



2004:22

Förstudierapport Standardmeddelanden Bilaga 1:

Riktlinjer för utveckling av
standardmeddelanden (förslag)



Bilaga 1: Riktlinjer för utveckling av standardmeddelanden (förslag)

Inledning

Detta dokument ger rekommendationer om både tekniska och juridiska principer för att definiera standardmeddelanden för frekvent utbyte av ärendedata, i första hand mellan myndigheter. Syftet är att stödja den samordnade utvecklingsprocess som behövs för informationsutbyte via konceptet för standardmeddelanden.

Bilaga 1.1, som behandlar konstruktionsprinciper för XML-scheman, skall stödja införandet av enhetliga tekniska ansatser vid utveckling av de standarddokument som samordnar informationsutbytet och bearbetningen av den utbytta informationen.

För att underlätta användningen bör myndigheterna även publicera Nyttjanderegler för utbytet av Standardmeddelanden. Nyttjandereglerna bör fungera som villkor i de fall utbytet sker mellan myndigheter och som ett avtal om mottagaren inte är en statlig myndighet. Det är en fördel om myndigheternas Nyttjanderegler kan utformas på ett likartat sätt. För att stödja villkorsutformningen formuleras vissa gemensamma mallvillkor i bilaga 1.2 och ges ett förslag till checklista i bilaga 1.3.

Innehåll:

Bilaga 1: Riktlinjer för utveckling av standardmeddelanden (förslag)

	1
<i>Inledning</i>	<i>1</i>
Bilaga 1.1 – vägledning för XML-schemakonstruktion	3
<i>Målgrupp</i>	<i>3</i>
<i>Relaterade dokument</i>	<i>3</i>
<i>Centrala egenskaper hos XML-baserade Standarddokument</i>	<i>4</i>
Enkla	4
Återanvändbara	4
Stabila	4
<i>XML standarder för schema konstruktion</i>	<i>5</i>
<i>Utvecklingsprocess av XML-scheman för standardmeddelanden</i>	<i>5</i>
<i>Internationella standarder och initiativ</i>	<i>5</i>
<i>Informationsmodellering</i>	<i>5</i>
<i>Myndighetsgemensamma termer</i>	<i>6</i>
<i>Designa informationsstrukturen/ schemahierarkin</i>	<i>6</i>
<i>XML-kodning</i>	<i>6</i>
<i>Publicering av XML-scheman</i>	<i>7</i>
<i>Schemahierarki</i>	<i>7</i>

<i>Schemanivåer</i>	7
Centrala typer och element	8
Paket	9
Arkitekturella scheman	9
Gränssnittsscheman eller toppnivå-scheman	10
<i>Hur många schemanivåer skall användas?</i>	10
<i>Datatyper, element och attribut</i>	10
Globala eller lokala	10
Svaga eller starka typ- och elementdefinitioner?	11
<i>Namespaces</i>	12
<i>Namngivningsregler</i>	13
<i>Dokumentation/Metadata</i>	13
<i>Kompatibilitet och versionshantering</i>	14
<i>Versionsbeteckning</i>	15
<i>Namespaces och versionshantering</i>	16
<i>Exempel på Standardmeddelande</i>	17
Infoutbyte i ett Standardmeddelande	17
Strukturering via XML Schema	17
<i>Några exempel på relaterade initiativ</i>	21
<i>Frågeställningar att beakta</i>	21
Begränsningar av informationsutlämning	21
Prestanda	21
Övergripande förvaltning myndighetsinternt samt centralt	22
<i>Hur kommer man igång?</i>	22
Bilaga 1.2 – vägledning – mallvillkor (förslag)	24
<i>Målgrupp</i>	24
<i>Relaterade dokument</i>	24
<i>Förslag till mallvillkor</i>	25
Bilaga 1.3 – vägledning nyttjandevillkor – checklista (förslag)	27
<i>Målgrupp</i>	27
<i>Relaterade dokument</i>	27
<i>Förslag till Checklista</i>	28

Bilaga 1.1 – Vägledning för XML-schemakonstruktion

Denna bilaga är en vägledning om tekniska principer för att definiera standardmeddelanden och konstruera XML-baserade standarddokument, dvs. leverans- och mottagningsformat för frekvent utbyte av ärendedata, i första hand mellan myndigheter. Modellen är dock generell och kan tillämpas även för informationsutbyte med privata organisationer. Utvecklingen av de riktlinjer som presenteras har skett tillsammans med representanter från ett antal av de myndigheter och IT-konsulter som kommer att producera standardmeddelanden.

Syftet med vägledningen är att stödja den tekniska utvecklingsprocess som hanteras lokalt på en IT-avdelning/myndighet eller via konsulter för att ta fram de informationsstrukturer som behövs för informationsutbyten via konceptet för standardmeddelanden. XML-vägledningen skall assistera enhetliga struktureringsansatser vid utveckling av standardmeddelanden, för att samordna informationsutbyten mellan myndigheter och bearbetning av den utbytta informationen.

Målgrupp

Vägledningen riktar sig till myndigheters IT-ledning, IT-arkitekter, systemansvariga för register, databaser och verksamhetssystem, systemdesigners och IT-konsulter. Eftersom den teknik som skall användas för att strukturera och definiera standardmeddelanden är XML, förutsätter vägledningen att läsaren har kunskap om system för utbyte av strukturerad data, XML, XML Schema och XML Namespaces. Vägledningen anvisar rekommenderade lösningar men förklarar inte dessa i detalj utan förutsätter att läsaren kan eller studerar föreslagna XML Schema-tekniker i annan, specialiserad litteratur.

Relaterade dokument

Dokumentet ”Standardmeddelanden – en förstudie” är skrivet med inriktning på målgruppen verksamhetsansvariga och beskriver nyttan med standardmeddelanden samt hanteringen av dessa hos myndigheter och gemensamt emellan myndigheter ur verksamhetsperspektiv. I övriga bilagor ges vägledning kring de juridiska aspekterna av ett dokumentutbyte med standardmeddelanden.

Förstudien kan med fördel läsas som introduktion till detta dokument, eftersom sammanhanget är viktigt att känna till innan den tekniska konstruktionen påbörjas.

Centrala egenskaper hos XML-baserade Standarddokument

Denna vägledning syftar till att stödja framtagande av XML-scheman för standardmeddelanden med följande utmärkande egenskaper.

Enkla

Standardmeddelandena skall vara lätta att förstå. Ett standardmeddelandes format och innehåll skall vara definierat och beskrivet på ett sådant sätt att det är enkelt för användare med begränsade erfarenheter av datastrukturering och XML schema att förstå och implementera tillämpningar som stödjer standardmeddelandet.

Återanvändbara

Standardmeddelanden skall implementeras på ett sådant sätt att återanvändning underlättas. Det är eftersträvansvärt att kunna återanvända tekniska lösningar och arbetsmetoder vid konstruktion av standardmeddelanden vid olika myndigheter. Detta underlättas av att definitionen av de olika standardmeddelandena följer enhetliga riktlinjer och rekommendationer. Det är önskvärt att en koordinering av typer, element och attribut görs mellan olika Standardmeddelanden, då detta möjliggör återanvändning av schema-definitioner samt underlättar att kombinera uppgifter från olika Standardmeddelanden. Koordineringen bör göras på en så hög nivå som möjligt, inte enbart internt inom en myndighet utan även gemensamt mellan flera myndigheter. På sikt eftersträvas att bygga upp ett gemensamt bibliotek med återanvändbara definitioner, modeller och scheman som sedan kan användas för att sätta ihop olika standarddokument.

Stabila

Designen av standardmeddelandena möjliggör en minimering av effekterna av framtida **ändringar** av gränssnitten mellan myndigheter. Ett standardmeddelandes innehållsdefinition, dvs. de XML-scheman som strukturerar meddelandet, ska vara stabil och det ska finnas en tydlig strategi för uppdateringar som möjliggör största möjliga kompatibilitet både framåt och bakåt. Hur detta löses praktiskt kan variera baserat på de olika myndigheternas tekniska lösningar.

Denna vägledning tar upp ett antal rekommendationer som syftar till att underlätta för myndigheter och konsulter att konstruera standardmeddelanden med dessa egenskaper.

XML standarder för schema konstruktion

XML-baserade Standardmeddelanden ska definieras enligt W3C:s XML-rekommendationer. Statskontoret har publicerat en vägledning ”SV 1:4: XML-teknik och metadata” som bland annat tar upp aktuella XML-rekommendationer. De standarder som i första hand berör konstruktion av XML-baserade Standardmeddelanden är:

- eXtensible Markup Language (XML) 1.0 (Third edition)
- XML Schema Version 1.0
- Namespaces in XML Version 1.0

Utvecklingsprocess av XML-scheman för standardmeddelanden

I denna vägledning beskrivs enbart konstruktionsprinciper för XML Schema för att strukturera standardmeddelanden. Vilken information som skall erbjudas som standardmeddelande antas redan vara bestämt av verksamhetsansvarig. Nedan beskrivs kort de olika steg som bör beaktas i utvecklingen av de XML-scheman som skall strukturera ett standard-meddelande. Schemakonstruktionen bör t.ex. bygga på en informationsmodell.

Internationella standarder och initiativ

Inom många områden finns det redan erkända internationella standarder eller branschinitiativ. Undersök då om dessa kan användas direkt eller anpassas till svenska krav. Exempel på relevanta sådana kan vara ebXML (e-affärer), XBRL (Redovisning och revision), HR-XML (personal och arbetsmarknad) och GML (geografisk information).

Informationsmodellering

Definiera och modellera på ”hög, konceptuell” nivå den information som skall utbytas. Använd gärna UML eller liknande modelleringspråk. Identifiera alla datatyper, element och attribut i den datamodell som ska ingå i Standardmeddelandet.

Myndighetsgemensamma termer

Ett stort antal termer i de uppgifter som skickas emellan myndigheter används av flertalet myndigheter, t.ex. namn och adressuppgifter. Därför bör dessa helst definieras gemensamt, alternativt ges en tydlig ansvarsfördelning, så att de kan definieras av ansvarig myndighet i återanvändbara XML-scheman som publiceras. Proceduren för att definiera centrala komponenter återstår att utforma, tills vidare får det hanteras av den närmast ansvariga myndigheten i samråd med berörda intressenter.

På sikt behöver ett myndighetsgemensamt öppet register inrättas, där systemutvecklare kan söka vilka standarddokument, scheman, centrala element och typer som redan är definierade och klara att återanvända eller anpassa.

Designa informationsstrukturen/ schemahierarkin

Bestäm den överordnade schemastrukturen baserat på schemadesignriktlinjerna i detta dokument. Hur många nivåer skall schemahierarkin ha, vad passar bäst för den aktuella tillämpningen? Hur skall komponenterna paketeras för att nå maximal nytta?

XML-kodning

Om man idag har en existerande informationsstruktur beskriven i ett annat format än XML Schema, t.ex. DTD:er, så bör man definiera ett XML Schema för de delar man önskar inkludera i ett Standardmeddelande. Detta garanterar att utvecklarna och systemen endast behöver förstå XML Scheman vid användning av Standardmeddelanden. Det är inte att rekommendera att man endast gör en direktkonvertering, då man bör se till att korrekt använda typdefinitioner, element, namespaces, moduleringsmöjligheter etc. Däremot kan man internt fortsätta att använda det existerande formatet.

Det är lämpligt att man inom en myndighet placerar de termer man finner centrala och myndighetsgemensamma i ett eller flera separata scheman som sedan kan göras tillgängliga för alla myndigheter, alternativt ersättas med gemensamma scheman från annan ansvarig myndighet eller gemensamt utvecklade scheman. Exempel på komponenter av denna typ som kan bli aktuella är adresser, personnummer, organisationsnummer mm.

Publicering av XML-scheman

För att utbyta data via XML-baserade Standardmeddelanden krävs att den myndighet som erbjuder standardmeddelandet har definierat och tillgängliggjort XML-scheman för meddelandeformaten. Inledningsvis bör detta ske på den egna myndighetens publika webbplats, på sikt bör en myndighetsgemensam sökfunktion och katalog inrättas.

Schemahierarki

Ett standardmeddelande skall definieras med XML Schema. Även om riktlinjerna är tydliga på det området finns det flera variationer på hur en XML-schemadefinition av ett standardmeddelande kan se ut. Eftersom återanvändning av XML-scheman internt på myndigheter, men även mellan myndigheter, är centralt för konceptet så skall detta vara i fokus vid schemadesignen och utvecklingen. Därför bör de termer som andra myndigheter kan tänkas behöva nyttja i konstruktion av andra standardmeddelande vara definierade i ett eget återanvändbart schema, alternativt tillsammans med angränsande termer. Termerna fromDatum och tomDatum skulle t.ex. kunna definieras i samma XML Schema.

XML-scheman kan grupperas i olika varianter med olika syften. Även om alla dessa varianter möjligtvis inte behövs för en första definition av ett Standardmeddelande så är det ändå lämpligt att tidigt skapa en strategi för hur organisationen skall strukturera sina framtida scheman. Följande varianter bör beaktas i en organisation som önskar skapa återanvändbarhet i sina informationsmodeller.

Schemanivåer

Följande figur visar ett exempel på en schemahierarki med fyra nivåer. De olika nivåerna beskrivs i efterföljande avsnitt.

<code><xs:element name="Kontaktperson" type="PersonNamn" /></code>
<code><xs:element name="Kontaktadress" type="PostAdress" /></code>
<code><xs:element name="Omfragad" type="PersonNr" /></code>
<code><xs:element name="Bidragmottagare" type="PersonNamn" /></code>
<code><xs:element name="KassaId" type="xs:positiveInteger" /></code>

Exempel 2 Exempel på tänkbara centrala element

Paket

Dessa scheman grupperar centrala typer och element, t.ex. utifrån olika kunskapsdomäner. Paketerna är tänkta att underlätta vid arbete med schema-konstruktion genom att inkludering av ett paket ger tillgång till samtliga däri refererade typer och element.

Exakt hur paket struktureras kommer att vara beroende av vilka struktureringsprinciper man väljer. Strukturen kan följa ansvarsområden, verksamhetens organisation, systemstruktur eller andra tydliga domäner.

Exempel på tänkbar paketering av ovanstående centrala typer och element skulle vara ett myndighetsövergripande "Individ"-paket som innehåller PostAdress, PersonNamn och PersonNr typerna som är definierade i egna scheman, se exemplet nedan.

```
<?xml version="1.0" encoding="UTF-8" ?>
<schema targetNamespace="http://schema.se/myndighet"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:ind="http://schema.se/myndighet"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified" version="1.0">
  <include schemaLocation="PostAdress.xsd" />
  <include schemaLocation="PersonNamn.xsd" />
  <include schemaLocation="PersonNummer.xsd" />
</schema>
```

Figur 2: VD_DigTypesPackage.xsd, xml-schema-exempel hämtat från den danska xml-schemadatabasen Infostrukturbasen.

Arkitekturella scheman

Dessa scheman bygger på centrala typer och element. Till skillnad från Paket lägger arkitekturella scheman till struktur och semantik och ska representera en specifik tillämpning av en domäns centrala typer och element. Denna schemanivå behöver i många fall inte tillämpas.

Gränssnittsscheman eller toppnivå-scheman

Ett gränssnittsschema definierar gränssnittet mellan två system, dvs. den struktur som informationen i ett standarddokument skall ha. Ett gränssnittsschema består oftast av delar från ett eller flera arkitekturella scheman och paket.

För att minska komplexiteten och undvika beroenden mellan Standardmeddelanden är det lämpligt att ha ett separat schema per Standarddokument på gränssnittsnivå. Eftersom ett standardmeddelande är definierat av en fråga eller ett svar, kommer det att krävas två scheman på gränssnittsnivå för att definiera en komplett transaktion, fråga och svar.

Hur många schemanivåer skall användas?

Inget generellt krav ställs på antalet nivåer, det får avgöras av varje myndighet vad som passar bäst. Huvudsaken är att behovet av återanvändning beaktas i designen av standardmeddelanden.

Under arbetet med att utveckla XML-scheman för standardmeddelanden kommer schemahierarkin att uppstå stegvis. Även om man initialt börjar med ett enstaka Standardmeddelande så är det lämpligt att tidigt identifiera centrala typer och element då återanvändning av dessa minimerar arbetsinsatsen vid kommande schemakonstruktion.

Hur många schemanivåer som bör användas mellan gränssnittsschemat på översta nivån och de scheman som innehåller centrala termer och element beror på komplexiteten. Beskrivningen av paket och arkitekturella scheman är till för att visa exempel på hur en schemastruktur kan se ut. I följande avsnitt med exempel på alternativa standardmeddelanden kan det tredje exemplet framhållas som den schemastruktur som har bäst fokus på återanvändning.

Datatyper, element och attribut

Globala eller lokala

Datatyper, element och attribut kan vara antingen globala eller lokala. De är globala om de är definierade direkt under XML Schema-elementet `schema` och lokala om de ingår i något annat element. Alla element som skall vara adresserbara skall vara definierade som globala element.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://schema.se/test/csn/studiestodFraga/1/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```

xmlns="http://schema.se/test/csn/studiestodFraga/1/"
version="1.0"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <!-- information borttagen i exemplet -->
    <xs:element name="studiestodFraga">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="kassaId"
type="xs:positiveInteger"/>
            <!-- information borttagen i exemplet -->
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>

```

I ovanstående exempel är elementet `studiestodFraga` globalt och `kassaId` lokalt. Globala datatyper, element och attribut är adresserbara och återanvändbara av andra scheman. Lokala datatyper, element och attribut är anonyma och kan endast användas inom det element där de definieras.

I scheman som är tänkta som bas för återanvändning skall datatyper alltid definieras som globala. Dessutom ska samtliga element som är tänkta för återanvändning deklarerats som globala.

För rena gränssnittsscheman, dvs. det schema som definierar själva standarddokumentet på översta nivån, kan det vara lämpligt att endast ha rotelementet deklarerat som globalt. Detta minskar risken för beroenden till andra scheman och gör schemat mer överblickbart.

Attribut bör endast användas för metadata. Använd alltid element för data som ska distribueras i Standardmeddelandet.

Svaga eller starka typ- och elementdefinitioner?

För varje datatyp, element och attribut bör det bestämmas vilka begränsningar som kan gälla. Begränsningarna implementeras sedan med hjälp av typkontroll där det är möjligt, då detta bland annat ger möjlighet att kontrollera innehållet i instanser redan vid validering. Typkontroll kan användas för kardinalitet antal (max, min och obligatoriskt), ordning, val och formatrestriktioner och definieras med hjälp av olika konstruktioner i XML Schema. Antal för element definieras med hjälp av XML Schema-attributen `minOccurs` och `maxOccurs` (default är att elementet är obligatoriskt och endast får förekomma en gång), antal för attribut kan definieras med XML Schema-attributet `use` som kan ges något av värdena `optional/required/prohibited`. För ordning på element används XML Schema-elementet `sequence` och för val `choice`. Formatrestriktioner för datatyper definieras med XML Schema-elementet

restriction i kombination med den formatbegränsning som önskas, exempelvis min- och maxvärden, mönstermatchning eller antal decimaler.

Ett annat sätt att begränsa innehållet i ett element är användandet av värdelistor. Värdelistor innebär listor och sammanställningar av tillåtna värden i ett element. Värdelistor som kan tänkas användas i flera element bör definieras som globala datatyper och på så vis göras återanvändbara.

Namespaces

Användning av namnrymder är central för möjligheten att återanvända typer och element utan att drabbas av namnkollisioner. Användning av kvalificerade namnrymder ökar möjligheten för användare att tolka den datamodell som beskrivs i ett schema då det tydligare framgår vilket ursprung ett element eller en typ har.

Namnrymder bör användas i alla scheman. Varje schema bör ha ett TargetNamespace. Ett TargetNamespace är den namnrymd som alla nydefinierade datatyper och element i schemat kommer att knytas till.

Det är önskvärt att alla namnrymder kvalificeras i användarnas tillämpade Standardmeddelanden och därför bör XML Schema-attributet elementFormDefault definieras som "qualified" i alla scheman.

```
<?xml version="1.0" encoding="UTF-8"?>
  <xs:schema
targetNamespace="http://schema.se/test/csn/studiestodFraga/1/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schema.se/test/csn/studiestodFraga/1/"
  version="1.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
    <!-- information borttagen i exemplet -->
    <xs:element name="studiestodFraga">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="kassaId"
type="xs:positiveInteger"/>
        <!-- information borttagen i exemplet -->
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

Default-namnrymder skall inte användas i XML scheman för standardmeddelanden. Definiera xs som namnrymdsprefix för XML Schema-name-

space-namnrymden (<http://www.w3.org/2001/XMLSchema>) och använd konsekvent detta prefix i alla scheman.

Namngivningsregler

Dokumentera valda namngivningsregler och se till att de används konsekvent i alla schemadokument. Allt eftersom myndighetsgemensamma namngivningsregler tas fram bör dessa snarast integreras med organisationens. Eventuella lokala avvikelser från de gemensamma reglerna måste tydligt förklaras och dokumenteras.

Namnregler behöver etableras för

- Namespaces
- Scheman
- Typer, element och attribut.

Utgå från eventuella namngivningsstandarder för datamodeller som organisationen redan använder. Om inga sådana finns kan ISO 11179 (Specification and Standardization of data elements) tas som en utgångspunkt. Detta är en standard framtagen för koordinering av metadatasamlingar. I denna standard ingår en del (del 5) som behandlar namngivnings- och identifikationsregler för dataelement. Även t.ex. ebXML Core Components och relaterade internationella initiativ innehåller förslag på namngivningsregler som kan användas som underlag.

Dokumentation och metadata

Bestäm hur scheman ska dokumenteras, utgående från ambitionen på återanvändbarhet och organisationen för schemautveckling. Utgå ifrån hur mycket information man behöver för att kunna skapa ett schema som passar in i organisationens schemastruktur samt hur mycket information man behöver för att kunna hitta, förstå och använda organisationens Standardmeddelanden. Ange gärna relevanta metadata enligt Dublin Core-modellen.

Scheman bör främst dokumenteras via användning av XML Schema elementet `annotation` i själva schemadokumentet samt kompletteras med exempel på instansdokument, dvs. xml-dokument som korrekt använder schemat.

```
<xs:annotation>
  <xs:documentation xml:lang="sv">
    Standardmeddelade för frågor om studiestöd till CSN
    2004-01-12 Skapad av: NN
  </xs:documentation>
  <xs:documentation xml:lang="en">
```

```
This would be a description that would be relevant to
use if the message was planned to be used by
english speaking users.
</xs:documentation>
</xs:annotation>
```

Även namngivningsregler och liknande bör göras tillgängliga då de underlättar förståelsen av strukturen i ett schemadokument. Vem som ansvarar för schemat bör också ingå i dokumentationen i schemat.

Kompatibilitet och versionshantering

Standardmeddelanden är gränssnitt och bör designas för att ge största möjliga stabilitet för användarna. Detta kräver att man bestämmer en ambitionsnivå vad gäller framåt- och bakåtkompatibilitet gentemot användarna, dvs. de myndigheter som använder standarddokument för informationsutbyte.

Framåtkompatibilitet innebär möjligheten för en mottagare att utan uppgradering klara att ta emot nya versioner av ett svardsdokument med samma resultat som vid mottagande av den gamla versionen. Detta innebär att mottagare av XML-dokument, givet att dokumentet passerar validering, måste ignorera attribut och element som de ej känner igen.

Bakåtkompatibilitet innebär möjligheten för en avsändare att kunna fortsätta skicka gamla versioner av ett dokument efter det att meddelandemottagarens system har uppgraderats. Detta uppnås genom att meddelanden kan utökas med valbara element och/eller attribut.

På sikt bör regler för publicering av nya schemaversioner för Standardmeddelanden definieras. Centralt är frågan kring hur nya schemaversioner skall nå alla användare samt hur inkompatibla ändringar ska introduceras.

Dessutom bör även vid kompatibla ändringar en plan för om och hur samtliga användare bör uppgradera till de nya versionerna tas fram. Att behöva stödja flera samtidiga versioner innebär extra arbete i form av verifiering att samtliga versioner fungerar.

När ambitionerna för kompatibilitet är klarlagda ska regler för versionshantering av scheman definieras och beskrivas. Provkör gärna reglerna manuellt för att avgöra möjligheten att upprätthålla ambitionerna gällande kompatibilitet.

Versionsbeteckning

XML Schema attributet `version` skall användas i samtliga scheman.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="http://schema.se/test/csn/studiestodFraga/1/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schema.se/test/csn/studiestodFraga/1/"
  version="1.3b"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <!-- information borttagen i exemplet -->

</xs:schema>
```

Versionsbeteckningar bör t.ex. ske enligt formen `Release.Revision`. Stega releasenumret vid inkompatibla ändringar, stega revisionsnumret vid kompatibla ändringar. Om det uppstår behov av att göra preliminära utgåvor kan dessa kompletteras med en utkastsbokstav. Exemplet 1.3b skulle då beteckna den andra (b) preliminära utgåvan av den tredje (.3) kompatibla ändringen sedan den första utgåvan (1) av schemat.

Det kan vara önskvärt att i schemadefinitionen tvinga fram angivande av använd schemaversion i dokumentinstanserna (de XML-dokument som skickas) för att möjliggöra spårbarhet vid framtida felsökning. Detta görs genom definition av ett obligatoriskt attribut med namnet `SchemaVersion` i Standardmeddelandets rotelement.

```
<xs:element name="studiestodFraga">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="kassaId"
type="xs:positiveInteger"/>
      <!-- information borttagen i exemplet -->
    </xs:sequence>
    <xs:attribute name="schemaVersion" type="xs:decimal"
use="required" />
  </xs:complexType>
</xs:element>
```

Denna konstruktion innebär att en elementet `studiestodfraga` måste innehålla attributet `SchemaVersion`. Det innebär ingen kontroll av att innehållet i attributet matchar `version` attributet i schemat vid valideringen men det ger en startpunkt för manuell felsökning. Det ger också en möjlighet att bygga systemstöd för en automatisk kontroll om så önskas, exempelvis om man systemmässigt vill möjliggöra bakåtkompatibilitet vid schema-

inkompatibilitet, genom att validera gamla xml-dokument mot den gamla schemaversionen. En studiestödfråga kan enligt riktlinjen ovan se ut som följande xml-dokument:

```
<?xml version="1.0" encoding="UTF-8"?>
<studiestodFraga xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:schemaLocation="http://schema.se/test/csn/studiestodFraga/1/
studiestodFraga.xsd"
xmlns="http://schema.se/test/csn/studiestodFraga/1/"
schemaVersion="1.0">
    <kassaId>1</kassaId>
    <kundPersonNr>19441231-0123</kundPersonNr>
    <fromDatum>2004-01-12</fromDatum>
</studiestodFraga>
```

Namespaces och versionshantering

Inkludering av versionsbeteckning i identiteten för en namnrymd innebär att en ny version skapar en ny namnrymd och scheman och instanser som refererar till den gamla namnrymden måste uppdateras. Detta kan vara ett önskvärt beteende i situationer då ändringar är inkompatibla, det vill säga att gamla implementationer av schemat inte längre ska fungera om en ny release av ett schema presenteras. Men i många fall är det dock inte önskvärt att tvinga fram uppdateringar av tillämpningar när en ny version av schemat skapas.

Om man exempelvis tar en studiestödsförfrågan och utökar dagens fromDatum med ett tomDatum.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://schema.se/test/csn/studiestodFraga/1.1/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://schema.se/test/csn/studiestodFraga/1.1/"
version="1.1"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:annotation>
    <!-- information borttagen i exemplet -->
  </xs:annotation>
  <xs:element name="studiestodFraga">
    <xs:complexType>
      <xs:sequence>
        <!-- information borttagen i exemplet -->
        <xs:element name="fromDatum"
type="xs:date"/>
        <xs:element name="tomDatum" type="xs:date"
minOccurs="0" maxOccurs="1"/>
```

```
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

Detta är tänkt att ge en utökad möjlighet att få studiestödsinformation för en avgränsad tidsperiod även före ett angivet datum. Om man genomför denna schemaändring och byter versionsbeteckning (stegning till 1.1 i exemplet ovan) i namnrymsidentiteten så kommer samtliga implementationer av denna fråga bli ogiltiga. Detta oavsett om eller när avsändaren tänkt börja använda den nya parametern. Om namnrymsidentiteten hade lämnats orörd hade avsändaren kunnat fortsätta använda sina existerande implementationer och själv välja när man vill börja använda den nya parametern.

Exempel på Standardmeddelande

Dessa exempel baserar sig på en förfrågan om studiestöd från en A-kassa till CSN.

Infoutbyte i ett Standardmeddelande

Nedan syns en instans av standarddokumentet ”studiestödsfråga”. Detta xml-dokument validerar mot schemat studiestodFraga.xsd, vars definition presenteras i tre olika alternativ i efterföljande avsnitt.

```
<?xml version="1.0" encoding="UTF-8"?>
<studiestodFraga xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schema.se/test/csn/studiestodFraga/1/ studiestodFraga.xsd"
  xmlns="http://schema.se/test/csn/studiestodFraga/1/">
  <kassald>1</kassald>
  <kundPersonNr>19441231-0123</kundPersonNr>
  <fromDatum>2004-01-12</fromDatum>
</studiestodFraga>
```

Exempel: Standardmeddelande för att utbyta information om en/flera individ/er har studiemedel eller ej.

Strukturering via XML Schema

Tre olika alternativ till xml-scheman presenteras som definierar standarddokumentet ovan. **Alternativ 3 är den lösning som är mest anpassad för återanvändning och passar därför bäst för implementering.** Anledningen till att även de andra alternativen presenteras är för att tydligt kunna illustrera och diskutera innebörden av att lösningen bör vara fokuserad på återanvändning.

1. **Alternativ 1** består av endast ett schema på gränssnittsnivå, utan återanvändningsmöjligheter.

2. **Alternativ 2** består av ett schema, men möjliggör adressering och återanvändning av samtliga element och typer.
3. **Alternativ 3** består av två olika scheman, ett (3a) som implementerar en global datatyp för adressering och återanvändning i andra scheman och ett annat (3b) som implementerar gränssnittsschemat med hjälp av den globala datatypen.

Alternativ 1: Detta schema är definierat helt utan möjlighet till återanvändning. Inga element eller typer utom själva elementet studiestodsfråga kan adresseras separat i dokument som baseras på det här schemat. Detta följer ett designmönster som kallas RussianDoll, eftersom det "yttersta" elementet döljer övriga element på samma sätt som den "yttersta" ryska dockan döljer de mindre dockorna inuti.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="http://schema.se/test/csn/studiestodFraga/1/"
  xmlns="http://schema.se/test/csn/studiestodFraga/1/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="1.0">
  <xs:annotation>
    <xs:documentation xml:lang="sv">
      Standardmeddelade för frågor om studiestöd till CSN (Alternativ 1)
      Detta schema är konstruerat enligt Russian-doll mönstret.
      2004-01-12 Skapad av: NN
    </xs:documentation>
  </xs:annotation>
  <xs:element name="studiestodFraga">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="kassald" type="xs:positiveInteger"/>
        <xs:element name="kundPersonNr">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="(19|20)[0-9]{6}-[0-9]{4}"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="fromDatum" type="xs:date" nillable="true"/>
        <xs:element name="tomDatum" type="xs:date" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="SchemaVersion" type="xs:decimal" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Exempel: Schemaalternativ 1

Alternativ 2: Detta gränssnittsschema definierar samtliga använda typer och element i schemat som globala och därmed adresserbara och återanvändbara av andra scheman. Detta innebär att man kan skapa ett eget schema som återanvänder detta standardmeddelandes definition av personnummer. Detta är dock troligen inte önskvärt, då syftet med Standardmeddelandet är att definiera ett systemgränssnitt, inte att publicera återanvändbara datatyper. Element som skall återanvändas skall definieras i ett eget schema, inte i ett gränssnittsschema.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="http://schema.se/test/csn/studiestodFraga/1/"
  xmlns="http://schema.se/test/csn/studiestodFraga/1/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="1.0">
  <xs:annotation>
    <xs:documentation xml:lang="sv">
      Standardmeddelade för frågor om studiestöd till CSN (Alternativ 2)
      2004-01-12 Skapad av: NN
    </xs:documentation>
  </xs:annotation>
  <xs:element name="kassald" type="xs:positiveInteger"/>
  <xs:element name="kundPersonNr" type="PersonNr"/>
  <xs:element name="fromDatum" type="xs:date"/>
  <xs:element name="studiestodFraga" type="StudiestodFraga"/>
  <xs:complexType name="StudiestodFraga">
    <xs:sequence>
      <xs:element ref="kassald"/>
      <xs:element ref="kundPersonNr"/>
      <xs:element ref="fromDatum" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="PersonNr">
    <xs:restriction base="xs:string">
      <xs:pattern value="(19|20)[0-9]{6}-[0-9]{4}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Exempel: Schemaalternativ 2

Lösningen i alternativ två gör det möjligt att implementera ett annat Standardmeddelandeschema xxFraga.xsd som innehåller elementet kundPersonNr från vårt schemaexempel. Detta skulle innebära att en ändring av elementnamnet till efterfragadPersonNr i vårt schema skulle medföra att alla användare av xxFraga.xsd måste genomföra denna namnändring. Denna extra administration i form av påverkansanalyser för omgivande scheman bör därför inte placeras på varje Standardmeddelande, utan på underliggande återanvändningsoptimerade scheman.

Alternativ 3, som bäst representerar hur standarddokument bör struktureras, visar på ett alternativ för definition av separata, återanvändbara element. Det ger en mer flexibel användning av scheman, där de uppgraderingar som behövs på grund av återanvändningen kan fokuseras till specifika scheman. I alternativ 3 har Standarddokumentet definierats med hjälp av två separata XML scheman, 3a och 3b.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  targetNamespace="http://schema.se/test/csn/personNr/1/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schema.se/test/csn/personNr/1/"
  version="1.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:annotation>
    <xs:documentation xml:lang="sv">
      Central typ för personnummer (Används i Alternativ 3)
      2004-01-13 Skapad av: NN
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType name="PersonNr">
    <xs:restriction base="xs:string">
      <xs:pattern value="(19|20)[0-9]{6}-[0-9]{4}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Exempel Schema 3a: Detta schema implementerar datatypen för termen personnummer och gör denna tillgänglig för återanvändning via personnummerschemats egen namnrymd.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schema.se/test/csn/studiestodFraga/1/"
  xmlns="http://schema.se/test/csn/studiestodFraga/1/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:pNr="http://schema.se/test/csn/personNr/1/"
  version="1.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import namespace="http://schema.se/test/csn/personNr/1/"
    schemaLocation="personNr.xsd"/>
  <xs:annotation>
    <xs:documentation xml:lang="sv">
      Standardmeddelade för frågor om studiestöd till CSN (Alternativ 3)
      2004-01-12
      Typen PersonNr importeras från ett fristående schema skapad av: NN
    </xs:documentation>
  </xs:annotation>
  <xs:element name="studiestodFraga" type="StudiestodFraga"/>
  <xs:complexType name="StudiestodFraga">
    <xs:sequence>
      <xs:element name="kassald" type="xs:positiveInteger"/>
      <xs:element name="kundPersonNr" type="pNr:PersonNr"/>
      <xs:element name="fromDatum" type="xs:date"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Exempel Schema 3b: I schema 3b adresseras och återanvänds datatypen för personnummer från schemat i exempel 3a. Dessutom har man gjort både datatypen och elementet för studiestödsfråga återanvändbara.

Exemplen Schema 3a+3b i alternativ tre visar på en början till en schemastruktur med olika schemanivåer som stödjer återanvändning. En sådan struktur baseras på antagandet att det finns vissa myndighetsgemensamma definitioner av scheman på nivån centrala element och typer, t.ex. för personnummer, organisationsnummer och liknande.

Dessa scheman med underliggande informationsmodeller skulle då vara publicerade för återanvändning av andra utvecklare.

Några exempel på relaterade initiativ

Det finns ett antal pågående initiativ runt om i världen som bland annat producerar rekommendationer, riktlinjer och exempel. Man kan exempelvis titta på:

UN/CEFACT - <http://www.ebxml.org/>

Danmark - <http://www.oio.dk/XML/CookBooks>

USA - <http://www.xml.gov/>

Storbritannien - <http://e-government.cabinetoffice.gov.uk/Resources/>

Frågeställningar att beakta

Begränsningar av informationsutlämning

I vissa fall kan en myndighet behöva begränsa vilken data som kan lämnas ut till en specifik mottagare. Till största delen är detta en fråga för de applikationer som genererar innehållet i standardmeddelandena. Standardmeddelandena måste dock definieras på ett sådant sätt att de element som innehåller den data som ska begränsas tillåts vara tomma. Detta görs lämpligen antingen genom att tillåta att elementet utesluts (definiera `minOccurs="0"`) eller att göra det möjligt att explicit visa att ett element som normalt ska innehålla en sträng är medvetet lämnat tomt genom att ge det värdet `xsi:nil` (från XML Schema Instance-namnrymden). Detta möjliggörs av att man i schemat definierar att elementet får innehålla `xsi:nil` genom att som sätta XML Schema-attributet `nillable` till `true`.

```
<xsi:element name="fromDatum" type="xs:date" nillable="true"/>
```

Prestanda

I vissa fall kan prestandakrav påverka schemadesignen. Om man exempelvis önskar att göra viss validering av värden redan vid dokumentvalideringen så kan långa värdelistor ta tid.

Fördelarna med att tidigt garantera korrekta värden brukar dock överväga eventuella prestandaförsämringar.

Övergripande förvaltning myndighetsinternt samt centralt

Organisationer som arbetar med schemastrukturer behöver en strategi förvaltning av gemensamma scheman och värdelistor. Om man redan har förvaltningsmetoder för existerande gränssnitt och datamodeller så bör mycket av detta kunna återanvändas.

Hur kommer man igång?

Genomför ett "Pilotprojekt"

Låt en liten grupp (kanske 2-4 personer) modellera och utveckla ett första Standardmeddelande.

Ge samtliga deltagare utbildning inom området XML Schema och konceptet Standardmeddelanden. Åtminstone en person i gruppen bör också ha tidigare erfarenhet av praktiskt arbete med XML scheman. Helst alla men åtminstone någon deltagare bör ha erfarenhet av datamodellering. Hela pilotprojektet bör betraktas som utbildning av personalen och organisationen.

1. Börja med att göra en ansats kring schemadesignregler, baserat på rekommendationerna i detta dokument.
2. Inventera lämpliga verktyg för schemakonstruktion och validering. Med lämpliga avses främst att de fungerar på rätt plattform och har rätt kostnadsbild för organisationen. Välj ut ett eller två av dessa att testa under projektets gång. De flesta verktyg av denna typ erbjuder testlicenser som möjliggör att man kan prova dem utan kostnad.
3. Välj ut ett standardmeddelande, lämpligen en fråga med svar, och skapa scheman och instanser av detta. Standardmeddelandet bör vara ett som bedöms vara enkelt att stödja med de existerande systemen, då pilotprojektets fokus bör ligga på schemadesignen och inte den tekniska implementationen.
4. Identifiera alla datatyper, element och attribut i den datamodell som ska ingå i Standardmeddelandet. Bestäm den överordnade schemastrukturen baserat på schemadesignriktlinjerna i detta dokument.
5. Bygg scheman med hjälp av utvalda verktyg. För att prova designreglerna kan det vara lämpligt att plocka in ytterligare någon projekt-

deltagare som får prova att implementera instanser baserat på dessa scheman och dess dokumentation.

6. Implementera systemstöd för Standardmeddelanden.
7. Efter att de första instanserna har gjorts görs lämpligen nya schema-versioner baserat på föreslagna versionsregler så att dess effekt på instanserna och systemimplementationen kan verifieras.
8. Följ upp hur schemadesignreglerna har fungerat under projektet och korrigera dem baserat på gjorda erfarenheter.

Syftet med ett pilotprojekt är att:

- bygga upp organisationens kompetens inom området
- ta fram en första version av designregler för XML Scheman
- sätta upp en utvecklingsmiljö
- få tillgång till ett eget exempel som visar att XML-baserade Standardmeddelanden fungerar ihop med organisationens system

Bilaga 1.2 – vägledning – mallvillkor (förslag)

För att underlätta samordningen bör myndigheterna publicera Nyttjanderegler för utbytet av Standardmeddelanden. Nyttjandereglerna bör fungera som villkor i de fall utbytet sker mellan myndigheter och som ett avtal om mottagaren inte är en statlig myndighet. Det är en fördel om myndigheters Nyttjanderegler utformas på ett likartat sätt.

Mot bakgrund av att olika myndigheter kan behöva ställa olika krav vid utbytet av Standardmeddelanden består vägledningen till utformningen av nyttjandevillkor av två delar. Dels de konkreta förslag till mallvillkor som presenteras här och som antas lämpliga för de flesta myndigheter, dels en ”checklista” med frågor som kan behöva regleras i vissa fall, se bilaga 1.3. Den exakta utformningen av Nyttjanderegler för olika typer av uppgifter ur olika register måste avgöras från fall till fall av respektive myndighet.

Valet av säkerhetsfunktioner vid godkännande av Nyttjanderegler samt vid utbyte av Standardmeddelanden bör t.ex., genom en prövning av respektive utlämnande myndighet, anpassas till den säkerhetsnivå som är lämplig för de uppgifter som är aktuella.

Målgrupp

Denna bilaga riktar sig till myndigheters jurister, verksamhetsarkitekter, verksamhetsansvariga och systemansvariga för register, databaser och verksamhetssystem.

Relaterade dokument

Dokumentet ”Standardmeddelanden – en förstudie” är skrivet med inriktning på målgruppen verksamhetsansvariga och beskriver nyttan med standardmeddelanden samt hanteringen av dessa hos myndigheter och gemensamt emellan myndigheter ur verksamhetsperspektiv

Förslag till mallvillkor

Tillämpningsområde och definitioner

1. Dessa Nyttjanderegler gäller för [Myndighetens] utbyte av information genom Standardmeddelanden.
2. Med Standardmeddelanden avses information som
 - i. görs tillgänglig eller tas emot genom [e-tjänsten/viss webbplats], och
 - ii. bygger på de fria tekniska format och strukturer, som [Myndigheten] bestämt [preciseras genom t.ex. en bilaga].¹

Säkerhet

3. Standardmeddelanden kan efter godkännande av dessa Nyttjanderegler utbytas med [behöriga mottagare].
4. Standardmeddelanden utbytes endast genom användande av [den eller de säkerhetslösningar som myndigheten godkänner].²

Elektronisk överföring

5. Standardmeddelanden får överföras elektroniskt till myndigheten av [behöriga personer] endast om handlingen överförs till myndigheten i enlighet med de säkerhetskrav som framgår av dessa Nyttjanderegler.

Behandling av personuppgifter

6. Utbytet av Standardmeddelanden skall ske i enlighet med [deltagande myndigheters] överenskommelse rörande behandling av personuppgifter.

Aviseringstid vid ändringar

7. [x] dagar innan en ändring av krav på format och struktur skall detta aviseras genom [publikation på myndighetens webbplats/meddelande till utsedd person].
8. [x] dagar innan en ändring av krav på säkerhet skall detta aviseras genom [publikation på myndighetens webbplats/meddelande till utsedd person].

¹ Jfr. ovan rörande rättigheter till XML-scheman.

² T.ex. lösenord eller legitimering med hjälp av e-legitimation.

9. [x] dagar innan en ändring av krav på tillgänglighet skall detta aviseras genom [publikation på myndighetens webbplats/meddelande till utsedd person].

Bilaga 1.3 – vägledning nyttjandevillkor – checklista (förslag)

För att underlätta samordningen bör myndigheterna publicera Nyttjanderegler för utbytet av Standardmeddelanden. Nyttjandereglerna bör fungera som villkor i de fall utbytet sker mellan myndigheter och som ett avtal om mottagaren inte är en statlig myndighet. Det är en fördel om myndigheters Nyttjanderegler utformas på ett likartat sätt.

Mot bakgrund av att olika myndigheter kan behöva ställa olika krav vid utbytet av Standardmeddelanden består vägledningen till utformningen av nyttjandevillkor av två delar. Dels de konkreta förslag till mallvillkor som presenterats i föregående bilaga, dels denna ”checklista” med frågor som kan behöva regleras i vissa fall. Den exakta utformningen av Nyttjanderegler för olika typer av uppgifter ur olika register måste avgöras från fall till fall av respektive myndighet.

Valet av säkerhetsfunktioner vid godkännande av Nyttjanderegler samt vid utbyte av Standardmeddelanden bör t.ex. anpassas till den säkerhetsnivå som är lämplig för de uppgifter som är aktuella genom regler för behörighetshantering inom de myndigheter som utbyter Standardmeddelanden mm. Det kan också behövas klargöranden över den rättsliga osäkerhet som gäller på området kring utlämnande av uppgifter genom direktåtkomst respektive på ADB-medium.

Målgrupp

Denna bilaga riktar sig till myndigheters jurister, verksamhetsarkitekter, verksamhetsansvariga och systemansvariga för register, databaser och verksamhetssystem.

Relaterade dokument

Dokumentet ”Standardmeddelanden – en förstudie” är skrivet med inriktning på målgruppen verksamhetsansvariga och beskriver nyttan med standardmeddelanden samt hanteringen av dessa hos myndigheter och gemensamt emellan myndigheter ur verksamhetsperspektiv

Förslag till Checklista

Tillgänglighet

1. Eventuellt åtagande av myndigheten att dimensionera och anpassa sin tekniska miljö så att viss tillgänglighet till databasen garanteras.
2. Vilka tider och veckodagar som databasen avses/skall finnas tillgänglig.
3. Om åtaganden görs enligt 1 och 2 kan det vara lämpligt att kombinera dessa med krav på mottagare/avsändare av Standardmeddelanden att meddela förväntade volymförändringar. Det kan t.ex. krävas att åtagandet görs inom visst antal dagar/god tid före en faktisk förändring.

Ansvarsbegränsning

4. Myndighetens åtagande om tillgänglighet gäller inte [under vissa omständigheter, t.ex. om mottagaren/avsändaren brustit i krav på meddelande av volymförändringar].
5. Begränsning av ansvar vid fel i samband med utbyte av Standardmeddelanden, avbrott, förändring av uppgifter eller liknande.

Hänvisning till lagstiftning

6. Tydliggörande av mottagarens skyldigheter enligt lag, exempelvis krav på att uppgifter skall behandlas i enlighet med personuppgiftslagen (1998:204).
7. Även i andra fall kan det vara lämpligt med hänvisning till lag och/eller annan författning.

Krav på mottagare av uppgifter

8. Krav på behörighetssystem hos mottagare av Standardmeddelanden. Ansvaret för identifiering av behöriga medarbetare och tillräcklig intern kontroll bör ligga på varje enskild mottagare av Standardmeddelanden.
9. Mottagarens skyldighet att hålla informationen aktuell och korrekt.
10. Begränsningar i rätten att återanvända, bearbeta eller andra inskränkningar i rätten att använda informationen samt författningsstöd för dessa begränsningar.

Upphovsrätt

11. Upphovsrättsliga aspekter på innehåll i Standardmeddelanden.
12. Upphovsrättsliga aspekter relaterade till utvecklade standarder för format och struktur, t.ex. XML-scheman.

Kostnadsfördelning

Kostnadsersättning, kostnadsfördelning och eventuella avgifter vid utbyte av Standardmeddelanden.