

E L M E R

ENKLERE OG MER EFFEKTIV RAPPORTERING

Middelthuns gate 27, Postboks 5250 Majorstua, N-0303 Oslo

Forslag til løsningsarkitektur for offentlig elektronisk innrapportering

Rapport fra ELMER-prosjektet 25. juli 2001

Forslag til løsningsarkitektur for offentlig elektronisk innrapportering

Rapport fra ELMER-prosjektet 25. juli 2001

Nærings- og handelsdepartementet har etablert Referansegruppen for elektronisk offentlig innrapportering med medlemmer fra blant annet alle de store etatene med hyppig skjema-basert næringslivskontakt. Referansegruppen skal være en pådriver for elektronisk innrapportering fra næringslivet til det offentlige.

Gruppen initierte prosjektet ELMER til å kartlegge rapporteringsrutiner og relevante datasystemer i noen bedrifter fra sommeren 2000 til sommeren 2001. Det var også et mål at ELMER i kartleggingsperioden skulle prøve ut enkle løsninger for elektronisk innrapportering fra de samme bedriftene.

ELMER er et samarbeid mellom Nærings- og handelsdepartementet, Næringslivets Hovedorganisasjon og Handels- og servicenæringens Hovedorganisasjon.

Rapporter fra ELMER-prosjektet:

Behandling av offentlige skjemaer i utvalgte bedrifter	juli 2000
Automatisk datafangst	juni 2001
Bedriftene og skjemaveldet – Observasjoner etter ett års kartlegging	juni 2001
Forslag til løsningsarkitektur for offentlig elektronisk innrapportering	juli 2001
Anbefalinger om videreutvikling av oppgaveregisteret	juli 2001
Skjemaet som lærte å lytte – Eksempel på elektronisk dialog	juli 2001

Innholdsfortegnelse

1	INNLEDNING	4
1.1	BAKGRUNN.....	4
1.2	UTFORDRINGER	4
1.3	DEFINISJONER.....	5
2	DATAFORMATER	5
2.1	SPESIFIKASJON AV DATAFORMATER	6
2.2	ET FRITT TILGJENGELIG SPRÅK FOR DATADEFINISJON - XML	6
2.2.1	<i>Bruk av XML</i>	6
2.3	MEN VÅRE SYSTEMER SKJØNNER IKKE XML !.....	8
2.4	OPPGAVEREGISTERETS ROLLE	8
3	ARKITEKTUREN	9
3.1	KRAV TIL EN IDEELL LØSNINGSARKITEKTUR.....	9
3.2	LEGOKLOSSPRINSIPPET.....	10
3.3	ARKITEKTURENS KOMPONENTER	11
3.3.1	<i>Oppgaveregistret (OR)</i>	11
3.3.1.1	Oppgaveregistrets betydning	11
3.3.1.2	Bruk av Oppgaveregistret i offentlig elektronisk innrapportering	12
3.3.2	<i>Mottak og distribusjon av elektroniske rapporter (MODER)</i>	13
3.4	GJENBRUK AV INNRAPPORTERTE DATA	14
3.5	OVERFØRING AV DATA	15
3.5.1	<i>Overføringsstandarder</i>	17
3.5.1.1	SOAP/Web Services	17
3.5.1.2	HTTP	18
3.5.1.3	SMTP (E-post).....	18
3.5.1.4	JMS.....	18
3.5.1.5	EDI	18
3.5.1.6	Oppsummering av overføringsstandarder	19
3.6	ARKITEKTURENS FORUTSETNINGER	19
4	SIKKERHET	20
5	OPPSUMMERING	21

1 Innledning

Den eksplosive utbredelsen av Internett og den standardiseringsbølgen som har oppstått i dette kjølvannet, har åpnet mange nye muligheter for kommunikasjon og informasjonsutveksling. Felles for de fleste standardene er enkelhet i definisjon og bruk. Dette gir muligheter for at man i dag med enkle midler kan løse det som tidligere har fortonet seg som uoverstigelige problemer. Denne enkelheten inviterer til en tankegang om å gjøre ting smått og lettbeint (komponentbasert og løsrevet) fremfor stort og tungt (spesialutviklet og integrert). Denne tankegangen kan også videreføres i nye løsninger for elektronisk offentlig innrapportering.

Dette dokumentet er ikke en detaljert teknisk beskrivelse av hvordan de enkelte problemstillinger konkret skal løses ved hjelp av ny teknologi. Dokumentet skal skissere et konsept, eller en måte å gjøre ting på, som *ikke* er avhengig av bestemte produkter eller leverandører.

Målsetningen er å skissere en løsningsarkitektur, grensesnitt i denne arkitekturen og metoder for datadefinisjon, som skal være et utgangspunkt for å sette alle aktører i stand til å etablere en elektronisk datautveksling uavhengig av hvilke eksisterende systemer og rutiner som benyttes i dag.

Vi tar ikke stilling til administrative vurderinger som dreier seg om hvem som bør gjøre hva i offentlig forvaltning, eller hvilke juridiske aspekter som kan være til hinder for en implementering av denne arkitekturen. Vi prøver å beskrive en arkitektur som etter vår mening er den ideelle for den problemsammensetningen som vi står ovenfor, og som er ganske uavhengig av administrative og juridiske rammebetingelser.

1.1 Bakgrunn

ELMER-prosjektet skal foreslå og teste ut enkle løsninger for elektronisk innrapportering basert på eksisterende teknologi. Testene som skal utføres må baseres på en arkitektur som ivaretar de behov og muligheter man har, både i næringslivet og i etatene.

Formålet med uttestingen er ikke bare å vise at man kan flytte data fra A til B, men å vise at man også kan gjøre dette i henhold til den arkitekturen som skisseres her. Pilotprosjektet vil også forhåpentligvis vise at flere aktører på en enkel måte kan passe sine produkter inn i denne arkitekturen basert på de spesifikasjoner vi stiller til rådighet.

ELMER-prosjektet har ideer om hvordan en slik arkitektur bør se ut for å dekke de behovene som eksisterer, og som tar hensyn til både bedriftenes og etatenes muligheter til å utnytte denne arkitekturen.

1.2 utfordringer

Opp mot 80% av bedriftene i Norge har en eller annen form for tilgang til Internett i dag, men tilgangsmåten og modenheten for bruk av nettet varierer sterkt.

Spennvidden i bedriftenes bruk av teknologi, og mangfoldet av løsninger er også meget stor. Nye løsninger må derfor designes slik at de må kunne tilpasses den enkelte bedrifts behov. Det er stor forskjell i behovene til en pølsebu og en bedrift med mange hundre ansatte.

Tilsvarende eksisterer det et mangfold av løsninger i offentlige etater, som på en eller annen måte må kobles med bedriftenes mulighet for leveranse.

På toppen av de løsninger man ønsker å etablere ligger det også i mange tilfeller sterke krav om sikkerhet, og alltid krav om entydig identifikasjon av bedriften.

Slik situasjonen fremstår i dag finner vi etatene (ca. 50 stk.) på den ene siden, med enten ikke eksisterende, eller med forskjellige "proprietære" grensesnitt. På den andre siden finner vi opp mot 400.000 potensielle oppgavepliktige enheter. Mellom disse er det et gap som må fylles. Måten dette gapet fylles på er avgjørende for om bedriftene vil kunne oppleve en reell forenkling i prosessen med å oppfylle sine oppgaveplikter.

1.3 Definisjoner

Skjema

I dette dokumentet benyttes begrepet "skjema" om en logisk samling data som skal dekke en mottakers informasjonsbehov i en bestemt sammenheng. Dette trenger nødvendigvis ikke være et papirskjema, eller et skjema som ser ut som det tilsvarende papirskjemaet. Når det er et papirskjema som omtales, så blir dette påpekt spesielt.

Aktører

I dette dokumentet brukes begrepet aktører om alle som er, eller kan bli involvert i offentlig elektronisk innrapportering. Dette gjelder selvfølgelig etatene og næringslivet, men også leverandører av lønns-, økonomi-, og produksjonssystemer, og også tredjepartsleverandører som kan tenkes å levere hele, eller deler av løsninger for dette formålet

2 Dataformater

La oss begynne med å se på hvordan man bør definere de data som skal utveksles mellom næringslivet og det offentlige.

For elektronisk datautveksling stilles det adskillig strengere krav til datadefinisjon enn ved utfylling av et papirbasert skjema, under forutsetning at data skal behandles maskinelt, og uten manuell inntasting. Utfylling av skjema på papir med påfølgende manuell avlesning gir rom for tolkning og vurdering av de utfylte verdiene hos mottaker.

Med et papirbasert skjema kan man faktisk vente til skjemaene har kommet i retur med å bestemme hvordan de enkelte elementene skal registreres i datasystemene, dersom dette ikke er gitt av et eksisterende system i forkant. Dataene blir ikke mindre eksakte av den grunn, men tidspunktet for vurderingen av dataformatet kan være forskjellig.

Maskinelt behandlede data må være eksakte for at man skal få en effektiviseringsgevinst. Dette betyr av man i forkant av en skjema utsendelse må bestemme nøyaktig hvordan respondentene skal spesifisere de enkelte elementene i skjemaet.

En av konsekvensene ved innføring av elektroniske "skjema" blir derfor at man kan oppleve at det brukes mer tid til spesifisering av skjemaet før det sendes ut. Denne tiden vil man mer enn spare inn på registreringstidspunktet, fordi registreringen av de mottatte skjemaene forutsettes å gå automatisk.

Ideelt sett bør det gå raskere å spesifisere et elektronisk skjema enn å designe et papirskjema. Hvis man gjør dette på riktig måte, og med de riktige hjelpemidler er dette mulig å få til.

2.1 Spesifisering av dataformater

Spesifisering av dataformater kan gjøres på flere måter:

- som en tekstlig beskrivelse
- som databasedefinisjoner (eks. ANSI SQL-92)
- som programspråkavhengige definisjoner (eks. COBOL datadefinisjon)
- ved å bruke et datadefinisjonsspråk

Hvis man legger til grunn at datadefinisjonene skal kunne brukes av alle aktører (respondenter, programvareleverandører og etater), bør ikke datadefinisjonene være knyttet til et bestemt databaseformat eller programmeringsspråk. En ren tekstlig beskrivelse er heller ikke å foretrekke fordi denne metoden umuliggjør en maskinell behandling av selve definisjonene.

2.2 Et fritt tilgjengelig språk for datadefinisjon - XML

Før man begynner å tenke på å lage et eget datadefinisjonsspråk bør man se på hva som allerede eksisterer. I dag behøver man ikke å gå så veldig langt før man støter på betegnelsen XML. XML står for eXtensible Markup Language, og er metode for å beskrive informasjon slik at både mennesker og maskiner skal kunne forstå innholdet.

XML har fått en voldsom utbredelse i det siste året. Det er først i denne perioden at man har fått de nødvendige verktøyene for å behandle XML effektivt. Selv om XML i utgangspunktet ikke stiller noe krav til verktøy (man kan skrive XML i Notepad om man vil), så er dette en forutsetning for effektiv bruk.

Både språket XML, og de nødvendige verktøy for å behandle XML-dokumenter er fritt tilgjengelige, og kan dermed benyttes av alle aktører uten lisenskostnader.

2.2.1 Bruk av XML

XML-standarden sier naturlig nok ikke noe om hva slags data *du* har, eller hvordan du vil beskrive disse. Dette er noe som må spesifiseres. Normalt bestemmes en slik spesifisering av den eller de som skal motta data. Dette kan spesifiseres av alle som skal motta data, men trenden er at grupperinger av interessenter som har behov for likelydende data går sammen om å lage felles spesifikasjoner.

Før man går videre på dette området kan det være nyttig å se på et eksempel på et XML-dokument og en spesifisering. Disse er ikke garantert 100% teknisk korrekte, men de gir en god ide om hvordan dette ser ut.

Eksempel på et XML-dokument som inneholder navn og adresse

```
<?xml version="1.0"?>
<adresseDefinisjon>
  <adresse land="Norge">
    <navn>Ola Nordmann</navn>
    <gateadresse>Nittedalsgata 234</gateadresse>
    <postnummer>2000</postnummer>
    <poststed>Lillestrøm</poststed>
  </adresse>
</adresseDefinisjon>
```

Selv om dokumentet åpner med en kryptisk linje, så ser man relativt greit at dette er en adresse. Så kommer spørsmålet; hva bestemmer at det skal hete "adresse", "navn", "postnummer" osv. Det er ingenting i veien for at man kan si at dette er bare noe som vi har blitt enige om, og så er det opp til hver enkelt som lager slike dokumenter å følge denne standarden.

Det er imidlertid en rekke ting å vinne på å lage en spesifikasjon som definerer dette:

Eksempel på en spesifikasjon av "adresseDefinisjon" (basert på XML Schema 1.0)

```
<xsd:schema xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <xsd:element name="adresseDefinisjon" type="AdresseDefinisjonsType"/>

  <xsd:complexType name="adresse">
    <xsd:element name="navn" type="xsd:string"/>
    <xsd:element name="gateadresse" type="xsd:string" minOccurs="0"/>
    <xsd:element name="postnummer" type="xsd:integer"/>
    <xsd:element name="poststed" type="xsd:string"/>

    <xsd:attribute name="land" type="xsd:NMTOKEN"
      use="fixed" value="Norge"/>
  </xsd:complexType>
</xsd:schema>
```

Dette er noe mer kryptisk, men til gjengjeld lages dette bare en gang for alle. Uten å gå i detalj kan det sies at denne spesifikasjonen bestemmer hva de forskjellige feltene må hete, at de må være utfylt (bortsett fra gateadresse), at postnummeret må være numerisk og at landet må hete "Norge" (en liste over alle tillatte land er ikke tatt med her).

Med XML Schema kan man legge inn alle tenkelige regler for hvordan data skal spesifiseres, struktureres og valideres. Ved hjelp av en spesifikasjon og ett av de fritt tilgjengelige verktøyene kan hvert enkelt XML-dokument valideres på en enkel måte.

Dette betyr også at en XML-spesifikasjon inneholder alt f.eks. en leverandør av et lønnsystem trenger å vite for å kunne lage et XML-dokument basert på denne spesifikasjonen. Hvordan leverandøren faktisk gjør dette har ingen betydning for mottaker.

2.3 Men våre systemer skjønner ikke XML !

Få bedrifter, leverandører av lønns-/økonomisystemer eller etater har i dag løsninger som "skjønner" XML. Det er heller ikke sikkert at det er fornuftig at de eksisterende systemene skal skrives om til å skjønne XML. Det kan være lurt å skille ut dataflyten i flere steg.

For en bedrift kan det se omtrent slik ut:

1. Lag et uttrekk av data fra økonomisystemet til en flat fil
2. Bruk et tredjepartsprodukt til å konvertere filen til et XML-dokument
3. Send XML-dokumentet til mottaker

En annen bedrift kan gjøre det på denne måten:

1. Hent data fra økonomisystemet inn i et regneark.
2. Last regnearket opp til en webtjeneste
3. Web-tjenesten konverterer regnearket til XML og sender det til mottaker

En tredje bedrift kan gjøre det slik:

1. En funksjon i økonomisystemet genererer XML-dokumentet og sender det direkte til mottaker.

Dette er tre av mange mulige måter å gjøre dette på. Bedriften selv må kunne bestemme i hvor stor grad den vil automatisere denne prosessen.

I enkleste fall kan bedriften finne frem alle data manuelt, og så benytte en sentral webtjeneste for å legge inn disse dataene. Denne tjenesten kan så generere XML-dokumentet på vegne av bedriften og sende dette til mottaker.

Poenget her er at dette ikke er noe teknisk problem. Anbefalingen fra vår side vil være å teste ut noen av de mest relevante måtene å gjøre dette på, men la leverandører og bedrifter selv avgjøre hvilke metoder som er mest hensiktsmessige for dem.

På tilsvarende måte kan etatene utnytte eksisterende systemer ved at mottakstjenesten omformer XML-dokumentet til et format som systemene skjønner.

Den store fordelen med skille produksjon av XML-dokumenter ut fra de eksisterende systemene, er at dagens systemer ikke behøver å endres. Dette skjærer også disse systemene for en del typer fremtidige endringer i spesifikasjonene for innrapportering.

2.4 Oppgaveregisterets rolle

Hvilken rolle kan Oppgaveregisteret spille for definisjon av data ?

Hvis vi tenker oss et scenario hvor vi benytter XML for all elektronisk offentlig innrapportering, så har vi allerede pekt på behovet av XML-spesifikasjoner for definisjon av data.

Når vi ser nærmere på innholdet i en XML-spesifikasjon ser vi at det som beskrives i stor grad samsvarer med innholdet i Oppgaveregistret. Oppgaveregistret inneholder i dag definisjoner for alle skjema for næringslivets innrapportering til staten. Dette betyr at man med visse tillegg i Oppgaveregistrets database kan generere XML-spesifikasjonene direkte fra denne. Om man i tillegg åpner Oppgaveregistrets data for maskinell tilgang utenfra, vil alle aktører kunne utnytte all denne informasjonen på elektronisk vis.

Med dette kan man oppnå en situasjon hvor definisjonen av offentlige skjema ligger ett sted, elektronisk tilgjengelig og maskinelt utnyttbart. Vi får et åpent, nasjonalt skjemaregister med definisjoner for hvert enkelt dataelement.

I tillegg til selve definisjonene av dataelementene kan man legge inn beskrivelser, og veiledninger for respondentene. Disse tekstlige beskrivelsene kan også benyttes direkte fra systemer som støtter opp under denne nye måten å rapportere på. Med dette har man også etablert en sentral database for beskrivelser og veiledninger for alle oppgaveplikter.

Med disse definisjonene på plass har man langt på vei også laget en spesifikaasjon for utforming av papirskjema. Dette vil faktisk også danne grunnlag for automatisk generering av papirskjema.

Flere etater og prosjekter har allerede en tid sett på Oppgaveregistret som en datadefinisjonsdatabase. Dette gjelder også Oppgaveregistret selv. Denne rollen bør videreutvikles og forsterkes.

3 Arkitekturen

Denne delen av dokumentet skal beskrive vårt forslag til arkitektur og de forskjellige elementene i denne. Vi forsøker å gjøre dette ved å holde oss på et konseptuelt nivå og uten å benytte for mange teknologiske begreper, selv om dette ikke kan unngås for enkelte områder.

3.1 Krav til en ideell løsningsarkitektur

La oss begynne med å stille en del krav til en ideell løsning:

1. Løsningen skal ikke være bundet til bestemte leverandører eller produkter. Dvs. at det skal være en åpen arkitektur.
2. De til enhver tid gjeldende spesifikasjoner av dataformater skal være åpent tilgjengelige og maskinelt utnyttbare.
3. Alle data fra avsender skal sendes ett, og bare ett sted, slik at mottakerløsningene bare må vedlikeholdes på dette stedet.
4. Avsender skal ha sikkerhet for at han sender data til riktig sted.
5. Mottaker skal ha sikkerhet for at data kommer fra riktig avsender.
6. Tredjepart skal ikke kunne lese eller på noen annen måte misbruke data som fanges opp under datatransporten.

7. Bedriftene skal ikke være nødt til å endre i, eller skifte sine eksisterende systemer for å kunne innrapportere elektronisk, men ha muligheten til dette dersom de ønsker det (basert på det definerte dataformatet).
8. Etatene skal ikke være nødt til å endre i, eller skifte sine eksisterende systemer for å kunne ta imot innrapporterte data elektronisk, men ha muligheten til dette dersom de ønsker det (basert på det definerte dataformatet).
9. Løsningen skal skape et marked for alle aktører som er interessert i å bidra med løsninger for datafangst og oversendelse fra næringslivet til det offentlige.
10. Løsningsarkitekturen bør møte næringslivets kommende behov for en infrastruktur som understøtter elektronisk forretningsdrift.

Denne listen er ikke i en prioritert rekkefølge.

Disse 10 kravene er i utgangspunktet ambisiøse, men vi skal prøve å illustrere at dette faktisk i dag er realiserbart og at utfordringene ikke er konseptuelle eller teknologiske.

3.2 Legoklossprinsippet

Dette er gammelt begrep som i det siste er tatt i bruk i en ny verden. Det man har tatt utgangspunkt i, er at brikkene (klossene) i en internettarkitektur selvfølgelig må passe sammen.

Det finnes store klosser, små klosser, gule, røde, grønne og blå klosser. Felles for dem alle er at de kan stables oppå hverandre, eller legges ved siden av hverandre for så å la en tredje kloss binde dem sammen. Dette gjør det mulig å lage svært ulike løsninger, samtidig som dette er fleksibelt, skalerbart, og kan vokse i takt med kommende behov i de retninger fremtiden viser det er fornuftig å gå. Hvis man benytter små klosser blir fleksibiliteten større og konsekvensen ved at en kloss foreldes, eller av andre grunner må skiftes, blir liten.

Det som gjør dette mulig er at grensesnittet mellom klossene er fast definert. Hvordan klossene er fabrikkert, eller hva de består av, betyr ingenting så lenge de implementerer det spesifiserte grensesnittet.

Hovedpoenget med denne analogien er følgende: Hele arkitekturen, og det potensialet den åpner for, dreier seg om definisjon av grensesnitt.

Det viktigste grensesnittet gjelder det offentliges mottak av data. Med andre ord; hvordan skal dataformatene se ut for de data som skal leveres det offentlige, og hvordan skal de oversendes. I kapittel 2 Dataformater i dette dokumentet, argumenteres det for at dataformatene bør være basert på en XML-spesifikasjon. Hvis dette blir gjeldende vil ELMER også hevde følgende:

Alle grensesnitt i en løsningsarkitektur for offentlig elektronisk innrapportering bør beskrives i henhold til XML-spesifikasjoner.

XML-spesifikasjonene er definisjoner av legoklossenes overside og underside. Disse spesifikasjonene vil sette enhver aktør i stand til å levere data til det offentlige.

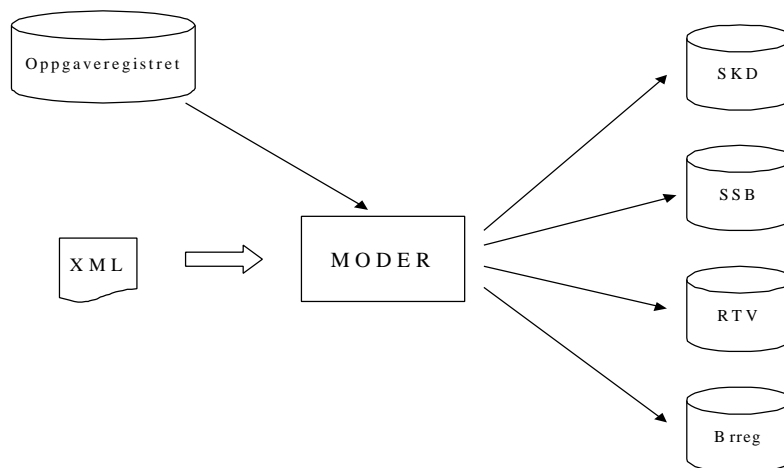
3.3 Arkitekturens komponenter

De eneste komponentene som man fra offentlig side *må* forholde seg til er Oppgaveregistret med sine datadefinisjoner, og mottakstjenesten. De øvrige komponentene kan man overlate til markedet å utvikle, basert på de grensesnitt som Oppgaveregistret og mottakstjenesten definerer.

Det vil nok allikevel være nødvendig at man fra offentlig side utvikler en tjeneste som bedriftene kan benytte dersom de selv ikke vil utvikle eller kjøpe støttesystemer for innrapportering. Denne tjenesten bør være en webbasert innrapporteringstjeneste som ikke forutsetter noe annet enn internett-tilgang fra bedriftene.

Hvordan bedriftene selv velger å løse sine innrapporteringsbehov bør være opp til bedriftene selv. Det som må være klart for dem, er hvilke datadefinisjoner og overleveringsmetode(r) de må forholde seg til. De komponentene som etableres på avgiversiden vil kunne være like ustandardiserte og mangslungne som de systemer som eksisterer i næringslivet i dag. Denne arkitekturen støtter dette.

Her følger en konseptuell skisse over mottakersiden i den arkitektur vi forsøker å beskrive:



Figur 1

De følgende underkapitlene vil beskrive de ulike komponentene.

3.3.1 Oppgaveregistret (OR)

3.3.1.1 Oppgaveregistrets betydning

Oppgaveregistret er en metadatabase som beskriver statlige skjema, så sant de er meldt til registret av etatene. Selv om opprettelsen av registret hadde et annet formål, har det blitt en datadefinisjonsdatabase. Det opprinnelige formålet er gitt av lov om Oppgaveregistret:

Lov om Oppgaveregisteret av 6. juni 1997 nr 35, §1.

”Oppgaveregisteret skal sørge for effektiv samordning og utnyttelse av oppgaveplikter som offentlige organer pålegger næringsdrivende. Registeret skal til enhver tid inneholde en oppdatert oversikt over alle slike oppgaveplikter og bidra med informasjon om disse og den samordning som finner sted.”

For å kunne utføre de oppgaver som er pålagt OR har man vært nødt til å gå et langt stykke på vei når det gjelder beskrivelse av skjema og dataelementer. Dette betyr at OR i alt annet enn en offisiell utnevning, har fått merkelappen datadefinisjonsregister. Det må understrekes at det pr. i dag ”kun” er registrert skjema for næringslivets innrapportering til staten, og at det pågår diskusjoner om eventuelle utvidelser av dette.

Datadefinisjoner er selve fundamentet for elektronisk kommunikasjon, og OR er kommet langt på vei til å utgjøre dette fundamentet for elektronisk offentlig innrapportering.

Hovedproblemet med definisjoner av dataformater er ikke å finne *måten* å gjøre dette på, men derimot å etablere en felles forståelse av *betydningen* av de enkelte dataelementene. Når man tidligere har etablert en datautveksling mellom to uavhengige parter, så har det vært nødvendig med omfattende bilaterale avklaringer for å oppnå denne felles forståelsen. Dette har kanskje vært den største flaskehalsen for utbredelsen av f.eks. EDI/EDIFACT.

Med Oppgaveregistret er staten i den posisjon at registret faktisk ene og alene definerer betydningen av alle elementer i alle skjemaer, og i hvilke sammenhenger de skal opptre. Slik man ser dette fra næringslivets side, så er ikke denne utviklingen å betrakte som ”regulerende” eller på andre måter negativ. Muligheten for elektronisk innrapportering fra næringslivet har i de fleste tilfeller ikke vært til stede, fordi det ikke har eksistert definisjoner av de data man kunne ønske å rapportere på elektronisk vis.

Oppgaveregistret kan derfor gi oss en unik mulighet for å lage elektroniske definisjoner av alle offentlige skjema (statlige pr. i dag). Samtidig vil bruken av OR som eneste lagringsplass for statens XML-spesifikasjoner sikre at OR faktisk *får* melding om all ny datainnsamling. Dermed øker også nytten av OR til sitt opprinnelige samordningsformål.

3.3.1.2 Bruk av Oppgaveregistret i offentlig elektronisk innrapportering

Det er nevnt som en forutsetning for vår arkitektur at Oppgaveregistret må åpnes for allment innsyn, og med mulighet for maskinell tilgang. I dette tilfellet vil det ikke være tilfredsstillende kun å gi tilgang til disse dataene via et HTML-basert webgrensesnitt. Dette gir ikke aktørene muligheten til f.eks. å hente ut definisjonen av ett bestemt dataelement, eller beskrivelsen av det.

En mulighet er å legge Oppgaveregistrets data inn i en LDAP-basert¹ katalog (Directory Service). Slike kataloger er optimalisert for lesing og kan håndtere veldig mange samtidige brukere. En elektronisk katalog er en hierarkisk database for lagring av objekter. Et objekt kan i denne sammenhengen være definisjonen av et dataelement, et skjema eller en beskrivelse. LDAP inneholder kraftige funksjoner for søking og oppslag. Data i en slik katalog kan

¹ LDAP – Lightweight Directory Access Protocol er en åpen standard for behandling av data i en elektronisk katalog.

aksesseres via LDAP fra alle typer programvareplattformer, og denne aksessen er ikke knyttet til programprodukter fra noen spesielle leverandører.

Det er også mulighet for å definere grensesnittet mellom brukerprogramvare og Oppgaveregistret i form av XML-spesifikasjoner, slik at både forespørsler til, og svar fra Oppgaveregistret opptrer i form av XML-dokumenter. Dette kan nok vise seg å være den mest fleksible løsningen, som også reduserer kompleksiteten ved at man til enhver tid kun trenger å forholde seg til XML-dokumenter.

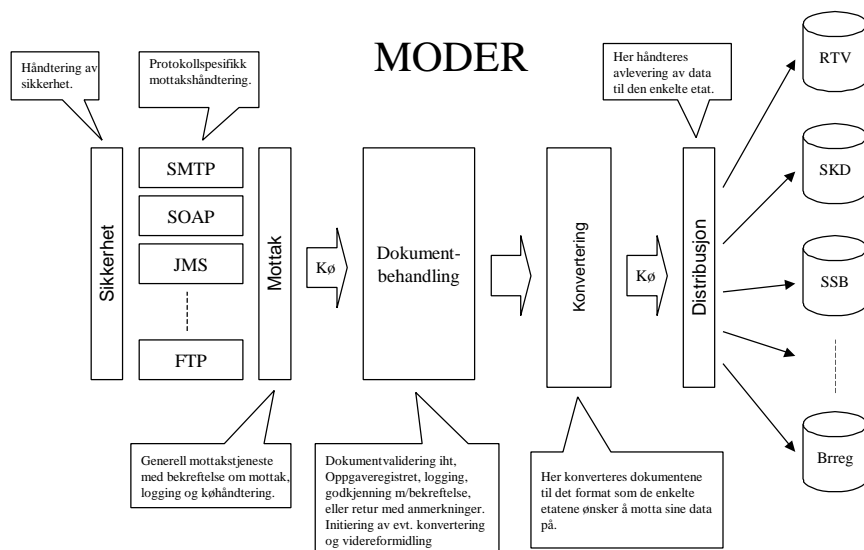
I tillegg til å åpne Oppgaveregistrets data via LDAP og/eller XML, vil det være nødvendig å gjøre skjemaene tilgjengelige som en eller flere XML-spesifikasjoner. Disse spesifikasjonene skal være tilgjengelige for referanse og praktisk bruk for aktører med programmer som genererer XML-dokumenter for innrapportering. Disse spesifikasjonene skal kunne genereres direkte fra Oppgaveregistrets katalog.

3.3.2 Mottak og distribusjon av elektroniske rapporter (MODER)

Dette mest effektive på kort sikt er å etablere en sentral mottakstjeneste som i første omgang skal være i stand til å motta data fra avgiverne, og videreformidle disse til relevante offentlige etater.

Denne tjenesten kan også ivareta den nødvendige sikkerhet som avgiverne har krav på, og som det offentlige er forpliktet til å tilby.

Vi skisserer her hvordan denne funksjonen kan se ut. Dette er selvfølgelig ingen fasit, men illustrerer hvordan man kan bygge opp en komponentbasert løsning med internett-teknologi.



Figur 2

Figur 2 illustrerer aspekter ved etablering av en slik tjeneste som vil ta høyde for eksisterende, og fremtidige muligheter og behov.

Sikkerhetshåndteringen bør så langt som mulig skilles fra resten av mottaksapparatet. Dette vil gi mulighet for å kunne endre sikkerhetstilbudet over tid, uten at dette berører resten av håndteringen av de mottatte dokumentene. Slik man sannsynligvis blir nødt til å gjøre dette i dag, er å knytte sikkerheten til den enkelte protokollen som ønskes benyttet.

Tilbudet av overføringsprotokoller vil kunne variere over tid. På samme måte som for sikkerhet bør disse kunne legges inn i arkitekturen uten at dette skal påvirke resten av mottaksapparatet.

Følgelig må mottak, logging, tilbakemelding og kjøing av dokumenter for viderebehandling være en generell funksjon, som ikke påvirkes av mottaks- eller distribusjonsmåte.

Dokumentbehandlingen omfatter validering/kontroll av de innkomne dokumentene, og med en eventuell retur med feilmeldinger hvis de ikke godkjennes. Hvis dokumentene godkjennes oversendes de for konvertering og distribusjon til de relevante etatene.

Vi har sagt at alle data skal leveres som XML-dokumenter. Dette vil i de fleste tilfeller ikke passe inn med de systemer og rutiner som etatene har i dag. Det vil derfor kreves en konvertering av XML-dokumentet til et dataformat som den aktuelle etatens systemer kan forstå. Denne konverteringsfunksjonaliteten legges inn i den sentrale mottakstjenesten slik at etatene kan motta data på den formen de ønsker, og på den måten de ønsker.

Rent praktisk kan en konvertering av XML-dokumenter til andre formater gjøres ved å benytte en teknologi som kalles Extensible Stylesheet Language Transformations (XSLT). Ved å benytte denne teknologien vil alle dataformater som etatene ønsker å benytte kunne spesifiseres i XSLT-dokumenter. Dette betyr at man ved endring i XML-dokumenter, eller i etatens dataformater, kun behøver å endre det aktuelle XSLT-dokumentet for å ivareta en korrekt konvertering.

Distribusjonen til etatene er det vi sier minst om i denne sammenhengen, men det bør kanskje bygges opp en arkitektur for dette som fungerer på samme måte som den protokollavhengige mottakshåndteringen, men i utgående retning.

3.4 Gjenbruk av innrapporterte data

Det har vært ønsket om å benytte tidligere innrapporterte data for preutfylling av nye skjema, eller bare som referanse for avgiver. I tillegg bør etatene bruke data som allerede er innrapportert til andre etater. Dette kan spare bedriftene for en del utfyllingsarbeid når de skal fylle ut skjema på web, og det kan også spare bedriftene for spørsmål dersom etatene kan hente tidligere innrapporterte data fra andre etater.

Dobbelrapportering er noe som næringslivet beklager seg over, og som har fått en del fokus i skjemaveldedebatten. Dobbelrapportering er selvfølgelig noe som bør unngås, men før man setter i gang et stort apparat for å bøte på dette så må man stikke fingeren i jorda og se på

realitetene. En gjennomgang av oppgaveregistret viser at den prosentvise andelen av identiske data som rapporteres direkte fra bedriftene til flere etater er svært liten. En del av dette skyldes at det er nyanseforskjeller i spørsmålene som ikke gjør dataene identiske, men heller ikke dette utgjør den store mengden. Dette er momenter som man bør ha i tankene før det lages store prosjekter for å for å unngå dobbelrapportering.

Et alternativ for å løse dette som mange ser for seg er å etablere en sentral database med all offentlig informasjon, som både etater og bedrifter kan forsyne seg fra (under forutsetning av at dataene er sikret slik at man bare får tilgang til den informasjonen man har rett til å se). Dette er en løsning som ELMER ikke vil anbefale. Etablering av en slik database medfører et stort sentralt utviklingsarbeid med mange fallgruber og liten fleksibilitet ved endringsbehov. I tillegg er det mange sikkerhetsaspekter og juridiske aspekter som må avklares dersom dette skal realiseres.

Vårt forslag er at etatene bør tilby webtjenester som leverer aktuelle data i XML-format til de som trenger, og har rett til disse dataene. Disse tjenestene kan så benyttes av de som utvikler løsninger innefor dette området, enten det er på offentlig eller privat side. Data som leveres fra næringslivet bør allikevel bare leveres én gang, og et sentralt *mottak* vil hjelpe på denne situasjonen. Arkitekturen som skisseres her tar imidlertid hensyn til at dette skal kunne implementeres. Når alle data passerer gjennom et sentralt mottak, vil man enkelt kunne dirigere alle data til de databaser man måtte ønske. Dette betyr at data kan sendes fra mottaket til flere etater, og ikke bare til den etaten som primært er mottaker av skjemaet.

Dette kan gjøres gjennomføres på offentlig side uten at det krever noe ekstra fra næringslivet., men et annet aspekt er at næringslivet selv kan ha behov for å samle og tilgjengeliggjøre deler av denne typen data. I en eBusiness-sammenheng er dette meget relevant, men krever samtidig mer av bedriften selv.

Bedriftene selv, eller tredjepartsleverandører, kan selvfølgelig selv legge opp til preutfylling av skjema med data fra egne systemer.

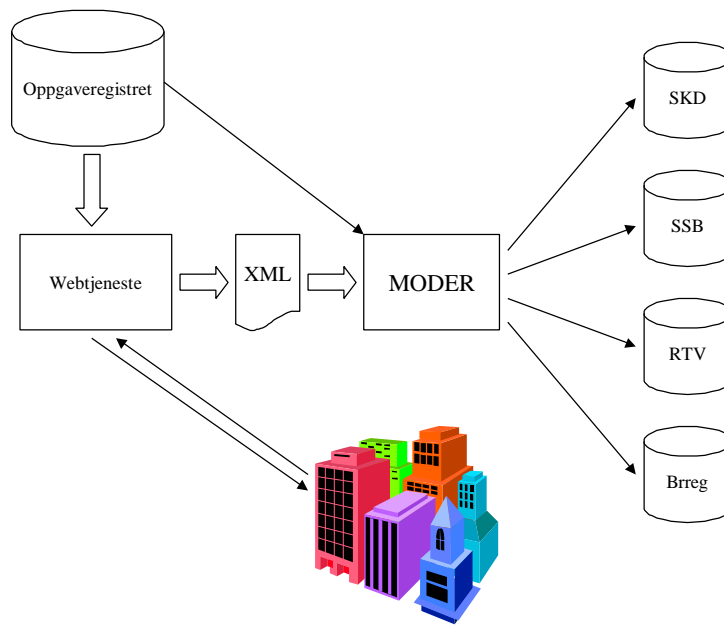
3.5 Overføring av data

Når man har et XML-dokument blir spørsmålet: Hvor skal dette dokumentet leveres og hvordan kan dette gjøres ?

Det har i flere sammenhenger vært indikert at det kan være lurt å samle mottak av all offentlig innrapportering ett sted. Dette forenkler problemstillingen for avgivere ved at det ikke må vurderes hvor data skal sendes. Samtidig samler man implementering og vedlikehold av slike tjenester på offentlig side på ett sted.

Måten en overlevering gjøres på har ingen betydning for dataene, men det er allikevel visse krav man må stille til en overføringsmetode. Disse kravene vil kunne være forskjellige for forskjellige typer data. Data som krever en høy grad av beskyttelse mot innsyn og med strenge krav til signatur, vil naturlig nok kreve systemer som støtter dette. I de fleste tilfellene vil man også stille krav til logging og bekreftelse om mottatt leveranse.

En annen type "overføring" er utfylling av webskjema. I mange tilfeller vil dette kunne være en god løsning. Her følger et eksempel på et slikt scenario:



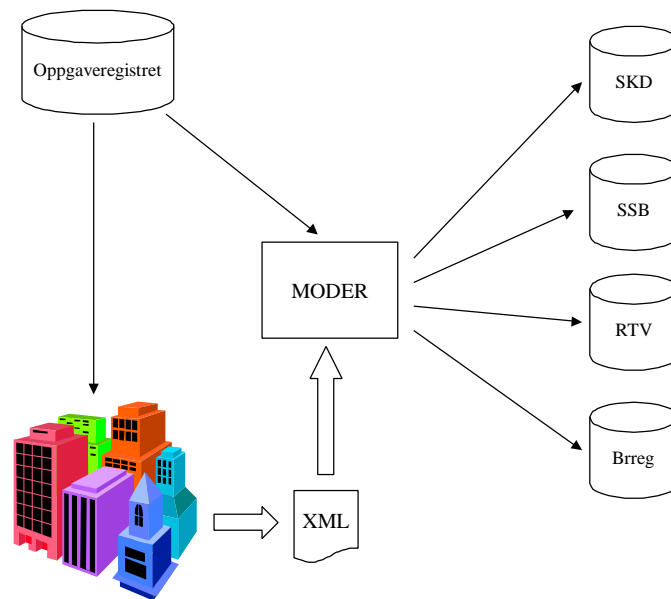
Figur 3

Av Figur 3 ser vi bedriftene som benytter en webtjeneste for ”overføring” av data for innrapportering. I et slikt scenario behøver vi ikke å bekymre oss om hvordan bedriften fremskaffer sine data (som i dag).

Webtjenesten benytter datadefinisjonene i oppgaveregistret til å generere og validere de aktuelle webskjema. Når bedriften har fått validert sin registrering vil webtjenesten generere et XML-dokument og sende dette til MODER på en foreskrevet kommunikasjonsmåte. MODER vil validere XML-dokumentet og videresende dette til riktig mottakeret i opprinnelig, eller omarbeidet form.

Sentralt er imidlertid datadefinisjonene i oppgaveregistret, og det faktum at man til enhver tid forholder seg til MODER som den endelige rapporteringsinstans.

En annen løsning vil kunne være at bedriftene selv på en eller annen måte maskinelt genererer XML-dokumentet og sender det direkte til MODER. Dette vil kunne se slik ut:



Figur 4

Figur 4 illustrerer at bedriften selv lager et XML-dokument i henhold til spesifikasjoner fra oppgaveregistret, og sender dette direkte til MODER.

3.5.1 Overføringsstandarder

Frem til nå har vi ikke nevnt hvordan dataene faktisk kan *overføres*.

Det finnes mange måter å overføre data på. Det finnes en rekke produkter og protokoller² som kan benyttes. Problemet med å benytte proprietære produkter er at dette normalt stiller krav til at både mottaker og avgiver må ha installert programvare fra samme leverandør.

Det eksisterer imidlertid programprodukter som benytter åpne standarder for meldingsutveksling, og som fint kan " snakke " med tilsvarende produkter fra andre leverandører. Vi skal her si litt om de mest aktuelle protokollene for datautveksling.

Dette vil bli noe teknisk, men det er en nødvendig del av arkitekturen som det ikke er så lett å beskrive på noen annen måte.

3.5.1.1 SOAP/Web Services

Frem til i dag har det ikke vært noen åpne standarder som har definert en datautvekslingsprotokoll til denne type formål. Et samarbeid mellom bla. IBM, Microsoft og andre har ført frem til en protokoll som kalles SOAP (Simple Object Access Protocol) og som baserer seg på HTTP og XML. Dette arbeidet har resultert i en spesifikasjon som er lagt fram for W3C (World Wide Web Consortium).

² En protokoll er i denne sammenheng en beskrivelse av hvordan data skal sendes mellom to eller flere maskiner. Det finnes mange slike beskrivelser; f.eks. HTTP, FTP og SMTP. Hver protokoll er som regel designet for bruk til spesielle formål.

SOAP er produktuavhengig og meldingene man sender med SOAP er definert av en XML-spesifikasjon. I og med at SOAP igjen kan basere seg på standard internettprotokoller (HTTP/SMTP), vil man kunne legge inn all nødvendig sikkerhet ved bruk av SOAP. Disse egenskapene gjør SOAP til en god kandidat for å spesifisere overføringsgrensesnittet mellom maskinelle løsninger på avgiversiden og mottakersiden.

Sammen med spesifikasjoner som UDDI (Universal Description Discovery and Integration) og WSDL (Web Services Description Language) utgjør SOAP det som kalles Web Services. Mange mener Web Services arkitekturen vil revolusjonere program til program kommunikasjonen over internett, og i teorien er denne arkitekturen som skapt for datautveksling mellom næringsliv og offentlig forvaltning. Erfaringene med denne arkitekturen er ennå ikke store, men det finnes løsninger på markedet som er lovende. Web Services passer også godt inn i den arkitekturen som ELMER legger frem her. I tillegg til å være en kilde for datadefinisjoner kan Oppgaveregistret på sikt også tenkes å bli en kilde for WSDL- og UDDI-spesifikasjoner.

3.5.1.2 HTTP

HTTP benyttes i dag av alle webservere og nettlesere, og er kjent for de fleste som jobber med webutvikling. Det er fullt mulig å spesifisere løsninger basert på ren HTTP, men man må gjøre en jobb med å standardisere HTTP-henvendelsene og publisere denne standarden. Da har man i prinsippet det samme som er gjort med SOAP. Da blir heller anbefalingen å benytte SOAP i stedet.

3.5.1.3 SMTP (E-post)

E-post har vært en leveringsmetode som mange mener er enkel å forholde seg til, og for noen typer data kan dette være en tilfredsstillende løsning.

Det er imidlertid noen aspekter ved det å benytte e-post som leveringsmetode. Normalt inneholder ikke e-postsystemer tilfredsstillende rutiner for hverken leveringsbekreftelse, logging, eller tilbakemelding. Dette betyr at man ikke kan være sikker på om dataene har kommet frem, eller om de har kommet til riktig sted. Den normale måten å løse dette på er at man har et mottakssystem som automatisk sender en bekræftelse tilbake til avsender via e-post. Men igjen har man ingen garanti for at tilbakemeldingen kommer frem.

Det finnes systemer som håndterer en del av disse problemstillingene, men det er viktig å være klar over at SMTP ikke primært er laget for denne typen formål.

3.5.1.4 JMS

Et eksempel på en standard som benyttes i flere og flere produkter er JMS (Java Message Service). JMS eies, og er spesifisert av Sun Microsystems inc. Dette gjør for såvidt ikke JMS til en "åpen" standard i ordets rette betydning, men JMS er i alle fall fritt tilgjengelig, både spesifikasjonen og de programbibliotekene som er nødvendige for å bruke JMS.

3.5.1.5 EDI

For bedrifter med eksisterende EDI-løsninger, eller bedrifter med store volumer data som skal utveksles, kan EDI vurderes som en transportmetode. Det må være opp til bedriftene selv å vurdere om de skal benytte EDI.

3.5.1.6 Oppsummering av overføringsstandarder

Man kan selvfølgelig også spesifisere egne datautvekslingsprotokoller, men etter vår oppfatning vil man ha mest å tjene på å benytte de mest hensiktsmessige eksisterende protokollene.

Hvilken metode som benyttes er ikke vesentlig for arkitekturen, eller for dataene. Her må driftsapparatet for en slik løsning vurdere hvilke protokoller man vil støtte.

Slik ståa er hos de fleste leverandører av programvare, og bedrifter med egne IT-avdelinger, vil nok SOAP være den protokollen som man er nærmest til å kunne utnytte. Microsoft har fått SOAP-støtte inn i sine verktøy, i tillegg til at dette også finnes bl.a. for Java³ og Perl⁴.

For de aller fleste vil det muligens være nok at de får tilgang til en webside hvor de kan laste opp sine XML-dokumenter. Dette fjerner behovet for å implementere kommunikasjonsløsninger i bedriftene. Hvis man derimot ønsker en eller annen form for automatikk i overføringsprosessen må man vurdere protokoller som SOAP, JMS eller EDI.

For ordens skyld kan vi nevne at det er utallige andre måter å gjøre dette på, men vi har prøvd å ta med de metodene vi ser at markedet mest sannsynlig kommer til å benytte.

3.6 Arkitekturens forutsetninger

Dette forslaget til løsningsarkitektur baserer seg i hovedsak på de ideer som ny teknologi inspirerer til. Det meste av ny (Informasjons- og Kommunikasjons-) teknologi i dag dreier seg om utnyttelse av Internett, og derfor tar denne arkitekturen også utgangspunkt i dette.

Internett skal benyttes som transportvei for de data som de oppgavepliktige skal levere til det offentlige i elektronisk form.

Den eneste kommentaren vi knytter til dette er at dette ikke er til hinder for å benytte andre måter (leide linjer ol.) for aktører med spesielle behov.

Oppgaveregistret blir en nasjonal datadefinisjonsdatabase, og spesifikasjoner for all statlig innrapportering skal legges der.

Oppgaveregistret har allerede registrert statlige skjema og brutt disse ned i enkelte dataelementer. For å kunne bruke disse definisjonene i en maskinell sammenheng må Oppgaveregistrets innhold utvides. Det som mangler er bl.a. definisjon av gruppestrukturer, datatyper og valideringsregler.. Hvis disse definisjonene gjøres åpent tilgjengelig, vil Oppgaveregistret bli en kilde for entydige spesifikasjoner for alle offentlige skjema i elektronisk form.

I tillegg bør man legge inn regelverk, veiledninger og beskrivelser på en sån måte at disse direkte kan benyttes som hjelpesystem i nye elektroniske innrapporteringsløsninger. Dette vil

³ Java er et fritt tilgjengelig plattformuavhengig programmeringsspråk, og er det hurtigst voksende på markedet i dag.

⁴ Perl er kanskje det mest utbredte programmeringsspråket for internettapplikasjoner på verdensbasis.

kunne redusere, og på sikt eliminere, behovet for å sende ut skjemaer, veiledninger og eventuelle spesifikasjoner av dataformater.

Alle data til det offentlige skal sendes til samme sentrale mottakstjeneste.

Hovedpoengene med dette er å gjøre det så enkelt som mulig for alle aktører. Med ett samlende leveringssted vil alle vite hvor de skal levere (og eventuelt hente) sine data. Rent administrativt og ressursmessig vil dette også være den mest gunstige løsningen. Det blir bare ett sted hvor man implementerer og vedlikeholder mottaksrutiner og sikkerhetsløsninger.

XML blir standarden for definisjon av dataformat. Spesifikasjonene baseres på XML Schema.

Argumentasjonen for dette er i hovedsak beskrevet i kapittelet 2 *Dataformater*, men et hovedpoeng med å velge XML er uten tvil den utbredelse og betydning XML har fått for eBusiness.

Alle brikker i denne arkitekturen skal ha grenseflater som er spesifisert via åpne standarder.

Hvis man skal ha noe håp om å kunne designe nasjonale løsninger som skal ta hensyn til næringslivets (og etatenes) mangfold, så må man basere løsningene på åpne standarder. Dette vil gi alle aktører samme vilkår, uavhengig av plattformer og systemer. Det er heller ingen ulempe at de kommende eBusinessløsninger primært baserer seg på åpne standarder. I tillegg til at dette er nyttig for offentlig elektronisk innrapportering, så gir man også programvareleverandører og bedrifter hjelp til å komme i gang med eBusinessløsninger.

4 Sikkerhet

Løsninger for elektronisk offentlig innrapportering vil kreve ulike grader av sikkerhet og autentisering. En arkitektur som skal favne hele innrapporteringsproblematikken må ha dette sikkerhetsaspektet med seg.

Det er vel ikke tvil om at digitale signaturer og PKI (Public Key Infrastructure) blir sentrale elementer for informasjonsutvekslingstjenester på Internett. Det finnes mange leverandøravhengige løsninger som fungerer i dag, men hvordan denne teknologien skal implementeres i stor skala i Norge diskuteres fremdeles.

Dette er et område hvor fravær av en defacto standard gjør det vanskelig å etablere en teknologi på bred basis. For ELMER sin del betyr dette at man ikke kan peke ut *den* teknologien som skal benyttes i løsninger for elektronisk innrapportering. Det en arkitektur derfor må ta høyde for, er at sikkerheten skal kunne implementeres slik at det i minst mulig grad påvirker resten av løsningen.

Ved å standardisere på åpne Internettprotokoller har man åpninger for å "plukke inn" de sikkerhetsløsninger som a) markedet selv ønsker å ta i bruk eller, b) offentlige myndigheter klarer å etablere som en standard.

Med den arkitekturen som er skissert i dette dokumentet, mener vi at det tas høyde for å implementere de sikkerhetsløsninger som kommer til å bli de gjeldende, uansett hvilke teknologivarianter som velges.

Med NOUen "Uten penn å blekk" som referanse, vil vi henstille om å legge så lite som mulig av sikkerheten til nivå 3. Lagring av sertifikater på eksterne media gir fremdeles kompliserte løsninger, og vi tror takten på elektronisk innrapportering blir betydelig redusert om man skal vente på standarder på dette området.

5 Oppsummering

Hovedmålsettingen med dette dokumentet er å beskrive en arkitektur for elektronisk offentlig innrapportering som krever minst mulig innsats fra offentlig side, og som gir størst mulig fleksibilitet for næringslivet.

Det er to hovedområder i dette problemområdet som er essensielle uansett arkitektur:

1. Definisjon av dataformat
2. Overføringsmetoder

Dette er områder man må etablere spesifikasjoner for, uansett hvordan man velger å designe en arkitektur. Hvis man tar utgangspunkt i at Internett skal benyttes som transportvei, så har man lagt til grunn en viktig forutsetning for valg av transportmetode.

Transportmetoder har igjen liten betydning hvis man ikke er enige om dataformatene som skal benyttes ved elektronisk formidling av data. Derfor vil definisjon av dataformater være det viktigste området for en løsning på elektronisk innrapportering.

Når vi skal beskrive disse dataformatene vil vi helt klart anbefale XML Schema som det språket som skal benyttes for dette formålet. XML Schema vil ta over som det primære datadefinisjonsspråket etter XML-DTD fordi det eliminerer alle svakheter som finnes i XML-DTD. XML Schema vil gi mulighet for å lage spesifikasjoner som ikke bare beskriver XML-dokumentenes felter og struktur, men også dataformater og valideringsregler for datainnhold.

Hvem skal i så fall lage disse XML Schema-spesifikasjonene? Oppgaveregistret inneholder allerede definisjoner for statlige skjema, skjemaenes felter, og skjemaenes struktur. Det som mangler er datatypene og valideringsreglene for de enkelte dataelementene. Vi anbefaler at dette blir en del av datadefinisjonene i Oppgaveregistrets database. Dette vil danne grunnlaget for XML Schema-spesifikasjoner av alle skjema som registreres i Oppgaveregistret.

Når man har laget et XML-dokument basert på definisjonene i Oppgaveregistret, må dette leveres til en mottakstjeneste ved å bruke en tilgjengelig transportmetode. Denne tjenesten skal så konvertere og videreformidle dataene til den aktuelle etaten.

Til sammen utgjør dette en komplett løsning for mottak av offentlig innrapporterte data.

Både mottaks- og avgiverløsninger som støtter denne arkitekturen finnes allerede på markedet, og arkitekturen legger også til rette for et utall av nye muligheter for løsninger fra de som måtte ønske å tilby dette.

Dette er ikke en arkitektur som er oppfunnet av ELMER, men som er satt sammen på grunnlag av løsninger man finner i B2B-systemer og EAI-systemer (Enterprise Application Integration).

Spesifikasjoner som XML Schema og SOAP har begynt å bre seg i markedet i dag. De funksjonene disse spesifikasjonene skal fylle har vært sårt savnet, og de legger på en måte kronen på verket i en B2B-, eller i dette tilfellet B2G-arkitektur.

Det vil selvfølgelig ligge mye arbeid i å utarbeide spesifikasjoner av alle grensesnitt, men vi mener fundamentet som ligger i Oppgaveregistret og den teknologien som eksisterer i dag, gir oss en unik mulighet til å faktisk implementere løsninger på relativt kort tid.

Kommentarer, spørsmål og innspill til dette dokumentet kan sendes til:

Prosjektleder for ELMER
Lars Uppheim
e-mail: lars.uppheim@nho.no
Tlf.: 23088410/91392868