denna version är inte bakåtkompatibla med version 1.0 och 1.1.	M Forsberg

Innehållsförteckning

SFTI Teknisk Transportprofil: Bas	1
1.1. Introduktion.....	4
2. Transportprotokoll.....	5
2.1. Säkerhet	5
3. Meddelanden	6
3.1. Paketering av dokument i ett meddelandekuvvert	6
3.1.1. Identifiering av avsändare och mottagare av meddelanden	6
3.1.2. Identifiering av tekniskt avtal.....	7
3.1.3. Identifiering av pågående konversation	7
3.1.4. Identifiering av vilken verksamhetstjänst som mottager meddelanden	7
3.1.5. Unik Identifiering av meddelande	8
3.1.6. Meddelandets innehållsförteckning.....	8
3.1.7. Synkronisering av meddelande med svar över samma förbindelse.....	9
3.1.8. Begäran om transportkvittens.....	10
3.2. Svarsmeddelanden	10
3.2.1. Transportkvittens från avsedd mottagare	10
3.2.2. Felmeddelanden och Felrapportering.....	11
3.2.2.1 HTTP/S fel.....	12
3.2.2.2 MIME Fel	12
3.2.2.3 SOAP Fel	12
3.2.2.4 Fel i Meddelandehantering.....	12
3.2.2.5 Rapportering av Inkompatibilitet med denna profil.....	12
3.3. Övrig information som EJ skall användas i meddelanden	12
3.3.1. <DuplicateElimination>	12
3.3.2. <Signature>	12
3.3.3. <MessageOrder>.....	12
3.4. Meddelandehanterarens övriga tjänster	13
4. Exempel på meddelande	14
4.1. Transportkvittens	15
5. Teknisk bilaga till transportprofil Bas	16
6. Realiseringsteknologier	17
6.1. Java – JAXM, SAAJ, Java Server Pages (JSP).....	17
6.2. Kodexempel i Java.....	18
7. Referenser.....	20
7.1. Normativa Referenser	20
7.2. Ej Normativa referenser	20
7.3. ebMS Säkerhetsprofiler	21
7.4. Definitioner av Core Components	22
7.4.1. Formaliserad beskrivning av transportprofilens detaljer	22

1.1. Introduktion

SFTI:s tekniska Transportprofil Bas är en specifikation som möjliggör enkel elektronisk kommunikation mellan affärspartners och/eller dess förmedlingstjänsteföretag. Bas-profilen är anpassad för förenklad kommunikation som innebär att ett eller flera dokument per meddelande översänds och en transportkvittens erhålls som svar.



Bas-profilen är baserad på ebXML ramverket och dess ebXML Message Service Specification [ebMS] Version 2.0. EbXML Ramverket har tagits fram av FN organisation UN/CEFACT i samarbete med OASIS och dess ingående delar är ISO standarder. UN/CEFACT är organisationen bakom EDIFACT och ebXML kan ses som EDIFACT:s efterträdare.

Dokument utbyts med hjälp av HTTP/S protokollet [SSL3].

”Collaboration Protocol Profile and Agreement” [ebCPA], ”Business Process Specification Schema” [BPSS] samt övriga ebXML specifikationer ingår EJ i Bas-profilen.

Denna specifikation av Bas-profilen skall ses som en strikt begränsad version av ebXML MSH specifikation [ebMS] där beskrivningen är konstruerad som ett tillägg till den fullständiga specifikationen och skall läsas tillsammans med denna.

ebXML Message Service Handler		
SOAP with Attachment		
HTTP/S	MIME	XML
TCP/IP		
...		

Bas-profilen utgår från ebXML ramverket där olika valbara alternativ har tagits bort. Den kräver ej en fullfjädrad [ebMS] meddelandehanteringstjänst utan kan realiserats med små medel baserad på tillgänglig kunskap och teknologier. Det finns även Open source implementationer att tillgå. Se bilaga för information om realiserings teknologier.

Genom att Bas-profilen utgör en strikt, begränsad del av ebXML ramverket kan parter som önskar kommunicera med stöd av mer funktionalitet enkelt skala upp *inom* ramverket exempelvis genom att använda fler tillförlitlighets- och säkerhetsmöjligheter.

Det kan noteras att XML formaterade dokument såväl som EDIFACT formaterade dokument kan översändas med hjälp av denna Bas-profil.

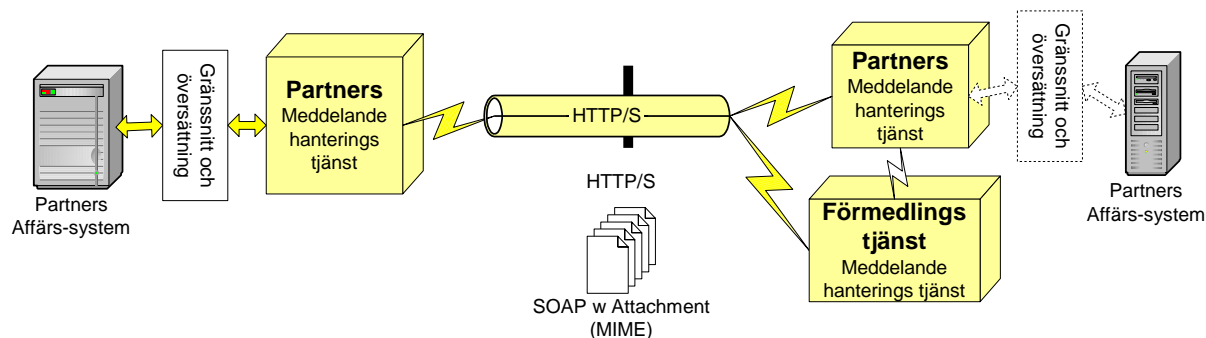
Om en mottagande meddelandehanterare (Message Service Handler – MSH) upptäcker en avvikelse från denna profil, exempelvis ett valbart alternativ som ej ingår, kan mottagande MSH välja, beroende på dess funktionalitet, om den ska fullfölja mottagandet eller rapportera ett fel som innebär att avsändande MSH agerat utanför profilens ramar. Se felrapporterings-sektionen för information hur denna typ av inkompatibilitetsproblem skall rapporteras.

2. Transportprotokoll

Meddelanden som innehåller dokument och svar transporteras mellan parter genom en begränsad användning av HTTP/S protokollet, dvs [SSL3].

Ett meddelande från en dokumentutställare skickas från partens meddelandetjänst (MSH) eller från en förmedlingstjänst som agerar på dokumentutställarens uppdrag.

Ett meddelande till en avsedd dokumentmottagare skickas till mottagandes parts meddelandetjänst (MSH) eller till en förmedlingstjänst som agerar åt den avsedda dokumentmottagaren.



Kommunikationen utförs synkront mellan avsändare och mottagare genom dessa meddelandehanteringstjänster, vilket innebär att en [SSL3], dvs HTTP förbindelse, ej kopplas ned innan svar erhållits eller timeout inträffat. En kvittens som indikerar att meddelandet mottagits måste således skickas tillbaka i samma förbindelse som avsänt meddelande. Kommunikationen mellan en parts affärssystem och meddelandehanteringstjänst (egen eller hos anlitad part) ligger under partens eget ansvar och kan utformas efter eget val.

För att definiera att synkron kommunikation skall användas måste `<syncReplyMode>` vara definierat i SOAP headern för varje meddelande. Se [ebMS] B.2.5 för mer information.

Meddelanden måste vara formaterade enligt MIME principerna definierade i sektion 19.4 i [RFC2616]. Det rekommenderas att implementationer tolkar närvaron av den obligatoriska HTTP headern 'SOAPAction' liberalt enligt riktlinjer i SOAP 1.1 specifikation, dvs att ett meddelande som ej innehåller 'SOAPAction' ej betraktas om inkompatibelt.

Om ett meddelande framgångsrikt översänds till en mottagande MSH skall HTTP svar 2XX returneras. Om fel uppstår skall HTTP status 3XX, 4XX eller 5XX returneras. Om SOAP fel uppstår skall '500 "Internal Server Error"' enligt SOAP 1.1 specifikationen rapporteras. Se felhanteringssektionen för mer information om felhantering och dess rapportering.

Se [ebMS] appendix B för mer och normativ information och detaljer gällande transportprotokollet.

2.1. Säkerhet

Den säkerhetsprofil som motsvarar säkerhetsnivån för Bas-profilen är Säkerhetsprofil 5. Endast servercertifikat skall användas för [SSL3] kommunikation.

Accesskontrollmekanismen "Basic and Digest Access Authentication" ingår EJ i profilen. Se [ebMS] sektion B.2.6 för mer information.

Obs! Server-certifikat används i transportprofilen för kryptering, inte autentisering. För detta syfte kan såväl egenutfärdade som tredjepartsutfärdade certifikat användas. Tänk dock på att de egenutfärdade certifikaten skapar varningar/frågor i klienterna vilket kan förvirra en ovan användare!

3. Meddelanden

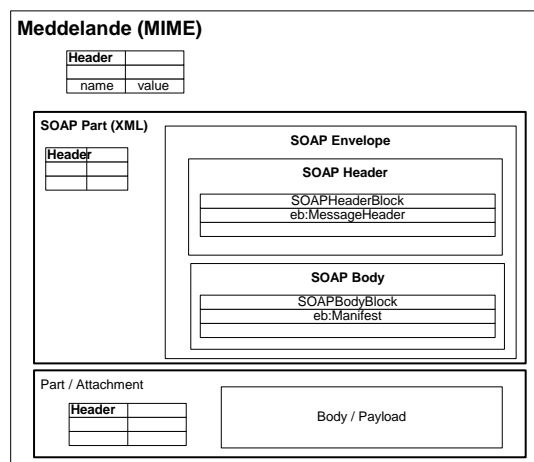
Bas-profilen är anpassad för att skicka ett eller flera dokument i ett meddelande. En transportkvittens returneras när meddelandet (och alla dess ingående delar) kommit fram i sin helhet.

3.1. Paketering av dokument i ett meddelandekuvvert

Dokument paketeras enligt MIME principer. Ett meddelande består av 2 eller fler delar, så kallade "MIME Parts". Ett dokument format bör vara beskrivet i en formatbeskrivning, ett s.k. dokumentschema.

1. SOAP v1.1 Envelope som formaterats i XML format
2. Payload, dvs ett eller flera dokument formaterat i ett godtyckligt format, exempelvis XML, EDIFACT. Varje dokument placeras i en egen MIME-del och refereras till i meddelandets innehållsförteckning.

Se sektion 3.1 i [ebMS] samt [SOAP] för detaljerade specifikationer samt nedan för ett exempel.



Regler för hur ett meddelande paketeras och hur en SOAP Envelope konstrueras beskrivs nedan.

3.1.1. Identifiering av avsändare och mottagare av meddelanden

Avsändare och avsedd mottagare identifieras genom att i varje SOAP header skriva in en text som unikt identifierare dessa.

I Bas-profilen kan man ange minst ett identifieringsbegrepp för avsändare och avsedd mottagare av meddelande. Om fler än ett identifieringsbegrepp anges måste dessa ange samma part.

Identifikationssättet definieras i samverkansspecifikationen (separat dokument).

```
<eb:From>
  <eb:PartyId eb:type='countrycode:organizationid'> SE1234567890 </eb:PartyId>
</eb:From>
<eb:To>
  <eb:PartyId eb:type='countrycode:organizationid'> SE9876543210 </eb:PartyId>
  <eb:PartyId eb:type='operatorid'> QWERTYUIOP </eb:PartyId>
</eb:To>
```

Ett <From> och <To> element måste ingå i varje meddelande och dess SOAP Header.

Rollbeskrivningar för respektive avsändare och mottagare ingår EJ i Bas-profilen och skall därför EJ ingå i varken <From> eller <To> elementen.

Möjligheten att ange fler än ett identifieringsbegrepp kan användas av förmedlingstjänster.

Se [ebMS] sektion 3.1.1 för mer information.

3.1.2. Identifiering av tekniskt avtal

I den tekniska Bas-profilen förutsätter inte nödvändigtvis att parterna upprättar ett tekniskt avtal skriftligen, utan avtalet om datautväxling kan formars muntlig eller konkludent genom att ena parten agerar i enlighet med andra partens erbjudande. Detta innebär för kommunikationsreglerna att den obligatoriska referensen till CPAId ej har någon betydelse för tolkningen av ivägskickade dokument. Om en CPA används måste parametrarna i denna ej motsäga denna profil. En CPAId bör skapas enligt följande definition:

```
CPAId ::= <datum> ':' <fromPartyId> ':' <toPartyId>
```

```
<eb:CPAId>20040510:SE1234567890:SE9876543210</eb:CPAId>
```

Ett <CPAId> element måste skickas med i varje meddelande och dess SOAP Header.

Se [ebMS] sektion 3.1.2 för mer information.

3.1.3. Identifiering av pågående konversation

För att kunna relatera meddelanden i en samverkansprocess, scenarion, till varandra i så kallade konversationer kräver [ebMS] att en unik konversationsidentifierare skickas med varje meddelande. Avsändaren av första meddelandet i konversationen bestämmer på egen hand vilket ID som kopplas till konversationen. Det rekommenderas dock att följande definition används.

Datum då första meddelandet i konversationen skickas slås samman (konkateneras) med ett unikt nummer. Numret måste vara unikt per datum. Exempelvis ett löpnummer som startar med noll (0) varje dag kl 00.00.00.

```
ConversationId ::= <datumNärMeddelandeÖversänds> ':' <uniktNummer> ':'  
<fromPartyId>
```

```
<eb:ConversationId>20040510:4567:SE1234567890</eb:ConversationId>
```

Ett <ConversationID> element måste skickas med i varje meddelande och dess SOAP Header.

Se [ebMS] sektion 3.1.3 för mer information.

3.1.4. Identifiering av vilken verksamhetstjänst som mottager meddelanden

Denna tjänst definieras i samverkansspecifikation baserat på mönstret nedan, vilket är beroende på samverkansprocessen och ingående dokumentnamn.

```
Service ::= 'urn:' <organization> ':services:documentprocessing'  
<samverkansProcess>  
Action ::= 'incoming' <dokumentNamn>
```

```
<eb:Service>urn:sfti:services:documentprocessing:BasicInvoice</eb:Service>  
<eb:Action>incomingBasicInvoice</eb:Action>
```

Se [ebMS] sektion 3.1.4, 3.1.5 för mer information.

3.1.5. Unik Identifiering av meddelande

För varje meddelande finns det möjlighet att koppla samman en globalt unik identifierare med meddelandet. Detta ID måste överensstämja med principerna beskrivna i [RFC2822]. Tidstämpeln kodifieras enligt UTC principer.

```
<eb:MessageData>
  <eb:MessageId>20040510-102030-28572@foretag.se</eb:MessageId>
  <eb:Timestamp>2004-05-15T11:12:12</eb:Timestamp>
</eb:MessageData>
```

Ett <MessageData> element måste skickas med i varje meddelande och dess SOAP Header. För avsända meddelanden exkluderas följande av dess element ur i Bas-profilen: <RefToMessageId> samt <TimeToLive>, som beskriver en absolut tidpunkt uttryckt i UTC när ett meddelande skall vara mottaget. <RefToMessageId> måste dock anges i felmeddelanden.

Se [ebMS] sektion 3.1.6 för mer information.

3.1.6. Meddelandets innehållsförteckning

I varje ebXML meddelande kan det ingå en beskrivning av innehållet i meddelandet, dvs dess paketerade dokument. I alla meddelanden med paketerade dokument (alltså ej transportkvittensen) måste i denna Bas-profil ha tillvalselementet, <Manifest> i SOAP Body. 'id'-attributet skall ej definieras. 'version'-attributet måste däremot definieras med konstanta värdet "2.0" (not: refererar till ebMS SOAP header extension, version 2.0).

<Manifest> elementet skall innehålla ett eller flera <Reference> element som pekar ut i meddelandet ingående dokument vilka har paketerats med hjälp av MIME principer. <Reference> elementet är en enkel länk [XLINK] där attributet 'type' har konstant värde enligt 'xlink:type="simple"'. Attributen 'id' och 'role', ingår ej i Bas-profilen och skall ej definieras.

'href'-attributet pekar ut ett specifikt dokumentet som är paketerat i en MIME del och referensen måste vara en URI och följa [XLINK] principerna. 'href' skall följa 'cid:' namnsättnings-principerna vilket innebär att MIME delen där dokument är paketerat måste innehålla en MIME header 'content-id' som motsvarar värdet i 'href'-attributet.

```
<SOAP:Body>
  <eb:Manifest eb:version="2.0">
    <eb:Reference xlink:href="cid:ebxmlpayload1@avsandare.com" xlink:type="simple">
      <eb:Description xml:lang="se"> Fritext beskrivning av meddelande </eb:Description>
      <eb:Schema eb:location="urn:se:sfti:collaborationprocesses:BasicInvoice.xsd"
eb:version="1.0"> </eb:Schema>
    </eb:Reference>
  </eb:Manifest>
</SOAP:Body>
```

Om 'href' ej pekar ut en existerande 'content-id' måste mottagande MSH rapportera detta genom att i svaret skriva in ett <Error> element med 'errorCode=MimeProblem' och 'severity=Error'.

Se [ebMS] sektion 3.2 för mer information.

Schema:

Dokument som utbyts i e-affärssammanhang beskrivs och definieras vanligtvis av ett schema. Ett dokument formaterat i XML-format beskrivs ofta av W3C XML Schema-dokument eller ISO Relax NG dokument. Dokument som skickas med hjälp av Bas-profilen bör vara beskrivna av ett schemadokument. Referensen till detta formatbeskrivningsdokument skrivs in i ett <Schema> element som ingår i <Reference> elementet. Två attribut skall definieras 'location' och 'version'.

- **location** innehåller en URI som är en referens till schema dokumentet.
- **version** innehåller en referens till schemadokumentets version.

```
<eb:Schema eb:location='urn:se:sfti:collaborationprocesses:BasicInvoice.xsd'  
eb:version='1.0'></eb:Schema>
```

Dessa två värden kan hämtas från dokumentationen som beskriver Samverkansprocessen där dokumenten definieras.

Se [ebMS] sektion 3.2.1.1 för mer information.

Det kan dock noteras att till exempel även EDIFACT-formaterade dokument kan översändas med hjälp av denna Bas-profil och <Schema> elementet bör då referera till en formatbeskrivning för EDIFACT-meddelandet. ~~formatbeskrivning~~.

Beskrivning:

Det är möjligt att beskriva ett meddelande genom att inkludera ett <Description> element i <Reference> elementet. Detta element kan inkluderas men är ej obligatorisk i Bas-profilen och skall ej påverka behandlingen av avsända och mottagna meddelanden.

```
<eb:Description xml:lang="se"> Fritext beskrivning av meddelande </eb:Description>
```

Attribut:

- 'xml:lang' är obligatorisk och måste definieras för varje <Description> element.

Se [ebMS] sektion 3.1.8 för mer information.

3.1.7. Synkronisering av meddelande med svar över samma förbindelse

För att ange, gentemot mottagare, att kvittensen skall skickas tillbaka över samma öppna förbindelse måste ett <SyncReply> element skickas med i varje meddelande och dess SOAP Header.

```
<eb:SyncReply eb:id='' eb:version='2.0' SOAP:mustUnderstand='1'  
SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next">  
</eb:SyncReply>
```

Attribut:

- 'id' kan anges för att unikt identifiera elementet
- 'version' är obligatorisk och skall anges med '2.0'
- 'SOAP:mustUnderstand' är obligatorisk och skall anges med '1'
- 'SOAP:actor' är obligatorisk och skall anges med "http://schemas.xmlsoap.org/soap/actor/next"

Se [ebMS] sektion 4.3 för mer information.

3.1.8. Begäran om transportkvittens

En avsändande part begär från en mottagande part en transportkvittens genom att i utgående meddelande inkludera ett <AckRequested> element i dess SOAP Header. AckRequested är obligatorisk i denna profil och får endast förekomma en gång i ett meddelande. Mottagande MSH svarar genom att inkludera ett <Acknowledgement> element i svaret. Signerade mottagningskvittenser får EJ enligt denna profil begäras av avsändare.

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

Attribut:

- 'id' kan anges för att unikt identifiera elementet
- 'version' är obligatorisk och skall anges med '2.0'
- 'SOAP:mustUnderstand' är obligatorisk och skall anges med '1'
- 'signed' är obligatoriskt och ska anges med 'false'
- 'SOAP:actor' är obligatorisk och skall anges med "urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH"

Se [ebMS] sektion 6.3.1 för mer information

3.2. Svarsmeddelanden

Ett svarsmmeddelande innehållande en transportkvittens eller felmeddelande skall skickas tillbaka över samma kommunikationsförbindelse som används för det initiala meddelandet.

Om ett meddelande framgångsrikt översänds till en mottagande MSH skall HTTP/S svar 2XX returneras tillsammans med svarsmmeddelandet som innehåller en ebMS Header med kvittens enligt avsnitt 3.2.1. I annat fall sänds felmeddelande enligt 3.2.2.

3.2.1. Transportkvittens från avsedd mottagare

En transportkvittens utgörs av ett <Acknowledgment> element i SOAP headern.

Meddelande för vilket transportkvittens lämnas skall uppfylla följande kriterier:

1. All information i meddelandet skall vara tillgänglig hos avsedd mottagare.
2. Meddelandet skall vara formaterat enligt ebMS paketerings- och meddelandepprinciper.

Verifiering och avsändande sker i avsedd mottagares MSH eller en förmedlingstjänst som agerar åt avsedd mottagare.

Ett meddelande anses enligt avtal ha kommit mottagaren **till handa** när transportkvittens har avsänts.

Ett transportkvittens skall uppfylla följande kriterier:

1. All information i meddelandet skall vara tillgänglig hos avsedd mottagare.
2. Meddelandet skall vara formaterat enligt ebMS paketerings- och meddelandepprinciper.
3. Kvittot skall ha ett format som avsedd mottagare enligt avtal kan hantera.

Regler för transportkvittens

Transportkvittens består enbart SOAP v1.1 Envelope som formaterats i XML format. För detta gäller reglerna i 3.1.1 – 3.1.5 med följande kompletteringar.

I kvittensmeddelandet skall elementet <Service> innehålla ”urn:oasis:names:tc:ebxml-msg:service” och <Action> skall innehålla ”Acknowledgment”

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

Elementet <ConversationId> skall innehålla samma konversationsidentitet som återfinns i det meddelande som kvitteras.

Elementet <Acknowledgment> skall finnas med i kvittensmeddelandets SOAP header.

```
<eb:Acknowledgment SOAP:mustUnderstand="1" eb:version="2.0"
  SOAP:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH">
  <eb:Timestamp>2005-10-28T17:29:01Z</eb:Timestamp>
  <eb:RefToMessageId>20040510-102030-28572@mittForetag.se</eb:RefToMessageId>
  <eb:From>
    <eb:PartyId eb:type="countrycode:organizationid">SE9876543210</eb:PartyId>
  </eb:From>
</eb:Acknowledgment>
```

Attribut:

- 'id' kan anges för att unikt identifiera elementet
- 'version' är obligatorisk och skall anges med '2.0'
- 'SOAP:mustUnderstand' är obligatorisk och skall anges med '1'
- 'SOAP:actor' är obligatorisk och skall anges med "urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH"

Det obligatoriska elementet <Timestamp> skall innehålla den tid som mottagande part anser vara tidpunkten när inkommande meddelande har kommit avsedd mottagare tillhanda. Tidpunkten anses gälla även för i meddelandet ingående dokument (ett eller flera).

Det obligatoriska elementet <RefToMessageId> skall innehålla samma meddelandeidentitet som återfinns i det meddelande som kvitteras.

Elementet <From> skall innehålla partsidentitet för den part som skapat svarsmeddelandet. Saknas elementet gäller <From> elementet enligt element <MessageHeader>.

I denna profil ingår EJ signerade mottagningskvittenser och därmed skall <Reference> element EJ anges.

Se [ebMS] sektion 6.3.2 för mer information

3.2.2. Felmeddelanden och Felrapportering

Felhantering och rapportering kan delas upp i 4 primära delar:

1. Kommunikationsprotokoll fel (HTTPS)
2. Paketerings problem (MIME)
3. SOAP problem
4. Meddelandehanteringsproblem [ebMS]

Se sektion 4.2 [ebMS] för detaljer.

3.2.2.1 HTTP/S fel

Om fel uppstår skall HTTP status 3XX, 4XX, eller 5XX returneras.

3.2.2.2 MIME Fel

Denna typ av fel rapporteras som genom att sända tillbaka ett SOAP meddelande med ett <ErrorList> element.

Se [ebMS] sektion 2.1.6 för mer information.

3.2.2.3 SOAP Fel

Om SOAP fel uppstår skall '500 "Internal Server Error"' enligt SOAP 1.1 specifikationen. En avsändande MSH måste vara beredd på att hantera SOAP Fault.

3.2.2.4 Fel i Meddelandehantering

Fel i meddelandehantering rapporteras genom att returnera <ErrorList> i SOAP headern i svarsmeddelandet. <ErrorList>, skall EJ vara med i avsända meddelanden utan endast i svar som innehåller felrapportering.

Se [ebMS] sektion 4.2 för mer information.

3.2.2.5 Rapportering av Inkompatibilitet med denna profil

Rapportering av inkompatibilitet med denna profil sker genom att mottagande MSH returnerar en <ErrorList> med felkoden "NotSupported".

3.3. Övrig information som EJ skall användas i meddelanden

ebXML ramverket innehåller ett antal valbara alternativ. Denna sektion beskriver de valbara möjligheter som EJ skall användas av meddelandehanteringstjänster som är kompatibla med denna Bas-profil.

3.3.1. <DuplicateElimination>

Genom att avsändaren skickar med ett <DuplicateElimination> element i SOAP headern kan avsändaren indikera att mottagaren skall kontrollera inkomna meddelande om dessa är en kopia av tidigare mottagna meddelande och om så är fallet skall dessa inte skickas vidare till bakomliggande affärssystem. Om <DuplicateElimination> elementet ej skickats med så hanteras överföringen enligt bästa möjliga insats (best effort). Denna valbara SOAP header skall EJ specificeras i denna Bas-profil.

Se [ebMS] sektion 3.1.7 för mer information.

3.3.2. <Signature>

Avsända meddelanden kan signeras digitalt genom att inkludera ett <Signature> element i SOAP headern. Denna valbara SOAP header skall EJ specificeras i denna Bas-profil.

Se [ebMS] sektion 4.1.1 för mer information.

3.3.3. <MessageOrder>

Avsändaren kan begära av mottagande MSH att avsända meddelanden skall skickas vidare till bakomliggande affärssystem i sekvensordning. Denna valbara SOAP header skall EJ specificeras i denna Bas-profil.

Se [ebMS] sektion 9 för mer information.

3.4. Meddelandehanterarens övriga tjänster

Meddelandehanteringsprogramvaror som realiserar denna profil skall EJ kräva av andra MSH att dessa stödjer Status eller Ping-Pong tjänsterna

Se [ebMS] sektion 7, 8 för mer information.

4. Exempel på meddelande

Följande exempel innehåller HTTP header, SOAP Header samt ett dokument paketerat m.h.a. MIME

```
POST /servlet/ebXMLhandler HTTP/1.1
Host: www.mottagare.se
SOAPAction: "ebXML"
Content-type: multipart/related; boundary="Boundary"; type="text/xml";
start="<ebxmhheader1@avsandare.com>"

--Boundary
Content-ID: <ebxmhheader1@avsandare.com>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"
  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="countrycode:organizationid">SE1234567890</eb:PartyId>
    </eb:From>
      <eb:To> <eb:PartyId eb:type="countrycode:organizationid">SE9876543210</eb:PartyId>
    </eb:To>
      <eb:CPAId>20040510:SE1234567890:SE9876543210</eb:CPAId>
      <eb:ConversationId>20040510:4567:SE1234567890</eb:ConversationId>
      <eb:Service>urn:sfti:services:documentprocessing:BasicInvoice</eb:Service>
      <eb:Action>incomingBasicInvoice</eb:Action>
      <eb:MessageData>
        <eb:MessageId>20051108-102030-28572@foretag.se</eb:MessageId>
        <eb:Timestamp>2005-11-08T11:12:12</eb:Timestamp>
      </eb:MessageData>
    </eb:MessageHeader>
    <eb:AckRequested SOAP:mustUnderstand="1" eb:version="2.0" eb:signed="false"
  SOAP:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH"/>
    <eb:SyncReply eb:id="" eb:version="2.0" SOAP:mustUnderstand="1"
      SOAP:actor="http://schemas.xmlsoap.org/soap/actor/next">
    </eb:SyncReply>
  </SOAP:Header>

  <SOAP:Body>
    <eb:Manifest eb:version="2.0">
      <eb:Reference xlink:href="cid:ebxmlpayload1@avsandare.com" xlink:type="simple">
        <eb:Description xml:lang="se"> Fritext beskrivning av meddelande </eb:Description>
        <eb:Schema eb:location="urn:se:sfti:collaborationprocesses:BasicInvoice.xsd"
      eb:version="1.0"> </eb:Schema>
      </eb:Reference>
    </eb:Manifest>
  </SOAP:Body>
</SOAP:Envelope>

--Boundary
Content-ID: <ebxmlpayload1@avsandare.se>
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<Faktura xmlns:sfti="urn:se:sfti">
</Faktura>

--Boundary--
```

4.1. Transportkvittens

```
HTTP/1.1 200 OK
SOAPAction: "ebXML"
Content-Type: text/xml

<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"
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" xmlns:eb="http://www.oasis-
open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
  <SOAP:Header xsi:schemaLocation="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">
    <eb:MessageHeader SOAP:mustUnderstand="1" eb:version="2.0">
      <eb:From>
        <eb:PartyId eb:type="countrycode:organizationid">SE9876543210</eb:PartyId>
      </eb:From>
      <eb:To>
        <eb:PartyId eb:type="countrycode:organizationid">SE1234567890</eb:PartyId>
      </eb:To>
      <eb:CPAId>20040514:SE1234567890:SE9876543210</eb:CPAId>
      <eb:ConversationId>20040514:4567:SE1234567890</eb:ConversationId>
      <eb:Service>urn:oasis:names:tc:ebxml-msg:service</eb:Service>
      <eb:Action>Acknowledgment</eb:Action>
      <eb:MessageData>
        <eb:MessageId>20051108-102030-28572@abc.se</eb:MessageId>
        <eb:Timestamp>2005-11-08T11:12:16</eb:Timestamp>
        <eb:RefToMessageId>20051108-102030-28572@foretag.se</eb:RefToMessageId>
      </eb:MessageData>
    </eb:MessageHeader>
    <eb:Acknowledgment SOAP:mustUnderstand="1" eb:version="2.0"
SOAP:actor="urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH">
      <eb:Timestamp>2005-11-08T11:12:16</eb:Timestamp>
      <eb:RefToMessageId>20051108-102030-28572@foretag.se</eb:RefToMessageId>
      <eb:From>
        <eb:PartyId eb:type="countrycode:organizationid">SE9876543210</eb:PartyId>
      </eb:From>
    </eb:Acknowledgment>
  </SOAP:Header>

  <SOAP:Body xsi:schemaLocation="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:Envelope>
```

5. Teknisk bilaga till transportprofil Bas

Bas-profilen kräver ej att parterna tar fram och använder ett ebXML Collaboration Protocol Agreement [ebCPPA] dokument. Dock kan det i vissa situationer vara av nytta för en eller båda parter att enkelt kunna skriva ned kommunikationsdetaljer i en teknisk bilaga, för automatiserad behandling och dokumentation. Bas-profilen erbjuder en möjlighet till detta genom profilbeskrivningen nedan. Innehållet i ett sådant dokument är endast en indikation på att en organisation kan hantera beskriven process, dokument etc. och skall ej anses som legalt bindande om ej annat avtalats.

I Bas-profilens tekniska bilaga kan följande information skrivas in:

- Identifierare [1..1] – Utställande parts identifierare på transportprofilen
- Start datum [0..1] – Datum när profilinformation börjar att gälla (fr o m kl 00.00.00)
- Slut datum [0..1] – Datum när profilininformation upphör att gälla (hela dygnet är inkluderat)
- Information om parten [1..1] – part vars process och tekniska information beskrivs i tekniska bilagan
 - Identifierare [1..1] på den organisation som vars funktion profilen beskriver (används i elementen <From> eller <To>, beroende på organisationens roll i samverkansprocessen)
 - Kontaktuppgifter [0..*]
 - Kontaktfunktion [0..1] – Kontaktens funktion/roll hos utställande part
 - Namn [1..1] – Namn på kontaktpunkt, person eller avdelning som hanterar frågor kring avsändande och mottagande av dokument enligt specificerad profil.
 - Kommunikationsdetaljer [0..*]
 - URI [0..1]
 - Kontaktsätt [0..1] – t ex telefon, fax, e-post, webb, SMS
 - Kontaktnummer [0..*] – kontaktnummer/kontaktadress vid angivet kontaktsätt
- Grundläggande Samverkansprocess [0..*]
 - Version [0..1] – samverkansprocessens version: <major> ’.’ <minor> ’.’ <revision>
 - Status [0..1] – samverkansprocessens status: [draft | standard]
 - Namn [1..1] – Identifikation av samverkansprocess
 - Roll [1..1] – Organisationens roll i samverkansprocessen [sändare | mottagare]
 - Mottagarens kommunikationsadress [0..1] (om mottagarroll)
 - Dokument schema [1..1] – för det dokument som ingår i samverkansprocessen
 - Service [1..1] – Mottagande tjänst
 - Action [1..1] – Specifik aktivitet inom mottagande tjänst

Se kapitlet om ”Definitioner av Core Components” (nedan) för mer information.

6. Realiseringsteknologier

6.1. Java – JAXM, SAAJ, Java Server Pages (JSP)

I Java ramverket finns det inbyggt stöd för ebXML kommunikation i ”Java Web Services Developer Pack”, se < <http://java.sun.com/webservices/webservicespack.html>> för mer information.

Se vidare information gällande JAXM Servlets om hur man enkelt kan implementera sändning och mottagning av meddelanden enligt denna profil. [JAXM] <<http://java.sun.com/xml/jaxm/>>

API och specifikation: < <http://java.sun.com/xml/downloads/jaxm.html>>

Se även ”SOAP with Attachments API for Java (SAAJ)”, <http://java.sun.com/xml/saa/index.jsp>

6.2. Kodexempel i Java

Följande kodexempel visar hur man enkelt kan skicka iväg ett dokument med hjälp av Open source programvaran Hermes.

```
package DemoEbxml;
import hk.hku.cecid.phoenix.message.handler.*;
import hk.hku.cecid.phoenix.message.packaging.*;
import java.net.URL;
import java.util.Date;
import javax.activation.DataHandler;

public class SftiSendEbxmlMessage implements MessageListener{

    public void SendEbXMLMessage(){
        try{
            ApplicationContext ac;
            ac = new ApplicationContext("20040514:SE1234567890:SE9876543210",
                "20040514:4567:SE1234567890",
                "urn:sfti:services:documentprocessing:BasicInvoice",
                "incomingBasicInvoice");

            // skapa en begäran
            Request mshReq = new Request(ac, new URL("https://localhost:8080/msh/"), this, "HTTP");

            // skapa ett meddelnde
            EbxmlMessage ebxmlMessage = new EbxmlMessage();
            MessageHeader msgHeader = ebxmlMessage.addMessageHeader();
            msgHeader.addFromPartyId("SE1234567890", "countrycode:organizationid");
            msgHeader.addToPartyId("SE9876543210", "countrycode:organizationid");
            msgHeader.setCpaId("20040514:SE1234567890:SE9876543210");
            msgHeader.setConversationId("20040514:4567:SE1234567890");
            msgHeader.setService("urn:sfti:services:documentprocessing:BasicInvoice");
            msgHeader.setAction("incomingBasicInvoice");
            String messageId = Utility.generateMessageId(new Date(), ebxmlMessage);
            msgHeader.setMessageId(messageId);
            msgHeader.setTimestamp(Utility.toUTCString(new Date()));

            // Koppla samman ett dokument med meddelandet
            AttachmentDataSource ads = new AttachmentDataSource("sfti.xml", "text/xml");
            DataHandler dataHandler = new DataHandler(ads);
            ebxmlMessage.addPayloadContainer(dataHandler, "contentId",
                "Fritextbeskrivning av meddelande");
            ebxmlMessage.addAckRequested(false);
            ebxmlMessage.addSyncReply();

            // Spara ändringar
            ebxmlMessage.saveChanges();

            // Skicka iväg meddelande.
            // Svaret kommer tillbaka genom anrop av onMessage()
            mshReq.send(ebxmlMessage);

            // Vänta på svar till onMessage()
            Thread.sleep(30000);
        }

        // felhantering
        catch(Exception e){
            System.err.println(e);
        }

        public URL getClientUrl(){
            return null;
        }

        //Svaret kommer tillbaka till denna s.k. Callback
        public void onMessage(EbxmlMessage ebxmlMessage){
            System.err.println("<<Message recived>>");
        }

        /**/
        public static void main(String[] args){
            SftiSendEbxmlMessage sendMessage = new SftiSendEbxmlMessage();
            sendMessage.SendEbXMLMessage();
        }
    }
}
```

}

7. Referenser

7.1. Normativa Referenser

- [CC] UN/CEFACT Core Components Technical Specification – Part 8 of the ebXML Framework, 15 November 2003, Version 2.01
- [ebCPPA] Oasis ebXML Collaboration-Protocol Profile and Agreement Specification Version 2.0, 23 September 2002
- [ebMS] ebXML Message Service Specification Version 2.0, 1 April 2002. OASIS ebXML Messaging Service Technical Committee
- [ISO3166] ISO 3166-1:1997 Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes. As amended by the ISO 3166 Maintenance agency
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol, HTTP/1.1", June 1999.
- [SSL3] A. Frier, P. Karlton and P. Kocher, "The SSL 3.0 Protocol", Netscape Communications Corp., Nov 18, 1996. 2780
- [SOAP] 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 08 May 2000, <<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>>
- [SOAPAttach] SOAP Messages with Attachments, John J. Barton, Hewlett Packard Labs; Satish Thatte and Henrik Frystyk Nielsen, Microsoft, Published Oct 09 2000 <<http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211/>>
- [XLINK] W3C XML Linking Recommendation, <<http://www.w3.org/TR/2001/REC-xlink-20010627/>>

7.2. Ej Normativa referenser

- [ebCPP] ebXML Collaboration Protocol Profile and Agreement specification, Version 1.0, published 10 May, 2001, <<http://www.ebxml.org/specs/ebCCP.doc>>
- [JAXM] Java API for XML Messaging, <<http://java.sun.com/xml/jaxm/>>
- [XMLSchema] W3C XML Schema Recommendation, <<http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/>>
<<http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>>
<<http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>>

7.3. ebMS Säkerhetsprofiler

Present in baseline MSH		Persistent digital signature	Non-persistent authentication	Persistent signed receipt	Non-persistent integrity	Persistent confidentiality	Non-persistent confidentiality	Persistent authorization	Non-persistent authorization	Trusted timestamp	Description of Profile
✓	Profile 0										no security services are applied to data
✓	Profile 1	✓									Sending MSH applies XML/DSIG structures to message
	Profile 2		✓						✓		Sending MSH authenticates and Receiving MSH authorizes sender based on communication channel credentials.
	Profile 3		✓				✓				Sending MSH authenticates and both MSHs negotiate a secure channel to transmit data
	Profile 4		✓		✓						Sending MSH authenticates, the Receiving MSH performs integrity checks using communications protocol
	Profile 5		✓								Sending MSH authenticates the communication channel only (e.g., SSL 3.0 over TCP/IP)
	Profile 6	✓					✓				Sending MSH applies XML/DSIG structures to message and passes in secure communications channel
	Profile 7	✓		✓							Sending MSH applies XML/DSIG structures to message and Receiving MSH returns a signed receipt
	Profile 8	✓		✓			✓				combination of profile 6 and 7
	Profile 9	✓								✓	Profile 5 with a trusted timestamp applied
	Profile 10	✓		✓						✓	Profile 9 with Receiving MSH returning a signed receipt
	Profile 11	✓					✓			✓	Profile 6 with the Receiving MSH applying a trusted timestamp
	Profile 12	✓		✓			✓			✓	Profile 8 with the Receiving MSH applying a trusted timestamp
	Profile 13	✓				✓					Sending MSH applies XML/DSIG structures to message and applies confidentiality structures (XMLEncryption)
	Profile 14	✓		✓		✓					Profile 13 with a signed receipt
	Profile 15	✓		✓						✓	Sending MSH applies XML/DSIG structures to message, a trusted timestamp is added to message, Receiving MSH returns a signed receipt
	Profile 16	✓				✓				✓	Profile 13 with a trusted timestamp applied Profile
	Profile 17	✓		✓		✓				✓	Profile 14 with a trusted timestamp applied
	Profile 18	✓							✓		Sending MSH applies XML/DSIG structures to message and forwards authorization credentials [SAML]
	Profile 19	✓		✓					✓		Profile 18 with Receiving MSH returning a signed receipt
	Profile 20	✓		✓					✓	✓	Profile 19 with the a trusted timestamp being applied to the Sending MSH message
	Profile 21	✓		✓		✓			✓	✓	Profile 19 with the Sending MSH applying confidentiality structures (XML-Encryption)
	Profile 22					✓					Sending MSH encapsulates the message within confidentiality structures (XML-Encryption)

7.4. Definitioner av Core Components

7.4.1. Formaliserad beskrivning av transportprofilens detaljer

I kapitel 5 visas på möjligheten arbeta med formaliserade ”visitkort” med transportprofilens detaljer, detta som ett frivilligt tillval vid implementering av transportprofilen. De som överväger implementering bör förslagsvis ta [ebCPA] som en generell utgångspunkt och därifrån transformera/reducera till nedanstående beskrivning, som bygger på preliminära Core Components publicerade av UN/Cefact 2004. (Not. Exemplet visar en struktur som passar rollen ”mottagare”.)

