

Design av løsning for automatisk måleravlesning av strøm (AMS)

Tor-Erik Steinsland

Master i kommunikasjonsteknologi

Oppgaven levert: Desember 2011

Hovedveileder: Poul Einar Heegaard, ITEM

Biveileder(e): Kjartan Mjøsund, Trådløse Trondheim
Lars Kulseng, Trådløse Trondheim

PROBLEMBESKRIVELSE

I løpet av 2016 skal alle norsk husstander ha automatisk måleravlesning av strøm. Strømforbruket skal leses av automatisk hver time, i tillegg skal energileverandøren eller sluttbrukeren kunne spørre måleren i sanntid for å få et bilde av strømforbruket. Det krever en løsning med robust toveis kommunikasjon. 2/3 av alle hjem har trådløst bredbånd. Er det mulig å benytte eksisterende infrastruktur for AMS? NVE har stadfestet at uavhengig av valgt kommunikasjonsinfrastruktur, skal målerdata gjøres tilgjengelig for tredjepart. Dette for å stimulere til tjenesteinnovasjon og økt konkurranse i markedet. Ved tilgjengeliggjøring av målerdata til tredjepart dukker det derimot opp en rekke utfordringer. Hvordan skal målerdata deles? Hvem skal få tilgang til målerdata? Hvordan kan datasikkerheten og hensynet til personvern ivaretas? I tillegg er det interessant å se på hvilke krav nettselskapene vil stille til en uavhengig kommunikasjonsleverandør, for at de skal godta en ikke-direkte adgang til målerdata. I oppgaven skal studenten belyse de spørsmålene nevnt over, i tillegg til å utvikle et testgrensesnitt for deling av målerdata. Rent praktisk skal følgende gjøres:

- Litteraturstudie av NVEs og nettselskaperes krav til en AMS-løsning
- Intervju med representanter fra utvalgte nettselskaper og NVE
- Litteraturstudie av sikker deling av personsensitive data
- Utvikling av et testgrensesnitt i Java for deling av målerdata

Oppgave gitt: 8. august 2011

Faglærer: Poul E. Heegaard

Student: Tor-Erik Steinsland

ABSTRACT

Smart metering should be fully deployed before January 1st 2017 in Norway. The smart meters should be able to report consumption on a hourly basis, meaning that the amount of meter data will increase drastically. This raises some challenges with regards to sharing of meter data. The meter data has to be shared with the power producer, and the regulator has also stated that meter data also should be made available to third parties at the consumers wish. Meter data is important financial data and is also considered as sensitive data, and should be shared in a secure and reliable manner.

This thesis gives an introduction to the smart metering technology, as well as the power industry in Norway. Further on, it presents some of the ICT solutions used in the industry today, as well as some proposed future solutions. Based on the proposed solutions and comments received from the industry actors, a test system with minimum centralization of business processes using Web Services technology has been developed.

A solution based on the existing system, Nubix, is proposed. The system works as a forwarding hub, meaning that it forwards request from power producers and third parties to the grid owners. By using this solution the development costs can be kept low, and the system can easily be extended if more centralization of meter data and business processes is needed.

FORORD

Denne oppgaven ble skrevet til faget TTM4905 - Nett og tjenester med fordypning i teleøkonomi. Oppgaven ble skrevet høsten 2011 og avslutter graden Master of Science i Kommunikasjonsteknologi ved Institutt for telematikk på Norges teknisk-naturvitenskapelige universitet (NTNU).

Veiledere for oppgaven har vært Lars Kulseng og Kjartan Mjøsund ved Trådløse Trondheim. Jeg ønsker å takke disse for hjelpen og innspillene jeg har fått underveis.

Jeg ønsker også å takke mine medstudenter for å ha bidratt til en uforglemmelig studietid.

TABELLER

6.1	Kundeforhold hos "GridA"	62
6.2	Kundeforhold hos "GridB"	62
6.3	Webtjenesteadresser	63
6.4	Oversikt over aktørenes nøkler, sertifikater og betrodde sertifikater	64
6.5	Meldingselementens størrelse i bytes	75
6.6	Utnyttelsesgrad	76
6.7	Krav til båndbredde	77
C.1	Klientenes "Keystores"	116
C.2	Brukernavn og passord for de ulike databasene	117
C.3	"Main"-metodene for de ulike klientene	118

FIGURER

2.1	AMS - forenklet arkitektur	5
2.2	AMS - Utvidet arkitektur	10
3.1	Nubix' arkitektur	22
4.1	SOAP "Envelope"	35
4.2	Forslag til arkitektur for sentralisert database	42
4.3	Forslag til Nubix-basert arkitektur	44
5.1	Nubix 2.0	50
5.2	Database - Nubix 2.0	52
6.1	Oppsett av HTTPS-sesjonen	69
6.2	Databasedesign hos "GridA" og "GridB"	71
C.1	Oppsett av Apache Tomcat	113
C.2	Oppsett av Axis2	114

AKRONYMER

AMS	Avanserte Måle- og Styresystemer
DIS	Datainnsamlingsystem
EDI	Electronic Data Interchange
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
JIP	Justert Innmatingsprofil
KIS	Kundeinformasjonssystem
MTM	Maskin-Til-Maskin
MVDB	Måleverdidatabase
NVE	Norges Vassdrags- og Energidirektorat
PLC	Power Line Communication
SMTP	Simple Mail Transfer Protocol
WSDL	Web Services Description Language
WTPR	Webtjenestepolicy-rammeverk
WTS	Webtjenestesikkerhet

XML Extensible Markup Language

INNHOOLD

1	Introduksjon	1
2	Avanserte Måle- og Styringssystemer	3
2.1	Bakgrunn	3
2.2	Kraftaktører og kraftnett i Norge	6
2.3	Krav til AMS	7
2.4	Arkitektur og kommunikasjonsteknologi	8
2.5	Verdivurdering	11
2.5.1	Kostnader	12
2.5.2	Nytte	13
2.5.3	Er det verdt det?	15
2.6	Oppsummering	18
3	Informasjons- og Kommunikasjonsteknologi i Kraftbransjen	19
3.1	Dagens IT-løsning	19
3.1.1	Ediel	19
3.1.2	Nubix	20
3.2	Rapport: Felles IKT-løsninger i det norske kraftmarkedet	23
3.2.1	Distribuert database	23
3.2.2	Felles måleverdidatabase	24
3.2.3	Felles behandlingsfunksjoner	25
3.2.4	Kommentarer fra bransjen	27
3.3	Oppsummering	29

4	Valg av IT-løsning	31
4.1	Webtjenester	31
4.1.1	WSDL	32
4.1.2	SOAP	34
4.1.3	Sikkerhet	36
4.1.4	Webtjenester - Fordeler og oppsummering	39
4.2	Valg av systemarkitektur	40
4.2.1	Sentralt lagret data	41
4.2.2	Nubixbasert meldingsformidler	43
4.2.3	Sentral lagring eller videresending?	45
4.3	Oppsummering	46
5	Design og Implementasjon	47
5.1	Krav	47
5.1.1	Arkitektur og funksjonalitet	47
5.1.2	Datastruktur	48
5.1.3	Sikkerhet	49
5.2	Design	49
5.2.1	Database	51
5.2.2	Metoder	51
5.3	Implementasjon	52
5.3.1	Utviklingsmiljø	53
5.3.2	Metodikk	53
5.3.3	Datastruktur	54
5.3.4	Meldingsdefinisjoner	55
5.3.5	Sikkerhet	56
5.4	Oppsummering	59
6	Resultater og Diskusjon	61
6.1	Resultater	61
6.1.1	Testoppsett	61
6.1.2	Meldingsutveksling	64

6.1.3	Sikkerhet	67
6.1.4	Oppsett hos nettselskapene	70
6.2	Diskusjon	72
6.2.1	Arkitektur	72
6.2.2	Datastruktur	73
6.2.3	Utnyttelsesgrad og krav til båndbredde	73
6.2.4	Sikkerhet og nøkkelhåndtering	77
6.2.5	Roller og autorisasjoner	78
6.2.6	Feilmeldinger	78
6.2.7	Hva med sluttbrukerne?	79
6.3	Oppsummering	79
7	Konklusjon	81
A	WSLD-definisjoner	87
B	Meldingsutveksling	101
C	Instruksjoner for oppsett av systemet	111
C.1	Nødvendig programvare	111
C.2	Oppsett av Eclipse	112
C.3	Sikkerhetsoppsett	115
C.3.1	Tjenerinstanser	115
C.3.2	Klienter	115
C.3.3	Nubix 2.0	116
C.3.4	Nettselskap	117
C.4	Databaseoppsett	117
C.5	Biblioteker	118
C.6	Testing	118

1

INTRODUKSJON

Innen 1. januar 2017 skal alle husstander i Norge ha byttet ut sine mekaniske strømmålere med nye strømmålere, heretter kalt smartmålere, som Leser av og sender inn strømforbruket automatisk. Avlesningsintervallene skal kunne være så korte som 15 minutter, i tillegg skal det legges til rette for toveiskommunikasjon mellom strømmåler og nettselskap. Med toveiskommunikasjon menes det at det skal være mulig utføre enkelte styringsfunksjoner, for eksempel struping av strømforbruket ved knapphetssituasjoner.

Norge har gjennom internasjonale forpliktelser et press på seg for å øke den totale energieffektiviteten. For å få til dette er det kritisk å bevisstgjøre sluttbrukere om sitt forbruksmønster. Med dagens manuelle kvartalsvise rapportering av målerstand har derimot ikke sluttbrukere det nødvendige datagrunnlaget for å gjøre en kvalitativ vurdering av sitt forbruk. AMS vil endre på dette, da det vil innebære at et detaljert datagrunnlag blir tilgjengelig. Sluttbrukere kan da få en detaljert oversikt over sitt forbruksmønster, og kan enklere vurdere og se effektene av energieffektiviserende tiltak.

Fokusområde

En AMS-utrulling introduserer flere problemstillinger; valg og utrulling av smartmålere, valg av kommunikasjonsinfrastruktur og utviklingen av sentrale IT-systemer som distribuerer målerdata, for å nevne noen. Denne oppgaven vil fokusere på den sistnevnte problemstillingen. NVE har satt som krav at målerdata skal kunne deles med tredjeparter hvis sluttbrukeren ønsker dette, i tillegg til at målerdata må deles med den kraftprodusenten som leverer strøm til sluttbrukeren. Dette medfører at det må foretas store investeringer på IKT-siden, og denne oppgaven vil blant annet se på ulike muligheter

for et system for deling av målerdata. Hvilke løsninger har blitt lansert? Hvilke krav stiller nettselskapene til en IT-organisering? Hvilke IT-systemer er allerede på plass for meldingsutveksling og rapportering i kraftbransjen, og er det aktuelt å basere fremtidige løsninger på en videreutvikling av disse?

Struktur

Resten av oppgaven er organisert som følger. Kapittel to gir en innføring i AMS og kraftbransjen i Norge. Kapitlet vil også presentere kravene som per dags dato er gjeldende for en AMS-implementering, samt gjøre en kort verdivurdering av AMS. Videre vil kapittel tre ta for seg IKT i kraftbransjen, hvilke systemer som allerede er på plass, og hvilke forslag til nye systemer som har blitt lansert. I kapittel fire presenteres Webtjenester, et omfattende rammeverk for meldingsutveksling over Internett. I tillegg blir det, basert på kraftaktørens kommentarer angående en fremtidig IT-løsning, gitt en overordnet spesifisering og vurdering av et mulig system for deling av målerdata. Kapittel fem vil definere en rekke funksjonelle og ikke-funksjonelle krav til systemet, før de ulike design- og implementasjonsspesifikke valgene som har blitt tatt blir presentert. Kapittel seks vil presentere resultatene av implementering, samt gå inn på en diskusjon omkring de konkrete valgene som har blitt tatt. Oppgaven konkluderes i kapittel syv.

2

AVANSERTE MÅLE- OG STYRINGSSYSTEMER

Dette kapitlet tar sikte på å gi en innføring i Avanserte måle- og styresystemer (AMS) fra et overordnet perspektiv. Kapitlet begynner med en rask beskrivelse av AMS og dets hensikter, etterfulgt av en beskrivelse av kraftmarkedet i Norge. Videre beskriver jeg enkelte av kravene norske og europeiske myndigheter stiller til en AMS-implementasjon og noen overordnede tekniske løsninger. Kapitlet avsluttes med en beskrivelse av potensielle nytteverdier og kostnader forbundet med AMS. For mer utfyllende informasjon om AMS, henviser jeg til de tidligere masteroppgavene [43], [37] og [38].

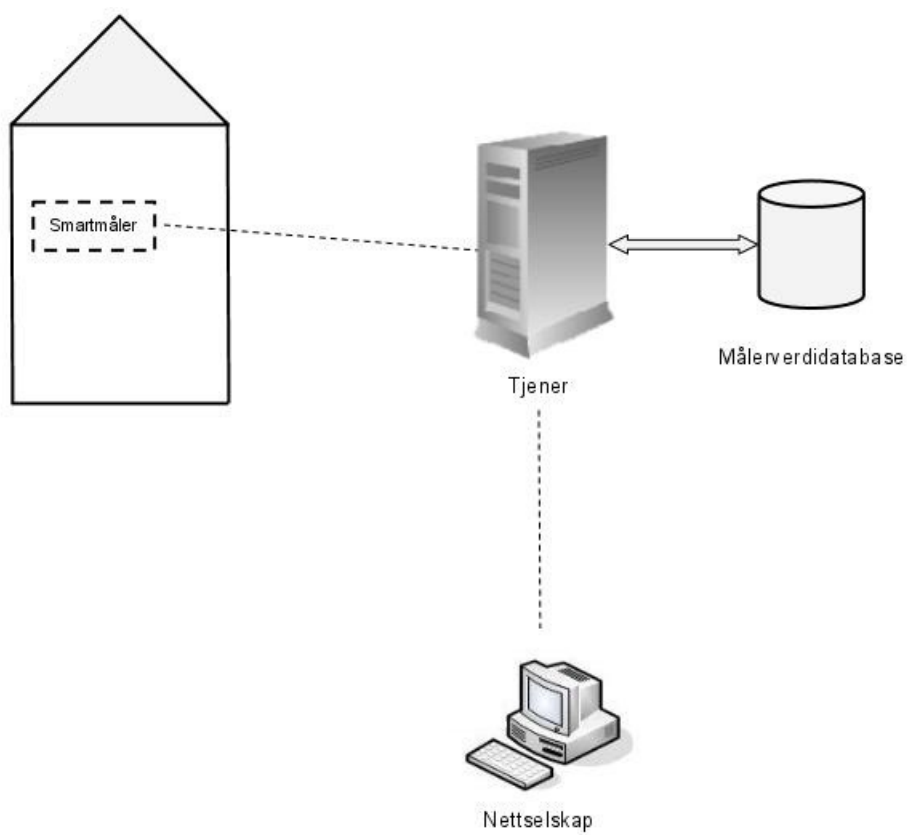
2.1 Bakgrunn

Det har de siste årene vært mye fokus på klimaproblematikk og et stadig stigende strømforbruk, nasjonalt, regionalt og globalt. En modernisering av kraftnettet, med mål om å legge til rette for et redusert og mer bærekraftig forbruk, er derfor et fokusområde. Begynnelsen på en slik moderniseringsprosess, som etter planen skal ende i det som kalles et smartgrid, er innføringen av AMS. AMS er et begrep tatt i bruk av Norges Vassdrags- og Energidirektorat (NVE), et organ underlagt Olje- og Energidepartementet med ansvar for å forvalte vassdrags- og energiresursene i Norge, for å beskrive teknologien som skal bli tatt i bruk for måling og avregning av sluttbrukerne i kraftmarkedet. I NVEs "Forslag til endring i forskrift 11. mars 1999 nr. 301", [30], setter de følgende overordnede mål med innføringen av AMS:

Hovedmålsettingen med innføringen av AMS er å bidra til et mer samfunnsøkonomisk rasjonelt kraftmarked gjennom å legge til rette for:

- *Mer effektiv avregning.*
- *Mer effektiv leverandørbytteprosess.*
- *Optimal tilpasning av forbruk og lokal produksjon.*

Forenklet kan vi si at AMS består av en smartmåler, kombinert med kommunikasjons-teknologi, en tjener som bl.a kjører et datainnsamlingsystem (DIS) og funksjonalitet for kommunikasjon med nettselskapenes avregningssystemer og en måleverdidatabase (MVDB), se figur 2.1. Målerdata sendes med jevne mellomrom automatisk fra hvert målepunkt over valgt kommunikasjonsmedium til et DIS, som lagrer målerdata i MVDBen. For nettselskapene legger AMS til rette for en mer nøyaktig avregning av sluttbrukerne samt en automatisering av avregningsprosessene, mens en sluttbruker kan ta glede av en mer detaljert oversikt over sitt forbruksmønster, samt han eller hun de slipper å selv rapportere målerstand til nettselskapene. For en mer utfyllende beskrivelse av de potensielle fordelene med AMS, se seksjon 2.5.



Figur 2.1: AMS - forenklet arkitektur

2.2 Kraftaktører og kraftnett i Norge

Det norske kraftmarkedet består av ulike typer fysiske nett, vi kan skille mellom tre hovedtyper, [16]: sentralnettet, regionalnettene og de lokale nettene, også kalt distribusjonsnettene. Sentralnettet er på mange måter hovednettet i Norge, da det binder sammen de ulike regionale og lokale nettene til et riksdekkende kraftnett. Sentralnettet er som regel et høyspenningsnett, og det driftes og utvikles av Statnett, [19]. De regionale nettene opererer kun på regionalt basis og fungerer som et bindeledd mellom sentralnettet og de lokale nettene. Industriverk med stort energibehov er ofte direkte koblet på de regionale nettene, [38]. De lokale nettene, eller distribusjonsnettene, er nettene som er fysisk bygd ut til sluttbrukerne. De eies og driftes av nettselskapene. Nettselskapene drives som naturlige monopol, og er strengt regulert av NVE for å hindre de negative aspektene som ofte kommer frem i monopolsituasjoner.

I tillegg til de tre fysiske typene nett, finnes det ulike aktører i kraftbransjen i Norge, vi kan skille mellom Statnett, nettselskap, kraftleverandører, også kalt produksjonsselskap, og Nord Pool.

Statnett er den norske systemoperatøren ansvarlig for utvikling og drifting av sentralnettet. Kontakten med sluttbruker er det primært nettselskapene som tar seg av, gjennom nettselskapenes ansvarsområder - måling, avregning og overføring av kraft til sluttbrukerne, [38]. Kraftleverandører, eller produksjonsselskap, produserer kraft, og selger og kjøper kraft via Nord Pool. Kraftselskapene leverer strøm til sluttbrukere via nettselskapene. Innenfor en region er det som regel kun ett nettselskap, men det kan være flere kraftprodusenter. Markedet for produksjon og salg av kraft er med andre ord ikke organisert som et naturlig monopol, slik som det er for transport av kraft, [11]. I dag deltar Norge i et felles kraftmarked med en felles kraftbørs, Nord Pool, med Sverige, Finland og Danmark, [12]. Det er i dette markedet nettselskapene kjøper strøm på vegne av sluttkundene. Markedet er organisert etter klassiske markedsøkonomiske prinsipper, det vil si at det er tilbud og etterspørsel som bestemmer prisen på kraft i dette markedet. Merk at selv om produksjons- og nettaktiviteter med tanke på kraft er forskjellige logiske enheter, så er det ikke uvanlig at enhetene er organisert under samme konsern, for eksempel at nettselskapet er et datterselskap av produksjonsselskapet, [38].

2.3 Krav til AMS

Som nevnt er NVEs ansvarsområde å forvalte vassdrags- og energiressursene i Norge. Innenfor dette mandatet ligger også kravspesifisering og standardisering av en AMS-løsning. Lignende prosjekter pågår verden over, og særlig arbeidet som pågår under EU-mandatet M/441 og i organisasjonen OPENmeter er av stor interesse for NVE. M/441 tar sikte på å standardisere funksjonaliteten for, og kommunikasjonen mellom, smartmålere i Europa, det være elektrisitets-, gass-, varme eller vannmålere. Det er et mål å opprette et felles og homogent kraftmarked på tvers av EU, og denne standardiseringsprosessen skal bidra til å legge det tekniske fundamentet for et slikt marked.

Organisasjonen OPENmeter har som ansvarsområde å spesifisere standarder som skal benyttes i AMS. Resultatet skal være et sett med åpne og offentlige standarder, basert primært på eksisterende og aksepterte standarder og protokoller, men også nye standarder skreddersydd for AMS. Eksisterende standarder som vil bli inkludert er bl.a. "Power line communication" (PLC) og DLMS/COSEM, se [38] for en mer utdypende beskrivelse av disse standardene.

NVE har tidligere uttalt at de følger det europeiske standardiseringsarbeidet nøye. Det er viktig at løsningene som blir valgt ikke blir ansett som særnorske, både fordi det vil være kostnadsøkende, men også fordi Norge også ønsker å delta i det fremtidige europeiske fellesmarkedet, noe som blir vanskeligere hvis standarder og protokoller brukt i Norge ikke er kompatible med de europeiske. Med dette som bakgrunn har NVE siden 2007 definert en rekke krav. Kravene er jevnlig ute til høring hos relevante aktører, og har blitt endret på enkelte punkter siden den gang. De kravene som har relevans for denne oppgaven kan oppsummeres med:

1. Kommunikasjon med eksternt utstyr

NVE krever at smartmålere skal kunne kommunisere med eksternt utstyr, for eksempel et display. Tidligere var det et krav om at denne kommunikasjonen skulle være IP-basert, men på grunn av innvendingene fra nettselskaper og utstyrsleverandører og det faktum at EUs standardiseringsarbeid ikke har offentliggjort en valgt kommunikasjonsstandard, har NVE valgt å gå bort fra kravet om IP, [31]. NVE

oppfordrer derimot til samarbeid mellom nettselskaper og utstyrsleverandører, for å unngå et u hensiktsmessig høyt antall ulike kommunikasjonsstandarder og for å spesifisere et felles format på data som sendes fra måler til eksternt utstyr.

2. Tredjepartstilgang

NVE stadfester at det skal åpnes for tredjepartstilgang til AMS-infrastrukturen. Med det menes det ikke nødvendigvis at tredjepart skal ha fysisk tilgang til kommunikasjonskanalen, men at nettselskapene plikter å formidle relevant informasjon mellom tredjepart og sluttbruker. Det er verdt å merke seg at NVE spesifiserer at det er sluttbruker som avgjør hvem som skal få tilgang til sine forbruksdata, [30]. Det vil si at tredjepart må ha samtykke fra sluttbrukerne, før datautveksling mellom nettselskap og tredjepart skal kunne skje.

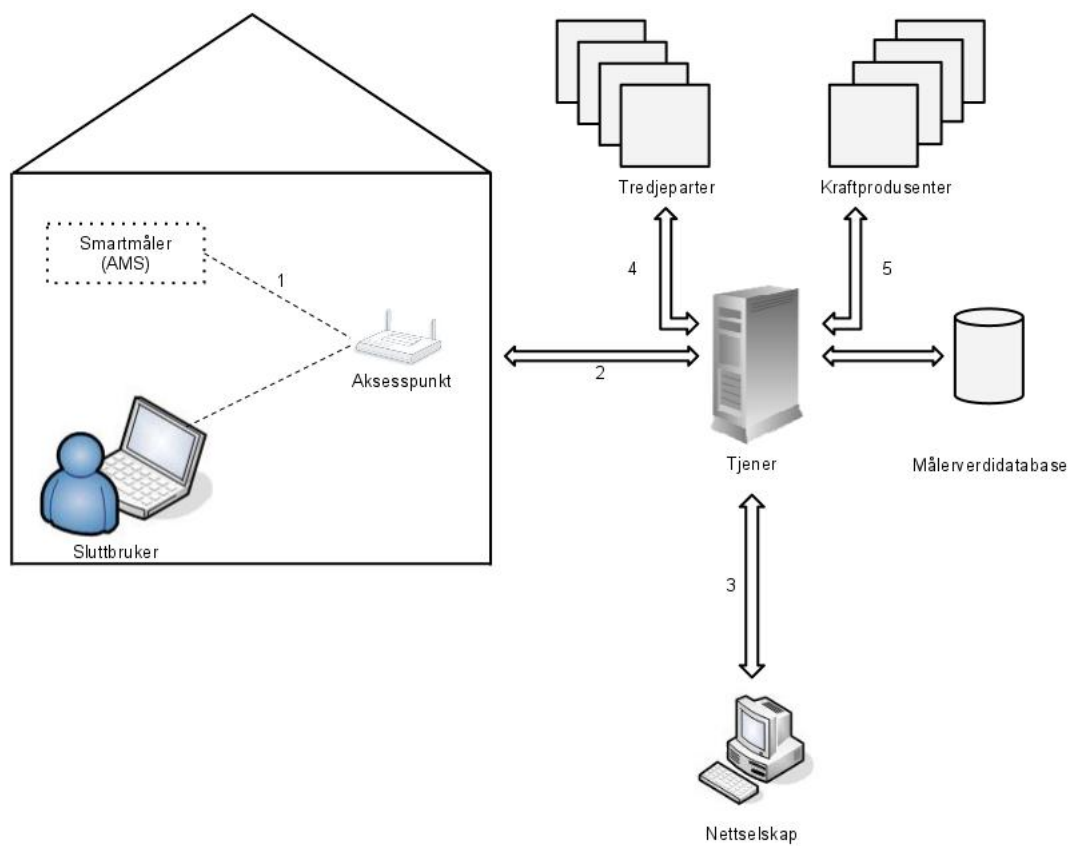
3. Sikkerhet

En stor utfordring ved innførsel av AMS er sikkerhet. Nettselskapene frykter at bruk av åpne kommunikasjonskanaler vil øke risikoen for misbruk og manipulering av målerdata. Nettselskapene ser også på tredjepartstilgang som en betydelig trussel for integriteten til målerdata. NVE understreker at AMS-implementeringene må sikres mot uautorisert tilgang, for å hindre lekkasje av kundedata eller manipulasjon av målerdata. Ansvar for sikkerheten legges på nettselskapene, [31]. NVE har stadfestet at kommunikasjonen skal baseres på åpne standarder, men åpner i nevnte høringsdokument for at proprietære løsninger kan benyttes hvis nettselskapene har tungtveiende sikkerhetsmessige argumenter for det. Merk at tredjepartstilgang er et ufravikelig krav, uavhengig av valgt kommunikasjonsløsning.

2.4 Arkitektur og kommunikasjonsteknologi

Det er mange utfordringer som må løses før AMS kan ruller ut i stor skala, ikke minst med tanke på valg av kommunikasjonsløsninger og arkitektur. Nettselskapene har tidligere ytret ønske om å benytte en kontrollert og proprietær PLC-basert løsning for transmisjon av målerdata. Rammen for denne oppgaven derimot, er at AMS skal ruller ut basert på eksisterende kommunikasjonsinfrastruktur, der det er mulig. Dette innebærer at et

komplekst økosystem av aktører og teknologier må samarbeide for å få en fungerende løsning. Vi kan se for oss en IP-basert arkitektur, illustrert i figur 2.2. Som leseren kan se, er tjeneren og MVDBen skissert separat fra nettselskapet. Dette er gjort for å illustrere de ulike elementene, og det betyr ikke at nettselskap, tjener og MVDB ikke kan være samlokalisert.



Figur 2.2: AMS - Utvidet arkitektur

Smartmåleren er koblet til husstandens aksesspunkt, kobling "1". Ulike protokoller kan benyttes her, for eksempel Ethernet, WiFi og Zigbee for å nevne noen. Videre sendes målerdata over Internett til en tjener, xDSL, WiFi og fiber er eksempler på mulig kommunikasjonsmedium, kobling "2". Denne tjeneren støtter samme funksjonalitet som tjeneren i figur 2.1, i tillegg til funksjonalitet for datautveksling med tredjepart.

Valg av kommunikasjonsløsning kan i stor grad påvirke kostnaden, nytteverdien og levetiden til en AMS-utrustning. [38] presenterer en grundig vurdering av ulike kommunikasjontechnologier. Rapporten kommer frem til at valg av kommunikasjontechnologi må tilpasses den enkelte situasjonen. Fiber, mobilteknologi (GSM, UMTS, HSPA, LTE), PLC og radiofrekvens (RF) er alle egnede teknologier under visse forutsetninger. For kommunikasjons internt, mellom smartmåleren og aksesspunktet og eventuelt annet eksternt utstyr, for eksempel et display, anbefaler rapporten Zigbee, med forbehold om at Zigbee blir standardisert av internasjonal organisasjoner. Se rapporten for mer utfyllende beskrivelse og evaluering av kommunikasjonsløsningene.

En annen utfordring er lokalisering og styring av tjeneren og MVDBen. Det er naturlig å tro at flere nettselskaper, på grunn av begrensede erfaring med IKT, vil outsource disse prosessene til andre aktører. Uansett løsning, er det kritisk for nettselskapene å få tilgang på målerdata, da disse er grunnlaget for nettselskapenes inntjening. I tillegg er det også et krav om at nettselskapene må dele målerdata med tredjepart.

I resten av denne oppgaven vil vi primært fokusere på kommunikasjonen mellom tjeneren og nettselskapet, "3", kommunikasjonen mellom tjeneren og ulike tredjeparter, "4", samt kommunikasjonen mellom tjeneren og kraftprodusenter,"5". Det antas at all kommunikasjon mellom smartmåleren og tjeneren er etablert og fungerende.

2.5 Verdivurdering

Det er knyttet store forventninger til effektene av en AMS-utrustning. NVE beskriver følgende overordnede fordeler med AMS på sine nettsider, [14]:

AMS gjør at strømkundene får bedre informasjon om kraftforbruket sitt, mer nøyaktig avregning og mulighet for automatisk styring av forbruket. AMS gir strømkundene mulighet til å ta styringen over strømforbruket, og vil kunne bidra til en bedre fordeling av

strømforbruket og et mer fleksibelt kraftmarked.

Dette avsnittet vil presentere kostnader og forventede nytteverdier ved en fullskala AMS-utrulling.

2.5.1 Kostnader

En AMS-utrulling vil medføre store kostnader for nettselskapene. Rådgivningsselskapet Econ Pöyry gjorde i 2007 en nytte- og kostnadsanalyse av en fullskala AMS-utrulling i Norge, [29]. Rapporten deler kostnadene opp i postene:

- Innkjøp av målere.
- Monteringskostnader.
- Kostnader til drift og vedlikehold i form av arbeidskraft og utstyr.
- Endringer i kundeinformasjonssystem
- Etablering av ny MVDB
- Automatisering av grensesnitt mellom datasystemer for innsamling og håndtering av informasjon
- Redesign av arbeidsprosesser for å håndtere store mengder data fra automatiske målere

Videre blir det presentert ulike tallverdier på hvor mye det vil koste per smartmåler. Rapporten viser til tidligere undersøkelser, samt data hentet inn fra nettselskaper som har implementert AMS. Rapporten viser til tallverdier fra tidsrommet 2004 til 2007, og investeringskostnad pr. målepunkt mellom henholdsvis 3150,- og 1850,- kroner, men vekten til de ulike kostnadspostene blir ikke forklart videre. Med en avskrivningstid på 10 år og 5 prosents rente blir den årlige kostnaden per målerpunkt henholdsvis mellom 640,- og 380,- kroner. Synkende kostnader for maskinvare og modning av teknologi brukes som forklaring på de synkende kostnadene. Merk at i 2007 var ikke timesmåling et aktuelt tema, kun måling på ukes- eller månedsbasis, så det forventes at investering i IT-systemer som håndteres timedata vil øke kostnadene betraktelig.

2.5.2 Nytte

For å gå tilbake til NVEs hovedmålsetning med innføring av AMS, nevnt i seksjon 2.1, skal AMS legge til rette for en mer effektiv avregning, en mer effektiv leverandørbytteprosess og en optimal tilpasning av forbruk og lokal produksjon.

Effektivisering av avregningsprosessen

Dette er nytteverdi som i stor grad tilfaller nettselskapene. De kan oppleve kostnadsbesparelser ved effektivisering av interne rutiner både for avregning, men også avlesning av målerdata. I tillegg nevner rapporten fra Econ Pöyry en redusert risiko for strømtveri og kreativ målerrapportering fra sluttbrukerne som en nytteverdi. Rapporten viser til samtaler med nettselskaper, som oppgir en gjennomsnittlig kostnadsbesparelse på omtrent 180,- kroner pr. måler pr. år, som følge av denne effektiviseringen. Besparelsene kommer av det at nettselskapene slipper å sende ut målerkort, i tillegg til nedbemanningen som følger av en større grad av automatisering av avregningsprosessen. En effektivisering av avregningsprosessen vil også komme sluttbrukerne til gode, da de vil slippe arbeidet med å manuelt lese av måleren, samt huske å rapportere målerstanden innen fristen.

Effektivisering av leverandørbytteprosessen

Ved bytte av kraftleverandør i dag kan det skje at sluttbruker må betale et korreksjonsoppgjør. Dette er nødvendig fordi nettselskapet er nødt til å estimere forbruket utifra når leverandørskiftet skal finne sted. En AMS-implementasjon skal kunne bidra med nødvendige data for at målerstanden kan leses av i det sluttbruker bytter kraftleverandør, og på den måte spare nettselskapene for estimering og kontroll av målerstanden. Denne effektiviseringen vil redusere grad av innlåsing av sluttbrukere hos nettselskapene, og på den måten bidra til økt konkurranse i kraftmarkedet.

Optimal tilpasning av forbruk

Dette er muligens den posten med størst sparepotensial. AMS legger til rette for at sluttbrukere kan få tilgang på en mye mer detaljert oversikt over deres strømforbruk enn tilfellet er i dag. Sluttbrukere vil få en forbedret mulighet til å få oversikt over hvor mye

strøm ulike apparater og oppvarmingskilder trekker, i tillegg til en generell bevisstgjøring om eget forbruk. Med dagens kvartalsvise rapportering er dette meget vanskelig. Et sitat fra Kempton og Layne, gjengitt i [38], illustrerer dagens situasjon godt:

Consider groceries in a hypothetical store totally without price markings, billed via a monthly statement. How could grocery shoppers economise under such a billing regime?

For å legge til rette for at sluttbrukere skal kunne ta større kontroll over forbruket deres, er det viktig at målerdata presenteres på en rask, fornuftig og individuelt tilpasset måte. Alternativer for presentasjon kan for eksempel være en mobil applikasjon, en individuell nettside eller et dedikert display montert hos sluttbruker. Tidligere var det et krav fra NVE at nettselskapene var pliktet til å montere display hos sluttbrukere som ønsket det, men det kravet gikk de bort i fra ved forrige høring¹, [31]. Rapporten "Smart Metering Guide - Energy Saving and the Customer", [41], fra "European Smart Metering Alliance" skiller mellom to alternativer mtp. hvordan sluttbrukere kan få feedback; direkte og indirekte. Indirekte feedback betyr feedback som har blitt behandlet før den når kunden, i tillegg til at den ikke kommer direkte fra smartmåleren. Nettsider, tradisjonelle regninger og mobile applikasjoner er eksempler på indirekte feedback. Direkte feedback er feedback som mottas i sanntid, direkte fra måleren og typisk presentert på et dedikert display.

[41] viser til forskning som hevder at forbedret indirekte feedback, lagt til rette av en AMS-utrulling, vil kunne bidra til en forbruksreduksjon på mellom 0 og 10 %, avhengig av kvaliteten på informasjonen. Det hevdes også at indirekte feedback vil presentere langtidsvirkningen av strømsparingstiltak best, for eksempel effektene av forbedret isolasjon, innføring av energieffektiviseringsystemer, strømstyring, etc.

Når det gjelder direkte feedback, hevdes det at relativt enkle display vil kunne bidra til en strømreduksjon på mellom 5 og 15 %, typisk rundt 10 %, [40]. Direkte feedback på et display legger til rette for at sluttbrukere kan få rask og effektiv oversikt over deres nåtidsforbruk, samt se effektene av merforbruk og eventuelle energisparingstiltak de iverksetter. Dedikerte display kan også brukes til å vise mer avanserte forbruksdata, se for deg et display som viser hvor mye strøm hvert enkelt rom i en bolig trekker. I tillegg kan man tenke seg at nettselskapet sender informasjon om sanntidsprisen på

¹Det er derimot fortsatt et krav om at målerne som installeres skal ha et grensesnitt som tillater tilkobling av display og annet eksternt utstyr.

strøm til displayet, slik at kunden kan få en nøyaktig oversikt over akkurat hvor mye sanntidsforbruket vil koste.

I tillegg til det uttalte målet å redusere forbruket generelt, er det også et mål å få strømforbruket fordelt jevnere utover døgnet. I typiske høylast-perioder i Norge må kraftprodusentene ofte ty til fordyrende løsninger som høylastkraftverk og driftsreserver, i tillegg til import av strøm fra Europa, [38]. Med dagens ordning om kvartalsvis innrapportering av målerdata er det vanskelig for nettselskapene å gjøre en rettfærdig avregning i slike situasjoner. Nettselskapene benytter i dag en modell som heter justert innmatingsprofil (JIP), som viser forbruksmønsteret til gjennomsnittskunden i et nettområde, [5]. Med AMS har nettselskapene mulighet til å avregne sluttbrukerne for akkurat det de bruker, så sluttbrukere som bruker mye i høylastperioder må med andre ord betale for det i større grad enn nå. På den andre siden, kan sluttbrukerne få informasjon om, f.eks via et display, at man går inn i en høylastperiode, og at strømprisene vil stige. Sluttbrukere har da en mulighet til å justere ned sitt forbruk og på den måten få lavere strømrregning.

Lokal produksjon

Den siste nytteverdien handler om lokal produksjon av strøm. I et fremtidig smartgrid vil sluttbrukere i enkelte strøk vil kunne dekke deler av sitt forbruk gjennom lokalt produsert fornybar energi, f.eks solkraft. En AMS-implementering skal legge til rette for at eventuell overskuddskraft fra den lokale produksjonen skal kunne selges tilbake i nettet. Sluttbrukere som benytter seg av dette vil da kunne få finansiert deler av strømforbruket sitt, i teorien også kunne gå i pluss.

2.5.3 Er det verdt det?

Man kan se på nytteverdien for AMS fra både et bedriftsøkonomisk og samfunnsnyttig perspektiv. For å begynne med det bedriftsøkonomiske perspektivet, som er enklere å tallfeste, skiller vi mellom nytte for nettselskaper og nytte for sluttbrukere.

For nettselskapene gir AMS primært en nytteverdi i at de kan effektivisere avregningsprosessen. Som nevnt i 2.5.2, oppga spurte nettselskaper i 2007 en gjennomsnittlig besparelse på om lag 180,- kroner per måler per år som følge av effektivisering. I 2.5.1 presenterte

vi kostnader forbundet med en AMS-utrulling, og viste til synkende kostnader per måler over tid. I 2007 var det i følge rapporten sitert en kostnad på ca. 380,- kroner pr. måler pr. år, noe som ville gitt et årlig tap pr. måler for nettselskapene på 200,- kroner. Tatt i betraktning at AMS sannsynligvis vil føre til at sluttbrukere reduserer sitt forbruk, noe som igjen vil innebære reduserte inntekter for nettselskapene, kan det tyde på at AMS rent bedriftsøkonomisk vil være negativt for nettselskapene. NVE fastsetter hvert år en inntektsramme for ulike nettselskaper, som regulerer hvor mye nettselskapene har lov til å tjene på overføring av kraft til kunder. NVE forklarer på sine nettsider, [9], hvordan de setter de årlige inntektsrammene:

NVE fastsetter inntektsrammene slik at inntekten over tid skal dekke kostnadene ved drift og avskrivning av nettet, samt gi en rimelig avkastning på investert kapital gitt effektiv drift, utnyttelse og utvikling av nettet.

Det er derfor naturlig å tro at nettselskapene vil få økte inntektsrammer i perioden for en storskala AMS-utrulling, og kostnadene vil da bli skjøvet i større eller mindre grad over på sluttbrukerne.

Sluttbrukere vil, fra et økonomisk perspektiv, ha nytte av AMS hvis det gjør at de klarer å redusere strømforbruket sitt. La oss se på et fiktivt og forenklet case på hvor store summer det er snakk om.

Snittprisen for strøm i Norge i 2010 var 46,8 øre/ kWh, [8], eksklusive nettleie og avgifter. [15] opererer med en snittpris, inkludert nettleie og avgifter, på ca. 1 krone pr. kWh i 2010. Det gjennomsnittlige energiforbruket i Norge for 2010 varierer endel utifra geografi. [2] oppgir, med bakgrunn i tall fra SSB, at det høyeste gjennomsnittsforbruket for en bolig i Norge er i fylkene Hedmark og Oppland, på totalt 23055 kWh. Det laveste gjennomsnittsforbruk finner vi i Oslo, 15054 kWh, pga. Oslos store andel av leiligheter som ofte bruker alternative og mer effektive oppvarmingskilder. For en gjennomsnittsbolig i Oppland eller Hedmark, vil da de årlige strømutfgiftene, inkludert nettleie og avgifter, beløpe seg til 23055,- kroner, eller 5763,- kroner per kvartal. For en gjennomsnittsbolig i Oslo vil de samme beløpene være hhv. 15054,- og 3764,- kroner. Hvis vi antar at majoriteten av boligene i Norge ikke vil installere et display, og en middels god datakvalitet, kan vi anta at en storskala AMS-utrulling vil medføre at sluttbrukere reduserer sitt strømforbruk med om lag 5 %, se seksjon 2.5.2. 5 % lavere

strømforbruk for en gjennomsnittlig bolig i Oppland eller Hedmark i 2010 ville betydd en kostnadsbesparelse på 1153,- kroner per år, eller ca. 288,- kroner per kvartal. For Oslo ville de tilsvarende beløpene vært på hhv. ca. 753,- og 188,- kroner. Kostnadsbesparelser forbundet med en skifting av forbruk fra høylast- til lavlastperioder er vanskeligere å tallfeste. Administrerende direktør i Fjordkraft, Sverre Gjessing, sier i et intervju i Teknisk Ukeblad, [32], at ved å flytte bruk av vaskemaskin, oppvaskmaskin og tørketrommel til den billigste timen i et normalår vil gi en gjennomsnittsfamilie årlige besparelser på om lag 45,- kroner. I et slikt regnestykke må man også ta høyde for at hvis mange endrer sitt forbruksmønster, vil forskjellen i pris mellom høy- og lavlastperioder bli redusert, og en eventuell kostnadsbesparelse bli enda mindre.

I mitt forenklete eksempel er det relativt beskjedne summer sluttbrukere kan spare. Jeg har brukt relativt forsiktige estimater mtp. sluttbrukernes reduksjon av forbruket. Rapporten "The potential of smart meter enabled programs to increase energy and systems efficiency: a mass pilot comparison", [39], viser til pilotprogrammer som har resultert i opptil 10 % reduksjon av energiforbruket, samt en 13 % reduksjon av den gjennomsnittlige strømregningen. Enkelte bransjeaktører opererer også med en energibesparelse på opptil 30 % gjennom økt måling og bevisstgjøring av sluttbrukerne, så AMS har potensiale til å bli kostnadsbesparende også for sluttbrukere.

På den andre siden er det stilt store forventninger til AMS fra et samfunnsnyttig perspektiv. Norge, og resten av verden forøvrig, er under press for å få strømforbruket ned mot et mer bærekraftig nivå. EU sine 20-20-20-mål sier bl.a. at man innen 2020 skal redusere energiforbruket med 20 % sammenlignet med forventet strømforbruk, dette skal skje ved hjelp av økt energieffektivitet. AMS spiller i denne sammenheng en stor rolle, i den forstand at det er et verktøy som gjør det enklere for sluttbrukere å effektivisere og redusere sitt forbruk. AMS i seg selv er også bare begynnelsen på en større endring i kraftnettet. AMS er ment som å være endepunktene i et fremtidig smartgrid, et begrep som innebærer en modernisering og effektivisering av dagens kraftnett, [38]. [44] hevder at en smartgrid-utbygging vil legge til rette for en reduksjon av CO₂-utslipp i USA på 12 til 18 %. Jeg viser til [38] for en videre presentasjon av smartgrid.

Som sagt, så er AMS lovende mtp. en fremtidig reduksjon av strømforbruk, i tillegg til en effektivisering av strømforbruket. Men, det er viktig å huske at det er hver enkelt

sluttbruker som tar beslutninger om sitt forbruk. Det er derfor viktig at sluttbrukere har sterke incentiver for både å redusere forbruket sitt generelt, men også å tilpasse forbruket sitt etter den gjeldene kraftsituasjonen. Norge har påtatt seg internasjonale forpliktelser angående redusert forbruk og økt energieffektivitet, og NVE, som den regulatoriske myndighet, spiller en viktig rolle her, mtp. at det er NVE som stiller krav som nettselskapene må oppfylle ved den fremtidige storskala AMS-utrulling.

2.6 Oppsummering

Dette kapitlet var ment som en innføring i AMS, bakgrunnsinformasjon, kraftmarkedet i Norge, krav, arkitektur og en enkel presentasjon av nytteverdi og kostnader forbundet med AMS. For en mer grundig innføring i AMS, og for vinklinger som denne oppgaven ikke har gått inn på, viser jeg til de tidligere masteroppgavene [38], [43] og [37]. Resten av oppgaven vil som nevnt fokusere på fremtidens IKT-løsninger i forbindelse med AMS i kraftbransjen. Det antas at AMS er på plass hos sluttbrukerne, og jeg vil fokusere på deling av målerdata mellom en Webtjener, kraftselskaper og tredjeparter. Sentrale problemstillinger vil bl.a. være, hvor Webtjeneren skal ligge og hvem som skal drifte den.

3

INFORMASJONS- OG KOMMUNIKASJONSTEKNOLOGI I KRAFTBRANSJEN

Forbundet med en storskala innføring av AMS i Norge er det store utviklingsbehov med tanke på IKT-løsninger i Norge. Systemene må utvides for å tåle en drastisk økning i trafikk- og datamengde, i tillegg til at systemene må kunne dele målerdata på en hensiktsmessig måte. Dette kapittelet skal gi en innføringen i dagens IT-systemer, samt presentere ulike muligheter for en fremtidig struktur.

3.1 Dagens IT-løsning

Det er allerede en stor grad av IT-systemer i bruk i kraftbransjen i dag. Ulike nettselskaper har vanligvis implementert egne kundeinformasjonssystemer (KISer), i tillegg til MVDBer for manuelt avleste målerdata, [38]. De ulike MVDBene, KISene og kommunikasjon mellom de er vanligvis basert på skreddersydde og proprietære løsninger for de ulike nettselskapene, [33].

Kommunikasjon og datautveksling mellom nettselskap og kraftprodusenter er derimot gjort via en felles plattform basert på åpne og velkjente standarder: Ediel og Nubix.

3.1.1 Ediel

Ediel er en standard som beskriver en EDI-basert meldingsutveksling over SMTP i kraftbransjen, [3]. EDI, Electronic Data Interchange", er en standard for elektronisk

utveksling av informasjon, f.eks. forretningsdokumenter, innberetninger og statistikk, [7]. Ediel spesifiserer format, typer og funksjoner til meldinger som utveksles, i tillegg til kommunikasjons- og sikkerhetsmekanismer, [3]. Ediel-meldingene er basert på EDIFACT-standarden, en internasjonalt akseptert standard for meldingsformat, men meldingene er skreddersydd for norske formål. Sammenlignet med XML, har Ediel fordelen at datamengden er redusert, [38] hevder at Ediel trenger mellom en tredjedel og halvparten av datamengden sammenlignet med en tilsvarende XML-melding. I tillegg har Ediel klare regler for mottaker- og avsender-ID, samt hvilken forretningsprosess meldingen dreier seg om. De to mest brukte Ediel-meldingene i dag er PRODAT OG MSCONS. PRODAT brukes som regel i forbindelse med leverandørbytteprosessen, og inneholder data som sjeldent blir endret, [27]. PRODAT-meldingen kan inneholde relativt mange felter, bl.a. meldings-ID, målepunkt-ID og kunde-ID. MSCONS, på den andre siden, er en meldingstype brukt for å rapportere målte verdier (transaksjonsdata), f.eks. produksjons- og forbruksdata, mellom aktører (nettselskap, kraftprodusenter og Statnett) i kraftbransjen, [27]. Dette er data som endres oftere enn data i PRODAT-meldingen, og meldingen er derfor mer hyppig brukt. MSCONS-meldingen brukes til flere konkrete forretningsprosesser, periodisk avlesning, leverandørbytte og flytting for å nevne noen, og til hver av prosessene kan meldingen inneholde flere felter. For eksempel, MSCONS-meldingen som inneholder informasjon om periodisk avlesning må inneholde målepunkt-ID, starttidspunkt for avregningsperiode, start-målerstand, sluttidspunkt for avregningsperiode og slutt-målerstand for å nevne noen.

Ediel er under stadig utvikling, og det er forventet¹ at Ediel fases ut mot en XML-basert standard i et fremtidig nordisk sluttbrukermarked.

3.1.2 Nubix

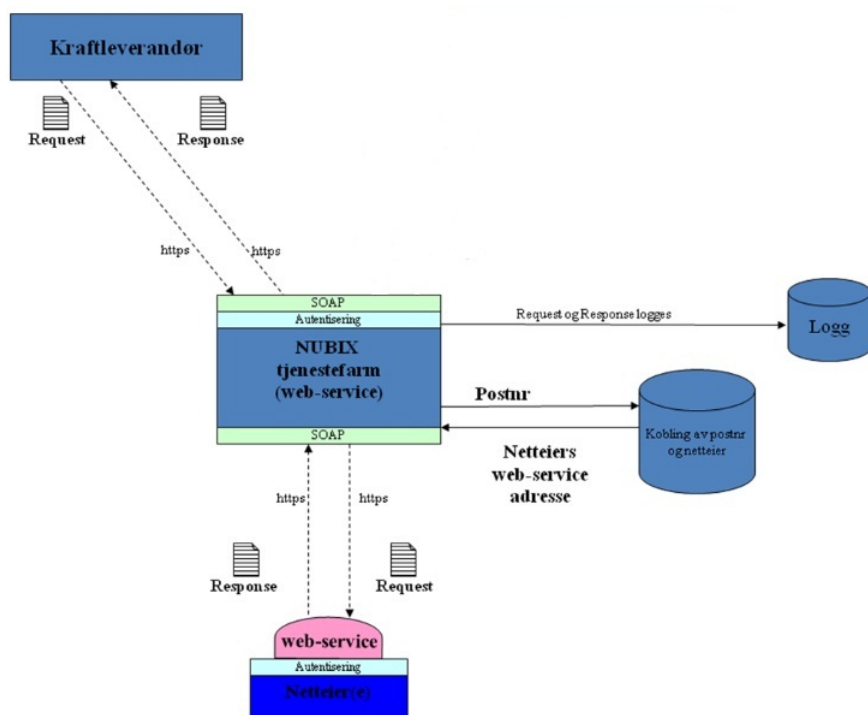
Hvert målepunkt i Norge har sin unike ID, målepunkt-ID. Målepunkt-ID er en 8-siffrers kode som unikt spesifiserer et elektrisk anlegg, [10]. IDen er standardisert på nasjonalt basis, og brukes bl.a. ved bytte av kraftleverandør. Som nevnt i seksjon 3.1.1, så er denne IDen et obligatorisk felt i de viktigste meldingene i Ediel-standarden. Denne IDen er en ID som sluttbrukere generelt har liten kjennskap til, og det er

¹Etter epostutveksling med Christian Le i SSE

derfor et behov for et automatisert system som utfører spørringer mot nettselskapenes KISer for å få funnet og bekreftet IDen. Nubix, "Nordic Utilities Business Information Exchange", er et slikt system, [13]. Nubix er en Webtjeneste, oversatt fra det engelske uttrykket "Web Services", driftet av Statnett, som tillater kraftprodusenter å hente ut informasjon fra nettselskapenes systemer. Nubix i seg selv opptrer som en meldingsformidler, det vil si at systemet tar i mot forespørsler fra kraftprodusenter, forespørslene må inneholde postnummer og minst to andre søkekriterier² for å bli akseptert av Nubix-tjeneren. Nubix-tjeneren gjør nå et oppslag i sin database basert på mottatt postnummer fra kraftleverandøren, finner nettselskap, og videresender forespørselen dit³, [13]. Forespørslene betjenes så videre av nettselskapenes systemer ved å gjøre oppslag i egen database, for så å returnere resultatet til kraftprodusenten via Nubix-tjeneren. For illustrasjon, se figur 3.1, fra [42].

²Navn, adresse, fødselsdato, boenhetsnummer, poststed eller målnummer for privatkunder

³Hvis det er flere nettselskap med samme postnummer, sendes forespørselen til alle



Figur 3.1: Nubix' arkitektur

Nubix bruker som nevnt Webtjeneste-teknologi, det vil si at meldingsutvekslinger til og fra tjeneren er basert på XML. På Edielportalen, [13], har man offentliggjort WSDL-dokumenter og XML skjemadefinisjoner (xsd-filer). WSDL-dokumenter beskriver hvordan man skal implementere klient- eller tjenersiden av en Webtjeneste, se seksjon 4.1.1, skjemadefinisjoner beskriver strukturen og hvilke felt som skal være med i en XML-melding. Basert på disse dokumentene er nettselskapene pliktet til å implementere en Webtjeneste som kan besvare disse forespørslene, men hvordan nettselskapene implementerer logikken for søk og oppslag i sine databaser er valgfritt, [33].

Nubix foretar også en autentiseringskontroll, man må være en registrert og godkjent bruker i Ediel-portalen, [13], for å kunne benytte Webtjenesten. I tillegg benytter Nubix HTTPS med SSL-kryptering for all kommunikasjon, som vanskeliggjør uautorisert tilgang til meldingene under transmisjon, [33].

3.2 Rapport: Felles IKT-løsninger i det norske kraftmarkedet

Som nevnt tidligere vil AMS medføre store investeringer og endringer i kraftbransjens IT-systemer. Det er allerede en stor grad av automatisering på flere av forretningsprosessene som involverer nettselskap, kraftleverandører og Statnett, forklart i 3.1, men forretningsprosessene mellom sluttbruker og nettselskapene er fortsatt preget av manuelle og proprietære løsninger. Konsulenthusene Thema Consulting Group og Devoteam daVinci har på oppdrag fra NVE skrevet en rapport om muligheter for felles IKT-løsninger i det norske kraftmarkedet, [33]. Rapporten presenterer og analyserer tre alternativer for et fremtidig IKT-system: distribuert database, felles måleverdidatabase og felles behandlingsfunksjoner. Jeg vil nå presentere de ulike alternativene, i tillegg til noen reaksjoner mottatt fra kraftindustrien.

3.2.1 Distribuert database

Dette alternativet er basert på en videreutvikling av dagens løsning. De ulike nettselskapene videreutvikler sine KISer og MVDBer til å ta i mot og prosessere større datamengder, meldingsutveksling mellom nettselskap og kraftleverandører og Statnett

vil fortsatt baseres på Ediel og Nubix. Nettselskapene vil da måtte rapportere relevant data med jevne mellomrom til de kraftprodusentene som deres kunder har avtale med. Rapporten påpeker at selv om dagens Ediel- og Nubix-løsning kan brukes som basis, så bør alle nye meldinger baseres på XML og Webtjenester.

Dette alternativet vil innebære en dobbeltlagring av målerdata, det vil si at målerdata må lagres både hos nettselskapene og hos kraftprodusentene. I tillegg til at det totalt sett vil kreve større ressurser å lagre data to steder, trekker rapporten også frem at det vil være en fare for inkonsistens i dataene. Dessuten må all funksjonalitet implementeres hos hvert enkelt nettselskap, og systemet vil være lite fleksibelt og effektivt med tanke på tilpasninger og endringer i funksjonaliteten.

For nettselskapene vil denne løsningen bety at de er selv ansvarlige for å etablere systemer for å gi kundene en mer detaljert oversikt over deres forbruk enn i dag, i tillegg til at nettselskapene selv må administrere tilgang til målerdata for tredjeparter.

Tredjeparter, på den andre siden, er prisgitt de ulike nettselskapers forskjellige løsninger for deling av målerdata, i motsetning til å kun ha en sentral enhet å forholde seg til. Rapporten hevder også at det er grunn til å tro at dette vil øke kompleksiteten i systemet, og at dette vil kunne gå utover kvaliteten til eventuelle tredjepartstjenester som vil bli tilbydd.

Fra et regulatorisk perspektiv, vil en slik løsning innebære at det er vanskelig å stille, samt følge opp, krav til tjeneste- og datakvalitet hos de ulike nettselskapene. Tilgang til data, samt kvaliteten på dataene, vil avhenge av implementeringen valgt hos hvert nettselskap, de samme rutinene rundt datavasking, leverandørskifte og lignende.

3.2.2 Felles måleverdidatabase

Dette alternativet innebærer etableringen av en felles nasjonal MVDB i tillegg til noe funksjonalitet for innhenting og kvalitetssikring av målerdata, samt beregne forbruk for hver enkelt måler og for hvert nettselskap totalt. Overføring av målerdata mellom lokalt mellomlagrede måleverdier hos nettselskapene og den sentrale databasen kan skje på ulike måter. Enten ved en direkte kopiering av data mellom databasene, initiert av nettselskapene på faste tider ("push") eller initiert av det sentrale systemet ("pull").

Brukere av et slikt sentralt system vil være kraftselskaper, systemansvarlig (Statnett),

nettselskap og autoriserte tredjeparter. Nettselskaper kan selv velge om de vil bruke den sentrale databasen som sitt hovedlager, eller om de vil basere seg på egne løsninger. Det sentrale systemet vil også implementere autentiserings- og sikkerhetsmekanismer. Systemet må sørge for konfidensialiteten til dataene, det vil si at dataene kun skal være tilgjengelig for de brukerne som har nødvendig autorisasjon. I tillegg må systemet sørge for dataintegritet, det vil si at dataene er fullstendige, nøyaktige og gyldige. Systemet må også sørge for at systemet er stabilt og at dataene er tilgjengelige ved behov. Det siste aspektet er spesielt viktig i forhold til det full-distribuerte alternativet. En feil på fellessystemet vil få konsekvenser for et betydelig større antall brukere enn en feil på et av de distribuerte systemene, i tillegg til at det vil generere negativ oppmerksomhet.

Sammenlignet med det distribuerte alternativet, vil en sentralisering av databasen innebære at dobbeltlagring kan unngås. Selv om det er grunn til å tro at noen nettselskaper vil velge å ha sin egen database, vil det være fullt mulig å kun basere seg på fellessystemet. I tillegg vil fellessystemet ha ansvaret for administrering og utveksling av data, slik at nettselskapenes IT-systemer kan forenkles. Dette vil også være en fordel for tredjeparter, som kun har en aktør å forholde seg til. Rapporten fremhever også at et sentralt system vil være en fordel ved en internasjonalisering av sluttbrukermarkedet, fordi data- og formatkonvertering vil kun måtte skje et sted for å tilpasse seg eventuelle ulikheter mellom standardene som brukes.

En slik løsning vil ha fordelene av at det er kun ett sted for alle data. Nettselskap, kraftprodusenter, tredjeparter og sluttbrukere vil kun ha et system å forholde seg til, noe som gir en udiskriminerende løsning. Et ankepunkt mot en slik løsning er at den vil medføre en relativt stor implementeringskostnad, men rapporten hevder at det på sikt vil bli et rimeligere alternativ enn at hvert nettselskap skal implementere egne løsninger for å tilpasse seg til AMS' økte krav til IT-funksjonalitet.

3.2.3 Felles behandlingsfunksjoner

Dette alternativet baserer seg på en videreutvikling av alternativet nevnt over. Der det forrige alternativet baserte seg på en felles MVDB, i tillegg til funksjonalitet for innhenting og kvalitetssikring av målerdata fra nettselskapene og forbruksberegning, så innebærer dette systemet at en større del av funksjonaliteten sentraliseres.

For det første vil det sentrale systemet samle inn målerdata direkte fra smartmålerne, så behovet for mellomagring hos nettselskapene forsvinner. Dataene må i tillegg kvalitetssikres, og informasjon om nettverksstatus bør deles med den ansvarlige for måleverdikjeden slik at vedlikehold og nødvendige utbedringer kan foretas. Dette systemet skal også kunne foreta fakturering på vegne av kraft- og nettselskapene. Systemet kan kombinere forbruksdata mottatt fra smartmålerne med strømpriser fra kraftselskapene og nettleie fra nettselskapene og produsere komplette fakturaer. Leverandørbytteprosessen vil også styres i det fulle av fellessystemet, men rapporten vektlegger at nettselskapene vil fortsette å vedlikeholde sine kunderelasjoner gjennom egne kanaler. Rapporten går ikke konkret inn på hvordan slik funksjonalitet skal implementeres, men foreslår rent generelt bruk av Webtjeneste-teknologi der det er mulig.

Sammenlignet med det forrige alternativet, innebærer som sagt en slik løsning en mye større grad av sentralisering av IT-systemene i kraftbransjen. Dette gir bl.a. fordelene at de samme dataene og tjenestene er tilgjengelige for alle, og at NVE har bedre muligheter til å stille, samt følge opp, konkrete krav til tjeneste- og datakvalitet. En slik løsning vil også være kostnadsbesparende for nettselskapene, hovedsaklig på grunn av at deres IT-systemer kan forenkles og at forretningsprosesser som fakturering og datadeling outsources til det sentrale systemet. I tillegg er det naturlig å tro at et slikt sentralt system vil redusere inngangsbarrierene for nye aktører i kraftbransjen, noe som kan bidra til å skape et mer innovativt og konkurransepreget miljø. Av ulemper legger rapporten frem at implementeringen av et slikt system vil være ressurskrevende, både mtp. tid og konkrete investeringer. I tillegg vil et slikt system kreve en sentral styring, og det er per dags dato ingen klar aktør i bransjen som innehar nødvendig kompetanse for dette.

Det er dette alternativet rapporten har som sin anbefalte løsning. Rapporten legger vekt på at at en slik sentral løsning vil tillate at alle berettigede aktører kan få enkel tilgang til den relevante dataen, og at man unngår et sett med proprietære løsninger hos de ulike nettselskapene. Fellesløsningen bør bygges modulært, slik at ny funksjonalitet enkelt kan legges på ved behov. Ny funksjonalitet bør utvikles med Webtjeneste-teknologi, men meldingsformatet bør så langt det er mulig baseres på allerede eksisterende Ediel-meldinger.

3.2.4 Kommentarer fra bransjen

Rapporten ble sendt ut på høring av NVE, og det har kommet inn totalt 22 svar fra ulike aktører i bransjen; Kraftselskaper, nettselskaper, kombinerte selskaper og ulike interesseorganisasjoner. De ulike aktørene har alle ulike motiver og fokusområder, og svarene de har gitt gjenspeiler til en viss grad dette, men på generelt basis har svarene vært kritiske til argumentasjonen og de forespeilede løsningene. Bransjen er generelt kritiske til en uklar rollefordeling i en felles IKT-løsning, i tillegg til kostnadsestimatene, etableringen av en sentral monopolistisk organisasjon, tidsaspektet for implementeringen og nedbygging av lokal kompetanse.

Angående manglende og uklar definisjon av rollefordelingen og ansvarsforholdet i en tenkt felles løsning, alternativ tre, påpeker Vokks Nett at det er nettselskapene som er direkte knyttet til kunden, via måleren, og bør derfor ha ansvaret for, og direkte adgang til, målerdata. Lyse Elnett går litt lengre i kritikken, og hevder at en sentralisering av behandlingsfunksjoner vil fjerne nettselskapenes rolle i måleverdikjeden. Det er nettselskapene som er ansvarlige for måling og avregning av sluttkundene, og de bør også ha det overordnede ansvaret for målerdata. Enkelte belyser også det problematiske ved at innføringen av en felles løsning vil bidra til å flytte kompetanse fra endepunktene av måleverdikjeden til det sentrale systemet. Hvem vil være ansvarlig for å håndtere brukerstøtte mot sluttbrukerne i tilfelle feil? Det er nettselskapene som er sluttbrukernes kontaktpunkt i dag, og det vil være problematisk for nettselskapene å gi brukerstøtte på et system de ikke har full råderett over.

Kostnadsestimatene som gis i rapporten settes det også spørsmålsteget ved. Rapporten anslår at etableringen av en felles måleverdidatabase, alternativ to, vil koste om lag 50 millioner. I Danmark er man i ferd med å etablere en tilsvarende løsning, DataHub-prosjektet [4]. DataHub var i begynnelsen estimert til å koste 30-40 millioner, men den reelle kontraktsfestede summen endte på 114 millioner i etableringskostnader, i tillegg til årlige driftskostnader på 20 millioner, [1], en sum Skagerak Nett hevder er mer realistisk for etableringen av en sentral måleverdidatabase. Skagerak Nett savner også en tydeligere forretningsmodell, samt en redegjøring for hvem som skal ta kostnadene for etablering og drifting av en slik fellesløsning. Også Lyse Elnett er kritiske til kostnadsaspektet, og hevder at siden nettselskapene uansett må videreutvikle sine egne KIS for å tilpasse seg

AMS, så vil utvikling av en slik fellesløsning medføre en betydelig ekstrakostnad.

Flere av selskapene er også kritiske til etableringen av en stor og tungrodd monopolorganisasjon som skal forvalte målerdata. NTE påpeker at en av fordelene med å basere seg på åpne standarder som TCP/IP, er at det fasiliterer stor grad av leverandøruavhengighet og konkurranse. Ved etablering av en monopolorganisasjon, vil denne uavhengigheten og konkurransesituasjonen bli kraftig redusert, og det vil være få incentiver blant aktørene for å ta i bruk innovative løsninger. Det påpekes også at ved en sentralisering av behandlingsfunksjonene, vil det være vanskeligere for aktører å differensiere seg på servicetilbud, noe som vil kunne resultere i en redusert konkurransesituasjon og mindre innovasjon i bransjen.

Tidsaspektet trekkes også frem som et argument mot innføring av fellesløsninger. NVE har satt som mål at AMS skal være på plass innen 1. januar 2017, og bransjen frykter at en forhastet beslutning på IT-siden vil medføre store feilinvesteringer for bransjen. Flere aktører tar til orde for at et fellessystem må kunne implementeres skrittvis, dvs. at i første omgang vil kun et minimum av funksjonalitet implementeres, og så vil systemet eventuelt kunne bygges ut ettersom behovene blir tydeligere. Det er urealistisk å tro at man før en AMS-utrulling vil klare å forutse alle konsekvensene det vil medføre, så å allerede nå låse seg fast til en IKT-løsning vil ikke være hensiktsmessig.

Enkelte aktører peker også på at hovedutfordringene på IKT-siden i dag er primært manglende målerstand ved leverandørbytte, samt generell tilgang på kvalitetssikrede måleverdier for aktørene i markedet. Aktørene argumenterer for at dette er utfordringer som i stor grad vil bli løst ved innføringen av AMS, og at innføring av felles IKT-løsninger alene vil gjøre lite for å forbedre situasjonen. Nå skal det sies at innføringen av AMS alene vil gjøre lite med tanke på tilgjengeliggjøring av målerdata for andre aktører enn nettselskapene alene, så hvis det blir et formelt krav om tredjepartstilgang til målerdata, må dette uansett implementeres. Problemet med manglende målerstand ved leverandørbytte vil derimot bli løst ved AMS, da målerstanden ved endringsøyeblikket simpelthen vil sendes automatisk til nettselskapet, nettselskapet kan så dele målerstanden med kraftleverandøren ved hjelp av allerede eksisterende løsninger.

Rapporten bruker skalerbarhet og oppdaterbarhet som et argument for en sentralisering av IT-systemene. Den påpeker at ved en distribuert løsning som i dag, må eventuelle

endringer gjennomføres i 130 forskjellige implementeringer. Flere aktører påpeker at dette ikke stemmer, da det kun finnes om lag fem forskjellige KIS- og MVDB-løsninger i det norske kraftmarkedet i dag. Eventuelle endringer i krav og funksjonalitet må da implementeres i de fem systemene, for så å oppdateres hos de ulike aktørene. Når det er sagt, vil en sentral løsning absolutt ha en fordel når det gjelder systemoppdateringer og endringer, men kanskje ikke i like stor grad som forespeilet i konsulentrapporten.

Responsen fra bransjen har som nevnt vært jevnt over negativ, spesielt hevder aktørene at rapporten ikke gir en god nok kvalitativ vurdering av dagens situasjon. Bransjen stiller seg derimot positiv til tiltak som kan effektivisere og optimalisere driften, men ønsker å ha en aktiv rolle i prosessen med å spesifisere nye løsninger. Selv om NVE har relativt frie tøyler til å stille krav til arkitektur og funksjonalitet, er det min mening at det er en fordel om bransjen kan komme frem til en løsning som de fleste er fornøyd med. Innføringen av AMS vil medføre store kostnader og vil være under mediernes søkelys, så NVE gjør klokt i å lytte til aktører med god kjennskap til kraftbransjens hverdag.

3.3 Oppsummering

Dette kapitlet ga en innføring i både dagens og mulige fremtidige løsninger på IT-siden i kraftbransjen. Det er allerede systemer på plass for datautveksling i kraftbransjen, Ediel og Nubix, men investeringer må til for å tilpasse seg en AMS-hverdag med en betydelig økning av datamengdene. Konsulentselskapene Thema og Devoteam daVinci har, på oppdrag fra NVE, levert en rapport som skisserer tre tenkte scenarier for fremtidens IKT-systemer i kraftbransjen, men responsen fra kraftbransjen har for det meste vært negativ. Jeg vil i neste kapittel ta utgangspunkt i konsulentrapporten, samt reaksjonene fra bransjen, og foreslå en arkitektur for en IT-løsning i en AMS-hverdag.

4

VALG AV IT-LØSNING

Som nevnt i forrige kapittel, har det kommet forslag til nye IKT-løsninger i kraftbransjen ved innføring av AMS. Systemene som er foreslått baserer seg i større grad på sentralisering av kjernefunksjoner enn det som er tilfellet i dag. Selv om bransjen i stor grad er positiv til initiativ som sikrer en mer effektiv forvaltning av måleverdier, er det stor uenighet om hvordan en slik løsning skal se ut, både fra et rent teknisk og et mer overordnet perspektiv.

Dette kapittelet tar sikte på å presentere Webtjenester, et rammeverk for kommunikasjon og meldingsutveksling mellom to applikasjoner. I tillegg vil jeg basert på konsulentrapporten og tilhørende kommentarer fra bransjen komme frem til en hensiktsmessig systemarkitektur for et system for deling av målerdata.

4.1 Webtjenester

Webtjenester er et omfattende rammeverk for maskin-til-maskin (MTM)-kommunikasjon basert på åpne standarder. Etersom Webtjenester er ment for MTM-kommunikasjon, er det designet med hensyn på å bli forstått av applikasjoner, og ikke mennesker [45]. Webtjenester opererer på toppen av eksisterende Web-standarder, bruker XML som dataformat og kan bli aksessert via typiske Internettprotokoller (HTTP, SMTP, FTP, etc.). Webtjenester er et mye beskrevet fagfelt, og flere formelle definisjoner er skrevet. Denne definisjonen, fra [28], mener jeg gir en lettforståelig og formell beskrivelse av Webtjenester:

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner predescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

4.1.1 WSDL

WSDL, "Web Services Description Language", er et XML-basert språk for å beskrive Webtjenester. WSDL er skapt for å gi nok teknisk informasjon om Webtjenesten til at man kan benytte dokumentet som basis for faktisk implementasjon. Informasjon som må beskrives er bl.a. URL, port, metodenavn, argumenter, datatyper, etc., [35]. WSDL bruker flere abstrakte elementer for å definere en Webtjeneste, bl.a. "message", "operation", "binding" og "service".

"Message"-elementet beskriver meldingene som sendes til og fra en Webtjeneste, samt deres attributter og returverdier, disse lagres i subelementet "part".

"Operation" beskriver et metodekall, med dets tilhørende inngående og utgående meldinger. Meldingene er koblet sammen i par, hhv. forespørsel og svar. Alle operasjoner som er tilbudt av en Webtjeneste samles sammen i WSDL-elementet "PortType", [45].

"Binding" beskriver hvordan, rent protokollmessig, en Webtjeneste kan bli aksessert, f.eks SOAP over HTTP eller SOAP over SMTP. Den samme porttypen kan ha opptil flere bindinger innenfor det samme WSDL-dokumentet, dvs. at en Webtjeneste kan i teorien være tilgjengelig over forskjellige protokoller, [35].

"Service"-elementet indikerer hvordan man kan aksessere Webtjenesten. Elementet inneholder opptil flere "port"-subelementer. Disse subelementene beskriver valgt "binding" og adressen i URL-notasjon for tjenesten.

En av de største fordelene med WSDL, er graden av standardisering. Tilbyder av en Webtjeneste kan publisere sitt WSDL-dokument, og brukere som ønsker å aksessere Webtjenesten kan benytte ulike verktøy for å implementere kode basert på gitt WSDL-dokument. I WSDL-dokumentet kan man også spesifisere regler for aksessering, f.eks at all kommunikasjon må gå over HTTPS eller at hodet i SOAP-meldingen må inneholde brukernavn- og passordfelder, [35]. Følgende er et eksempel på et relativt enkelt WSDL-

dokument, kopiert fra [26]. Som man kan se, så inneholder dokumentet også et "types"-element, dette inneholder XML skjemadefinisjoner, som brukes til å definere de ulike input- og output-meldingene med informasjon om hvilke datatyper og attributter som brukes av de ulike meldingene.

```

<wsdl:types>
  <xsd:schema targetNamespace="http://namespaces.snowboard-info.com"
    xmlns:xsd="http://www.w3.org/1999/XMLSchema">
    <xsd:element name="GetEndorsingBoarder">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="manufacturer" type="string"/>
          <xsd:element name="model" type="string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="GetEndorsingBoarderResponse">
      <xsd:complexType>
        <xsd:all>
          <xsd:element name="endorsingBoarder" type="string"/>
        </xsd:all>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="GetEndorsingBoarderFault">
      <xsd:complexType>
        <xsd:all>
          <xsd:element name="errorMessage" type="string"/>
        </xsd:all>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="GetEndorsingBoarderRequest">
  <wsdl:part name="body" element="esxsd:GetEndorsingBoarder"/>
</wsdl:message>
<wsdl:message name="GetEndorsingBoarderResponse">
  <wsdl:part name="body" element="esxsd:GetEndorsingBoarderResponse"/>
</wsdl:message>
<wsdl:portType name="GetEndorsingBoarderPortType">
  <wsdl:operation name="GetEndorsingBoarder">
    <wsdl:input message="es:GetEndorsingBoarderRequest"/>
    <wsdl:output message="es:GetEndorsingBoarderResponse"/>
    <wsdl:fault message="es:GetEndorsingBoarderFault"/>
  </wsdl:operation>
</wsdl:portType>

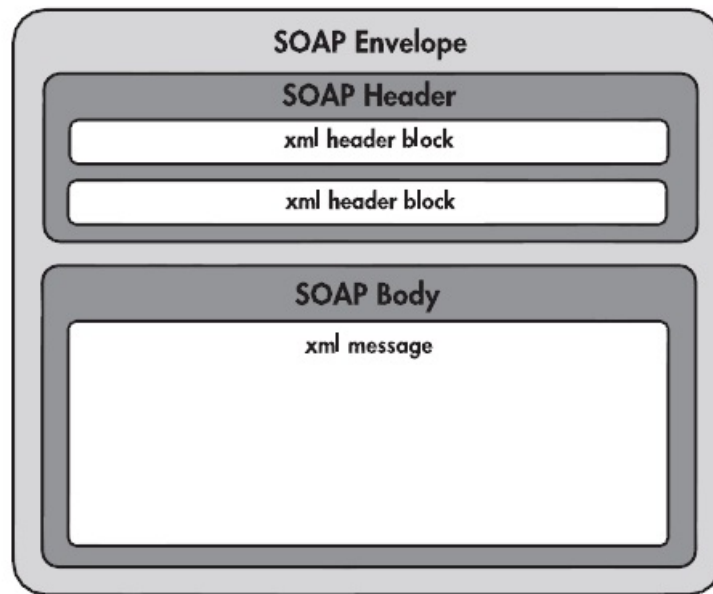
```

```
<wsdl:binding name="EndorsementSearchSoapBinding"
              type="es:GetEndorsingBoarderPortType">
  <soap:binding style="document"
                transport="http://schemas.xmlsoap.org/soap/
                http"/>
  <wsdl:operation name="GetEndorsingBoarder">
    <soap:operation soapAction="http://www.snowboard-info.com/
    EndorsementSearch"/>
    <wsdl:input>
      <soap:body use="literal"
                namespace="http://schemas.snowboard-info.com/
                EndorsementSearch.xsd"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal"
                namespace="http://schemas.snowboard-info.com/
                EndorsementSearch.xsd"/>
    </wsdl:output>
    <wsdl:fault>
      <soap:body use="literal"
                namespace="http://schemas.snowboard-info.com/
                EndorsementSearch.xsd"/>
    </wsdl:fault>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="EndorsementSearchService">
  <wsdl:port name="GetEndorsingBoarderPort"
            binding="es:EndorsementSearchSoapBinding">
    <soap:address location="http://www.snowboard-info.com/
    EndorsementSearch"/>
  </wsdl:port>
</wsdl:service>
```

4.1.2 SOAP

SOAP er en XML-basert modell for meldingsutveksling som brukes i Webtjenester. SOAP-meldinger blir sendt som nyttelast i valgt applikasjonslagsprotokoll (HTTP, SMTP, FTP, etc.), og er ikke å regne som en separat protokoll.

En SOAP-melding er et standardisert XML-dokument, bestående av rotelement "Envelope", som inneholder to subelementer, "header" (hode) og "body"(kropp). Figur 4.1, fra [35], illustrerer SOAPs meldingsformat. Hodet er et valgfritt subelement, og inneholder applikasjonsspesifikk informasjon, [45], f.eks informasjon om valgt sikkerhetsløsning, som



Figur 4.1: SOAP "Envelope"

jeg vil gå mer inn på i 4.1.3.

Kroppen er obligatorisk og inneholder XML-meldingen ment for mottakeren av forespørselen, som bl.a. inkluderer tjenestenavn og eventuelle attributter, [35]. Under kommer eksempler på hhv. forespørsel- og svar-meldinger som implementerer Webtjenesten spesifisert i WSDL-dokumentet i seksjon 4.1.1. Meldingene er kopiert fra [26]

SOAP Request:

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
    encoding/">
  <SOAP-ENV:Body>
    <m:GetEndorsingBoarder xmlns:m="http://namespaces.snowboard-
      info.com">
      <manufacturer>K2</manufacturer>
      <model>Fatbob</model>
    </m:GetEndorsingBoarder>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Som man kan se over, inneholder denne SOAP-meldingen ikke et hode-element. Meldingen kaller operasjonen "GetEndorsingBoarder" med attributtene, som spesifisert i meldingen "GetEndorsingBoarderRequest", "manufacturerer" og "model".

SOAP Response:

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
    encoding/">
  <SOAP-ENV:Body>
    <m:GetEndorsingBoarderResponse xmlns:m="http://namespaces.
      snowboard-info.com">
      <endorsingBoarder>Chris Englesmann</endorsingBoarder>
    </m:GetEndorsingBoarderResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Her kan man se den korresponderende "GetEndorsingBoarderResponse"-meldingen, som inneholder attributtet "endorsingBoarder" av typen String.

4.1.3 Sikkerhet

Webtjenester baserer seg som sagt på en underliggende protokoll for transport, og er derfor i utgangspunktet sårbar mot de samme truslene som den underliggende protokollen. Webtjenestesikkerhet (WTS), "Web Service Security", er derfor blitt definert. WTS definerer sikkerhetsmekanismer på meldingsnivå som tar for seg et bredt utvalg av typiske sikkerhetsutfordringer; identifisering og autentisering, autorisering, integritet, ikke-fornektelse og konfidensialitet, [34].

"Web Service Policy (WS-Policy) Framework" (WTPR) er et rammeverk som beskriver hvordan Webtjenesten skal bruke WTS. WTPR kan brukes til å spesifisere egenskaper, krav og karakteristikk ved en Webtjeneste, [34]. Rent praktisk spesifiserer WTPR et nytt element i WSDL-definisjonen, "wsp:Policy", som inneholder subelementer som spesifiserer akkurat hvilke sikkerhetsmekanismer som skal tas i bruk, f.eks brukernavn- og passordautentisering eller digital signering. Under er et eksempel på et "policy"-element som spesifiserer at SOAP-meldinger skal ha med brukernavn og et passord i hodet. Passordet skal ikke sendes i klartekst, men som et meldingsekstrakt. Eksempelet er kopiert fra [18].

```
<wsp:Policy wsu:Id="UsernameToken" xmlns:wsu=
  "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
    wssecurity-utility-1.0.xsd"
```

```

    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsp:ExactlyOne>
    <wsp>All>
      <sp:SupportingTokens
        xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-
          securitypolicy/200702">
        <wsp:Policy>
          <sp:UsernameToken sp:IncludeToken="http://docs.oasis-
            open.org/ws-sx/ws-securitypolicy/200702/
              IncludeToken/AlwaysToRecipient">
            <wsp:Policy>
              <sp:HashPassword/>
            </wsp:Policy>
          </sp:UsernameToken>
        </wsp:Policy>
      </sp:SupportingTokens>
    </wsp>All>
  </wsp:ExactlyOne>
</wsp:Policy>

```

Dette er et eksempel på den enkleste formen for "policy". Merk URLen som slutter på "AlwaysToRecipient" i attributtene til "sp:UsernameToken"-elementet, dette spesifiserer at brukernavn og passord **må** være inkludert i meldinger sendt fra klient til tjener, men **ikke** i meldinger sendt fra tjener til klient.

Som nevnt tidligere, så kan SOAP-hodet brukes til bl.a. sikkerhetsinformasjon. Hodet vil da inneholde de feltene som er spesifisert i "policy"-definisjonen i WSDL-dokumentet til Webtjenesten. Et eksempel på et hode som samsvarer med "policy"-definisjonen ovenfor er gitt under, [18].

```

<soapenv:Header>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss
    /2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
    soapenv:mustUnderstand="1">
    <wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.
      org/wss/2004/01/oasis-200401-wss-wssecurity-utility
        -1.0.xsd" wsu:Id="UsernameToken-26752749">
      <wsse:Username>libuser</wsse:Username>
      <wsse:Password Type="http://docs.oasis-open.org/wss
        /2004/01/oasis-200401-wss-username-token-profile
          -1.0#PasswordDigest">/Wt/2yDdZwa8a5qd7U70hrp29/w=</
        wsse:Password>
      <wsse:Nonce>4ZQz5ytME/RXfChuKJ03iA=</wsse:Nonce>
      <wsu:Created>2009-03-17T11:20:57.467Z</wsu:Created>
    </wsse:UsernameToken>
  </wsse:Security>

```

```
</wsse:Security>  
</soapenv:Header>
```

Som man kan se inneholder SOAP-hodet feltene "Username", "Password", "Nonce" og "Created", som alle er subelementer av "UsernameToken", og som er hhv. et vilkårlig generert tall og et tidsstempel som indikerer når meldingen ble generert. Passordet blir sendt som et ekstrakt. I Axis2 blir ekstraktet generert av SHA1-funksjonen, med konkatasjonen av "Created", "Nonce" og passordet i klartekst som input, [18].

I eksempelet vist over er kryptering ikke spesifisert. Det vil si at hvis meldingen skal krypteres, må dette skje på et underliggende lag. Et eksempel er HTTPS som er en mye brukt protokoll for sikker meldingsutveksling over HTTP-protokollen. HTTPS, som er en kombinasjon av HTTP og TLS¹/SSL², baserer seg på asymmetrisk kryptering, også kalt offentlig nøkkelpkryptering, for å opprette en sikker kanal mellom to kommuniserende parter. En HTTPS-tilkobling starter med at klienten ber tjeneren om å opprette en sikker tilkobling. Tjeneren sender så sitt digitale sertifikat, som inneholder tjenerens navn, offentlige nøkkel og en signatur fra en sikkerhetsautoritet, [25]. En sikkerhetsautoritet er en betrodd tredjepart som utleverer digitale sertifikater mot betaling. Med betrodd menes det at en klient som mottar et digitalt sertifikat signert av en kjent sikkerhetsautoritet stoler på integriteten til sertifikatet. GlobalSign og VeriSign er eksempler på kjente sikkerhetsautoriteter. Klienten vil så generere et vilkårlig tall, kryptere det med tjenerens offentlige nøkkel og sende resultatet til tjeneren. Tjeneren vil deretter dekryptere meldingen med sin private nøkkel, og benytte det vilkårlige tallet generert av klienten som en symmetrisk nøkkel for fremtidig meldingsutveksling, [6].

HTTPS sørger for punkt-til-punkt-kryptering, som vil si at eventuelle mellomledd ikke vil kunne tyde informasjonen som sendes. Webtjenester er ofte implementert i en verdikjede av Webtjenester som kaller hverandre, i slike situasjoner er det ofte et behov for å tyde deler av informasjonen som sendes også hos mellomleddene. Et alternativ er at det opprettes en HTTPS-tilkobling mellom hver og en av mellomleddene, men når det blir opptil flere mellomledd vil det medføre mye overhead ifb. med opprettelse av HTTPS-tilkoblingene samt sertifikathåndtering. WTS spesifiserer også mekanismer som gjør at

¹"Transport Layer Security"

²"Secure Socket Layer"

krypteringen kan foregå på meldingsnivå, og man kan i WSDL-dokumentet definere nøyaktig hvilke deler av SOAP-meldingen som skal krypteres, samt detaljer om valgt algoritme, nøkkellengde etc. Fordelen med dette er at data som ikke anses som nødvendig å kryptere kan sendes i klartekst, og kan leses av eventuelle mellomledd og behandles etter ønsket funksjonalitet.

Presentasjonen av sikkerhetsutfordringer og -mekanismer i forbindelse med Webtjenester har vært på et innføringsnivå, for en mer detaljert og teknisk beskrivelse, se [34].

4.1.4 Webtjenester - Fordeler og oppsummering

Webtjenester baserer seg som nevnt på XML, noe som gjør at Webtjenester er plattform- og produsentuavhengig, [24]. Aktører som ønsker å implementere en Webtjenesteklient trenger kun WSDL-dokumentet som beskriver tjenesten, og selve implementeringen på klientsiden er opp til den enkelte. Bruken av WSDL er også positivt når det gjelder gjenbruk av kode. En Webtjeneste blir som regel aksessert av opptil flere klienter, som benytter Webtjenesten til ulike formål og forretningsprosesser. I stedet for at tilbyderer må skreddersy tjenesten til de ulike klientenes formål, kan klientene selv benytte de delene av Webtjenesten de har behov for. Ettersom et Webtjenestebasert system for deling av målerdata vil bli aksessert av ulike aktører med ulike motiver, er dette en tiltalende fordel. Webtjenester er også fleksible med tanke på bruk. Webtjenester kan bli aksessert av sluttbrukere via f.eks en nettleser, en applikasjon eller en annen Webtjeneste. Fra et AMS-perspektiv betyr dette at Webtjenester både kan brukes til å presentere målerdata til en sluttbruker på en nettside, eller at kraftprodusenter kan bruke Webtjenester til å hente målerdata som brukes som faktureringsgrunnlag i deres IKT-systemer.

Som man kan se, er Webtjenester forbundet med fordeler som er tiltalende for et system for deling av målerdata. Ettersom de er basert på åpne og velfungerende standarder, er det gode muligheter for utnyttelse av eksisterende infrastruktur og kompetanse. Det er derfor naturlig å tro at å benytte Webtjenester for et slikt system vil være mer kostnadseffektivt enn etableringen av et proprietært og skreddersydd system.

Presentasjonen av Webtjenester har vært relativt kort og ikke spesielt detaljert, jeg viser til kildene [35], [28] og [36] for mer utfyllende informasjon.

4.2 Valg av systemarkitektur

Som nevnt tidligere, ble forslagene som kom frem i konsulentrapporten sendt ut på høring, og særlig forslaget om løsningen som innebar en stor grad av sentralisering av behandlingsfunksjoner fikk mye kritikk. Aktørene i bransjen er bl.a. skeptiske til at kritiske forretningsprosesser skal gjøres felles. En slik situasjon vil fjerne konkurransefortrinnet til aktører som spesialiserer seg på effektiv styring. Dessuten er nettselskapene er skeptiske til at de vil miste sin rolle som ansvarlig for målerdata som genereres på strømmettene de er ansvarlig for å drifte. I tillegg er bransjen kritisk til etableringen av en felles IKT-løsning før man ser de virkelige behovene som AMS vil skape.

Jeg synes bransjen for det meste har tungtveiende og gode argumenter. Historien har vist at markedssituasjoner som i stor grad er preget av konkurranse og leverandøruavhengighet skaper et innovativt miljø som belønner aktører med de mest effektive løsningene. Etter min mening er det liten grunn til å tro at en stor sentral og byråkratisk styrt organisasjon skal legge til rette for et mer effektivt og teknisk velfungerende system enn hvis markedskreftene i større grad får definere systemene og løsningene.

På den andre siden ser jeg absolutt fordelene med at aktører, det være kraftleverandører, regulatoriske myndigheter eller kommersielle tredjeparter, kun har ett sted å henvende seg for å få nødvendige data. Innovative tredjepartstjenester kan bidra til å få utnyttet potensialet til AMS, og da er det viktig at det etableres et velfungerende system for at tredjeparter får tilgang på data de trenger. Dette argumentet blir også ytterligere forsterket når visjonene om et felles europeisk sluttbrukermarked kommer inn i bildet. Skal utenlandske kraftleverandører og tredjepartsaktører levere sine tjenester i det norske markedet, er man avhengig av et effektivt, felles og standardisert kontaktpunkt for utveksling av data.

Oppsummert ser jeg for meg følgende overordnede krav til et system:

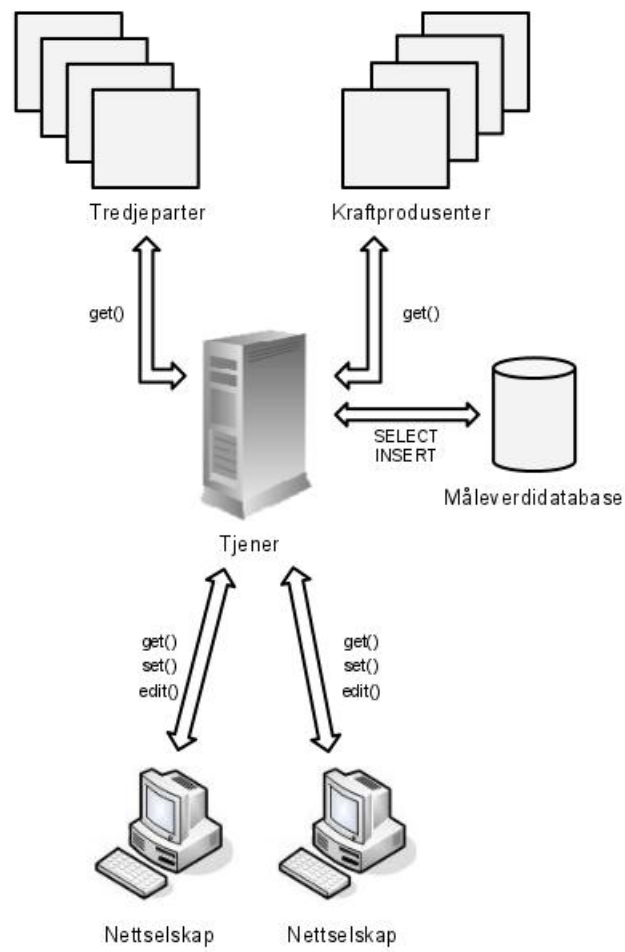
- Ett felles kontaktpunkt for kraftleverandører og tredjeparter, basert på åpne og velkjente standarder.
- Fritt valg av IKT-løsning for nettselskapene, for å legge til rette for innovasjon og effektivisering av driften.

- Nettselskapene har ansvaret for måleverdienes kvalitet og korrekthet.

Spørsmålet blir da, hvordan skal kontaktpunktet implementeres og forvaltes slik at kravene oppfylles?

4.2.1 Sentralt lagret data

Slik jeg ser det er det å etablere en felles database en aktuell mulighet, som beskrevet i seksjon 3.2.2. Funksjonalitet vil derimot være noe forenklet sammenlignet med nevnte beskrivelse. Ansvaret for kvalitetssikring av målerdata bør etter min mening ligge hos nettselskapene, da det er de som har direkte kontakt med målepunktet. Nettselskapene står fritt til å benytte den sentrale databasen som sin primære database, men velger de å ikke gjøre det må det stilles strenge krav til at den sentrale databasen holdes oppdatert med kvalitetssikrede data.

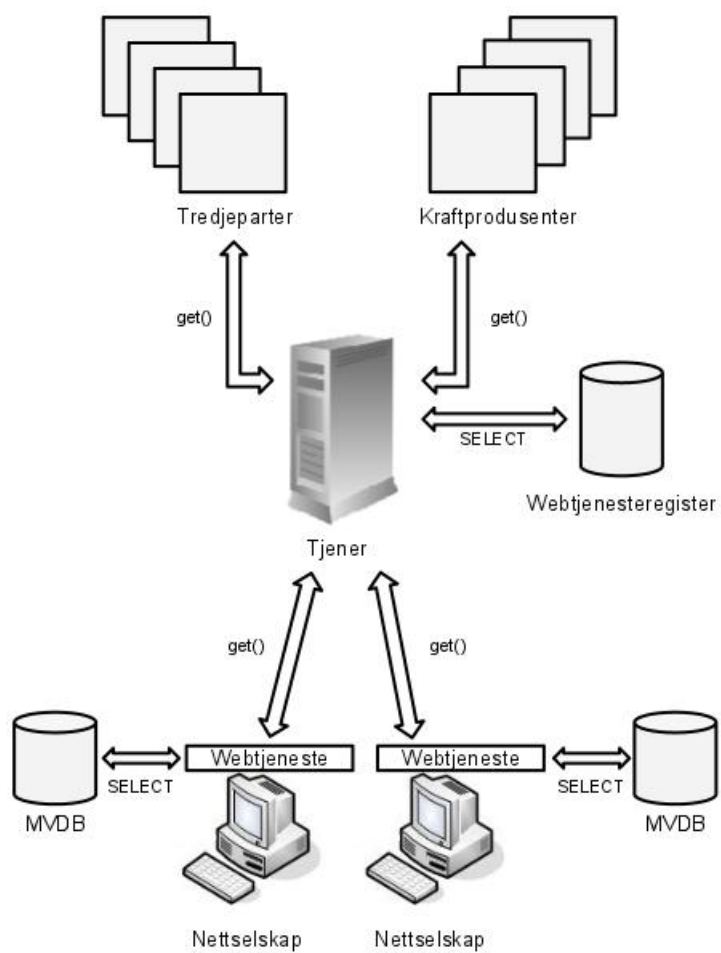


Figur 4.2: Forslag til arkitektur for sentralisert database

En overordnet skissering av den tenkte løsningen er gitt i figur 4.2. Tredjeparter og kraftleverandører har mulighet til å hente ut måleverdier fra Webtjenesten som kommuniserer med en sentralisert database. Aktørene gjør et oppslag på målepunktID, starttid og sluttid, og får samtlige måleverdier mellom start- og sluttiden i retur. Kommunikasjonen går over HTTPS, og Webtjenesten må foreta en autorisasjonskontroll for å sikre at aktørene kun får hente ut data fra målepunkter de har tillatelse til. Se seksjon 4.1.3 for en beskrivelse av hvordan autentisering kan skje ved bruk av Webtjenester. På den andre siden må nettselskapene ha mulighet til å sette, endre og hente ut data fra et målepunkt. På samme måte som for kraftprodusenter og tredjeparter, må nettselskapene også ha mulighet til å hente ut verdier for et målepunkt. Denne funksjonaliteten er først og fremst for nettselskapene som bruker den sentrale databasen som sin primære database. Også her bør kommunikasjonen gå over HTTPS, samt at Webtjenesten må foreta en autentiseringskontroll av nettselskapene.

4.2.2 Nubixbasert meldingsformidler

En annen mulighet for deling av målerdata er en videreutvikling av Nubix-systemet, beskrevet i 3.1.2. Nubix vil da implementere funksjonalitet for å videresende forespørsler fra kraftprodusenter og tredjepartsaktører til korrekt nettselskap. En overordnet arkitekturskisse er gitt i figur 4.3. Den sentrale Webtjenesten inneholder en database som inneholder en oversikt over hvilke målepunkt-IDer som hører til hvilke nettselskap, samt hvilke aktører, kraftleverandører og tredjeparter, som har tillatelse til å lese av et gitt målepunkt. Forespørselen videresendes så, hvis tillatt, til korrekt nettselskap som gjør oppslag i sin MVDB. Som i Nubix, må nettselskapene implementere en Webtjeneste som tar imot og svarer på slike forespørsler, men nettselskapene står fritt til å implementere logikken bak tjenesten. Med en slik løsning legger man opp til at nettselskapene f.eks kan samarbeide regionalt om en felles MVDB, eller å benytte tjenester i Nettskyen for lagring av målerdata. For tredjeparter og kraftleverandører vil en slik løsning prinsipielt sett være lik den forrige løsningen, da de kun vil ha ett kontaktpunkt å forholde seg til for å lese måleverdier.



Figur 4.3: Forslag til Nubix-basert arkitektur

4.2.3 Sentral lagring eller videresending?

Det finnes gode argumenter både for og i mot begge alternativene jeg har skissert. Ved å bruke en sentral MVDB vil alle målerdata, uavhengig av om sluttbrukeren benytter tjenester fra en tredjepart eller benytter en kraftleverandør fra utenfor nettselskapets organisasjon, bli lastet opp til den sentrale databasen. Dette vil si at i tilfeller hvor en sluttbruker ikke har noe kundeforhold med en ekstern aktør, som man kan anta vil gjelde for flere sluttbrukere, vil ikke denne opplastingen ha noen nevneverdig hensikt ettersom flere av nettselskapene har sagt at de uansett vil ha sin egen MVDB. Hvis man derimot baserer seg på en Nubixbasert meldingsformidler, vil kun de målerdata som blir etterspurt sendes over nettet. Gitt at i snitt hvert målepunkt leses av mer enn en aktør, vil et distribuert alternativ medføre høyere total trafikkmengde ut fra nettselskapet, men som jeg kommer tilbake til i seksjon 6.2.3, så vil ikke datamengden være urovekkende stor.

Som nevnt over bør ikke nettselskapene være forpliktet til å bruke en eventuell sentralisert database som sin primære. Det betyr ikke at flere nettselskaper vil benytte den sentrale databasen som sin primære, gitt at den tilfredsstillter deres krav. En sentral database som driftes av en kommersiell, men regulert, aktør, vil kunne konkurrere med andre tilbydere av datalagring og også ta i bruk innovative og kostnadsbesparende teknologier for lagring og deling av data. På den måten er det ikke utenkelig at et slikt system vil være attraktivt for nettselskaper som ønsker å outsource denne delen av driften.

Hovedfordelen, slik jeg ser det, med et Nubixbasert alternativ, er at det er et betydelig enklere system å implementere, forutsatt at man baserer seg på det arbeidet som allerede er gjort, samt aktivt utnytter kompetansen og erfaringene som finnes. Det er derfor grunn til å forvente at en Nubixbasert løsning vil bli vesentlig billigere enn å etablere en sentralisert database, i tillegg til at man unngår unødvendig dobbeltlagring og transmisjon av målerdata, samt utfordringen med å holde databasene synkronisert.

Uavhengig av valgt løsning, må kommunikasjonen foregå kryptert for å ivareta personvernet og hindre at uvedkommende kan aksessere data. Nubix bruker som nevnt over HTTPS for transport, og det er naturlig å fortsette med dette. Som jeg har beskrevet i seksjon 4.1.3, så kreves det endel prosessering for å opprette en HTTPS-tilkobling mellom to parter. Hvis man benytter en sentral database må det kun opprettes en HTTPS-tilkobling for hver forespørsel, mellom klient og tjener. Hvis man derimot benytter det

Nubixbaserte alternativet, må man først opprette en HTTPS-tilkobling mellom klient (kraftleverandør eller tredjepart) og tjener (det sentrale systemet), før det opprettes en ny HTTPS-tilkobling mellom klient (det sentrale systemet) og tjener (nettselskap). Med andre ord må det opprettes to HTTPS-tilkoblinger, noe som gir dobbelt så mye prosessering for hvert kall på målerdata.

Etter min mening så fremstår det Nubix-baserte alternativet som det mest fornuftige. Hovedsaklig på grunn av at det baserer seg på videreutvikling av et allerede eksisterende og velfungerende system. Implementerings- og utviklingskostnadene kan da holdes relativt lave. Det fremstår fornuftig å ha en avventende holdning til etablering av en omfattende fellesløsning på IT-siden før man ser behovene AMS vil skape. I tillegg er bransjen generelt kritiske til en stor grad av sentralisering av IKT-løsninger, og det virker uklokt å tvinge igjennom en løsning som vil skape misnøye blant brukerne. Det er derimot viktig at et Nubixbasert meldingsformidlingssystem implementeres slik at det kan videreutvikles til å bli et mer omfattende system. Det vil spare kostnader hvis man på et senere tidspunkt ser at det er et reelt behov for en slik organisering.

4.3 Oppsummering

Dette kapitlet startet med en introduksjon til Webtjenester, et standardisert rammeverk for applikasjon-til-applikasjon-kommunikasjon. Senere presenterte jeg etter min mening de to mest realistiske løsningene for deling av målerdata mellom nettselskap, tredjeparter og kraftleverandører. Jeg tok utgangspunkt i en konsulentrapport som tok for seg en tenkt fremtidig IKT-løsning i kraftbransjen, samt kommentarene rapporten fikk fra ulike bransjeaktører. Grunnet aspekter som implementeringskostnad og -tid, fleksibilitet og det faktum at det allerede finnes kompetanse forbundet med utvikling og driftning av et tilsvarende system, kom jeg frem til at et Nubix-basert system for deling av målerdata vil være mest hensiktsmessig.

5

DESIGN OG IMPLEMENTASJON

I dette kapitlet vil jeg gå gjennom de viktigste detaljene forbundet med Nubix 2.0, et system for deling av målerdata, både fra et overordnet designperspektiv og et mer detaljert implementasjonsperspektiv. Jeg vil begynne med å spesifisere kravene jeg har satt til systemet, deretter fortsette med en overordnet designbeskrivelse, før jeg beveger meg nedover mot et mer detaljert nivå mot slutten av kapitlet. En mer utfyllende diskusjon rundt de tekniske løsningene jeg har valgt kommer i kapittel 6. Merk at Nubix 2.0 er navnet jeg har gitt systemet som har blitt utviklet, og at det ikke har noen formell tilknytning til det eksisterende Nubix-systemet.

5.1 Krav

I seksjon 4.2 beskrev jeg tre overordnede krav til systemet; ett felles kontaktpunkt for aktørene i bransjen, fritt valg av IKT-løsning for nettselskapene, samt at selve målerdataene forvaltes av nettselskapene. Jeg vil her presentere og begrunne de konkrete kravene jeg stilte til Nubix 2.0 med hensyn til arkitektur og funksjonalitet, datastruktur og sikkerhet.

5.1.1 Arkitektur og funksjonalitet

Nubix 2.0 vil kommunisere med et ubestemt antall aktører; nettselskaper, kraftprodusenter, regulatoriske myndigheter og andre tredjeparter. En veldokumentert og åpen arkitektur vil kunne bidra til å øke fleksibiliteten og redusere kompleksiteten til systemet. En slik arkitektur vil også bidra til å holde utviklingskostnadene nede, samt gjøre det

enklere for aktører å kommunisere med systemet. Rent konkret resulterer dette i følgende krav:

- Systemet skal benytte IP som grunnleggende nettverksprotokoll.
- Systemet skal tilby et grensesnitt basert på åpne og veldokumenterte standarder.
- Systemet skal der det er mulig utnytte allerede eksisterende funksjonalitet.
- Systemet skal kunne håndtere et stort antall forespørsler.
- Systemet skal ikke legge sterke føringer på valg av utviklingsmiljø hos aktørene.

Kravet om å benytte IP som nettverksprotokoll grunner i at IP er den desidert mest brukte nettverksprotokollen, og ved å benytte en godt kjent, mye brukt og standardisert protokoll reduseres barrierene for aktørene som ønsker å benytte systemet.

Smartmålere vil, i tillegg til detaljert målerdata, også kunne tilby avlesning av aktivt forbruk. Det fører til følgende krav til systemets tilbudte metoder:

- Systemet skal tilby en metode for innhenting av forbruksdata mellom to tidspunkt for et gitt målepunkt.
- Systemet skal tilby en metode for innhenting av aktivt forbruk for et gitt målepunkt.

5.1.2 Datastruktur

Ettersom AMS vil medføre en drastisk økning i tilgjengelig forbruksdata må det defineres en datastruktur for overføring av dataen. For å minske den totale datamengden som overføres bør ikke datastrukturen inneholde overflødig data. Datastrukturen bør også lages mtp. at den vil prosesseres av andre applikasjoner, ikke nødvendigvis leses direkte av mennesker. Oppsummert fører dette til følgende krav:

- Datastrukturen skal være på et veldokumentert og kjent format.
- Datastrukturen skal basere seg på en hierarkisk oppbygging.
- Datastrukturen skal utvikles primært for å forstås av maskiner, ikke mennesker.

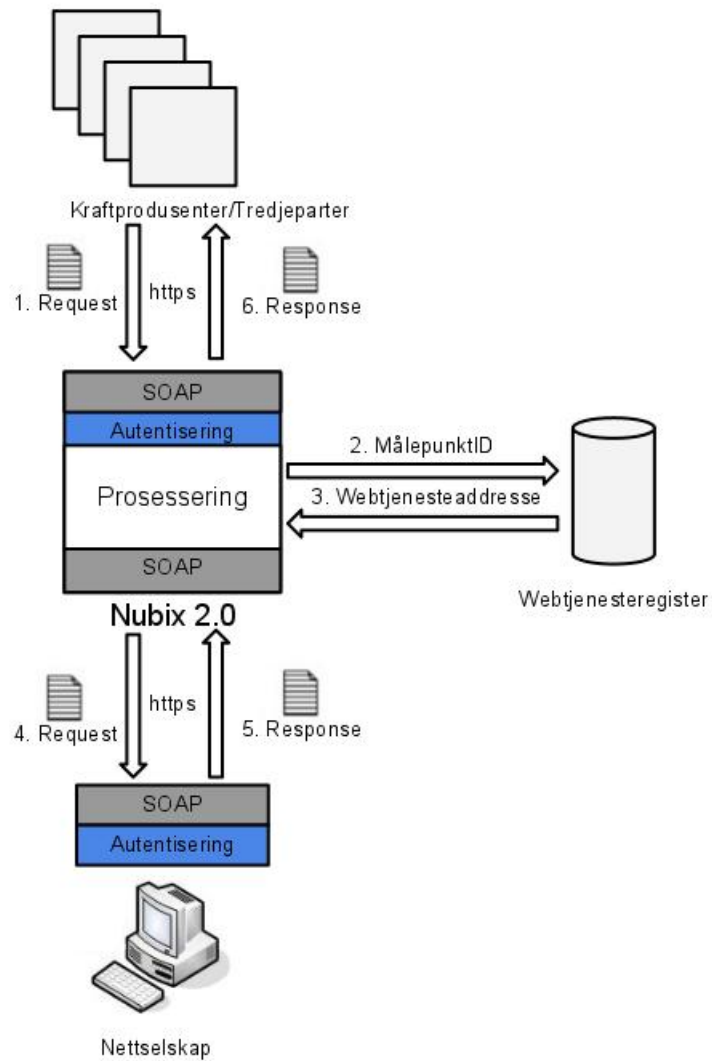
5.1.3 Sikkerhet

Detaljert informasjon om historisk og aktivt strømforbruk kan tenkes brukt til å danne seg et bilde av om en bruker er i hjemmet eller ikke, og er på den måten å anse som personsensitiv data. Det er derfor viktig at Nubix 2.0 ivaretar sikkerheten på en tilfredsstillende måte, både ved å sørge for at kun autoriserte brukere får tilgang til målerdata, samt at uvedkommende hindres adgang til målerdata under transmisjon. Rent konkret skal:

- All dataoverføring være kryptert etter gjeldende krav.
- Kun registrerte og autoriserte aktører ha tilgang til målerdata.

5.2 Design

Som nevnt i 4.2.3 har jeg kommet frem til at en videreutvikling av det eksisterende Webtjenestebaserte Nubix-systemet fremstår som det mest hensiktsmessige alternativet. Som nevnt i seksjon 4.1.4 har Webtjenester den fordel at det er en plattform- og produsentuavhengig teknologi. Webtjenester er i tillegg basert på IP, benytter XML som dataformat og er allerede til en viss grad tatt i bruk av kraftbransjen. Slik jeg ser det er Webtjenester det beste alternativet basert på kravene jeg stilte i seksjon 5.1.1. Figur 5.1 viser en overordnet designskisse av hvordan Nubix 2.0 fungerer. Merk nummereringen på meldingsutvekslingene og operasjonene.



Figur 5.1: Nubix 2.0

Meldingene som utveksles er som følger.

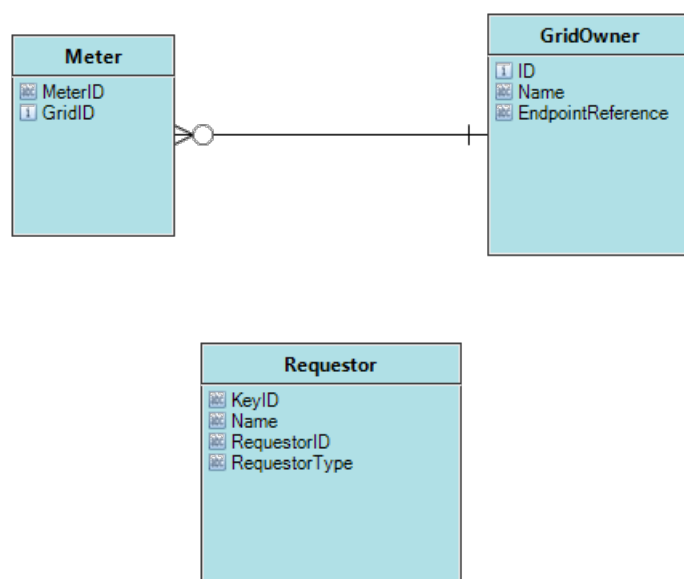
1. Kraftprodusenten eller tredjeparten, heretter kalt klienten, sender forespørsler som SOAP-meldinger til Webtjenesten.
Forespørslene inneholder bl.a. autentiseringsinformasjon, IDen til måleren klienten ønsker å lese forbruket til samt ønsket tidsrom klienten ønsker målerdata fra, og sendes som nyttelast i en HTTPS-melding. Ved ankomst blir klienten autentisert.
2. Nubix 2.0 søker opp Webtjenesteadressen til korrekt nettselskap basert på mottatt målepunkt-ID.
3. Webtjenesteadressen returneres og en ny SOAP-melding genereres.
4. SOAP-meldingen sendes over HTTPS til korrekt nettselskap.
Nettselskapet autentiserer nå klienten (Nubix 2.0) og behandler forespørselen.
5. Nettselskapet sender svaret tilbake til Nubix 2.0
6. Svaret blir videresendt tilbake til klienten.

5.2.1 Database

Databasen jeg har satt opp er illustrert i figur 5.2. Databasen er tenkt til å inneholde en oversikt over alle målerne i Norge, samt hvilket nettselskap de tilhører. For hvert nettselskap lagres navnet, samt adressen ("EndpointReference") til Webtjenesten nettselskapet har etablert. I tillegg inneholder databasen en tabell som inneholder IDen til nøkkelen som klienten har signert meldingen med, beskrevet i seksjon 5.3.5. Denne IDen er unik, og brukes derfor som indeks i tabellen. I tillegg har jeg valgt å inkludere en unik ID for hver registrerte klient, samt et flagg ("RequestorType") som er tenkt til å definere rollen til klienten (f.eks kraftprodusent, regulatorisk myndighet, kommersiell tredjepart, etc.).

5.2.2 Metoder

I henhold til kravene i 5.1.1 tilbyr Nubix 2.0 to metoder til klienter som ønsker å lese målerdata; "getActiveMeterData" og "getReportedMeterData". Den førstnevnte metoden



Figur 5.2: Database - Nubix 2.0

tar kun den ønskede målerens ID som parameter og returnerer, hvis måleren finnes, hvis klienten er registrert i Nubix og hvis klienten er autorisert for å lese data fra denne måleren, momentanforbruket til denne måleren.

”GetReportedMeterData” returnerer forbruksdata samlet inn innenfor et gitt tidsrom. Metoden tar parametrene målerpunkt-ID, starttid og sluttid som input, og returnerer tidsstemplet målerdata innenfor dette intervallet, gitt at klienten har tillatelse til å lese målerdataen.

Merk at Nubix 2.0 kun foretar en kontroll på om hvorvidt klienten i det hele tatt har tillatelse til å gjøre oppslag mot nettselskapenes databaser, mens funksjonalitet for å sørge for at klienten kun får lese data fra målere den har et kundeforhold til må implementeres hos nettselskapet.

5.3 Implementasjon

I denne seksjonen vil jeg gå mer teknisk inn på implementeringsløsninger, og da med fokus på WSDL-definisjonene forbundet med Nubix 2.0. WSDL-definisjonene er relativt verbale, så jeg viser til vedlegg A for en full gjengivelse av WSDL-definisjonene.

Nubix 2.0 vil operere både som klient og som tjener; kraftleverandører og tredjeparter vil se på Nubix 2.0 som en tjener, mens fra nettselskapenes perspektiv vil Nubix 2.0

være en klient som utfører spørringer mot deres systemer. Dette medfører at man trenger to WSDL-dokumenter for å definere det totale systemet; ett dokument for å beskrive tjenerfunksjonaliteten til Nubix 2.0 rettet mot kraftleverandører og tredjeparter, og ett dokument for å beskrive Webtjenesten som nettselskapene må implementere.

5.3.1 Utviklingsmiljø

Det finnes flere typer programvare og utviklingsmiljøer som støtter utvikling av Webtjenester. Jeg har valgt å benytte meg av Eclipse med tilleggspakken "Web tools platform", Apache Axis2, en java-basert modul som støtter implementering av både klient- og tjenersiden av en Webtjeneste. For mer informasjon om Axis2, se [21]

I tillegg til Webtjenesteprogramvare, trengs også programvare for å kjøre en eller flere Webtjenerinstanser. Apache Tomcat er en populær tjenerimplementasjon basert på åpen kildekode. Tomcat og Axis2 kommer i tillegg fra samme stiftelse, så de er relativt enkle å integrere. For mer informasjon om Tomcat, se [23].

Axis2 er modulbasert, og det finnes flere ulike moduler man kan legge til for å få utvidet funksjonalitet. Apache Rampart, [22], er en slik modul, og blir brukt for å implementere Webtjenestesikkerhet i en Axis2-basert Webtjeneste. Rampart støtter funksjonalitet beskrevet i WTPR, som inkluderer signering, kryptering, med mer.

5.3.2 Metodikk

Det er to prinsipielt forskjellige metoder å utvikle en Webtjeneste på; "top-down" eller "bottom-up". Med det menes hhv. at man enten starter med å definere et WSDL-dokument, for så å benytte et verktøy for å generere rammeverket for kommunikasjon med tjenesten, eller at man starter med å lage kode, for så å benytte et tilsvarende verktøy for å generere et WSDL-dokument. Webtjenesteklienter utvikles primært etter første prinsipp, dvs. at man bruker et WSDL-dokument for å generere klasser og funksjonalitet som tar seg av selve kommunikasjonen med tjenesten. Det er fordeler og ulemper forbundet med hvert prinsipp. Fordelen med "bottom-up" er at det er en relativt rask måte å utvikle en Webtjeneste utifra et eksisterende system. Det er også en gunstig metode hvis man utvikler en Webtjeneste ment å fungere innad i en organisasjon, da man som regel har full

kontroll over både klient og tjener, samt miljøet de opererer i. "Top-down", på den andre siden, har den fordelen at det skaper en Webtjeneste uavhengig av programmeringsspråk og utviklingsmiljø, og er på den måten et mer interoperatibelt alternativ.

For å gå tilbake til kravene jeg stilte i 5.1.1, særlig til det punktet som omhandler det å legge føringer på implementeringen hos nettselskapene, er det tydelig at "top-down"-metoden vil være mest hensiktsmessig for Nubix 2.0. Det vil også være mer hensiktsmessig fra et implementeringsperspektiv. Ved å kun tilby nettselskapene et WSDL-dokument for å beskrive tjenesten de må implementere unngår man at utviklerne må implementere det samme i de ulike nettselskaperes systemer.

Axis2 tilbyr et Eclipse-integrert verktøy, "WSDL2JAVA", som jeg har benyttet for å generere rammeverk fra WSDL-dokumentene. All kommunikasjon med databasen, autentisering av brukere, samt videresending til riktig nettselskap må implementeres separat.

5.3.3 Datastruktur

Jeg har spesifisert en datastruktur jeg mener er hensiktsmessig for å utveksle målerdata i henhold til kravene jeg stilte i 5.1.2. Et sett med tidsstemplede målerdata, "ReportedData", er pakket inn i den overordnede klassen "MeterData". På denne måten får man en hierarkisk og fleksibel datastruktur som er uavhengig av størrelsen på tidsintervallet for målingene. Det at datastrukturen er basert på XML er også en fordel i MTM-kommunikasjon, da det finnes flere biblioteker og metoder for prosessering av XML. XML-skjemadefinisjonen for "*MeterData"-datastrukturen er gitt under.

```
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://logic/xsd">
  <xs:complexType name="MeterData">
    <xs:sequence>
      <xs:element minOccurs="0" name="meterID" nillable="true" type="xs:string"/>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="reportedData" nillable="true" type="ax21:ReportedData"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ReportedData">
    <xs:sequence>
```



```

<xs:element minOccurs="0" name="consumption" type="xs:double" />
<xs:element minOccurs="0" name="endTime" nillable="true" type="xs:dateTime" />
<xs:element minOccurs="0" name="startTime" nillable="true" type="xs:dateTime" />
</xs:sequence>
</xs:complexType>
</xs:schema>

```

For aktivt forbruk blir datastrukturen enklere, da det kun er en tallverdi per melding, i tillegg til målepunktID og en tidsstempling. Skjemadefinisjon er gitt under.

```

<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified" targetNamespace="http://logic/xsd">
  <xs:complexType name="ActiveMeterData">
    <xs:sequence>
      <xs:element minOccurs="0" name="activeConsumption" type="xs:double" />
      <xs:element minOccurs="0" name="meterID" nillable="true" type="xs:string" />
      <xs:element minOccurs="0" name="timeStamp" nillable="true" type="xs:dateTime" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

5.3.4 Meldingsdefinisjoner

Som beskrevet i seksjon 4.1.1 må meldinger som utveksles i en Webtjeneste beskrives av et "message"-element i WSDL-definisjonen, i tillegg til at operasjonene eller metodene tilbud av en Webtjeneste må samles i separate "operation"-elementer under elementet "PortType". Nubix 2.0 tilbyr to metoder, som totalt gir fire forespørsel- og svar-meldinger. WSDL-definisjonen er gitt under.

```

<wsdl:message name="getReportedMeterDataRequest">
  <wsdl:part name="parameters" element="ns:getReportedMeterData" />
</wsdl:message>

<wsdl:message name="getReportedMeterDataResponse">
  <wsdl:part name="parameters" element="ns:getReportedMeterDataResponse" />
</wsdl:message>

```

```
</wsdl:message>

<wsdl:message name="getActiveMeterDataRequest">
  <wsdl:part name="parameters" element="ns:getActiveMeterData"></wsdl:part>
</wsdl:message>

<wsdl:message name="getActiveMeterDataResponse">
  <wsdl:part name="parameters" element="
    ns:getActiveMeterDataResponse"></wsdl:part>
</wsdl:message>

<wsdl:portType name="GetMeterDataServicePortType" wsdl:PolicyURIs
  ="#SigOverHTTPS">

  <wsdl:operation name="getActiveMeterData">
    <wsdl:input message="ns:getActiveMeterDataRequest"
      wsaw:Action="urn:getActiveMeterData"></wsdl:input>
    <wsdl:output message="ns:getActiveMeterDataResponse"
      wsaw:Action="urn:getActiveMeterDataResponse"></wsdl:output>
  </wsdl:operation>

  <wsdl:operation name="getReportedMeterData">
    <wsdl:input message="ns:getReportedMeterDataRequest"
      wsaw:Action="urn:getReportedMeterData"></wsdl:input>
    <wsdl:output message="ns:getReportedMeterDataResponse"
      wsaw:Action="urn:getReportedMeterDataResponse"></wsdl:output>
  </wsdl:operation>
</wsdl:portType>
```

Definisjonen er relativt intuitiv. Hver metode har to meldinger forbundet med seg, hhv. en input ("request") og et output ("response"). Webtjenesten tilbyr som nevnt totalt to metoder, "getActiveMeterData" og "getReportedMeterData", som begge er definert under elementet "portType". "portType" har også et attributt, "PolicyURI", som her spesifiserer IDen til sikkerhetsspesifikasjonen, beskrevet i neste avsnitt, som operasjonene må følge.

5.3.5 Sikkerhet

Som jeg nevnte i seksjon 5.1.3 er detaljert strømmålerdata samt aktivt forbruk å anse som personsensitiv data, og bør beskyttes mot uautorisert tilgang. Det vil si at data må bli kryptert før de sendes, samt at aktørene må autentiseres før datautveksling kan finne sted.

Når det gjelder kryptering så har jeg valgt å benytte HTTPS, som er de-facto standarden for sikker overføring på Internett. Når det gjelder autentisering er det primært to metoder dette gjøres på i dag; brukernavn- og passordautentisering eller autentisering ved hjelp av digitale signaturer. Ettersom sistnevnte er et sikrere autentiseringsverktøy, gitt at det implementeres og forvaltes riktig, samt at mekanismen i tillegg kan brukes til å verifisere meldingsintegritet, fremstår det som det mest logiske valget for autentisering.

I seksjon 4.1.3 presenterte jeg såvidt et rammeverk for å definere sikkerhetsmekanismer på meldingsnivå til bruk i Webtjenester. Rent praktisk gjøres det ved at man legger til et "Policy"-element i WSDL-dokumentet som spesifiserer hva klienter må oppfylle sikkerhetsmessig for å kommunisere med tjenesten. For Nubix 2.0 har jeg, inspirert av [20], definerte følgende "Policy"¹ for å sørge for at meldingsutvekslingen blir signert digitalt:

```
<wsp:Policy wsu:Id="SigOverHTTPS" xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <wsp:ExactlyOne>
    <wsp:All>
      <sp:Wss10>
        <sp:Policy>
          <sp:MustSupportRefKeyIdentifier />
          <sp:MustSupportRefIssuerSerial />
        </sp:Policy>
      </sp:Wss10>
      <sp:TransportBinding>
        <wsp:Policy>
          <sp:TransportToken>
            <wsp:Policy>
              <sp:HttpsToken RequireClientCertificate="false" />
            </wsp:Policy>
          </sp:TransportToken>
          <sp:AlgorithmSuite>
            <wsp:Policy>
              <sp:Basic256 />
            </wsp:Policy>
          </sp:AlgorithmSuite>
          <sp:Layout>
            <wsp:Policy>
              <sp:Lax />
            </wsp:Policy>
          </sp:Layout>
        </wsp:Policy>
      </sp:TransportBinding>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

¹Merk at jeg har tatt bort XML navnerom-spesifikasjoner av layoutsmessige årsaker

```
        </wsp:Policy>
    </sp:Layout>
    <sp:IncludeTimestamp />
</wsp:Policy>
</sp:TransportBinding>
<sp:AsymmetricBinding>
    <wsp:Policy>
        <sp:InitiatorToken>
            <wsp:Policy>
                <sp:X509Token>
                    <wsp:Policy>
                        <sp:WssX509V3Token10 />
                    </wsp:Policy>
                </sp:X509Token>
            </wsp:Policy>
        </sp:InitiatorToken>
        <sp:RecipientToken>
            <wsp:Policy>
                <sp:X509Token>
                    <wsp:Policy>
                        <sp:WssX509V3Token10 />
                    </wsp:Policy>
                </sp:X509Token>
            </wsp:Policy>
        </sp:RecipientToken>
        <sp:AlgorithmSuite>
            <wsp:Policy>
                <sp:Basic256 />
            </wsp:Policy>
        </sp:AlgorithmSuite>
    <sp:Layout>
        <wsp:Policy>
            <sp:Strict />
        </wsp:Policy>
    </sp:Layout>
    <sp:IncludeTimestamp />
    <sp:OnlySignEntireHeadersAndBody />
</wsp:Policy>
</sp:AsymmetricBinding>
<sp:SignedParts>
    <sp:Body />
</sp:SignedParts>
</wsp:All>
</wsp:ExactlyOne>
</wsp:Policy>
```

”Policyen” ovenfor består av et sett med ”Policy Assertions”, samt to operatører. De to operatørene er ”ExactlyOne” og ”All”, og når de står i den rekkefølgen betyr de simpelthen at samtlige ”assertions” må oppfylles. Det er to ”assertions” av interesse; ”TransportBinding” og ”AsymmetricBinding”. Den førstnevnte definerer at Webtjenesten kun er tilgjengelig over HTTPS, samt at melding skal krypteres med 256-biters AES-kryptering². ”AsymmetricBinding” definerer for det første at det skal foregå en kryptografisk operasjon basert på asymmetriske nøkler. I Nubix 2.0 benytter jeg ”AsymmetricBinding” for å spesifisere at meldingsutvekslingen skal signeres av begge parter, toveissignering spesifiseres ved inkluderingen av subelementene ”InitiatorToken” og ”RecipientToken” under ”AsymmetricBinding”. Feltet ”SignedParts” definerer hvilke deler av SOAP-meldingen som skal signeres, i dette tilfellet kroppen (”sp:Body”). Meldingene som utveksles vil da inneholde et hodeelement som inneholder chifftereksten til kroppen på meldingen. Kryptering, dekryptering og signaturverifisering blir gjort automatisk av det underliggende Rampart-biblioteket.

For at aktørene kan verifisere mottatte signaturer, må de være i bestittelse av den andre partens offentlige nøkler. Som nevnt i seksjon 4.1.3 inneholder et sertifikat aktørens navn, dets offentlige nøkkel og en signatur fra en sikkerhetsautoritet som går god for sertifikatets integritet. I testsystemet jeg har satt opp har jeg benyttet verktøyene Java Keytool og Openssl for generering av nøkkelpar, samt X509-sertifikater signert av en fiktiv sikkerhetsautoritet. [17] gir en god gjennomgang av hvordan dette kan gjøres.

5.4 Oppsummering

Dette kapitlet presenterte Nubix 2.0, et tenkt system for sikker deling av målerdata mellom nettselskap og kraftleverandører og tredjeparter. I første delen av kapitlet definerte jeg kravene som ble satt til systemet. Siste delen av kapitlet ga en innføring i de konkrete tekniske design- og implementasjonsløsningene jeg har valgt for å tilfredstille de nevnte krav. Neste kapittel vil gå nærmere inn på testoppsettet, samt en mer utfyllende diskusjon rundt de valgene jeg har tatt og hvilke implikasjoner de vil kunne få.

²Spesifisert av feltet ”Basic256” under elementet ”AlgorithmSuite”

6

RESULTATER OG DISKUSJON

Dette kapitlet vil starte med å presentere systemet jeg har satt opp for å teste Nubix 2.0, beskrevet i forrige kapittel, ut fra de kravene som ble satt i seksjon 5.1. Videre vil jeg gå inn på en diskusjon rundt valgte løsninger, samt gå inn på aspekter jeg ikke har fokusert på. Merk at alle tallverdier som returneres av systemet er fiktive verdier, da kommunikasjon med smartmålere har blitt sett bort ifra i denne masteroppgaven.

6.1 Resultater

6.1.1 Testoppsett

Aktører og Kundeforhold

For å teste funksjonaliteten i Nubix 2.0 har jeg satt opp totalt seks aktører. To nettselskaper, "GridA" og "GridB", to kraftprodusenter, "Powerplant" og "Waterfall", og to tredjepartsaktører, "Energysystems" og "Energysolutions". I tillegg har jeg lagt til totalt fire fiktive smartmålere i databasen til Nubix med målerID på hhv. "12345678", "87654321", "23456789" og "98765432", hvor de to første målerne tilhører "GridB" og de to siste tilhører "GridA". I nettselskapenes databaser har jeg lagt inn fiktive målerdata på timebasis for alle målerne for hele november. Tabell 6.1 viser en oversikt over kundeforholdene hos "GridA". Målepunkt "23456789" fikk strøm fra kraftprodusent "Waterfall" hele november, i tillegg til at tredjepart "Energysolutions" har fått tillatelse til å lese av aktivt forbruk f.o.m. 10. november kl. 1100.

Målepunkt "98765432" fikk strøm fra "Powerplant" t.o.m. 19. november, men byttet til

Målepunkt	Powerplant	Waterfall	Energysolutions	Energysystems
23456789		1. t.o.m 30. november	Aktiv f.o.m. 10. november kl. 1100	
98765432	1. t.o.m. 19. november	20. t.o.m. 30. november		Alt f.o.m. 20. november kl. 1100

Tabell 6.1: Kundeforhold hos "GridA"

Målepunkt	Powerplant	Waterfall	Energysolutions	Energysystems
12345678	1. t.o.m 30. november		Alt f.o.m. 17. november kl. 1000	
87654321	1. t.o.m. 9. november	10. t.o.m. 30. november	Forbruk f.o.m. 16. november kl 0000	

Tabell 6.2: Kundeforhold hos "GridB"

"Waterfall" f.o.m. 20. november. Tredjeparten "Energysystems" fikk tillatelse til å lese alt, dvs. både aktivt og rapportert forbruk f.o.m. 20. november kl. 1100.

Tilsvarende viser tabell 6.2 kundeforholdene hos "GridB". Målepunkt "12345678" fikk strøm fra "Powerplant" hele november og tillot "Energysolutions" å lese samtlige målerdata f.o.m. 17. november kl. 1000. Målepunkt "87654321" på den andre siden fikk strøm fra "Powerplant" t.o.m. 9. november, men byttet til "Waterfall" f.o.m. 10. november. Dette målepunktet har også et kundeforhold til tredjeparten "Energysolutions" som tillater tredjeparten å lese av rapportert forbruk f.o.m. 16. november kl 0000.

Tjerneroppsett

Hver av tjenerne, Nubix 2.0, "GridA" og "GridB", kjører på sin egen instans av en lokal Tomcat-tjener og er konfigurert til å lytte på forskjellige porter, hhv. 8443, 8445 og 8447. Av testårsaker har jeg også gjort Webtjenestene tilgjengelige over HTTP på portnummer 8080, 8090 og 8095. Tabell 6.3 viser full adresse ("EndpointReference") for de tre Webtjenestene.

Nubix 2.0	https://localhost:8443/Nubix20/services/GetMeterDataService.GetMeterDataServiceHttpsSoap11Endpoint/
GridA	https://localhost:8445/GridA/services/GetMeterDataService.GetMeterDataServiceHttpsSoap12Endpoint/
GridB	https://localhost:8447/GridB/services/GetMeterDataService.GetMeterDataServiceHttpsSoap12Endpoint/

Tabell 6.3: Webtjenesteadresser

Nøkler og Sertifikater

For å aksessere en HTTPS-basert Webtjeneste må klienten stole på integriteten til tjenerens sertifikat. I et reelt scenario kan man tenke seg at tjeneren har fått utstedt et sertifikat fra en betrodd sertifikatautoritet, men dette testsystemet baserer seg kun på selvsignerte sertifikater. Det innebærer at klientene må ha importert tjenerens sertifikater i sine "truststores", filer som inneholder betrodde tjeneres sertifikater. I tillegg, ettersom aktørene autentiserer seg ved hjelp av digitale signaturer, må tjeneren også ha klientenes sertifikater i sin "truststore".

En full oversikt over aktørene, deres sertifikater og deres betrodde sertifikater er gitt i tabell 6.4. Som man kan se må alle kraftprodusenter og tredjeparter ha Nubix 2.0 sitt sertifikat i sin "truststore". Nubix 2.0, på den andre siden, opererer som tjener for kraftprodusenter og tredjeparter, og må derfor ha deres sertifikater i sin "truststore" for å kunne autentisere dem, men må også ha nettselskapenes sertifikater i sin "truststore" både for å kunne opprette en HTTPS-tilkobling mot, og for å kunne verifisere deres signaturer. Til sist er det viktig at nettselskapene har Nubix 2.0 sitt sertifikat i sine "truststores" for å kunne verifisere dets digitale signatur. Jeg har brukt Java Keytool og Openssl for å lage nøkler og sertifikater. Jeg har brukt Keytool-kommandoen "genkey" for å lage "keystores" for hver aktør, Openssl for å lage tilhørende selvsignerte x509-sertifikater samt Keytool-kommandoen "import" for å importere tjenerens x509-sertifikater inn i klientenes "truststores". "Truststores" kan enten være separate filer, eller være integrert i klientens "keystore", som er metoden jeg har brukt for dette systemet. Ved bruk av egne "truststores" må man fortelle Java-applikasjonen hvordan den aksesseres, dette kan gjøres på følgende

Aktør	"Keystore"	Sertifikat	Betrodde sertifikater
Powerplant	powerplant.jks	powerplant.cert	nubix.cert
Waterfall	waterfall.jks	waterfall.cert	nubix.cert
Energysolutions	energysolutions.jks	energysolutions.cert	nubix.cert
Energysystems	energysystems.jks	energysystems.cert	nubix.cert
Nubix 2.0	nubix.jks	nubix.cert	powerplant.cert waterfall.cert energysolutions.cert energysystems.cert gridA.cert gridB.cert
GridA	gridA.jks	gridB.cert	nubix.cert
GridB	gridA.jks	gridB.cert	nubix.cert

Tabell 6.4: Oversikt over aktørenes nøkler, sertifikater og betrodde sertifikater

måte:

```
System.setProperty("javax.net.ssl.trustStore", "Path/to/truststore.jks");
System.setProperty("javax.net.ssl.trustStorePassword", "truststore_password");
```

6.1.2 Meldingsutveksling

Jeg vil nå beskrive meldingene som utveksles ved et kall på "GetReportedMeterData"-metoden fra kraftprodusenten "Powerplant". Av layouts- og lesbarhetsmessige årsaker har jeg hverken tatt med XML navnerom for feltene eller innholdet til hode-elementet "<wsse:Security>", jeg vil komme tilbake til sikkerhetsaspektet i seksjon 6.1.3. Meldingene i sin helhet er gitt i vedlegg B.

```
<soapenv:Envelope>
  <soapenv:Header>
    <wsse:Security>
      ...
```

```

</wsse:Security>
<wsa:To>https://localhost:8443/Nubix20/services/
  GetMeterDataService.
  GetMeterDataServiceHttpSoap11Endpoint</wsa:To>
<wsa:MessageID>urn:uuid:7a850930-a0a2-4321-a42e-41
  cad0383486</wsa:MessageID>
<wsa:Action>urn:getReportedMeterData</wsa:Action>
</soapenv:Header>
<soapenv:Body wsu:Id="Id-20812468">
  <ns2:getReportedMeterData xmlns:ns2="http://main">
    <ns2:meterID>87654321</ns2:meterID>
    <ns2:startTime>2011-11-01T00:00:00.000+01:00</
      ns2:startTime>
    <ns2:endTime>2011-11-01T01:59:59.000+01:00</
      ns2:endTime>
  </ns2:getReportedMeterData>
</soapenv:Body>
</soapenv:Envelope>

```

Forespørselen sendt fra klienten til Nubix 2.0 er gitt over, og er relativt lett-forståelig. ”<wsa:To>”-angir adressen til mottakeren, i Webtjenesteterminologi kalt ”EndpointReference”. Merk også elementet ”<wsa:MessageID>”, som inneholder en identifikator for akkurat denne meldingen. Meldingsnavn, ”GetReportedMeterData”, og meldingsparametere er, som man kan se, gitt i meldingskroppen. Som leseren kan se har meldingskroppen blitt merket med en ID, ”wsu_Id=Id-20812468”, årsaken blir tydeliggjort i seksjon 6.1.3. Starttid og sluttid er representert som et ”DateTime”-objekt i XML. Meldingen blir så behandlet av Nubix 2.0, og videresendt til korrekt nettselskap, det fiktive nettselskapet ”GridB”:

```

<soapenv:Envelope>
  <soapenv:Header>
    <wsse:Security>
    ...
  </wsse:Security>
  <wsa:To>https://localhost:8447/GridB/services/
    GetMeterDataService.
    GetMeterDataServiceHttpSoap11Endpoint</wsa:To>
  <wsa:MessageID>urn:uuid:6a71b244-a8ca-4f76-b4ba-2
    dc03800e871</wsa:MessageID>
  <wsa:Action>urn:getReportedMeterData</wsa:Action>
</soapenv:Header>
<soapenv:Body wsu:Id="Id-10601348">
  <ns2:getReportedMeterData xmlns:ns2="http://main">
    <ns2:meterID>87654321</ns2:meterID>

```

```

    <ns2:requestorID>1</ns2:requestorID>
    <ns2:requestorType>1</ns2:requestorType>
    <ns2:startTime>2011-11-01T00:00:00.000+01:00</
      ns2:startTime>
    <ns2:endTime>2011-11-01T01:59:59.000+01:00</
      ns2:endTime>
  </ns2:getReportedMeterData>
</soapenv:Body>
</soapenv:Envelope>

```

Feltene "requestorID" og "requestorType" har blitt lagt til, i tillegg til at adressen i elementet "<wsa:To>" har blitt endret. Nettselskapet, "GridB", autentiserer så Nubix 2.0, foretar en autorisasjonskontroll av klienten med ID "1", før et svar blir sendt tilbake til Nubix 2.0. Meldingene fra nettselskapet til Nubix 2.0 og fra Nubix 2.0 tilbake til klienten er tilnærmet identiske, så jeg viser kun den sistnevnte meldingen.

```

<soapenv:Envelope>
  <soapenv:Header>
    <wsse:Security>
      ...
    </wsse:Security>
    <wsa:Action>urn:getReportedMeterDataResponse</wsa:Action>
    <wsa:RelatesTo>urn:uuid:7a850930-a0a2-4321-a42e-41
      cad0383486</wsa:RelatesTo>
  </soapenv:Header>
  <soapenv:Body>
    <ns2:getReportedMeterDataResponse>
      <ns2:return>
        <ns1:meterID >87654321</ns1:meterID>
        <reportedData>
          <consumption>1669.657651765012</consumption>
          <endTime>2011-11-01T00:59:59.000+01:00</endTime>
          <startTime>2011-11-01T00:00:00.000+01:00</
            startTime>
        </reportedData>
        <reportedData>
          <consumption>1877.8715625789725</consumption>
          <endTime>2011-11-01T01:59:59.000+01:00</endTime>
          <startTime>2011-11-01T01:00:00.000+01:00</
            startTime>
        </reportedData>
      </ns2:return>
    </ns2:getReportedMeterDataResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

I hodet på meldingen har vi bl.a. elementet "<wsa:RelatesTo>". Dette elementet inneholder den samme IDen som ble sendt i elementet "<wsa:MessageID>" i forespørselsmeldingen. Ved å sammenligne dette feltet med IDen i den utgående meldingen, kan klienten foreta en korrekt kobling av forespørsel og svar. Kroppen på meldingen er også relativt lettforståelig, og samsvarer med XML-skjemadefinisjonen jeg beskrev i seksjon 5.3.3. Hos klientene i systemet har jeg implementert en enkel metode for å ta i mot dataene, samt skrive disse til en XML-fil.

6.1.3 Sikkerhet

Nedenfor vises innholdet til elementet "<wsse:Security>" til den første meldingen nevnt over, igjen har jeg valgt å korte ned enkelte felter av hensyn til lesbarheten. Dette elementet er tilnærmet identisk i samtlige meldinger som utveksles, så det har ingen hensikt å inkludere samtlige.

```
<wsse:Security soapenv:mustUnderstand="1">
  <wsu:Timestamp wsu:Id="Timestamp-1">
    <wsu:Created>2011-12-11T17:13:04.045Z</wsu:Created>
    <wsu:Expires>2011-12-11T17:18:04.045Z</wsu:Expires>
  </wsu:Timestamp>
  <ds:Signature Id="Signature-2">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm=".../xml-exc-
        c14n#" />
      <ds:SignatureMethod Algorithm=".../xmldsig#rsa-sha1"
        />
      <ds:Reference URI="#Id-20812468">
        <ds:Transforms>
          <ds:Transform ../xml-exc-c14n#" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm=".../xmldsig#sha1" />
        <ds:DigestValue>pC2ppUdEJhHsSTYEoSzviRMVwaY=</
          ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#Timestamp-1">
        <ds:Transforms>
          <ds:Transform Algorithm=".../xml-exc-c14n#" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm=".../xmldsig#sha1" />
        <ds:DigestValue>lLNjBfw9NkrmtqCjEI0ae1i1288=</
          ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
  </ds:Signature>
</wsse:Security>
```

```

</ds:SignedInfo>
<ds:SignatureValue>....</ds:SignatureValue>
<ds:KeyInfo Id="KeyId-98B9EFAAF3A946C0F013236235841592"
>
  <wsse:SecurityTokenReference>
    <wsse:KeyIdentifier>2aaEUpmJe3lWYY9jdA7+9BHlPog=
    </wsse:KeyIdentifier>
  </wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>

```

Det første som er verdt å merke seg er attributtet "mustUnderstand=1", som betyr at dette SOAP-hodeelementet **må** prosesseres av mottakeren. Videre er det første subelementet er et tidsstempel. Dette er inkludert for å sørge for at "gjentakelses"-angrep¹ blir vanskeliggjort ved at meldinger mottatt etter tidspunktet angitt i "Expires" ikke blir prosessert. Lengden på intervallet kan endres, fem minutter er standardverdien gitt av Rampart-biblioteket. For å verifisere tidsstempelets integritet er også meldingsekstraktet kalkulert av SHA1-funksjonen inkludert i hodet, dette ligger under feltet "Reference URI = "#Timestamp-1". Den faktiske signaturen kommer i feltet "SignatureValue", som jeg av layoutsmessige årsaker har endret til "...". Merk også elementet "KeyInfo", som brukes til å identifisere nøkkelen som er brukt til å lage signaturen. Jeg benytter verdien i "KeyIdentifier" for å koble forespørsel med rett aktør (kraftprodusent eller tredjepart). Hvis verifiseringen av signaturen feiler, vil Rampart automatisk returnere en feilmelding til klienten.

Som nevnt er det Rampart som tar seg av den faktiske verifiseringen av signaturen. Under har jeg gitt logg-meldingene som printes til konsollen i eclipse etter meldingen er mottatt.

```

[INFO] Verification successful for URI "\#Id-20812468"
[INFO] Verification successful for URI "\#Timestamp-1"

```

I verifiseringsbekreftelsene over kan man kjenne igjen feltet "Id-20812468" fra SOAP-kroppen i den første meldingen nevnt over, i tillegg til feltet "Timestamp-1" fra tidsstempelet i hodeelementet "wsse:security". Dette er altså beskjeden fra Rampart-implementasjonen om at de refererte signaturfeltene er blitt verifisert. Ved feil i verifisering

¹Oversatt fra det engelske uttrykket "Replay attack"

blir en feilmelding returnert. Under er et eksempel på et utdrag fra feilmeldingen som gis hvis klienten ikke er autorisert, altså at tjeneren ikke har klientens sertifikat i sin "truststore", feilmeldingen har jeg framprovosert ved å fjerne "powerplant.cert" fra "nubix.jks".

```
[ERROR] The signature or decryption was invalid
org.apache.axis2.AxisFault: The signature or decryption was
invalid
```

Når det gjelder HTTPS, så fungerer det også i henhold til spesifikasjonen. Jeg har fanget opp pakkestrømmen ved hjelp av Wireshark og kun chiffterkst blir sendt. Figur 6.1 viser et utkast av meldingsutvekslingene forbundet med etableringen av en HTTPS-sesjon mellom "Powerplant" og Nubix 2.0. Innholdet i pakkene som sendes inneholder kun chiffterkst. Hvis "transportBinding"-elementet er inkludert i policy-definisjonen, som spesifisert i seksjon 5.3.5, og klienten prøver å kommunisere over HTTP, blir følgende feilmelding gitt:

```
[ERROR] Expected transport is "https" but incoming transport
found : "http"
```

I koden som er vedlagt har jeg derimot kommentert ut "transportBinding"-elementet, dette har jeg gjort av testårsaker for å kunne studere meldingene som sendes i verktøyet Tcpmon.

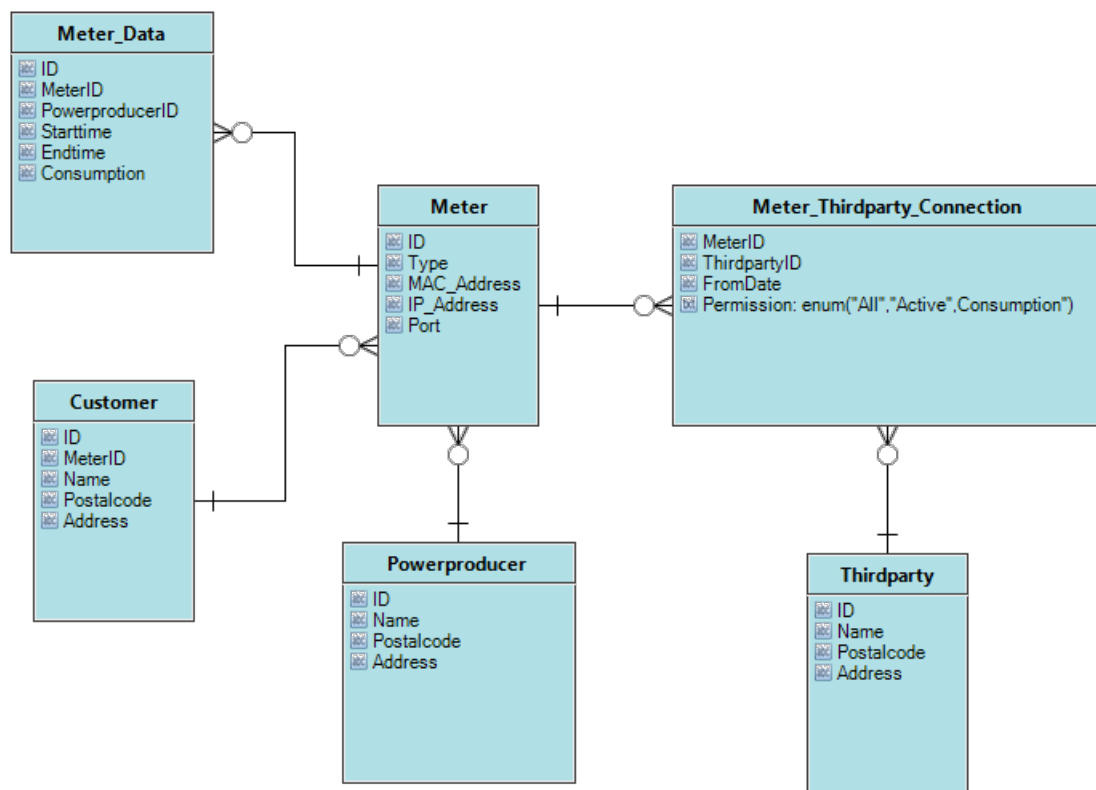
1	0.000000	127.0.0.1	127.0.0.1	TCP	52	52063	>	pcsync-https	[SYN]
2	0.001000	127.0.0.1	127.0.0.1	TCP	52	52063	>	pcsync-https	[SYN, ACK]
3	0.001000	127.0.0.1	127.0.0.1	TCP	40	52063	>	pcsync-https	[ACK]
6	0.126007	127.0.0.1	127.0.0.1	TCP	208	52063	>	pcsync-https	[PSH, ACK]
7	0.126007	127.0.0.1	127.0.0.1	TCP	40	52063	>	pcsync-https	[ACK]
8	0.239014	127.0.0.1	127.0.0.1	TCP	1500	52063	>	pcsync-https	[ACK]
9	0.239014	127.0.0.1	127.0.0.1	TCP	881	52063	>	pcsync-https	[PSH, ACK]
10	0.239014	127.0.0.1	127.0.0.1	TCP	40	52063	>	pcsync-https	[ACK]
13	0.290017	127.0.0.1	127.0.0.1	TCP	115	52063	>	pcsync-https	[PSH, ACK]
14	0.290017	127.0.0.1	127.0.0.1	TCP	40	52063	>	pcsync-https	[ACK]
17	0.341020	127.0.0.1	127.0.0.1	TCP	46	52063	>	pcsync-https	[PSH, ACK]
18	0.341020	127.0.0.1	127.0.0.1	TCP	40	52063	>	pcsync-https	[ACK]
21	0.343020	127.0.0.1	127.0.0.1	TCP	93	52063	>	pcsync-https	[PSH, ACK]
22	0.343020	127.0.0.1	127.0.0.1	TCP	40	52063	>	pcsync-https	[ACK]
25	0.346020	127.0.0.1	127.0.0.1	TCP	46	52063	>	pcsync-https	[PSH, ACK]
26	0.346020	127.0.0.1	127.0.0.1	TCP	40	52063	>	pcsync-https	[ACK]
27	0.347020	127.0.0.1	127.0.0.1	TCP	93	52063	>	pcsync-https	[PSH, ACK]

Figur 6.1: Oppsett av HTTPS-sesjonen

6.1.4 Oppsett hos nettselskapene

For å teste funksjonaliteten har det også blitt satt opp en database for hvert nettselskap, samt implementert funksjonalitet for å hente data ut av databasen. Selve implementering hos nettselskapene har ikke vært et fokus i denne oppgaven, så graden av funksjonalitet er kun på et tilstrekkelig nivå for testing. Selve databasedesignet, inspirert av arbeidet i [43], er illustrert i figur 6.2.

En unik måler har generert opp til flere sett med målerdata. Hver måling er identifisert med en autogenerated ID, og har i tillegg IDen til kraftprodusenten som har levert strømmen. I tillegg ser jeg for meg en mange-til-mange-relasjon mellom målere og tredjeparter; en måler kan ha et kundeforhold til mange tredjeparter, og en tredjepart kan ha et kundeforhold til mange målere. For dette forholdet har jeg inkludert tabellen "Meter_Thirdparty_Connection", som foruten IDen til måleren og tredjeparten inneholder datoen kundeforholdet ble opprettet, samt hvilke type data tredjeparten har lov til å lese; alt, kun aktivt forbruk eller kun rapportert forbruk. Databasene er som tidligere nevnt fylt med fiktive data, og kan brukes som en basis for videre arbeid.



Figur 6.2: Databasedesign hos "GridA" og "GridB"

6.2 Diskusjon

I dette avsnittet vil jeg gå inn på en diskusjon rundt de tekniske valgene som har blitt gjort, samt presentere temaer jeg ikke har prioritert i oppgaven.

6.2.1 Arkitektur

Nubix 2.0 baserer seg som nevnt på en videresending av forespørsler fra klienter til korrekt nettselskap. Det har blitt foreslått arkitekturer som baserer seg på en større grad av sentral lagring av målerdata, men bransjen er kritiske til hvordan dette skal forvaltes. Bransjeaktørene har derimot sine egne interesser å beskytte, og det er ikke helt utenkelig at det har farget høringssvarene de sendte til NVE. På den andre siden kan man heller ikke se bort ifra at konsulenthusene ansvarlige for rapporten jeg har brukt som grunnlag har økonomiske interesser forbundet med utviklingen av et større sentralisert system.

Det som er verdt å merke seg er at fra klientenes ståsted, dvs. kraftprodusenter og tredjeparter, vil Nubix 2.0 fremstå tilnærmet likt uavhengig om data ligger lagret hos nettselskapene eller sentralt. Klientene vil fortsatt kun ha ett kontaktpunkt for å lese av målerdata for alle målepunktene de har et kundeforhold til.

Metodene tilbudt av Nubix 2.0 krever at klientene kjenner til målepunktIDen til måleren de ønsker å lese data fra. Et spørsmål som da blir tydelig er hvordan klientene skal få kjennskap til denne IDen, da det som nevnt tidligere er en ID som sluttbrukere har relativt lite kjennskap til. Det eksisterende Nubix-systemet har derimot allerede funksjonalitet for å søke opp et målepunkt basert på kundens navn, fødselsnummer og postnummer, så det bør benyttes videre. En annen utfordring er derimot at Nubix 2.0 må ha en oversikt over samtlige målepunkt i Norge og hvilke nettselskap de tilhører. Med 2.6 millioner målepunkt i Norge og 8 bytes per målepunkt-ID, vil derimot ikke datamengden bli urovekkende stor², og ettersom det er å anta at målepunkt-IDene og deres tilhørende nettselskap ikke vil endre seg betraktelig over tid, ser ikke jeg på det som en uoverkommelig utfordring.

Jeg har også valgt å legge til en unik ID for hver klient, denne blir søkt opp basert på klientens nøkkelinformasjon og sendes til nettselskapenes for oppslag i deres databaser. Hvis dette skal være en unik ID kun for dette formålet må den forvaltes og distribueres

²2.6 millioner 8-bytes IDer gir totalt rett under 20 Megabyte

av Nubix 2.0, men en kan også se for seg å f.eks benytte klientens organisasjonsnummer her. Ettersom det er et langsiktig mål å etablere et felles europeisk sluttbrukermarked kan man også godta REID ³, som er et initiativ for å etablere en globalt gyldig ID for å identifisere foretak. Jeg la også til et flagg, "requestorType", som ble sendt med meldingene fra Nubix 2.0 til nettselskapene, dette var i utgangspunktet tenkt som en måte å plassere klienter inn i noen fastsatte grupper f.eks p.g.a. QoS—årsaker, men dette flagget kan like godt brukes til å fastsette hvilken type klient-ID som blir sendt.

6.2.2 Datastruktur

For dette systemet har det blitt definert en datastruktur etter kravene listet i seksjon 5.1.2. Datastrukturen ble definert med tanke på at den skulle inneholde minimalt med informasjon som sjeldent blir endret og ble basert på XML. XML er derimot, som vil bli vist i neste avsnitt, ikke spesielt effektivt med tanke på nyttelast per melding. For å øke effektivitetsgraden kunne man benyttet en bytebasert notasjon, f.eks med en bytesekvens for å skille mellom ulike sett målerdata, samt basert datastrukturen i større grad på de eksisterende Ediel-meldingene omtalt i seksjon 3.1.1. I tillegg har det ikke blitt definert hvilken benevning som sendes i "consumption"-elementet. En slik løsning innebærer at aktørene må bli enige om benevning på forhånd, f.eks Watt per time for rapportert forbruk og Watt for aktivt forbruk, men det er kanskje mer realistisk at datastrukturen inkluderer et flagg som spesifiserer nøyaktig hvilken benevning som blir benyttet.

6.2.3 Utnyttelsesgrad og krav til båndbredde

Man vet at XML og Webtjenester ikke er spesielt effektive mtp. nyttelast per melding. I hver av meldingene som sendes i Nubix 2.0 går f.eks 2405 bytes med til SOAP-hodeelementet. Tabell 6.5 gir en oversikt over størrelsen i bytes på nyttelasten til HTTP-meldingene som sendes til og fra Nubix 2.0 i forbindelse med noen kall på "GetReportedMeterData" fra kraftprodusenten "Powerplant". Kolonnen "måleverdier" viser antall timesdata som hentes ut. "Powerplant - Nubix 2.0" gir størrelsen i bytes på forespørselsmeldingen som kommer til Nubix 2.0 fra Powerplant, "Nubix 2.0 - GridB" gir størrelsen i bytes på meldingen som sendes fra Nubix 2.0 til "GridB" og de to neste

³Registered Entity Identifier

kolonnene gir størrelsen på hhv. meldingene som går fra "GridB" til Nubix 2.0 og fra Nubix 2.0 tilbake til "Powerplant". Merk at 720 er antall timeverdier per måler for november måned.

Måleverdier	Powerplant - Nubix 2.0	Nubix 2.0 - GridB	GridB - Nubix 2.0	Nubix 2.0 - Powerplant
1	3020	3093	3067	3067
2	3020	3093	3266	3266
3	3020	3093	3465	3465
24	3020	3093	7644	7644
720	3020	3093	146130	146130

Tabell 6.5: Meldingselementens størrelse i bytes

Som man kan se, så er størrelsen på forespørselsmeldingene uavhengig av antall målinger som forespørres, mens størrelsen på svaret øker selvsagt proporsjonalt med antall målinger som forespørres. Hvert "ReportedData"-element vil kreve ca. 200 bytes, litt avhengig av antall desimalers nøyaktighet i "Consumption"-feltet. I tillegg kommer feltet som inneholder målepunkt-IDen, "meterID", som med en 8-bytes målepunkt-ID krever totalt 64 bytes. Hvis vi da antar at nyttelasten per melding er samtlige "ReportedData"-elementer, samt "meterID"-elementet, kan den totale nyttelasten uttrykkes som:

$$\text{Nyttelast} = 64 + 200n$$

Der n uttrykker antall måleverdier.

Som nevnt inneholder hver av meldingene et hodeelement på ca 2400 bytes⁴, i tillegg kommer litt ekstra overhead i forbindelse med deklarerer av "SOAP envelope"- og "SOAP body"-elementene. Dette vil si at effektiviteten mtp. nyttelast per melding øker jo større tidsintervall man etterspør målerdata fra. Tabell 6.6 uttrykker utnyttelsesgraden som forholdet mellom nyttelast og totallast.

NVE har stadfestet at målerdata skal være tilgjengelig innen klokken 0900 neste virkedag, så for analysen om krav til båndbredde vil jeg anta at størsteparten av kallene på "GetReportedMeterData" vil spørre etter målinger for det siste døgnet.

Som man se i tabell 6.5, så vil hvert kall på måleverdier for det siste døgnet resultere i

⁴Avhenger bl.a. av lengden på Webtjenesteadressen til tjeneren

Måleverdier	Effektivitetsgrad
1	0.09
2	0.14
3	0.19
24	0.63
720	0.99

Tabell 6.6: Utnyttelsesgrad

totalt 10664 bytes ⁵ inngående trafikk til Nubix 2.0 og 10737 ⁶ bytes i utgående trafikk. Tabell 6.7 viser en sammenheng mellom antall kall på målerdata, total datatrafikk inn og ut i Megabyte, samt hva datatrafikken vil si mtp. båndbredde inn og ut i Megabit/s.

⁵3020 + 7644 bytes

⁶3096 + 7644 bytes

Antall kall	Data inn	Data ut	Båndbredde inn	Båndbredde ut
200 000	2132.8 MB	2147.4 MB	0.20 MBit/s	0.20 MBit/s
500 000	5332 MB	5368.5 MB	0.49 MBit/s	0.50 MBit/s
1 000 000	10664 MB	10737 MB	0.99 Mbit/s	0.99 Mbit/s
1 500 000	15996 MB	16105.5 MB	1.48 Mbit/s	1.49 Mbit/s
2 600 000	27726.4 MB	27916.2 MB	2.57 MBit/s	2.58 Mbit/s

Tabell 6.7: Krav til båndbredde

Hverken datamengden eller minimumskravet til båndbredde i seg selv er urovekkende høyt. Hvis alle målepunktene i Norge, 2.6 millioner, leses av en gang daglig vil det resultere i en total datatrafikk på 2.57 MBit/s inngående og 2.58 MBit/s utgående datatrafikk. Merk at disse tallene kun betegner nyttelasten i HTTP-meldingene som utveksles i forbindelse med "getReportedMeterData"-meldinger, og at "getActiveMeterData"-meldinger, samt ekstra overhead i forbindelse med HTTP-hodeelementer og oppsett av HTTPS-tilkobling, kommer i tillegg. Ettersom en betydelig andel av sluttbrukerne i Norge benytter kraftleverandør fra samme konsern som nettselskapet, er det derimot ikke sannsynlig at forbruk fra samtlige målere vil leses av hver dag. Det er derimot sannsynlig at forespørsler vil komme i store bolker, f.eks fordi systemene hos kraftleverandørene vil iterere over samtlige målere og sende et stort antall forespørsler ila. kort tid, det er derfor viktig at systemet dimensjoneres langt utover minimumskravet. Et av kravene i seksjon 5.1.1 var at systemet skal kunne håndtere et stort antall forespørsler. I Tomcat-konfigurasjonen kan man spesifisere antall tråder som maks skal benyttes av tjeneren. I systemet har jeg satt en øvre grense på 200 tråder, og testet med å sende mange forespørsler rett etter hverandre, men har ikke stresstestet systemet over en lengre periode.

6.2.4 Sikkerhet og nøkkelhåndtering

Nubix 2.0 benytter asymmetrisk kryptering både for å sette opp en HTTPS-tilkobling for å sende data kryptert og for å signere meldinger frem og tilbake for å sørge for autentisering og meldingsintegritet. Systemet i denne oppgaven er basert på selvsignerte sertifikater, men i en reell situasjon må en annen løsning benyttes. Et alternativ er at samtlige aktører

benytter sertifikater signert av en betrodd sertifikatautoritet, f.eks Verisign, og enten inkluderer dette sertifikatet i meldingene som sendes eller at de utveksles på en annen sikker måte. En slik løsning innebærer derimot en ekstra utgift for aktørene, da signerte sertifikater er relativt dyre produkter⁷, og vil på denne måten være en inngangsbarriere for små og innovative aktører.

En annen løsning er at Nubix-systemet selv kan signere sertifikatene til aktører som har blitt autorisert for å lese av målerdata. Dette innebærer at man ikke er avhengig av en fordyrende tredjepart, men at det må utvikles en sikker måte for autoriserte aktører å få signert sine sertifikater.

6.2.5 Roller og autorisasjoner

I testsystemene jeg satt opp hos nettselskapene "GridA" og "GridB" har jeg lagt til tre autorisasjonsnivåer for tredjeparter; fullt, aktivt eller rapportert forbruk. Med det menes hhv. at tredjepartene har tillatelse til å lese alle data, kun aktivt forbruk eller kun rapportert forbruk. En kan derimot se for seg at det vil bli etablert mer avanserte autorisasjonsnivåer. F.eks kan en se for seg at noen aktører kun har tillatelse til å lese månedsforbruk samt at enkelte aktører kun har tillatelse til å lese anonyme målerdata, altså ikke forbundet med en spesifikk måler, men f.eks samtlige målerdata fra alle målerne innenfor en region. Det siste eksempelet er særlig interessant for innsamling av statistikk. Mer avansert rolleinndeling fører derimot til mer komplekse systemer, og det bør gjennomføres en grundig analyse av nøyaktig hvilken funksjonalitet som er nødvendig for komplekse og kostbare prosjekter igangsettes.

6.2.6 Feilmeldinger

Webtjenester tillater også at man definerer skreddersydde feilmeldinger. Det har ikke blitt gjort i dette testoppsettet. Funksjonalitet som sørger for at klienter ikke får lese målerdata de ikke er autorisert til er blitt implementert hos både "GridA" og "GridB": Gitt at en klient ber om målerdata den ikke er autorisert for returnerer systemet kun null. I et reelt scenario burde det vært spesifisert et sett med standardiserte feilmeldinger for de scenariene som kan tenkes å oppstå, for eksempel at klienten ikke har autorisasjon til å lese

⁷3000 USD for et sertifikat gyldig i to år fra Verisign

målerdata, at klienten ikke har autorisasjon til å lese alle data innenfor gitt tidsintervall, og lignende.

6.2.7 Hva med sluttbrukerne?

En løsning som Nubix 2.0 vil ikke få noen direkte konsekvenser for sluttbrukerne, de vil fortsatt være priggitt nettselskapenes, kraftleverandørens og eventuelle tredjeparters nettsider og systemer. Ettersom det er nettopp det at sluttbrukere får tilgang til en detaljert oversikt over sitt strømforbruk som holdes som den største fordelen med AMS, er det viktig at det utvikles gode systemer for å gi feedback til sluttbrukerne. Ettersom nettselskapene er organisert som naturlige monopoler, har de relativt få incentiver for å gjøre investeringer i sine systemer rettet mot kunden, så det er viktig at det blir stilt klare krav til hva de må tilby kundene i form av feedback.

På den andre siden så vil system som Nubix 2.0 være portalen hvor kraftleverandører og tredjeparter henter sine data, så det er essensielt at systemet er stabilt og leverer etter gjeldende krav. For at det skal bli et marked for tredjepartstjenester til sluttbrukere basert på detaljerte målerdata og/eller momentanforbruk, er det kritisk at datagrunnlaget er på plass og at det er til å stole på. Uavhengig av hva som er årsaken til en eventuell feil, vil et ustabil og upålitelig system for deling av målerdata gå hardt ut over eventuelle kommersielle tredjeparter, da ingen sluttbrukere vil betale for noe som ikke fungerer.

6.3 Oppsummering

Dette kapitlet startet med å presentere testsystemet for Nubix 2.0, Webtjenesten jeg har definert for sikker deling av målerdata mellom nettselskaper og kraftleverandører/tredjeparter. Videre gikk jeg mer inn på en diskusjon rundt enkelte av de tekniske valgene som har blitt tatt, samt introduserte temaer jeg ikke har fokusert nevneverdig på.

7

KONKLUSJON

Det er per dags dato et krav at AMS skal være ferdig utrullet innen 1. januar 2017. Før den tid må store investeringer foretas, ikke bare i innkjøp og installering av smartmålere i de tusen hjem, men også i IKT-infrastrukturen for å håndtere en eksplosjon i tilgjengelige måleverdier.

I tillegg til at nettselskapene må utveksle målerdata med kraftprodusentene, er det også et krav fra NVE sin side om at målerdata skal bli gjort tilgjengelige for tredjeparter hvis sluttbruker ønsker det. Ettersom målerdata både er viktige av avregningsårsaker, samt kan anses som personsensitive data, er det viktig at det utvikles et sikkert og pålitelig system for utveksling. Fokuset i denne oppgaven har vært å se på ulike arkitekturer og teknologier som kan være aktuelle for et slikt system.

Uavhengig av valgt løsning på IKT-siden er det viktig å huske på hva som er det overordnede målet med AMS; Øke energieffektiviteten gjennom detaljert måling og bevisstgjøring av sluttbrukerne. Fra NVE sin side er det på den ene siden viktig å etablere gode incentivordninger for sluttbrukere, men også å stille strenge og hensiktsmessige krav til datakvalitet og datatilgjengelighet som nettselskapene må oppfylle.

Nubixbasert meldingsformidling

Oppgaven anbefaler en videreutvikling av dagens Nubix-løsning for deling av målerdata, og det har blitt spesifisert en rekke funksjonelle og ikke-funksjonelle krav til hvordan systemet bør se ut. Videre har det blitt utviklet et testgrensesnitt basert på Webtjenester med fiktive aktører i henhold til kravene. Webtjenester er en veldokumentert og mye anvendt teknologi som kan oppfylle kravene som har blitt definert, men det har også blitt

vist at Webtjenester har enkelte utfordringer mtp. effektivitetsgraden av nytteinformasjon i meldingsutvekslingen.

Veien videre

Gjennom arbeidet i denne oppgaven har det dukket opp flere problemstillinger det kunne vært interessant å se nærmere på. For det første ville det vært interessant å gå inn i en skalerbarhets- og ytelsesanalyse av systemet som har blitt spesifisert. For at systemet skal bli vellykket er det nødt til å kunne håndtere et stort antall forespørsler, hvilke krav må stilles til maskinvare- og nettinfrastruktur?

Videre ville en grundigere analyse av sikkerhetsutfordringene være interessant. Hvilken sikkerhetsnivå bør man legge seg på, og er det svakheter forbundet med WTS som kan bli utnyttet? Det hadde også vært spennende å gjennomføre et mindre pilotprosjekt, basert på denne oppgaven og oppgaven til Per-Kristian Helland og Åshild Thorrud [43]. Det forutsetter derimot at man får til kommunikasjon med en smartmåler, samt finne en produsent som er villig til å dele tekniske dokumenter som beskriver kommunikasjonen med måleren. Avslutningsvis hadde det også vært interessant å se på et av premissene for denne oppgaven; nemlig om det vil være et rimeligere alternativ å benytte eksisterende nettinfrastruktur for kommunikasjon med smartmåleren. Selv om et PLC-basert alternativ vil medføre investeringer med tanke på utbygging av selve infrastrukturen, ville det vært interessant å sammenligne denne investeringen med kostnadene og arbeidet forbundet med installasjonen og konfigurasjonen av smartmålere som skal være kompatible med sluttbrukernes eksisterende utstyr.

BIBLIOGRAFI

- [1] <http://www.energinet.dk/DA/El/Nyheder/Sider/Dataregister-for-elmarkedets-akt%C3%B8rer-et-vigtigt-skridt-n%C3%A6rmere.aspx>.
- [2] Bruker mest strøm på Østlandet. <http://www.vg.no/dinepenger/artikkel.php?artid=10012207>.
- [3] Brukerveiledninger & Internformat. <http://www.ediel.no/EdielPortal/showDocument.aspx?url=http://www.edisys.no/ediel/Brukerveiledninger%20og%20Internformat.htm>.
- [4] DataHub. <http://www.energinet.dk/DA/El/DataHub/Sider/DataHub.aspx>.
- [5] Fortum - hva er en standardprofil? <http://www.fortum.no/no/Strom/Kundeservice/Maleravlesning/Hva-er-en-standardprofil/>.
- [6] How Stuff Works: How Encryption Works. <http://computer.howstuffworks.com/encryption4.htm>.
- [7] Hva er EDI? <http://www.evenex.no/Leksikon/>.
- [8] Høyere strømpriser i 2010. <http://www.ssb.no/vis/emner/10/08/10/elkraftpris/arkiv/art-2011-01-11-01.html>.
- [9] Økonomisk regulering av nettselskap. <http://www.nve.no/no/Kraftmarked/Regulering-av-nettselskapene/>.
- [10] Målepunkt ID. <http://www.bkk.no/privat/nettleie/malepunkt-id/>.
- [11] Monopol på transport av strøm. <http://www.statnett.no/no/Om-Statnett/Statnett-pa-1-2-3/Monopol-pa-transport-av-strom/>.

- [12] Nordisk marked, nord pool. <http://www.kraftkartet.no/default.asp?gruppe=3306>.
- [13] Nubix - Webtjeneste for Oppslag av MålepunktIT. <http://www.ediel.no/dokumenter/Nubix/nubix.htm>.
- [14] Nves nettside. <http://nve.no/AMS>.
- [15] Rekordhøyt energiforbruk. <http://www.ssb.no/energiregn/>.
- [16] Sentralnett, regionalnett, lokalnett. <http://www.statnett.no/no/0m-Statnett/Statnett-pa-1-2-3/Sentralnett-regionalnett-lokalnett/>.
- [17] Setting Up Keystores for a Client and a Service. <http://wso2.org/library/174>.
- [18] Setting up WS-Security with Rampart module in Apache Axis web services framework. <http://krangsquared.blogspot.com/2010/03/setting-up-ws-security-in-axis.html>.
- [19] Statnett på 1-2-3. <http://www.statnett.no/no/0m-Statnett/Statnett-pa-1-2-3/>.
- [20] Web Service Security with Apache Rampart - Part 2 (Message-Level Security). <http://ws02.org/library/3415>.
- [21] Webservice: Apache Axis2. <http://axis.apache.org/axis2/java/core/>.
- [22] Webservice: Apache Rampart. <http://axis.apache.org/axis2/java/rampart/>.
- [23] Webservice: Apache Tomcat. <http://tomcat.apache.org/>.
- [24] WhitePaper. Web services: Benefits, challenges, and a unique, visual development solution. <http://www.altova.com/whitepapers/webservices.pdf>.
- [25] Wikipedia: Transport Layer Security. http://no.wikipedia.org/wiki/Transport_Layer_Security.
- [26] WSDL Examples. <http://www.w3.org/2001/03/14-annotated-WSDL-examples>.

-
- [27] Norsk brukerveiledning for implementering av PRODAT-meldingen. www.edisys.no/ediel/brukerveil/PRODAT_18_A.doc, Juli 2003.
- [28] Web Services Architecture. <http://www.w3.org/TR/ws-arch/>, Februar 2004.
- [29] Nye måleteknologier. http://nve.no/PageFiles/808/ECON_Ny_maalerteknologier.pdf?epslanguage=no, 2007.
- [30] Avanserte måle-og styringssystem (AMS). <http://www.nve.no/PageFiles/808/AMS%20siste%20versjon.pdf?epslanguage=no>, Desember 2009.
- [31] Avanserte måle-og styringssystem (AMS). Oppsummering av høringsuttalelser og endelig forskriftstekst. <http://www.nve.no/PageFiles/808/dokument1-11-oppsummering.pdf>, Juli 2011.
- [32] Du kan spare 45 kroner i året. <http://www.tu.no/energi/article276337.ece>, Februar 2011.
- [33] Felles IKT-løsninger i det norske kraftmarkedet. <http://www.nve.no/Global/Publikasjoner/Publikasjoner%202011/Diverse%202011/Felles%20%20IKT-1%C3%B8sninger%20i%20det%20norske%20kraftmakedet.pdf>, April 2011.
- [34] Anoop Singhal, Theodore Winograd, Karen Scarfone. Guide to Secure Web Services. <http://csrc.nist.gov/publications/nistpubs/800-95/SP800-95.pdf>, August 2007.
- [35] Brian Suda. SOAP Web Services. <http://suda.co.uk/publications/MSc/brian.suda.thesis.pdf>, 2003.
- [36] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana. Unraveling the web services web: an introduction to soap, wsdl, and uddi. mar/apr 2002.
- [37] Kristian Engan. Bruksområde for data registrert i MV/LV nettstasjon for bruk i distribusjonsnett. <http://www.sintef.no/project/M-AMS/Ferdige%20dokumenter/Kristian%20Engan%20-%20Masteroppgave%20A4.pdf>, Juni 2010.

- [38] Christian Haugen. Vurdering av kommunikasjonsalternativer for informasjonsutveksling med AMS mellom smarte hus og et smart kraftnett. http://www.sintef.no/project/M-AMS/Ferdige%20dokumenter/Masteroppgave_Kommunikasjonsalternativer%20AMS.pdf, Juni 2010.
- [39] J. Stromback, C. Dromacque, M. H. Yassin. The potential of smart meter enabled programs to increase energy and systems efficiency: a mass pilot comparison. <http://www.esmig.eu/press/filestor/empower-demand-report.pdf>, 2011.
- [40] Josco Kester and Maria Jose Gonzales Burgos and John Parsons. The Effectiveness of Feedback on Energy Consumption. <http://www.eci.ox.ac.uk/research/energy/downloads/smart-metering-report.pdf>, April 2006.
- [41] Josco Kester and Maria Jose Gonzales Burgos and John Parsons. Smart Metering Guide. Energy Saving and the Customer. <http://www.ecn.nl/docs/library/report/2011/o11004.pdf>, Januar 2011.
- [42] Lars Olav Fosse. Forretningsprosesser i kraftmarkedet. Forslag til meldingsutveksling og internettbasert søketjeneste administrert av Statnett. http://www.sunndalenergi.no/Pdf/dokument3_07.pdf, Mars 2007.
- [43] Per Kristian Helland and Åshild Thorrud. Design av løsning for automatisk måleravlesning av strøm (AMS) basert på Wi-Fi, Juni 2011.
- [44] PNNL. The Smart Grid: An Estimation of the Energy and CO2 Benefits. http://energyenvironment.pnnl.gov/news/pdf/PNNL-19112_Revision_1_Final.pdf, Januar 2010.
- [45] Prof. Do van Thanh. XML Web Services. http://www.item.ntnu.no/_media/academics/courses/ttm4128/lectures/08-xml_web_services.pdf.



WSDL-DEFINISJONER

Under kommer WSDL-definisjon til Nubix 2.0 rettet mot kraftprodusenter og tredjeparter.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://main"
  xmlns:ns1="http://org.apache.axis2/xsd" xmlns:wsp="http://
  schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:ns="http://main" xmlns:wsaw="http://www.w3.org/2006/05/
  addressing/wsdl"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns:http="
  http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:ax21="http://logic/xsd" xmlns:xs="http://www.w3.org
  /2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:mime=
  "http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/">
<wsdl:documentation>Please Type your service description here
</wsdl:documentation>
<wsdl:types>
  <xs:schema attributeFormDefault="qualified"
    elementFormDefault="qualified" targetNamespace="http://
    logic/xsd">
    <xs:complexType name="MeterData">
      <xs:sequence>
        <xs:element minOccurs="0" name="meterID" nillable="
          true"
          type="xs:string" />
        <xs:element maxOccurs="unbounded" minOccurs="0"
          name="reportedData" nillable="true" type="
          ax21:ReportedData" />
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ReportedData">
      <xs:sequence>
        <xs:element minOccurs="0" name="consumption" type="
          xs:double" />
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
</wsdl:types>
```

```
<xs:element minOccurs="0" name="endTime" nillable="
  true"
  type="xs:dateTime" />
<xs:element minOccurs="0" name="startTime" nillable="
  true"
  type="xs:dateTime" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="ActiveMeterData">
  <xs:sequence>
    <xs:element minOccurs="0" name="activeConsumption"
      type="xs:double" />
    <xs:element minOccurs="0" name="meterID" nillable="
      true"
      type="xs:string" />
    <xs:element minOccurs="0" name="timeStamp" nillable="
      true"
      type="xs:dateTime" />
  </xs:sequence>
</xs:complexType>
</xs:schema>
<xs:schema attributeFormDefault="qualified"
  elementFormDefault="qualified" targetNamespace="http://
  main"
  xmlns:ax22="http://logic/xsd">
  <xs:import namespace="http://logic/xsd" />
  <xs:element name="getReportedMeterData">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="meterID" nillable="
          true"
          type="xs:string" />
        <xs:element minOccurs="0" name="startTime" nillable="
          true"
          type="xs:dateTime" />
        <xs:element minOccurs="0" name="endTime" nillable="
          true"
          type="xs:dateTime" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="getReportedMeterDataResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="return" nillable="
          true"
          type="ax21:MeterData" />
      </xs:sequence>
    </xs:complexType>
```

```

</xs:element>
<xs:element name="getActiveMeterData">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="meterID" nillable="
        true"
        type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="getActiveMeterDataResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" name="return" nillable="
        true"
        type="ax21:ActiveMeterData" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="getActiveMeterDataResponse">
  <wsdl:part name="parameters" element="
    ns:getActiveMeterDataResponse">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="getReportedMeterDataResponse">
  <wsdl:part name="parameters" element="
    ns:getReportedMeterDataResponse">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="getActiveMeterDataRequest">
  <wsdl:part name="parameters" element="ns:getActiveMeterData"
  >
  </wsdl:part>
</wsdl:message>
<wsdl:message name="getReportedMeterDataRequest">
  <wsdl:part name="parameters" element="
    ns:getReportedMeterData">
  </wsdl:part>
</wsdl:message>
<wsdl:portType name="GetMeterDataServicePortType"
  wsp:PolicyURIs="#SigOverHTTPS">
  <wsdl:operation name="getActiveMeterData">
    <wsdl:input message="ns:getActiveMeterDataRequest"
      wsaw:Action="urn:getActiveMeterData">
    </wsdl:input>
    <wsdl:output message="ns:getActiveMeterDataResponse"
      wsaw:Action="urn:getActiveMeterDataResponse">

```

```
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="getReportedMeterData">
  <wsdl:input message="ns:getReportedMeterDataRequest"
    wsaw:Action="urn:getReportedMeterData">
    </wsdl:input>
    <wsdl:output message="ns:getReportedMeterDataResponse"
      wsaw:Action="urn:getReportedMeterDataResponse">
      </wsdl:output>
    </wsdl:operation>
  </wsdl:portType>
<wsdl:binding name="GetMeterDataServiceSoap11Binding"
  type="ns:GetMeterDataServicePortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="getActiveMeterData">
    <soap:operation soapAction="urn:getActiveMeterData"
      style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getReportedMeterData">
    <soap:operation soapAction="urn:getReportedMeterData"
      style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="GetMeterDataService">
  <wsdl:port name="GetMeterDataServiceHttpSoap11Endpoint"
    binding="ns:GetMeterDataServiceSoap11Binding">
    <soap:address
      location="http://localhost:8080/Nubix20/services/
        GetMeterDataService.
        GetMeterDataServiceHttpSoap11Endpoint/" />
  </wsdl:port>
  <wsdl:port name="GetMeterDataServiceHttpsSoap11Endpoint"
    binding="ns:GetMeterDataServiceSoap11Binding">
    <soap12:address
      location="https://localhost:8443/Nubix20/services/
        GetMeterDataService."/
```

```

        GetMeterDataServiceHttpsSoap11Endpoint/" />
    </wsdl:port>
</wsdl:service>
<wsp:Policy wsu:Id="SigOverHTTPS"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
  <wsp:ExactlyOne>
    <wsp:All>
      <sp:Wss10 xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
        <sp:Policy>
          <sp:MustSupportRefKeyIdentifier />
          <sp:MustSupportRefIssuerSerial />
        </sp:Policy>
      </sp:Wss10>
      <sp:TransportBinding
        xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
        <wsp:Policy>
          <sp:TransportToken>
            <wsp:Policy>
              <sp:HttpsToken RequireClientCertificate="false" />
            </wsp:Policy>
          </sp:TransportToken>
          <sp:AlgorithmSuite>
            <wsp:Policy>
              <sp:Basic256 />
            </wsp:Policy>
          </sp:AlgorithmSuite>
          <sp:Layout>
            <wsp:Policy>
              <sp:Lax />
            </wsp:Policy>
          </sp:Layout>
          <sp:IncludeTimestamp />
        </wsp:Policy>
      </sp:TransportBinding>
      <sp:AsymmetricBinding
        xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
        <wsp:Policy>
          <sp:InitiatorToken>
            <wsp:Policy>
              <sp:X509Token
                sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/Never"
              >

```

```

        <wsp:Policy>
            <sp:WssX509V3Token10 />
        </wsp:Policy>
    </sp:X509Token>
</wsp:Policy>
</sp:InitiatorToken>
<sp:RecipientToken>
    <wsp:Policy>
        <sp:X509Token
            sp:IncludeToken="http://schemas.xmlsoap.org/ws
                /2005/07/securitypolicy/IncludeToken/Never"
            >
            <wsp:Policy>
                <sp:WssX509V3Token10 />
            </wsp:Policy>
        </sp:X509Token>
    </wsp:Policy>
</sp:RecipientToken>
<sp:AlgorithmSuite>
    <wsp:Policy>
        <sp:Basic256 />
    </wsp:Policy>
</sp:AlgorithmSuite>
<sp:Layout>
    <wsp:Policy>
        <sp:Strict />
    </wsp:Policy>
</sp:Layout>
    <sp:IncludeTimestamp />
    <sp:OnlySignEntireHeadersAndBody />
</wsp:Policy>
</sp:AsymmetricBinding>
<sp:SignedParts
    xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/
        securitypolicy">
    <sp:Body />
</sp:SignedParts>
</wsp>All>
</wsp:ExactlyOne>
</wsp:Policy>
</wsdl:definitions>

```

Nå følger WSDL-definisjonen som nettselskapene må implementere. Merk at de faktiske adressefeltene må endres for å samsvare med hvert nettselskaps faktiske Web-adresse. Eksempelet gitt under er gyldig for "GridA".

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

```

<wsdl:definitions targetNamespace="http://main"
  xmlns:ns1="http://org.apache.axis2/xsd" xmlns:wsp="http://
  schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:ns="http://main" xmlns:wsaw="http://www.w3.org/2006/05/
  addressing/wSDL"
  xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/" xmlns:http="
  http://schemas.xmlsoap.org/wSDL/http/"
  xmlns:ax21="http://logic/xsd" xmlns:xs="http://www.w3.org
  /2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wSDL/soap/" xmlns:mime=
  "http://schemas.xmlsoap.org/wSDL/mime/"
  xmlns:soap12="http://schemas.xmlsoap.org/wSDL/soap12/">
<wsdl:documentation>Please Type your service description here
</wsdl:documentation>
<wsdl:types>
  <xs:schema attributeFormDefault="qualified"
    elementFormDefault="qualified" targetNamespace="http://
    logic/xsd">
    <xs:complexType name="MeterData">
      <xs:sequence>
        <xs:element minOccurs="0" name="meterID" nillable="
          true"
          type="xs:string" />
        <xs:element maxOccurs="unbounded" minOccurs="0"
          name="reportedData" nillable="true" type="
          ax21:ReportedData" />
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ReportedData">
      <xs:sequence>
        <xs:element minOccurs="0" name="consumption" type="
          xs:double" />
        <xs:element minOccurs="0" name="endTime" nillable="
          true"
          type="xs:dateTime" />
        <xs:element minOccurs="0" name="startTime" nillable="
          true"
          type="xs:dateTime" />
      </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ActiveMeterData">
      <xs:sequence>
        <xs:element minOccurs="0" name="activeConsumption"
          type="xs:double" />
        <xs:element minOccurs="0" name="meterID" nillable="
          true"
          type="xs:string" />
        <xs:element minOccurs="0" name="timeStamp" nillable="
          true"

```

```

        type="xs:dateTime" />
    </xs:sequence>
</xs:complexType>
</xs:schema>
<xs:schema attributeFormDefault="qualified"
  elementFormDefault="qualified" targetNamespace="http://
  main">
  <xs:import namespace="http://logic/xsd" />
  <xs:element name="getReportedMeterData">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="meterID" nillable="
          true"
          type="xs:string" />
        <xs:element minOccurs="0" name="requestorID"
          nillable="true"
          type="xs:string" />
        <xs:element minOccurs="0" name="requestorType" type="
          xs:int" />
        <xs:element minOccurs="0" name="startTime" nillable="
          true"
          type="xs:dateTime" />
        <xs:element minOccurs="0" name="endTime" nillable="
          true"
          type="xs:dateTime" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="getReportedMeterDataResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="return" nillable="
          true"
          type="ax21:MeterData" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="getActiveMeterData">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="meterID" nillable="
          true"
          type="xs:string" />
        <xs:element minOccurs="0" name="requestorID"
          nillable="true"
          type="xs:string" />
        <xs:element minOccurs="0" name="requestorType" type="
          xs:int" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```



```

    </xs:complexType>
  </xs:element>
  <xs:element name="getActiveMeterDataResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="return" nillable="
          true"
          type="ax21:ActiveMeterData" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="getActiveMeterDataResponse">
  <wsdl:part name="parameters" element="
    ns:getActiveMeterDataResponse">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="getReportedMeterDataResponse">
  <wsdl:part name="parameters" element="
    ns:getReportedMeterDataResponse">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="getActiveMeterDataRequest">
  <wsdl:part name="parameters" element="ns:getActiveMeterData"
  >
  </wsdl:part>
</wsdl:message>
<wsdl:message name="getReportedMeterDataRequest">
  <wsdl:part name="parameters" element="
    ns:getReportedMeterData">
  </wsdl:part>
</wsdl:message>
<wsdl:portType name="GetMeterDataServicePortType"
  wsp:PolicyURIs="#SigOnly">
  <wsdl:operation name="getActiveMeterData">
    <wsdl:input message="ns:getActiveMeterDataRequest"
      wsaw:Action="urn:getActiveMeterData">
    </wsdl:input>
    <wsdl:output message="ns:getActiveMeterDataResponse"
      wsaw:Action="urn:getActiveMeterDataResponse">
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getReportedMeterData">
    <wsdl:input message="ns:getReportedMeterDataRequest"
      wsaw:Action="urn:getReportedMeterData">
    </wsdl:input>
    <wsdl:output message="ns:getReportedMeterDataResponse"
      wsaw:Action="urn:getReportedMeterDataResponse">
  </wsdl:operation>

```

```
        </wsdl:output>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="GetMeterDataServiceSoap11Binding"
    type="ns:GetMeterDataServicePortType">
    <soap:binding style="document"
        transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="getActiveMeterData">
        <soap:operation soapAction="urn:getActiveMeterData"
            style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getReportedMeterData">
        <soap:operation soapAction="urn:getReportedMeterData"
            style="document" />
        <wsdl:input>
            <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="GetMeterDataServiceHttpBinding" type="
    ns:GetMeterDataServicePortType">
    <http:binding verb="POST" />
    <wsdl:operation name="getActiveMeterData">
        <http:operation location="getActiveMeterData" />
        <wsdl:input>
            <mime:content part="parameters" type="text/xml" />
        </wsdl:input>
        <wsdl:output>
            <mime:content part="parameters" type="text/xml" />
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="getReportedMeterData">
        <http:operation location="getReportedMeterData" />
        <wsdl:input>
            <mime:content part="parameters" type="text/xml" />
        </wsdl:input>
        <wsdl:output>
            <mime:content part="parameters" type="text/xml" />
        </wsdl:output>
    </wsdl:operation>
</wsdl:operation>
```

```

</wsdl:binding>
<wsdl:binding name="GetMeterDataServiceSoap12Binding"
  type="ns:GetMeterDataServicePortType">
  <soap12:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="getActiveMeterData">
    <soap12:operation soapAction="urn:getActiveMeterData"
      style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getReportedMeterData">
    <soap12:operation soapAction="urn:getReportedMeterData"
      style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="GetMeterDataService">
  <wsdl:port name="GetMeterDataServiceHttpSoap11Endpoint"
    binding="ns:GetMeterDataServiceSoap11Binding">
    <soap:address
      location="http://localhost:8090/GridA/services/
        GetMeterDataService.
        GetMeterDataServiceHttpSoap11Endpoint/" />
  </wsdl:port>
  <wsdl:port name="GetMeterDataServiceHttpSoap12Endpoint"
    binding="ns:GetMeterDataServiceSoap12Binding">
    <soap12:address
      location="http://localhost:8090/GridA/services/
        GetMeterDataService.
        GetMeterDataServiceHttpSoap12Endpoint/" />
  </wsdl:port>
  <wsdl:port name="GetMeterDataServiceHttpEndpoint" binding="
    ns:GetMeterDataServiceHttpBinding">
    <http:address
      location="http://localhost:8090/GridA/services/
        GetMeterDataService.GetMeterDataServiceHttpEndpoint/"
      />
  </wsdl:port>
  <wsdl:port name="GetMeterDataServiceHttpsSoap12Endpoint"

```

```

        binding="ns:GetMeterDataServiceSoap12Binding">
        <soap12:address
            location="https://localhost:8445/GridA/services/
                GetMeterDataService.
                GetMeterDataServiceHttpsSoap12Endpoint/" />
    </wsdl:port>
    <wsdl:port name="GetMeterDataServiceHttpsSoap11Endpoint"
        binding="ns:GetMeterDataServiceSoap11Binding">
        <soap:address
            location="https://localhost:8445/GridA/services/
                GetMeterDataService.
                GetMeterDataServiceHttpsSoap11Endpoint/" />
    </wsdl:port>
    <wsdl:port name="GetMeterDataServiceHttpsEndpoint" binding="
        ns:GetMeterDataServiceHttpBinding">
        <http:address
            location="https://localhost:8445/GridA/services/
                GetMeterDataService.GetMeterDataServiceHttpsEndpoint/
                " />
    </wsdl:port>
</wsdl:service>
<wsp:Policy wsu:Id="SigOnly"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis
        -200401-wss-wssecurity-utility-1.0.xsd">
    <wsp:ExactlyOne>
        <wsp:All>
            <sp:Wss10 xmlns:sp="http://schemas.xmlsoap.org/ws
                /2005/07/securitypolicy">
                <sp:Policy>
                    <sp:MustSupportRefKeyIdentifier />
                    <sp:MustSupportRefIssuerSerial />
                </sp:Policy>
            </sp:Wss10>
            <sp:TransportBinding
                xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/
                    securitypolicy">
                <wsp:Policy>
                    <sp:TransportToken>
                        <wsp:Policy>
                            <sp:HttpsToken RequireClientCertificate="false"
                                />
                        </wsp:Policy>
                    </sp:TransportToken>
                    <sp:AlgorithmSuite>
                        <wsp:Policy>
                            <sp:Basic256 />
                        </wsp:Policy>
                    </sp:AlgorithmSuite>
                </wsp:Policy>
            </sp:TransportBinding>
        </wsp:All>
    </wsp:ExactlyOne>
</wsp:Policy>

```

```
<sp:Layout>
  <wsp:Policy>
    <sp:Lax />
  </wsp:Policy>
</sp:Layout>
<sp:IncludeTimestamp />
</wsp:Policy>
</sp:TransportBinding>
<sp:AsymmetricBinding
  xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/
  securitypolicy">
  <wsp:Policy>
    <sp:InitiatorToken>
      <wsp:Policy>
        <sp:X509Token
          sp:IncludeToken="http://schemas.xmlsoap.org/ws
            /2005/07/securitypolicy/IncludeToken/Never"
          >
          <wsp:Policy>
            <sp:WssX509V3Token10 />
          </wsp:Policy>
        </sp:X509Token>
      </wsp:Policy>
    </sp:InitiatorToken>
    <sp:RecipientToken>
      <wsp:Policy>
        <sp:X509Token
          sp:IncludeToken="http://schemas.xmlsoap.org/ws
            /2005/07/securitypolicy/IncludeToken/Never"
          >
          <wsp:Policy>
            <sp:WssX509V3Token10 />
          </wsp:Policy>
        </sp:X509Token>
      </wsp:Policy>
    </sp:RecipientToken>
    <sp:AlgorithmSuite>
      <wsp:Policy>
        <sp:Basic256 />
      </wsp:Policy>
    </sp:AlgorithmSuite>
  <sp:Layout>
    <wsp:Policy>
      <sp:Strict />
    </wsp:Policy>
  </sp:Layout>
  <sp:IncludeTimestamp />
  <sp:OnlySignEntireHeadersAndBody />
</wsp:Policy>
```

```
</sp:AsymmetricBinding>
<sp:SignedParts
  xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/
    securitypolicy">
  <sp:Body />
</sp:SignedParts>
</wsp:All>
</wsp:ExactlyOne>
</wsp:Policy>
</wsdl:definitions>
```

B

MELDINGSUTVEKSLING

Dette vedlegget viser SOAP-meldingene forbundet med et kall på "GetReportedData"-metoden fra kraftprodusenten "Powerplant" på målepunktet "87654321". Først, forespørselen fra "Powerplant":

```
<?xml version='1.0' encoding='UTF-8'?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
    soap/envelope/">
    <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/
      addressing">
      <wsse:Security xmlns:wsse="http://docs.oasis-open.org/
        wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.
        xsd" soapenv:mustUnderstand="1">
        <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org
          /wss/2004/01/oasis-200401-wss-wssecurity-utility
          -1.0.xsd" wsu:Id="Timestamp-1">
          <wsu:Created>2011-12-11T17:13:04.045Z</
            wsu:Created>
          <wsu:Expires>2011-12-11T17:18:04.045Z</
            wsu:Expires>
        </wsu:Timestamp>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/
          xmldsig#" Id="Signature-2">
          <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://
              www.w3.org/2001/10/xml-exc-c14n#" />
            <ds:SignatureMethod Algorithm="http://www.w3.
              org/2000/09/xmldsig#rsa-sha1" />
            <ds:Reference URI="#Id-20812468">
              <ds:Transforms>
                <ds:Transform Algorithm="http://www.w3.
                  org/2001/10/xml-exc-c14n#" />
              </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.
              org/2000/09/xmldsig#sha1" />
          </ds:Reference>
        </ds:SignedInfo>
      </ds:Signature>
    </soapenv:Header>
  </soapenv:Envelope>
```

```

        <ds:DigestValue>pC2ppUdEJhHsSTYEoSzviRMVwaY
          =</ds:DigestValue>
    </ds:Reference>
    <ds:Reference URI="#Timestamp-1">
      <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.
          org/2001/10/xml-exc-c14n#" />
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.
        org/2000/09/xmldsig#sha1" />
      <ds:DigestValue>lLNjBfw9NkrmtqCjEI0ae1i1288
        =</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>
  LC6JVgA8wK0jTd648OK/
  e56reNvsMLjsEbo8C+v/sLweQvSyp2NG
  2zjpktm+HHEuscsc57mO6MaKancZ rGzH
  SfWe5DNg+LUV5CIeBmJeukFzmeRPu1Ap
  p9WUnKWpGKxq/uQTUOzdBn/NBxGOJZFc/gk
  5m6ttUo45/M8H3rDrfhDcWdYOLiHWQSm
  UatnCVz1DfaxoFMX7xwsivWLH
  7FpeRoTiSawvJqjOAYePFOIK9KyRTXs
  SixwaITNHZF1KNvEkhtSw2vOC82EWSaP3
  1WVij9EG0uM754GZsJU00xFq//BF4grx
  OHfVD8fNnlu/x+cMUI+2rMnTZkl5mvMT6
  wjRA==
  </ds:SignatureValue>
  <ds:KeyInfo Id="KeyId-98
    B9EFAAF3A946C0F013236235841592">
    <wsse:SecurityTokenReference xmlns:wsu="http:
      //docs.oasis-open.org/wss/2004/01/oasis
      -200401-wss-wssecurity-utility-1.0.xsd"
      wsu:Id="STRId-98
      B9EFAAF3A946C0F013236235841643">
      <wsse:KeyIdentifier EncodingType="http://
        docs.oasis-open.org/wss/2004/01/oasis
        -200401-wss-soap-message-security-1.0#
        Base64Binary" ValueType="http://docs.
        oasis-open.org/wss/2004/01/oasis-200401-
        wss-x509-token-profile-1.0#
        X509SubjectKeyIdentifier">2
        aaEUpmJe3lWYY9jdA7+9BHlPog=</
        wsse:KeyIdentifier>
    </wsse:SecurityTokenReference>
  </ds:KeyInfo>
</ds:Signature>
</wsse:Security>
<wsa:To>http://localhost:8079/Nubix20/services/

```



```

    GetMeterDataService .
    GetMeterDataServiceHttpSoap11Endpoint/</wsa:To>
    <wsa:MessageID>urn:uuid:7a850930-a0a2-4321-a42e-41
    cad0383486</wsa:MessageID>
    <wsa:Action>urn:getReportedMeterData</wsa:Action>
  </soapenv:Header>
  <soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss
  /2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  wsu:Id="Id-20812468">
    <ns2:getReportedMeterData xmlns:ns2="http://main">
      <ns2:meterID>87654321</ns2:meterID>
      <ns2:startTime>2011-11-01T00:00:00.000+01:00</
      ns2:startTime>
      <ns2:endTime>2011-11-01T01:59:59.000+01:00</
      ns2:endTime>
    </ns2:getReportedMeterData>
  </soapenv:Body>
</soapenv:Envelope>

```

Videre, meldingen fra Nubix 2.0 til "GridB", nettselskapet ansvarlig for avregninga av målepunkt "87654321":

```

<?xml version='1.0' encoding='UTF-8'?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
  soap/envelope/">
    <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/
    addressing">
      <wsse:Security xmlns:wsse="http://docs.oasis-open.org/
      wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.
      xsd" soapenv:mustUnderstand="1">
        <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org
        /wss/2004/01/oasis-200401-wss-wssecurity-utility
        -1.0.xsd" wsu:Id="Timestamp-9">
          <wsu:Created>2011-12-11T17:13:04.359Z</
          wsu:Created>
          <wsu:Expires>2011-12-11T17:18:04.359Z</
          wsu:Expires>
        </wsu:Timestamp>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/
        xmldsig#" Id="Signature-10">
          <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://
            www.w3.org/2001/10/xml-exc-c14n#" />
            <ds:SignatureMethod Algorithm="http://www.w3.
            org/2000/09/xmldsig#rsa-sha1" />
            <ds:Reference URI="#Id-10601348">
              <ds:Transforms>

```

```

        <ds:Transform Algorithm="http://www.w3.
            org/2001/10/xml-exc-c14n#" />
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.
        org/2000/09/xmldsig#sha1" />
    <ds:DigestValue>+r6LTq6TOFDA47ntxiVYoobuk+Y
=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#Timestamp-9">
    <ds:Transforms>
        <ds:Transform Algorithm="http://www.w3.
            org/2001/10/xml-exc-c14n#" />
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.
        org/2000/09/xmldsig#sha1" />
    <ds:DigestValue>WIZaxvfmYbwhQqB/13bhDt745lk
=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
    R4cTE4j4/Uk/BKoXYGj5IN4Uh2WgI1t4L
    86CFQ9yDmdudY+RY4haYd4sxxO0
    I2lh80/YSeo5yUIbHAhzbjLx6e4
    mxgvGLVzEkgB8rxq
    SGFwb1KyRVb3RAr9lv
    YbZMIWmGYpiGxP5z
    LmMbtbYKPIFXl0qt
    SrUTO7RJO5rQI
    ZdnAnxYAHdQqpma/N8rkKw
    cS2Ipn2T9l/PuQcwZatKZMNz
    Vd3PCTNFIVTw
    6wYFB/7Qmpqte+4RGE
    oab/S7XSw8eb9XRpG
    aIFTOT6+2BAIzKX
    qGRHZye69zMsIIrS
    kXkkjc/RTXIMq
    nRs3pCwxyboPzvzu9o2eOS+A5P+
    AhJAyeTO8hWA==
    </ds:SignatureValue>
<ds:KeyInfo Id="KeyId-8645926
    F5ADC635962132362358435914">
    <wsse:SecurityTokenReference xmlns:wsu="http:
        //docs.oasis-open.org/wss/2004/01/oasis
        -200401-wss-wssecurity-utility-1.0.xsd"
        wsu:Id="STRId-8645926
        F5ADC635962132362358435915">
    <wsse:KeyIdentifier EncodingType="http://
        docs.oasis-open.org/wss/2004/01/oasis
        -200401-wss-soap-message-security-1.0#

```

```

        Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509SubjectKeyIdentifier">
        OQF6TvD2yq03o2wxZzKxrJ15N0k=</
        wsse:KeyIdentifier>
    </wsse:SecurityTokenReference>
    </ds:KeyInfo>
    </ds:Signature>
</wsse:Security>
<wsa:To>http://localhost:8094/GridB/services/
    GetMeterDataService.
    GetMeterDataServiceHttpSoap11Endpoint</wsa:To>
<wsa:MessageID>urn:uuid:6a71b244-a8ca-4f76-b4ba-2
    dc03800e871</wsa:MessageID>
<wsa:Action>urn:getReportedMeterData</wsa:Action>
</soapenv:Header>
<soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    wsu:Id="Id-10601348">
    <ns2:getReportedMeterData xmlns:ns2="http://main">
        <ns2:meterID>87654321</ns2:meterID>
        <ns2:requestorID>1</ns2:requestorID>
        <ns2:requestorType>1</ns2:requestorType>
        <ns2:startTime>2011-11-01T00:00:00.000+01:00</
            ns2:startTime>
        <ns2:endTime>2011-11-01T01:59:59.000+01:00</
            ns2:endTime>
    </ns2:getReportedMeterData>
    </soapenv:Body>
</soapenv:Envelope>

```

Svarmeldingen fra "GridB" til "Nubix 2.0" :

```

<?xml version='1.0' encoding='UTF-8'?>
    <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
        soap/envelope/">
        <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/
            addressing">
            <wsse:Security xmlns:wsse="http://docs.oasis-open.org/
                wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.
                xsd" soapenv:mustUnderstand="1">
                <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org
                    /wss/2004/01/oasis-200401-wss-wssecurity-utility
                    -1.0.xsd" wsu:Id="Timestamp-7">
                    <wsu:Created>2011-12-11T17:13:04.440Z</
                        wsu:Created>

```

```

    <wsu:Expires>2011-12-11T17:18:04.440Z</
      wsu:Expires>
  </wsu:Timestamp>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/
    xmldsig#" Id="Signature-8">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://
        www.w3.org/2001/10/xml-exc-c14n#" />
      <ds:SignatureMethod Algorithm="http://www.w3.
        org/2000/09/xmldsig#rsa-sha1" />
      <ds:Reference URI="#Id-23919296">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.
            org/2001/10/xml-exc-c14n#" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.
          org/2000/09/xmldsig#sha1" />
        <ds:DigestValue>UmJ4fHzgJNW2kJ78EJ0DNepqOhY
          =</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#Timestamp-7">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.
            org/2001/10/xml-exc-c14n#" />
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.
          org/2000/09/xmldsig#sha1" />
        <ds:DigestValue>Qyu288ihQiQvW3OXn9EZwJ70A9c
          =</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
    Ap0TUj1a69vKHUiFqkbC
    Wn0RckFti68oakptgZM
    1Xo+XiOv2u6Js8jIZoq0e7wK
    q42QdRAsKVYc6GNnRWVj
    mkU5GgDZPsfbal2dMckruBhM2
    vE1aFXnnkKfGsb9jP6HvifL6
    0u3tR3xWDz0aYc5J
    WnrGNtA7qn8IJTtZYJK
    zBpk99pNF6nYBqpU6tqMutPxX9h
    ODGfBpaM0di/4ZxhX/8WRm
    tNQgfmQk0OqBMKthRDuLWJLxiCtaZ8
    zurMj/W2lSwB0zilMR/EVwopDIWF
    CIuiRQ6HGsluzPZdhR3E1dOUuQu/A
    tRBsuml4SGBxw5k7Q1o2TFYJD89ivzV
    42I4in0w==
    </ds:SignatureValue>

```

```

<ds:KeyInfo Id="KeyId-7
  D16E8E3A9EAEEC46F132362358444111">
  <wsse:SecurityTokenReference xmlns:wsu="http://
    docs.oasis-open.org/wss/2004/01/oasis-
    200401-wss-wssecurity-utility-1.0.xsd"
    wsu:Id="STRId-7
      D16E8E3A9EAEEC46F132362358444112">
        <wsse:KeyIdentifier EncodingType="http://
          docs.oasis-open.org/wss/2004/01/oasis-
          200401-wss-soap-message-security-1.0#
          Base64Binary" ValueType="http://docs.
          oasis-open.org/wss/2004/01/oasis-200401-
          wss-x509-token-profile-1.0#
          X509SubjectKeyIdentifier">
          aMhuitud8Lr2TMQ3EL3XCY1juhg=</
            wsse:KeyIdentifier>
          </wsse:SecurityTokenReference>
        </ds:KeyInfo>
      </ds:Signature>
    </wsse:Security>
    <wsa:Action>urn:getReportedMeterDataResponse</
      wsa:Action>
    <wsa:RelatesTo>urn:uuid:6a71b244-a8ca-4f76-b4ba-2
      dc03800e871</wsa:RelatesTo>
  </soapenv:Header>
  <soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss
    /2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    wsu:Id="Id-23919296">
    <ns2:getReportedMeterDataResponse xmlns:ns2="http://
      main">
      <ns2:return>
        <ns1:meterID xmlns:ns1="http://logic/xsd">
          87654321</ns1:meterID>
        <reportedData xmlns="http://logic/xsd">
          <consumption>1669.657651765012</consumption>
          <endTime>2011-11-01T02:59:59.000+01:00</
            endTime>
          <startTime>2011-11-01T02:00:00.000+01:00</
            startTime>
        </reportedData>
        <reportedData xmlns="http://logic/xsd">
          <consumption>1877.8715625789725</consumption>
          <endTime>2011-11-01T02:59:59.000+01:00</
            endTime>
          <startTime>2011-11-01T02:00:00.000+01:00</
            startTime>
        </reportedData>
      </ns2:return>
    </ns2:getReportedMeterDataResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

```

    </soapenv:Body>
</soapenv:Envelope>

```

Og til sist meldingen fra Nubix 2.0 tilbake til "Powerplant":

```

<?xml version='1.0' encoding='UTF-8'?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/
    soap/envelope/">
    <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/
      addressing">
      <wsse:Security xmlns:wsse="http://docs.oasis-open.org/
        wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.
        xsd" soapenv:mustUnderstand="1">
        <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org
          /wss/2004/01/oasis-200401-wss-wssecurity-utility
          -1.0.xsd" wsu:Id="Timestamp-11">
          <wsu:Created>2011-12-11T17:13:04.489Z</
            wsu:Created>
          <wsu:Expires>2011-12-11T17:18:04.489Z</
            wsu:Expires>
        </wsu:Timestamp>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/
          xmldsig#" Id="Signature-12">
          <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://
              www.w3.org/2001/10/xml-exc-c14n#" />
            <ds:SignatureMethod Algorithm="http://www.w3.
              org/2000/09/xmldsig#rsa-sha1" />
            <ds:Reference URI="#Id-11464042">
              <ds:Transforms>
                <ds:Transform Algorithm="http://www.w3.
                  org/2001/10/xml-exc-c14n#" />
              </ds:Transforms>
              <ds:DigestMethod Algorithm="http://www.w3.
                org/2000/09/xmldsig#sha1" />
              <ds:DigestValue>BH3Bea34Hgs+aYxXdARInzlvngxk
                =</ds:DigestValue>
            </ds:Reference>
            <ds:Reference URI="#Timestamp-11">
              <ds:Transforms>
                <ds:Transform Algorithm="http://www.w3.
                  org/2001/10/xml-exc-c14n#" />
              </ds:Transforms>
              <ds:DigestMethod Algorithm="http://www.w3.
                org/2000/09/xmldsig#sha1" />
              <ds:DigestValue>PocV4CCTJ5kyyF8a7gIwtdSn3lw
                =</ds:DigestValue>
            </ds:Reference>
          </ds:SignedInfo>
        </ds:Signature>
      </wsse:Security>
    </soapenv:Header>
  </soapenv:Envelope>

```

```

</ds:SignedInfo> <ds:SignatureValue>YNHv7j
cymBK8WpdAbuF/p4YvJe2M3f/z8kGU+
KBjkwvww9VjvLNFvdi1jQC9kmLrVR6BiYKN9N
QzhEq3yJJlo+izm
gNhm/6AVP9QOU3pgdfQVHBZwa4xZMKhV9OUvwy
wCjZ+rp8ku7v1MCed+N5aDRWla52KTrCJ7fPnwlUT
p5m55WFDyfiRv6uqJ4N5VGViD4VUXmChY7qmOqf
bfMMi2KIZvTYQgkyokBco
dH2Xrf74Jd3VpGzlJ58/Gw0a5aMG/PzdtG0te4aQ7
0DeuBKwdaQ0ta9AuPiYjFPO/y3Wh+6FCXLb9cMRe
LXpANyAJjy2qpdjCb33i4FcWG4LHPmqWQ==
</ds:SignatureValue>
<ds:KeyInfo Id="KeyId-8645926
F5ADC635962132362358449017">
  <wsse:SecurityTokenReference xmlns:wsu="http:
//docs.oasis-open.org/wss/2004/01/oasis
-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="STRId-8645926
F5ADC635962132362358449018">
  <wsse:KeyIdentifier EncodingType="http://
docs.oasis-open.org/wss/2004/01/oasis
-200401-wss-soap-message-security-1.0#
Base64Binary" ValueType="http://docs.
oasis-open.org/wss/2004/01/oasis-200401-
wss-x509-token-profile-1.0#
X509SubjectKeyIdentifier">
OQF6TvD2yq03o2wxZzKxrJ15N0k=</
wsse:KeyIdentifier>
  </wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>
<wsa:Action>urn:getReportedMeterDataResponse</
wsa:Action>
<wsa:RelatesTo>urn:uuid:7a850930-a0a2-4321-a42e-41
cad0383486</wsa:RelatesTo>
</soapenv:Header>
<soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss
/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="Id-11464042">
<ns2:getReportedMeterDataResponse xmlns:ns2="http://
main">
  <ns2:return>
    <ns1:meterID xmlns:ns1="http://logic/xsd">
87654321</ns1:meterID>
    <reportedData xmlns="http://logic/xsd">
      <consumption>1669.657651765012</consumption>
      <endTime>2011-11-01T00:59:59.000+01:00</
endTime>
  </ns2:return>
</ns2:getReportedMeterDataResponse>
</soapenv:Body>
</soapenv:Envelope>

```

```
        <startTime>2011-11-01T00:00:00.000+01:00</  
          startTime>  
    </reportedData>  
    <reportedData xmlns="http://logic/xsd">  
      <consumption>1877.8715625789725</consumption>  
      <endTime>2011-11-01T01:59:59.000+01:00</  
        endTime>  
      <startTime>2011-11-01T01:00:00.000+01:00</  
        startTime>  
    </reportedData>  
  </ns2:return>  
</ns2:getReportedMeterDataResponse>  
</soapenv:Body>  
</soapenv:Envelope>
```




INSTRUKSJONER FOR OPPSETT AV SYSTEMET

Koden og nødvendige filer finnes som vedlegg til denne rapporten. Følgende mapper er inkludert:

- Workspace. Her ligger alle klassene og konfigurasjonsfilene.
- Keystores. Her ligger alle "keystore"-filene.
- DB. Her ligger sql-filer for de tre databasene i bruk.
- Biblioteker. Inneholder de ulike .jar- og .mar-filene som blir brukt.

Jeg anbefaler å benytte Xampp for å starte SQL-tjeneren, samt Phpmyadmin for å sette opp databasene. Phpmyadmin følger med Xampp. Instruksjoner for oppsett av databasene er gitt i C.4

C.1 Nødvendig programvare

Denne guiden vil basere seg på følgende programvare, merk versjonsnumre:

- Eclipse IDE for Java Developers version **3.7.1** (Indigo)
- Apache Tomcat **7.0.22**
- Apache Axis2 **1.6.0**
- Apache rampart **1.6.0**

C.2 Oppsett av Eclipse

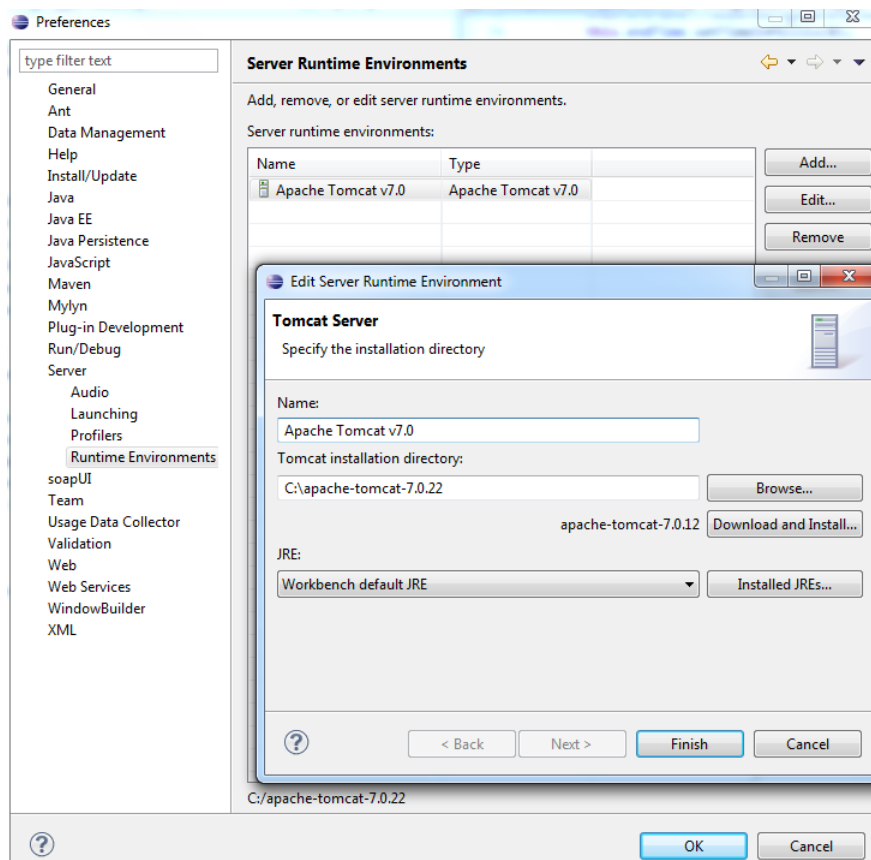
Start Eclipse, bruk et vilkårlig "Workspace", ikke benytt den vedlagte "Workspace"-katalogen.

Tilleggsmodulen "Web Tools Platform" (WTP) må bli installert. Dette gjøres ved å:

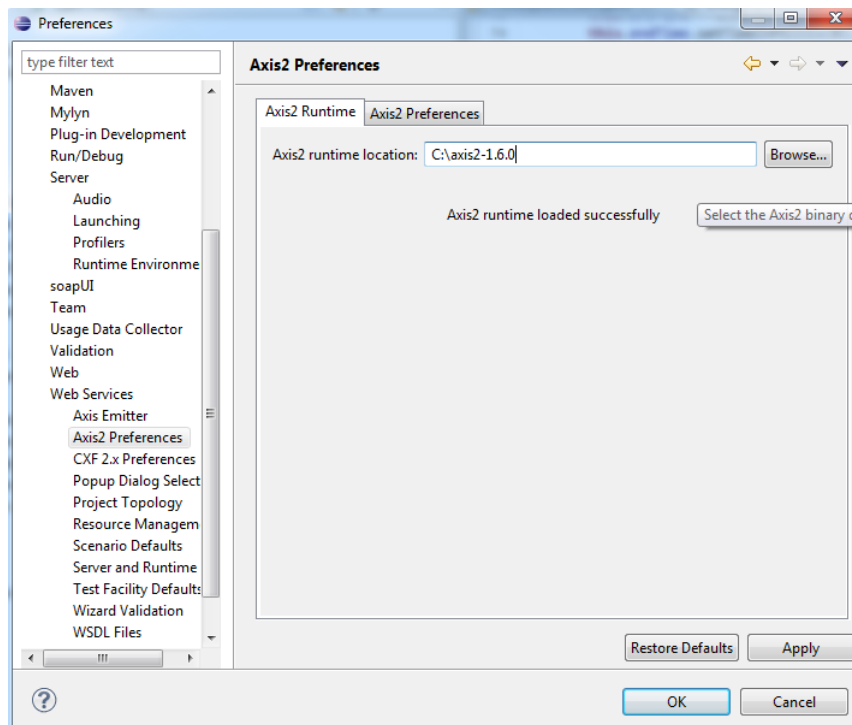
- Klikk "Help" - "Install new software".
- I "Work with", lim inn: <http://download.eclipse.org/webtools/repository/indigo/>
- Velg "Web Tools Platform (WTP) 3.3.1"

Videre må Eclipse bli fortalt hvor installasjonen av Apache Tomcat og Apache Axis2 ligger. For Tomcat:

- Klikk "Windows" - "Preferences" - "Server" - "Runtime Environments" - "Add".
- Velg Apache Tomcat v.7.0
- Sett inn stien til Apache Tomcat-installasjonen, se figur C.1 for et eksempel.



Figur C.1: Oppsett av Apache Tomcat



Figur C.2: Oppsett av Axis2

Det samme må gjøres for Axis2:

- Klikk "Windows" - "Preferences" - "Web Services" - "Axis2 Preferences" - "Axis2 runtime".
- Sett "Axis2 runtime location" til installasjonskatalogen til Axis2, se figur C.2

Klikk nå "File"->"Switch Workspace" og velg den vedlagte "Workspace"-katalogen.

C.3 Sikkerhetsoppsett

I filen "java.security", som ligger i mappen "JAVA_HOME/lib/security" må følgende sikkerhetstilbyder legges til:

```
security.provider.99=org.bouncycastle.jce.provider.  
BouncyCastleProvider
```

I "java.security" er det allerede spesifisert en rekke med andre sikkerhetstilbydere, sørg for at "BouncyCastle" får et annet nummer enn de som allerede er definert (99 i mitt eksempel).

I tillegg må systemet også modifiseres med korrekt sti til samtlige "Keystores" som blir benyttet. "Keystorene", jks-filer, ligger i den vedlagte katalogen "Keystores".

C.3.1 Tjenerinstanser

Som sagt så er det satt opp tre tjenerinstanser. Som ligger i prosjektet "Servers". Hver tjener har en konfigurasjonsfil, "server.xml", som må endres for å peke på korrekt "Keystore" for oppsett av HTTPS-sesjoner. I hver "server.xml" er det definert et "connector"-element for å sette en rekke SSL-parametre, bl.a. hvilken "keystore" som skal benyttes. Stien til denne "keystoren" må endres slik:

```
keystoreFile="STI_TIL_Keystores\keystore.jks"
```

Hvor "keystore" er hhv. "nubix", "gridA" og "gridB" for tjenerinstansene "Tomcat v7.0 Server at localhost-config", "Tomcat v7.0 Server at localhost-config (2)" og "Tomcat v7.0 Server at localhost-config (3)".

C.3.2 Klienter

Her følger instruksjoner for oppsett av nøkkelinformasjon hos klientene, "Powerplant", "Waterfall", "Energysystems" og "Energysolutions". Se tabell C.1 for oversikt over

klientenes "Keystores", heretter kalt "nøkkel". For hver av klientene må to endringer gjøres i deres individuelle prosjektkataloger, funnet i "Project Explorer" i Eclipse.

1. Konfigurer rampart til å benytte riktig nøkkel. Dette gjøres ved å åpne klientens "axis2.conf", funnet i mappen "WebContent/WEB-INF/conf", og endre stien i "ramp:property"-elementet som peker på "keystore"-filen til den vedlagte "Keystores"-katalogen. Merk at klientens egen "keystore" må forbli uendret, f.eks blir den fulle stien for "Powerplant" "STI_TIL_KEYSTORE/powerplant.jks"
2. Endre stien i "trustStorePath-variabelen" i klientens "main-klasse" til "Keystores"-katalogen.

C.3.3 Nubix 2.0

Totalt tre endringer må i "Nubix20"-prosjektet. "Keystore"-filen som dette prosjektet benytter heter "nubix.jks".

1. Konfigurer rampart til å benytte riktig nøkkel for utgående meldinger. Dette gjøres tilsvarende som for klientene.
2. Konfigurer rampart til å benytte riktig nøkkel for innkommende meldinger. Dette gjøres ved å endre stien i elementet "ramp:property" i filen "services.xml". Denne filen ligger i mappen "WebContent/WEB-INF/services/GetMeterDataService/META-INF". Stien blir da "STI_TIL_KEYSTORE/nubix.jks".
3. I klassen "GetMeterDataServiceSkeleton.java" må stien endres i metoden "initSSL()".

Klient	Nøkkel
Powerplant	powerplant.jks
Waterfall	waterfall.jks
Energysystems	energysystems.jks
Energysolutions	energysolutions.jks

Tabell C.1: Klientenes "Keystores"

C.3.4 Nettselskap

I prosjektene for nettselskapene, "GridA" og "GridB", må stien i "ramp:property"-elementet i services.xml endres, tilsvarende som for Nubix 2.0. For "GridA" blir stien "STI_TIL_KEYSTORE/grida.jks"

C.4 Databaseoppsett

Databasene "grid_owner_lookup", "meter_data_grid_owner_a" og "meter_data_grid_owner_b" må opprettes. I tillegg må det lages brukere med leserettigheter for de tre databasene. Tabell C.2 viser brukernavn- og passordet som må benyttes.

Endres brukernavn- og passordkombinasjonen må dette også endres i de ulike klassene som håndterer databasekommunikasjon. Disse klassene ligger i pakken "data" i de ulike prosjektene. I den vedlagte katalogen "DB" ligger sql-filer de tre databasene som må importeres inn i sine respektive databaser. Phpmyadmin tilbyr en enkel og grei måte å gjøre dette på.

Database	Brukernavn	Passord
grid_owner_lookup	nubix	n2EsHRUSDsbVynFY
meter_data_grid_owner_a	GridA	rpyMDFC35wYZaRHX
meter_data_grid_owner_b	GridOwnerB	7Hn5smueaYVYNAdJ

Tabell C.2: Brukernavn og passord for de ulike databasene

C.5 Biblioteker

De nødvendige bibliotekene skal allerede være på plass, men de er lagt ved i katalogen "Biblioteker" i hvis det skulle være et behov. Korrekt plassering av bibliotekene for et prosjekt er som følger:

- Filene i underkatalogen "lib" skal kopieres til mappen "WebContent/WEB-INF/lib.
- Filene i underkatalogen "modules" skal kopieres til mappen "WebContent/WEB-INF/modules"
- Filene i underkatalogen "modules" må også legges til "build path" for prosjektet.

C.6 Testing

Systemet er nå klart for testing. Tabell C.3 viser en oversikt over de ulike klientens hovedklasser. I "main"-metoden kan man endre ønsket målepunkt-ID, start- og sluttidspunkt samt om man ønsker å kalle "getReportedMeterData" eller "getActiveMeterData". Resultatet av spørringen blir skrevet til XML-fil som vil vises under klientprosjektet. Det er mulig prosjektet må "reloads" for at filen skal vises.

Klient	Hovedklasse
Powerplant	PowerplantClient.java
Waterfall	WaterfallClient
Energysystems	EnergySystemsClient.java
Energysolutions	EnergySolutionsClient.java

Tabell C.3: "Main"-metodene for de ulike klientene