



Jarl Kvanli



Infrastruktur for elektronisk handel

En avhandling med fokus på
katalogtjenester i elektronisk handel

Oktober 2001



Institutt for datateknikk og informasjonsvitenskap

Norges teknisk-naturvitenskapelige universitet

Forord

Internett gir mulighet for å kommunisere mellom kjøper og selger på en ny måte, og mediet gir mulighet for elektronisk handel (e-handel) mellom kunder, bedrifter og det offentlige. Ved etablering av e-handel må vi ha en teknologisk infrastruktur som understøtter kommunikasjon mellom kjøper og selger.

Bakgrunn for avhandlingen er økende bruk av Internett som kommunikasjons- og salgskanal. Økt utbredelse av løsninger for e-handel, i nye virksomheter, gir et konkurransefortrinn i forhold til eksisterende virksomheter. Om eksisterende virksomheter ønsker å bruke Internett som salgs- og kommunikasjonskanal er det nødvendig å endre nåværende infrastruktur. Avhandlingen fokuserer på dette arbeidet.

Avhandlingen presenterer teori for e-handel, katalogtjenester, og katalogtjenester i e-handel. På bakgrunn av teori, og et praktisk forsøk i en virksomhet, presenterer avhandlingen en anbefaling av infrastruktur for e-handel. Anbefalingen beskriver hvordan vi kan konstruere en løsning for e-handel ved å integrere nye komponenter til eksisterende infrastruktur.

Avhandlingen er skrevet i forbindelse med hovedfagsstudiet ved Norges teknisk-naturvitenskapelige universitet (NTNU). Hovedfagsstudiet ble gjennomført i perioden våren 2000 til høsten 2001, og arbeidet med selve avhandlingen ble gjennomført i perioden høsten 2000 til høsten 2001.

Studiet er gjennomført i samarbeid med IKON AS. I denne sammenheng vil jeg takke Truls Nergård og Egil Helland for teknisk bistand. Jeg vil spesielt takke Ole Jens Dalen og Stein Helge Riise for administrativ og teknisk bistand gjennom hele prosjektet. Jeg vil også rette en stor takk til veileder, professor Ingeborg Sølvberg, ved NTNU.

Til slutt vil jeg også rette en påskjønnelse til Telenor/ZebZign, SUN/Netscape og MaXware International. Deres produkter og kompetanse har vært til stor hjelp i studiet.

Trondheim, 1 oktober 2001

Jarl Kvanli

Innhold

Sammendrag.....	5
1 Innledning	7
1.1 MÅL	8
1.2 PROBLEMSTILLING OG AVGRENSNING	8
1.3 TILNÆRMING.....	10
1.4 MÅLGRUPPE	10
1.5 OVERSIKT OVER AVHANDLINGEN	10
2 Om elektronisk handel.....	13
2.1 INNLEDNING	13
2.1.1 Begreper om e-handel.....	13
2.2 MARKEDSFØRING OG STRATEGI FOR E-HANDEL	14
2.2.1 Markedsføring og kommunikasjon	14
2.2.2 Strategier for e-handel.....	16
2.3 ARKITEKTUR	18
2.3.1 Kategorier	19
2.3.2 Sikkerhet.....	21
2.4 STANDARDER OG SPESIFIKASJONER.....	23
2.4.1 EDI og EDIFACT	23
2.4.2 XML	27
2.4.3 Integrasjon EDI og XML	30
2.4.4 OBI.....	32
2.4.5 UDDI.....	34
2.5 SYSTEMER FOR E-HANDEL	37
2.6 OPPSUMMERING	38
3 Om katalogsystemer og katalogtjenester.....	39
3.1 INNLEDNING	39
3.1.1 Begreper om katalogsystemer og katalogtjenester	40
3.2 ARKITEKTUR	42
3.2.1 Design og struktur	42
3.2.2 Egenskaper	47
3.2.3 Klassifisering av kataloger.....	52
3.2.4 Sikkerhet.....	54
3.3 STANDARDER OG SPESIFIKASJONER.....	58
3.3.1 X.500	58
3.3.2 LDAP	64
3.3.3 DNS	68
3.4 SYSTEMER FOR KATALOGTJENESTER	71
3.5 OPPSUMMERING	72
4 Om katalogtjenester i elektronisk handel.....	75
4.1 INNLEDNING	75
4.2 ARKITEKTUR	75
4.2.1 Metakatalog	76
4.3 GEVINSTER MED KATALOGTJENESTER I E-HANDEL.....	79
4.4 ULEMPER MED KATALOGTJENESTER I E-HANDEL	81

4.5 SUKSESSKRITERIER FOR E-HANDEL	81
4.6 OPPSUMMERING	82
5 Strategi, metode og modellering	83
5.1 INNLEDNING	83
5.2 STRATEGI FOR SYSTEMUTVIKLING	83
5.2.1 Strategi for utvikling i prosjektet.....	85
5.3 METODE FOR INNSAMLING OG BEARBEIDING AV DATA	87
5.3.1 Metode i prosjektet	88
5.4 VERKTØY FOR MODELLERING	89
5.4.1 Verktøy for modellering i prosjektet	90
5.5 OPPSUMMERING	91
6 Systemutvikling	93
6.1 OM SAMARBEIDSPARTNER	93
6.2 NÅ-SITUASJON I IKON	93
6.2.1 Om bedrift	94
6.2.2 Om salg og markedsføring	94
6.2.3 Om infrastruktur	95
6.3 KRAV TIL NY LØSNING	96
6.3.1 Krav til funksjonalitet.....	96
6.3.2 Krav til infrastruktur	97
6.4 UTVIKLING AV NY LØSNING	97
6.5 DOKUMENTASJON AV NY LØSNING	99
6.5.1 Klargjøring og konfigurasjon av server, datakilder og katalogsystem	100
6.5.2 Integrasjon datakilder og katalogsystem	101
6.5.3 Integrasjon web og katalogsystem	102
6.6 OPPSUMMERING	102
7 Diskusjon og anbefaling	103
7.1 ALTERNATIVE VALG FOR UTVIKLING OG LØSNING	103
7.1.1 Strategi, metoder og modeller	103
7.1.2 Systemer for e-handel	104
7.1.3 Teknologisk infrastruktur.....	105
7.2 RESULTAT OG ANBEFALLING	107
7.2.1 Resultat.....	107
7.2.2 Anbefaling.....	108
8 Evaluering	109
8.1 EVALUERING AV MÅLOPPNÅELSE, SYSTEMUTVIKLING OG SUKSESSKRITERIER.....	109
8.1.1 Oppnåelse av hoved- og delmål	109
8.1.2 Systemutvikling.....	110
8.1.3 Oppnåelse av suksesskriterier.....	111
8.2 EVALUERING AV PROSJEKTARBEID	112
8.3 VALG AV SAMARBEIDSPARTNER.....	112
8.4 FORSLAG TIL VIDERE ARBEID.....	113
9 Referanser	115

Vedlegg

- Vedlegg A – Om Unified Modeling Language.
- Vedlegg B – Om IKON AS.
- Vedlegg C – Intervjuguide for datainnsamling av nå-situasjon i IKON.
- Vedlegg D – Modellering av nå-situasjon for salg av tjenester i IKON.
- Vedlegg E – Modellering av ny løsning for salg av tjenester i IKON.
- Vedlegg F – Klargjøring og konfigurasjon av server, datakilder og katalogsystem.
- Vedlegg G – Integrasjon datakilder og katalogsystem.
- Vedlegg H – Integrasjon web og katalogsystem.
- Vedlegg I – Utvikling av LDAP- og SQL-operasjoner i PHP.

Figurer

Figur 1-1 – Skisse over problemstilling, avgrensning og tilnærming.	9
Figur 2-1 – Tradisjonell kommunikasjonsmodell [TD00].	15
Figur 2-2 – Ny kommunikasjonsmodell for web [TD00].	15
Figur 2-3 – Eksempel på arkitektur for B2C [TD00].	19
Figur 2-4 – Eksempel på arkitektur for B2B [CT00].	20
Figur 2-5 – Eksempel på arkitektur for B2P [TES00].	21
Figur 2-6 – Eksempel på utveksling av EDI-meldinger mellom organisasjoner [NR01].	24
Figur 2-7 – Komponenter i en EDI-løsning [SIN01].	24
Figur 2-8 – Eksempel på EDI-melding (opprinnelig eksempel er lengre) [EP01].	26
Figur 2-9 – Eksempel på en uformatert EDIFACT-melding [EP01].	26
Figur 2-10 – Eksempel på XML-syntaks [ERH99].	27
Figur 2-11 – Eksempel på syntaks i XSL for periodisk tabell [ERH99].	28
Figur 2-12 – Sammenkobling av XML og XSL gir mulighet for visning med HTML [NR01].	28
Figur 2-13 – HTML-kode produsert ved hjelp av XML- og XSL-kode [ERH99].	29
Figur 2-14 – Web-side (i HTML) produsert ved hjelp av XML og XSL [ERH99].	29
Figur 2-15 – Eksempel på handelssystem med EDI og XML [RMH01].	31
Figur 2-16 – Oversetting mellom EDI og XML [RMH01].	31
Figur 2-17 – Modell/arkitektur for OBI [OBI01].	33
Figur 2-18 – UDDI i relasjon til andre standarder og spesifikasjoner [UD01].	36
Figur 2-19 – Sammenheng mellom spesifikasjon, skjema og tjenester i UDDI [UD01].	36
Figur 2-20 – Bruk av UDDI [UD01].	37
Figur 3-1 – Komponenter i infrastruktur for e-handel [NED00].	40
Figur 3-2 – Kjernen i katalogsystemer [RA00].	42
Figur 3-3 – Eksempel på katalogtre [SS00].	43
Figur 3-4 – Fysisk navnemodell [SS00].	44
Figur 3-5 – Logisk navnemodell [SS00].	45
Figur 3-6 – Partisjonering av en global organisasjon (the mythical) [SS00].	47
Figur 3-7 – Egenskaper for katalogtjenester.	48
Figur 3-8 – Forholdet mellom replikering og synkronisering [RA00].	50
Figur 3-9 – Nivå av katalogsystemer [AGI99].	53
Figur 3-10 – Klassifisering av katalogsystemer [AGI99].	54
Figur 3-11 – SSL i relasjon til TCP/IP [NET01].	57

Figur 3-12 – Forholdet mellom protokoller og agenter i X.500 [SS00].	60
Figur 3-13 – Forholdet mellom DN og RDN i DIT [SS00].	62
Figur 3-14 – LDAP-server som gateway til X.500-server [JBH98].	65
Figur 3-15 – LDAP-klient og LDAP-server [JBH98].	65
Figur 3-16 – Poster, attributter og verdier [JBH98].	67
Figur 3-17 – Oppslag i DNS [HIO01].	71
Figur 4-1 – Fokus på integrasjon av katalogsystemer i e-handel.	75
Figur 4-2 – Eksempel på sammenkobling av separate kataloger til metakatalog [RA00].	77
Figur 4-3 – Eksempel på sammenkobling av attributter fra kataloger til metakatalog [RA00].	77
Figur 5-1 – Utviklingsstrategier [AES98].	84
Figur 5-2 – Grunnlag for valg av strategier for utvikling.	85
Figur 5-3 – Valgte strategier for prosjekt.	87
Figur 6-1 – Logo for IKON AS [IK01].	93
Figur 6-2 – Variabler som påvirket utviklingen av ny infrastruktur.	96
Figur 6-3 – Ny infrastruktur.	97
Figur 6-4 – Skjerm bilde for søk etter ønsket kompetanse.	99
Figur 6-5 – Skjerm bilde for tilbakemelding på søk etter kompetanse.	100
Figur 6-6 – Modell over datastrukturer.	101
Figur 7-1 – Prosjektets resultat - prototyp.	107

Tabeller

Tabell 2-1 – Strategier for brukererfaringer [MT00].	16
Tabell 2-2 – Strategier for struktur på web-sider [MT00].	17
Tabell 2-3 – Strategier for forbedring av prosesser [MT00].	17
Tabell 2-4 – Sammendrag av strategier for e-handel fra Charles Trepper [CT00].	18
Tabell 2-5 – Oppbygging av EDI EDIFACT meldingsformat [EP01].	25
Tabell 2-6 – Sammenligning av løsninger for e-handel basert på XML eller EDI [RMH01].	30
Tabell 2-7 – Forskjell mellom OBI- og XML-formatet [BKS01].	34
Tabell 2-8 – Underliggende systemer i en B2C-løsningen for e-handel [TD00].	37
Tabell 2-9 – Leverandører av systemer for e-handel.	38
Tabell 3-1 – Standarder for X.500 [SS00].	59
Tabell 3-2 – Operasjoner i X.500 [SS00].	63
Tabell 3-3 – RFC-dokumenter for LDAP versjon 3.	64
Tabell 3-4 – Operasjoner i LDAP [SS00, MET00].	68
Tabell 3-5 – RFC-dokumenter for DNS [IET01, UIT01].	69
Tabell 3-6 – Rot-domener i USA og resten av verden [HIO01].	69
Tabell 3-7 – Nye top-domener [IAHC01].	70
Tabell 3-8 – Leverandører og produkter av katalogsystemer.	72
Tabell 4-1 – Leverandører og produkter av metakataloger.	79
Tabell 4-2 – Gevinster med katalogsystemer.	79
Tabell 5-1 – Type visninger (View) og diagrammer for distribuerte systemer i UML.	90
Tabell 6-1 – Valg av nye komponenter til infrastruktur for e-handel.	99
Tabell 8-1 – Oppnåelse av suksesskriterier for IKON.	111

Sammendrag

Hensikten med avhandlingen er å anbefale en teknologisk infrastruktur, med katalogsystemer, for e-handel. Avhandlingen fokuserer på å gjenbruke data om produkter og tjenester, i etablerte virksomheter, og gjøre de tilgjengelig på Internett.

Studiet viser at vi kan bruke katalogsystemer i infrastruktur for e-handel. Vi kan bruke slike systemer til å integrere og synkronisere data med etablerte datastrukturer. Studiet viser også at data i katalogsystemer kan gjøres tilgjengelige på Internett, for salg og markedsføring, gjennom et web-grensesnitt.

Infrastruktur for e-handel kan bestå av datamaskin(er) med egenskaper som server, web-server og katalogserver. Katalogsystem bør være utformet som metakatalog, og støtte etablerte standarder og spesifikasjoner for katalogtjenester, kommunikasjon og sikkerhet. Foruten katalogsystem bør infrastrukturen bestå av verktøy for integrasjon av data mellom web, katalogsystem og datakilder.

Avhandlingen omfatter en teoretisk fremstilling av e-handel, katalogtjenester og katalogtjenester i e-handel. Dokumentet presenterer strategier for utvikling, metoder for innsamling og bearbeiding av data, og verktøy for modellering ved systemutvikling. Avhandlingen dokumenterer utviklingsprosessen frem til prototyp. Dokumentet omtaler også alternative valg av metoder for systemutvikling, løsning og teknologi. Til slutt presenterer avhandlingen studiets resultat og anbefaling.

Studiet brukte kvalitativ metode ved innsamling og bearbeiding av data om bedriften, og litteraturanalyse for å fremskaffe data for den teoretiske fremstillingen i avhandlingen. Utvikling av prototyp ble gjennomført med bakgrunn i valgte strategier for systemutvikling. Teoretisk bakgrunn og prototyp ble brukt som grunnlag for studiets anbefaling.

Studiet er gjennomført i samarbeid med IKON AS. Selskapet bør på bakgrunn av prototyp og studiets anbefaling evt beslutte endelig løsning for e-handel.

1 Innledning

Handel mellom mennesker har pågått i flere tusen år. Historien viser at fysisk tilgjengelighet har vært et av kriteriene for handel mellom mennesker. Mennesker møtte hverandre på en felles arena, og kommuniserte sine varer og tjenester for potensielle kjøpere. For de som var interessert ble varer byttet mot andre produkter og tjenester.

Det er også et bytteforhold ved dagens handel, men gjennom et universalt medium kalt penger. Og penger står mer og mer sentralt i økende internasjonalisering og global handel. Den økende globaliseringen påvirker måten vi handler på. De "gamle" møteplassene er erstattet med en ny arena – Internett. Møteplassen er hjemme, på jobben eller på reise, og er tilgjengelig 24 timer i døgnet.

Det er liten tvil om at e-handel vil endre måten vi utveksler varer og tjenester på. For å følge utviklingen vil økende krav til kompetanse om teknologi være meget sentral. Disse endringene vil for mange bli oppfattet som dramatisk og skremmende, men samtidig spennende og utfordrende.

E-handel er her nå, og skal vi overleve denne form for konkurranse, må vi søke å etablere et konkurransefortrinn i forhold til våre konkurrenter. Vi må kunne forsyne våre omgivelser med data - med riktig mengde og innhold, til riktig tid. Data skal være tilgjengelig for våre kunder, samarbeidspartnere og ansatte nå. De må kunne leses, endres og slettes i flere systemer samtidig, og systemene må kunne utveksle flere typer dataformat – tekst, bilde, lyd og video.

Her er det mange utfordringer. En av disse er å tilrettelegge for e-handel i organisasjoner eller bedrifter. Utfordringen i denne sammenheng er å integrere e-handel med eksisterende forretningsprosesser, eller kaste eksisterende prosesser og bygge de opp igjen i det nye perspektivet. Holdningen i måten vi kommuniserer med våre kunder må også endres. Fra fysisk kontakt til kontakt ved hjelp av teknologi.

Teknologisk infrastruktur står meget sentralt i denne sammenheng. Lagringskapasitet for enorme mengder data, båndbredde til kommunikasjon med en hel verden, databaser med meget god ytelse og som tåler hard belastning 24 timer i døgnet, maskinvare "store som berg" og sikkerhetsmekanismer som skal tåle en trøkk. Systemarkitekter har en helt annen hverdag i dag i jaget etter å designe infrastruktur for denne type handel.

Uavhengig av hvordan de teknologiske løsningene er oppbygd, så er data om produkter og tjenester det mest sentrale. Det er dette som er utgangspunktet for prosjektet. Nøkkelen for å lykkes er å fokusere på data og informasjon, som skal være oppdatert, riktig og tilgjengelig.

1.1 Mål

Tema for studiet er infrastruktur i e-handel. Avhandlingen fokuserer på katalogsystemer, som sentral teknologi, i infrastruktur for e-handel.

Hovedmål for prosjektet er:

Anbefale en infrastruktur, med katalogtjenester, for e-handel i små og mellomstore bedrifter.

Følgende *delmål* skal understøtte hovedmål:

Introdusere og beskrive tema e-handel, katalogtjenester og katalogsystemer i relasjon til infrastruktur for virksomheter.

Dokumentere fremgangsmåte for utvikling og løsning (prototyp) for e-handel.

Analysere og evaluere løsning for e-handel.

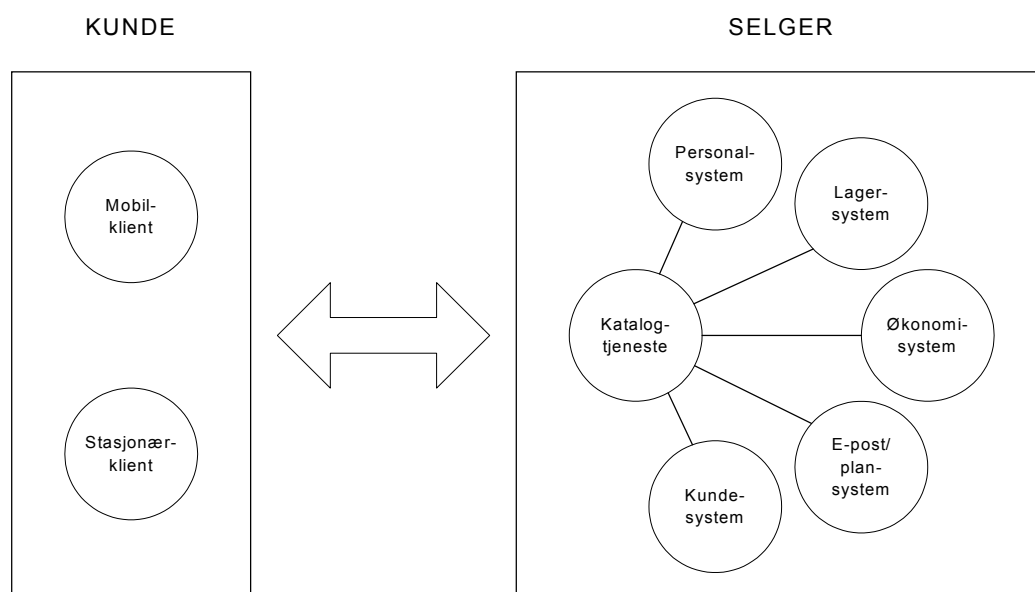
1.2 Problemstilling og avgrensning

Undertegnede har gjennom tidligere utdanning og yrkeserfaring fått innblikk i muligheter og begrensninger med IT/IS. Gjenbruk av data/informasjon på tvers av fagområder og systemer har gitt og gir fortsatt store muligheter for effektivisering. Dette gjelder både i privat og offentlig sektor [STA96].

Private bedrifter i Norge har, med Næringslivets Hovedorganisasjon (NHO) [NHO00] i ryggen, startet flere initiativ for å klargjøre næringslivet for e-handel. NHO ser blant annet store gevinster i å forenkle produksjons- og innkjøpsprosesser, redusere kostnader og forenkle måten vi inngår avtaler på tvers av landegrensene [NHO00].

Offentlig sektor har gjennom mange år implementert nye informasjonssystemer (IS) for å understøtte eksisterende, endrede eller nye oppgaver. Dette har blant annet gitt gevinster i form av reduserte kostnader, bedre samarbeid og bedre service for kunder og publikum [STA96].

Det er fortsatt mulighet for forbedringer. Det har både privat og offentlig sektor erkjent. Det offentlige har utarbeidet et eget program for e-handel. Hensikten med programmet er å legge til rette for e-handel i det offentlige, og på bakgrunn av dette bidra til effektivisering av sektoren [AA99a]. NHO00 har programfestet tema e-handel gjennom eNHO. I programmet søker NHO å heve basiskompetansen innen e-handel for sine medlemmer [NHO00].



Figur 1-1 – Skisse over problemstilling, avgrensning og tilnærming.

Nevnte føringer fra politiske miljø og organisasjonen bekrefter utviklingen innen e-handel i Norge. Denne utviklingen er en utfordring for både nye og etablerte virksomheter. For etablerte virksomheter vil disse føringene forsterke behovet for at de må endre eksisterende infrastruktur om den skal fungere for e-handel.

Studiets *problemstilling* (se figur 1-1) er forankret i å forstå hvordan nåværende infrastruktur fungerer, og hva vi evt må gjøre for å tilpasse den for e-handel.

Studiets *avgrensning* (se figur 1-1) er blant annet knyttet til hvem anbefalingen passer for. Studiet fokuserer på hvordan små og mellomstore virksomheter kan endre eksisterende infrastruktur for å klargjøre den for e-handel.

Det eksisterer mange pakkeløsninger (hyllevare) for e-handel i dag, men å bruke slike pakker krever ofte at vi må nytte andre systemer og applikasjoner fra samme leverandør. Eksempel på slike systemer og applikasjoner er operativsystemer, nettlesere og spesielle databaser.

Studiet tar utgangspunkt i å bruke leverandøruavhengige systemer og applikasjoner.

Vi kan bruke forskjellig type teknologi i infrastruktur for e-handel. Prosjektet søker å bruke katalogtjenester som aktiv komponent i infrastruktur. På denne måten kan vi undersøke om katalogsystemer passer i infrastruktur for e-handel.

Studiet er ikke et rent utviklingsprosjekt for en virksomhet, men et prosjekt i forbindelse med cand scient-studiet ved NTNU.

1.3 Tilnærming

Studiet retter søkelys på infrastruktur for e-handel i etablerte virksomheter. Prosjektet fokuserer på gjenbruk av data, om produkter og tjenester, i eksisterende datastrukturer, og gjøre disse tilgjengelig på web. Dette gir igjen mulighet for salg og markedsføring av produkter og tjenester over Internett.

Det foregår en hel del forskning innen e-handel og katalogtjenester. Mye av forskningen vurderer e-handel og katalogtjenester *isolert*, men et fåtall aktører har vurdert katalogtjenester *integrert* i infrastruktur for e-handel. De miljøer som omtaler katalogsystemer i e-handel mener slike systemer er nøkkelen i en god infrastruktur.

Min *tilnærming* til studiet (se figur 1-1) er å ta utgangspunkt i kunnskap om katalogsystemer og e-handel isolert, og deretter se emnene i relasjon til hverandre. Kunnskapen er deretter bakgrunn for utvikling av en prototyp. Kunnskap, om e-handel i relasjon til katalogtjenester, og prototyp er grunnlag for diskusjon og endelig anbefaling for studiet.

Teoretisk grunnlag, analyse av bedrift, utvikling av prototyp og resultat fra selve forsøket er grunnlag for diskusjon og anbefaling av infrastruktur for e-handel.

1.4 Målgruppe

Målgruppen for avhandlingen er Institutt for datateknikk og informasjonsvitenskap (IDI) ved Norges teknisk-naturvitenskapelige universitet (NTNU).

Dokumentet kan være interessant for ledere, mellomledere i små og mellomstore bedrifter, og organisasjoner som ønsker å etablere seg innen e-handel.

Avhandlingen kan også være av interesse for virksomheter som planlegger å innføre katalogsystemer og katalogtjenester.

1.5 Oversikt over avhandlingen

Avhandlingen er delt i tre deler:

Teoretisk forankring	- kapittel 2, 3 og 4.
Analyse og utvikling av prototyp	- kapittel 5 og 6.
Diskusjon, anbefaling og evaluering	- kapittel 7 og 8.

Teoretisk forankring

Første del av avhandlingen omtaler teori innen infrastruktur for e-handel, og omtaler strategier, arkitektur, standarder og spesifikasjoner, sikkerhet og tilgjengelige systemer for e-handel og katalogtjenester. Kapitlene er delt opp

slik at først presenteres tema for e-handel og katalogsystemer hver for seg, og deretter i relasjon til hverandre.

Analyse og utvikling av prototyp

Med bakgrunn i den teoretisk forankringen fortsetter avhandlingen med å presentere bakgrunn for og valg av strategier, metoder og modeller for systemutvikling. Avhandlingen dokumenterer også utviklingsprosessen frem til prototyp og selve prototypen.

Diskusjon, anbefaling og evaluering

Med utgangspunkt i prototypen fortsetter avhandlingen med diskusjon om alternative løsninger med påfølgende anbefaling av infrastruktur for e-handel. Til slutt følger en kort evaluering av prosjektet som omtaler grad av måloppnåelse, våre erfaringer med prosjektarbeid og forslag til videre arbeid.

2 Om elektronisk handel

Kapitlet introduserer begrepet e-handel og tema som strategi, brukerområder, arkitektur, sikkerhet, standarder og spesifikasjoner for e-handel.

2.1 Innledning

Mange har meninger om e-handel, og mange har oppfatninger om hvilken fremtid den har. Flere nasjonale og internasjonale prognoser fremhever at flere forbrukere og bedrifter vil nytte denne form for handel, og omsetningen vil øke betydelig de nærmeste årene [AN99, MM00, MT00, NHO00, TD00]. I Norge er både offentlig og privat sektor bevisst dette potensialet [AA99a, AA99b, NHO00].

Uansett fremtid, den nye økonomien introduserer et vell av nye begreper. Og om vi kobler informasjonsteknologi (IT), med den nye økonomien, nytter vi ofte begreper som e-handel, e-commerce og e-business. Men begrepene blir ofte brukt om hverandre og kan derfor virke mer forvirrende enn oppklarende.

2.1.1 Begreper om e-handel

I avsnittene nedenfor har jeg presentert definisjonen for noen av begrepene innen e-handel:

E-commerce - is any electronic exchange of information to conduct business [MM00].

Electronic business - is the transformation of an organization's business and functional processes to deliver additional customer value through the application of technologies, philosophies, and computing paradigms of the new digital economy [UNI00].

E-handel - er kommersielle transaksjoner mellom aktører ved hjelp av et elektronisk medium [AA99b].

Med utgangspunkt i nevnte definisjoner blir begrepet e-handel gjennomgående brukt om elektronisk e-handel i avhandlingen.

Det er forskjell mellom begrepene e-commerce, electronic business og e-handel. Det er også forskjell på definisjoner innen hvert begrep. Dette viser oss at fagområdet er i kontinuerlig endring.

Andersen mfl [AN99] inndeler aktørene innen e-handel i tre hovedgrupper, og består av følgende:

1. Myndigheter (på statlig og lokalt plan).
2. Næringslivsaktører (dvs forskjellige former for bedrifter og foretak).
3. Innbyggere i egenskap av konsumenter/forbrukere.

I denne sammenheng blir hovedgruppe 2 og 3 kalt *markedsaktører*. På bakgrunn av definisjonene kan vi dele markedet inn i følgende hovedkategorier:

Business to Business (BtB eller B2B) - samhandling mellom bedrifter.

Business to Consumer (BtC eller B2C) - samhandling mellom bedrifter og kunder.

Business to Public (BtP eller B2P) - samhandling mellom myndigheter og foretak/kunder.

Andersen fremhever at vi har kommet lengst innen utvikling av løsninger for B2B og B2C. Men det skjer også mye innen B2P. Utviklingen av B2P påvirker, til en viss grad, utviklingen av de andre områdene. Dette fordi myndighetene (nasjonalt og internasjonalt) kan bestemme standarder og føringer for å oppnå politiske mål.

2.2 Markedsføring og strategi for e-handel

Innføring av e-handel i bedrifter og organisasjoner bør være forankret i visjon, mål og strategi for virksomheten. Vi bør ikke definere strategi for e-handel som en isolert strategi, men vi bør innarbeide den som en del av den totale strategien for virksomheten [MT00, TD00].

2.2.1 Markedsføring og kommunikasjon

Utgangspunktet for all e-handel, uavhengig av visjon, mål og strategi, er å [TD00]:

1. Gjøre potensielle kunder kjent med at bedriften er tilstede på nettet. Dette forutsetter at vi formidler adressen til kundene og motiverer dem til å besøke bedriften på Internett.
2. Fokuserer på å beholde interessen og oppmerksomheten hos kundene.

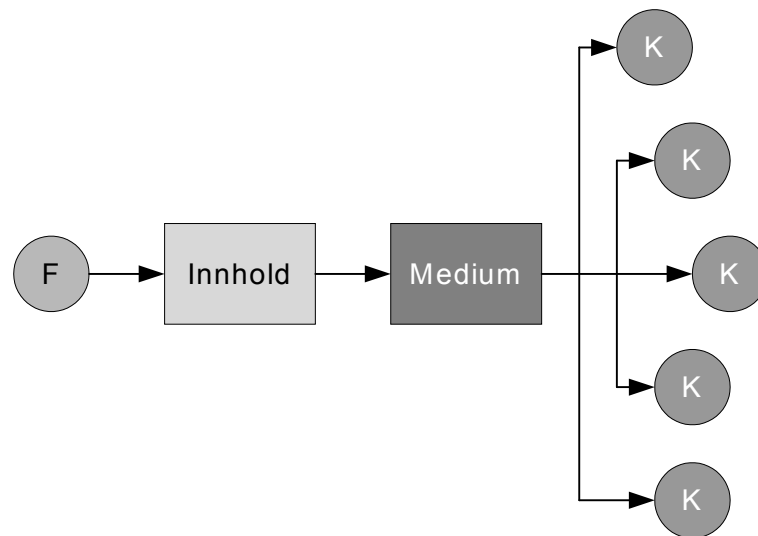
For å gjøre adressen kjent for kunden kan vi nytte flere former for markedsføring. Vi kan bruke tradisjonelle medier for markedsføring: TV, radio, aviser, magasiner etc. Vi kan også bruke internett for å nå potensielle kunder.

Vi kan gjøre adressen kjent på nettet gjennom etablerte søkemotorer som for eksempel AltaVista, Fast Search, Kvasir eller Yahoo! [ALT01, FAS01, KVA01, YAH01]. Dette forutsetter at tilstrekkelig metadata er integrert i hjemmesidene for virksomheten. Vi kan også gjøre adressen kjent vha bannere og knapper på andre hjemmesider som har stort antall besøkende (for eksempel nettaviser, nettbanks osv).

Om vi har fanget interessen, hos potensielle kunder, vil neste skritt være å gjøre det vi kan for å beholde den. Om vi skal beholde interessen må vi ta hensyn til egenskapene i nettet. Dette inkluderer blant annet nettes mulighet for interaktiv kommunikasjon.

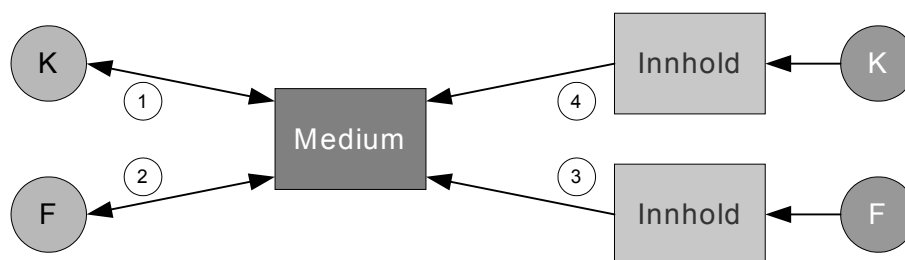
Tepfers og Davidsen [TD00] fremhever at tradisjonelle modeller for kommunikasjon har vært forankret i kommunikasjon med *en-til-mange*. Forhandler har et budskap som blir formidlet gjennom et medium til mange mottakere. Budskapet blir sendt til kunden og kunden kan ikke kommunisere med avsender. Dette kaller vi enveis kommunikasjon.

I figur 2-1 er en modell skissert for denne type kommunikasjon. I modellen er F forhandler og K kunde. Vi kan lese modellen slik: F sender sitt Innhold gjennom Medium til K. Innhold i Medium kan være både statisk (tekst og bilder) og dynamisk (video og lyd).



Figur 2-1 – Tradisjonell kommunikasjonsmodell [TD00].

Kommunikasjon med *en-til-mange* er også mulig på Internett. Men siden mediet gir mulighet til å kommunisere interaktivt, og kommunisere *mange-til-mange* (se figur 2-2), gir det mange nye muligheter. Forholdet mellom forhandler og kunde er ikke lenger det sentrale. Men det er selve mediet i mellom som er sentralt.



Figur 2-2 – Ny kommunikasjonsmodell for web [TD00].

Vi kan lese figur 2-2 slik: Kunder K samhandler med mediet (1). Kunder K kan for eksempel surfe på Internett. Det kan også forhandlere F gjøre (2). I denne sammenheng kan vi benytte markedsføring mellom bedrifter. Forhandler F kan i tillegg opprette egne tjenester som gir innhold om produkter og tjenester til mediet (3). Dette kan også kunder K gjøre, ved å samle inn informasjon på egne hjemmesider (4) [TD00].

Å forstå egenskaper for mediet, med muligheter og begrensninger, er viktig ved utvikling av løsninger for e-handel på web.

2.2.2 Strategier for e-handel

For å lage en god strategi, som understøtter e-handel i en virksomhet, er det altså nødvendig å forstå mediet og dets egenskaper. I prosessen med å utarbeide slike strategier bør vi vurdere tre områder [MT00]:

- *brukererfaringer* ved interaksjon med innholdet.
- *struktur* på web-sider som understøtter brukererfaringer.
- *interne prosesser* – arbeidsflyt og relasjoner i produksjon som støtter brukererfaringer og struktur på web-sidene.

De neste tre tabellene gir en oversikt over underliggende strategier innen *brukererfaring*, *struktur* og *prosesser*.

Tabell 2-1 – Strategier for brukererfaringer [MT00].

Strategi	Beskrivelse	Fordeler
Tilby medlemskap	Tilbyr registrering på aktuell web-side som gir tilgang til spesielt innhold og tjenester	"Bygge opp samfunn", målrettet markedsføring og mulighet til påmelding for medlemskap
Personalisere brukers erfaringer	Presenterer innhold som passer brukers tidligere klikk og visninger på web	"Bygge opp samfunn", målrettet handelstilbud, og kunde og web-side lojalitet
Støtte bruker	Tilby rask og tilpasset kundeservice	"Bygge opp samfunn", kunde og web-side lojalitet, og repeterende handel
Kommunisere via samfunn	Tilby bruker å kommunisere med hverandre og publisere gjennom web-side	"Bygge opp samfunn", og kunde og web-side lojalitet
Belønne bruker	Tilby bruker belønning for å besøke og bruke web-side	Kunde og web-side lojalitet, fremme mulighet for salg og kryssalg
Effektiv markedsføring	Fremme web-sidens innhold og produkter uten å støte fra seg nåværende eller potensielle kunder	Kunde og web-side lojalitet, fremme salg og mulighet for kryssalg
Sette opp smarte relasjoner	Etablere relasjoner med både private (kunder) og kommersielle publisere på web	Kunde og web-side lojalitet og ny inntektsstrøm

I tabell 2-1 er sammenheng mellom brukererfaringer og valg av strategi skissert [MT00]. Tabellen viser at strategiene søker å markedsføre web-siden og "bygge et samfunn" omkring hjemmesiden. På denne måten kan vi skape lojalitet til web-siden, som igjen gir mulighet for salg og evt kryssalg av produkter og tjenester.

Tabell 2-2 – Strategier for struktur på web-sider [MT00].

Strategi	Beskrivelse	Fordeler
Bruk av konsistent og klar navigasjon	Tilby konsistent design- og navigasjonsstrategi for web-side	Kunde og web-side lojalitet
Støtte personalisering	Kategorisere innhold som formulerer en persons profil	Kunde og web-side lojalitet, fremme mulighet for produkt- og kryssalg
Strømlinjeformet prosess	Minimalisere anstrengelser ved betaling av produkter eller registrering på web-side	Handel og medlemsprogrammer
Praktisk åpenhet om bruk av informasjon	Presentere nettbruker, med en kunngjøring, om hva du har tenkt å bruke informasjon til	Web-side og medlemslojalitet
Passende administrasjon for rettigheter	Velge passende metode for administrasjon av rettigheter	Økt handel

Tabell 2-2 viser alternative strategier for struktur på web-sider. Tabellen viser at vi bør unngå at bruker blir frustrert over navigasjonen på sidene. På denne måten kan vi motivere bruker til flere nye besøk, som igjen gir mulighet for økt markedsføring, presentasjon og salg.

Tabell 2-3 – Strategier for forbedring av prosesser [MT00].

Strategi	Beskrivelse	Fordeler
Forstå løsningen for innhold og e-handel	Forstå hvordan publisering og systemer for e-handel løser ulike problemer for kunder	Større grunnlag for evaluering av verktøy
Effektiv administrasjon av prosessrelasjoner	Automatisere manuelle prosesser (arbeidsflyt) i produksjonen	Raskere oppdatering av innhold, mindre feil ved leveranse
Utnytte fordel av forretningsinformasjon	Påvirke resultatet fra surfing og kjøp til finjustering av navigasjon og strategier for handel	Handel fra målrettet tilbud, proaktiv planlegging
Utvikle skalerbare administrasjonsprogrammer for forbindelser og syndikater	Tilsluttede relasjoner er lønnsomme bare om de kan administreres automatisk	Handel for innholdssider, større gjenkjenning av merke

Tabell 2-3 viser sammendrag av strategier for forbedring av interne prosesser eller arbeidsflyt i virksomheten. Sammendraget viser hvilke fordeler vi kan oppnå ved å forstå løsningen både for kunde og virksomhet. Å gjennomføre slike strategier gir store muligheter for å automatisere manuelle oppgaver.

Det eksisterer flere alternative strategier for e-handel. Charles Trepper [CT00] tar utgangspunkt i "Microsoft's E-Commerce Strategy". Disse strategiene har nær kobling til Microsofts produktportefølje for e-handel. Ved utforming av strategier skiller Trepper mellom B2C og B2B. Tabell 2-4 gir en oversikt over hva Trepper retter søkelyset mot i arbeidet med strategier for e-handel.

Tabell 2-4 – Sammendrag av strategier for e-handel fra Charles Trepper [CT00].

Strategi	Beskrivelse	Fordeler
Personalisering	Analysere og forstå hvordan bruker nytter hjemmeside	Kan bruke data om kunder til målrettet reklame, annonsering per e-post, prisanalyse etc
Enkel navigering	Utarbeide en hjemmeside som er funksjonell og som gir rask navigering	Bruker oppnår rask kontakt med bedrift og bruker kommer tilbake senere
Kundetilpasninger	Lage en hjemmeside med fokus på kundens behov, og som kunden kan bruke umiddelbart	Kunde får et positivt inntrykk av hjemmeside, og kommer tilbake til siden
Konsistens	Gjennomgående design og navigering for alle nivå på hjemmesiden	Kunde oppfatter hjemmesiden som profesjonell
Kunnskapsforvaltning	Lage hjemmeside som gir mulighet for å tilføre virksomheten ny kunnskap	Utvikler virksomheten til å bli bedre, og gir konkurransefortrinn i forhold til konkurrenter
Merkebygging	Skape et nytt eller videreutvikle eksisterende merkenavn	Skaper lojalitet til hjemmesiden

Om vi sammenligner nevnte strategier for e-handel [MT00, CT00] er det flere likheter. Dette viser, til en viss grad, at slike strategier er akseptert, og at de blir brukt av flere store internasjonale og nasjonale aktører.

Vi bør ikke se e-handel isolert i en virksomhet eller konsern. Vi bør også være bevisst hvilke mål og strategier som gjelder nasjonalt og internasjonalt. Dette har betydning for hvordan vi bør utforme løsninger for e-handel i en virksomhet.

Et eksempel på dette er resultatet av et prosjekt i offentlig sektor i Sverige. I rapporten "Strategier for elektronisk handel i offentlig sektor [SI99a]" fremhever forfatter at det har vært en storstilt og ambisiøs satsning i Sverige. Satsningen har også vært stor i internasjonal målestokk. Rapporten fremhever at vi kan lære mye av erfaringene fra Sverige. Forfatteren av rapporten mener at også Norge bør fokusere på å:

- utarbeide felles standard for elektronisk handel.
- tilpasse e-handel til rettslige reguleringer.
- sette i gang opplysningsvirksomhet/kampanje om e-handel rettet mot små og mellomstore bedrifter (SMB).

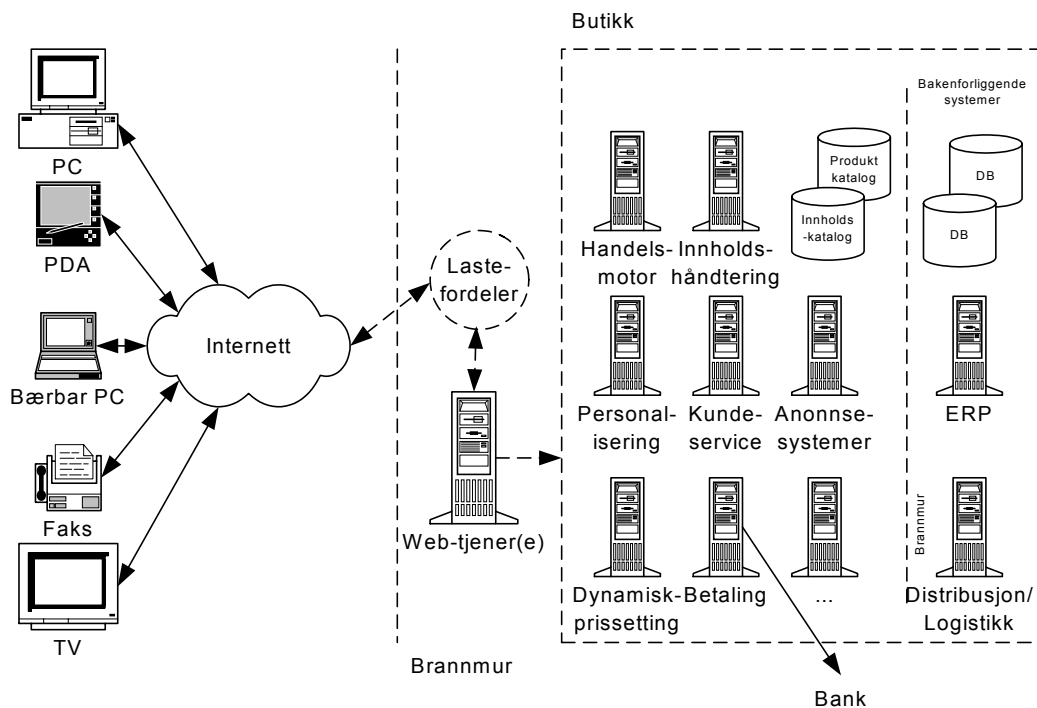
2.3 Arkitektur

Systemer for e-handel består ofte av flere underliggende systemer. Hensikten med systemene er direkte eller indirekte å understøtte salgsprosessen. Utfordringen i denne sammenheng er å identifisere hvilke komponenter vi har behov for og sette de sammen til et helhetlig system.

2.3.1 Kategorier

Ulike behov skaper ulike løsninger for e-handel, og behovene former igjen arkitekturen for underliggende systemer. Systemer, som er basert på B2C, B2B eller B2P, krever forskjellig arkitektur. Eksempler på underliggende systemer er varekataloger, handlekurver, ordrebehandling, betaling og kunderegistrering. I tillegg må vi ha systemer for administrasjon og vedlikehold.

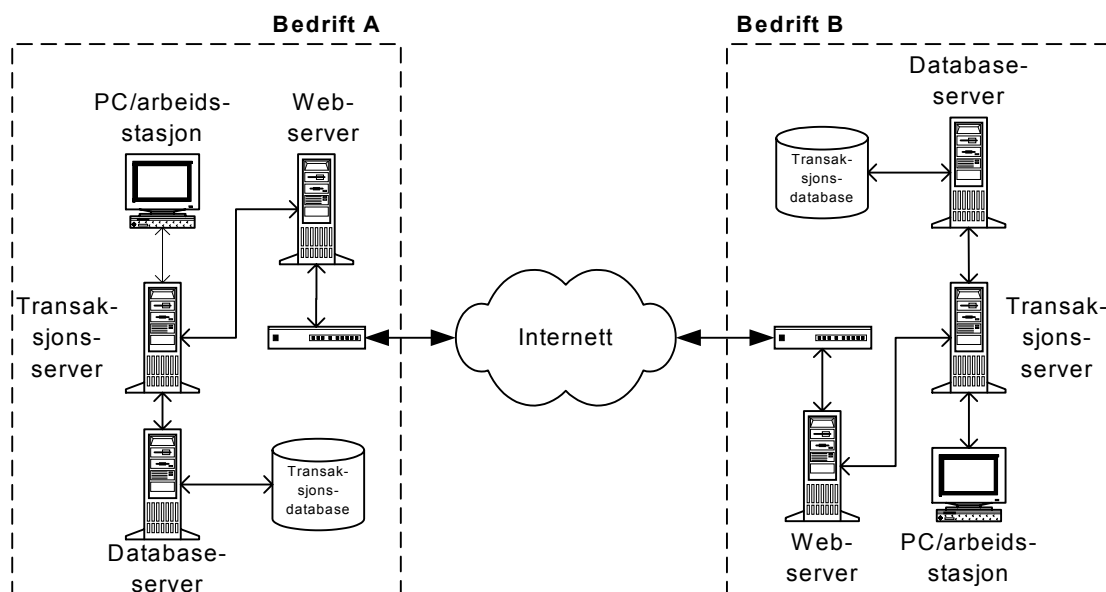
Tepfers og Davidsen [TD00] har skissert en mulig løsning for e-handel og en evt sammenheng mellom underliggende systemer. Figur 2-3 gir en oversikt over hvilke elementer slike systemer *kan* bestå av og hvordan vi *kan* sette disse sammen.



Figur 2-3 – Eksempel på arkitektur for B2C [TD00].

Tepfers og Davidsen fremhever at kunden, i slike løsninger, må kunne kommunisere med nettbutikken gjennom flere forskjellige kanaler. Eksempler på slike kanaler er TV, faks, telefon, mobiltelefon, PC eller PDA.

Trepper [CT00] skisserer et annet forslag til arkitektur. Den er basert på handel mellom organisasjoner (B2B). Figur 2-4 viser en enkel skisse over en B2B-arkitektur.



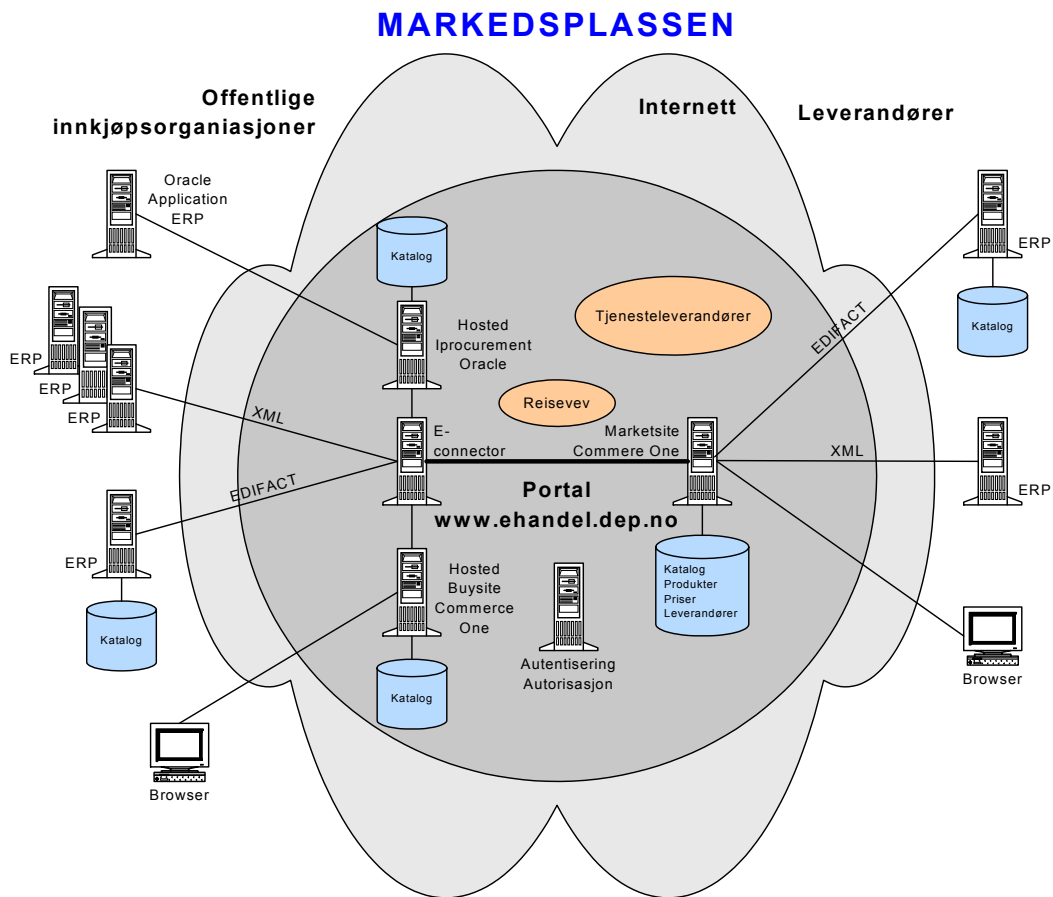
Figur 2-4 – Eksempel på arkitektur for B2B [CT00].

B2B-arkitektur skiller seg noe fra arkitekturen for B2C. Løsningene bruker samme elementer (PC-er, servere, rutere, kommunikasjonslinjer etc), men er koblet sammen på en annen måte. I tillegg kan elementene ha forskjellige oppgaver. Bedrifter med B2B kommuniserer med hverandre gjennom få medier (kanaler), og det er normalt at det på forhånd er avtalt hvordan bedriftene skal kommunisere.

Det eksisterer også andre forslag til arkitektur for systemer innen e-handel. Skissen i figur 2-3 er et eksempel på et system for samhandling mellom bedrift og kunde (B2C) og figur 2-4 viser et system som fokuserer på samhandling mellom bedrifter (B2B). Et tredje eksempel på arkitektur er samhandling mellom bedrifter og myndigheter (B2P).

Telenor ASA [TEL01] har gjennom arbeidet med Markedsplassen [AA99a] utformet et forslag til arkitektur for en slik B2P-løsning [TES00]. Markedsplassen vil være et møtested for offentlige kjøpere og leverandører (se figur 2-5).

Forslaget er basert på utvidet bruk av kommersielle tilgjengelige løsninger. I tillegg er den basert på flere etablerte standarder og spesifikasjoner. En annen interessant observasjon, i dette forslaget, er hvordan Telenor ser for seg bruk av kataloger.



Figur 2-5 – Eksempel på arkitektur for B2P [TES00].

Telenor skisserer også utbredt bruk av systemer for Enterprise Resource Planning (ERP).

ERP er en samling av applikasjoner. Applikasjonene binder sammen flere funksjoner i et selskap (for eksempel finans, produksjon, salg, personal etc) inn til en database [SAP01b].

I denne sammenheng er ERP brukt både av offentlige organisasjoner for innkjøp og leverandører. Vi kan bruke systemer for ERP til å koble sammen flere underliggende systemer med ulike behov for data og format. Om vi tar i bruk ERP, for å integrere datakilder, må vi foreberede oss på å gjøre noen tilpasninger selv [CT00].

Vi kan altså bruke ERP-systemer for å koble sammen flere datakilder med ulike format. Sett i relasjon til katalogsystemer og e-handel er denne observasjonen meget interessant.

2.3.2 Sikkerhet

Foruten økonomiske, tekniske og juridiske problemstillinger er sikkerhet sannsynligvis den største bekymringen for virksomheter innen e-handel. Charles Trepper [CT00] fremhever at problemet med sikkerhet i e-handel er at

mennesker tenker først og fremst lokal sikkerhet, og forstår ikke risikoen ved å gjøre virksomheten tilgjengelig på Internett. Et annet problem er at ledere benekter for at insidere utøver en sikkerhetsmessig risiko.

I de neste avsnittene vil jeg omtale noen mekanismer som vi kan benytte for å gjøre transaksjoner mellom kunde og kjøper mer sikker:

- Secure Electronic Transaction standard
- Entrust/Direct
- Entrust/TruePass
- Smartkort

Secure Electronic Transaction standard (SET)

SET er en åpen teknisk standard for handel. Spesifikasjonen er utviklet av VISA [VIS01] and MasterCard [MAS01]. Hensikten med standarden er gjøre transaksjoner, med betalingskort over Internett, sikrere [SET01].

Sertifikater og kryptering blir brukt, til å gi tillit (trust), gjennom hele kjeden med transaksjoner. Standarden fokuserer på konfidensialitet¹ og autentisering. SET har en egen ordning for sertifisering. For å få denne må virksomheten bestå en SET Compliance Test [LAN01].

Entrust/Direct

Entrust/Direct er en web-basert applikasjon. Den forsterker tradisjonell sikkerhet på web gjennom et sett med proxy-er som etablerer en autentisert og sikker sesjon mellom klient (nettleser) og server (web-tjener). Etableringen skjer vha digitale sertifikater [TES00].

Ved bruk av digitale sertifikater skjer lokal innlogging mot profil for sertifikater og autentisering skjer mot web-tjener. Nøkkel- og sertifikathåndtering, i applikasjoner fra Entrust, er automatisk og transparent² for bruker.

Entrust/Direct gir også mulighet for å nytte permanente digitale signaturer. Signaturen for bruker blir lagret i "bakre" systemer sammen med innhold i transaksjonen. Dette gir blant annet mulighet for sporbarhet i ettertid.

Direct gir en sentralisert løsning for brukeradministrasjon. Administrasjon skjer sentralt gjennom administrasjon av passordregler, bestemmelse av lovlig algoritmer for kryptering, bestemmelse for metoder for identifikasjon mm.

Entrust/TruePass

TruePass gir, på samme måte som Entrust/Direct, en forsterkning av sikkerheten på web. Applikasjonen bruker en Java-komponent som blir tilknyttet bruker. Dette gjør at det ikke er behov for installasjon av dedikert programvare. Java-komponenten gjør bruker maskinuavhengig.

¹ Konfidensialitet - "konfidensiell art (el. beskaffenhet)" eller "evne til å holde konfidensielle opplysninger for seg selv" [KUN01]

² Transparent - "gjennomsiktig", "opplagt" eller "åpenbar" [KUN01]

TruePass er godt egnet for mobile brukere, hvor vi ønsker å opprettholde permanente digitale signaturer og kryptering, og sentral administrasjon av rettigheter for bruker [TES00].

Smartkort

Nærings- og handelsdepartementet (NHD) [NHD01] definerer og forklarer smartkort slik:

Plastkort (typisk bankkortstørrelse) som inneholder en eller flere mikroprosessorer, og som kan kommunisere med datamaskiner o.l. gjennom en smartkortleser. Et smartkort kan lagre informasjon på en sikker måte, og kan inneholde funksjoner for kryptografi.

Smartkort vil både kunne beskytte hemmelige nøkler og gjennomføre sikkerhetskritiske funksjoner for kryptografi. ISO 7816 [ISO01] spesifiserer de fysiske egenskapene for kortet og kommunikasjon med kortterminal. Per i dag eksisterer det flere typer smartkort – fra minnekort for enkel lagring av data (telefonkort) til prosesserbare kort med en eller flere applikasjon på kortet.

Det foregår et arbeid for standardisering på området. En ny versjon av SET (C-SET) vil være basert på Smartkort. Microsoft [MS01], Netscape [NET01] mfl er enige om et felles grensesnitt for kommunikasjon med slike kort. På bakgrunn av dette kan vi forvente at smartkortlesere blir integrert som en del av en standard PC.

Sikkerhet er et meget sentralt tema i e-handel. Utvikling og implementasjon av standarder og mekanismer som SET, Entrust/Direct og TruePass, og Smartkort gir oss større trygghet og bedre sikkerhet ved handel på Internett.

2.4 Standarder og spesifikasjoner

Det eksisterer flere spesifikasjoner og standarder som har påvirket og fortsatt påvirker utviklingen av tekniske løsninger for e-handel. Uavhengig av hvilke løsninger, som en virksomhet vurderer, bør nasjonale og internasjonale spesifikasjoner og standarder være med i vurderingen. Dette avsnittet vil omtale *noen* sentrale spesifikasjoner og standarder for e-handel i dag.

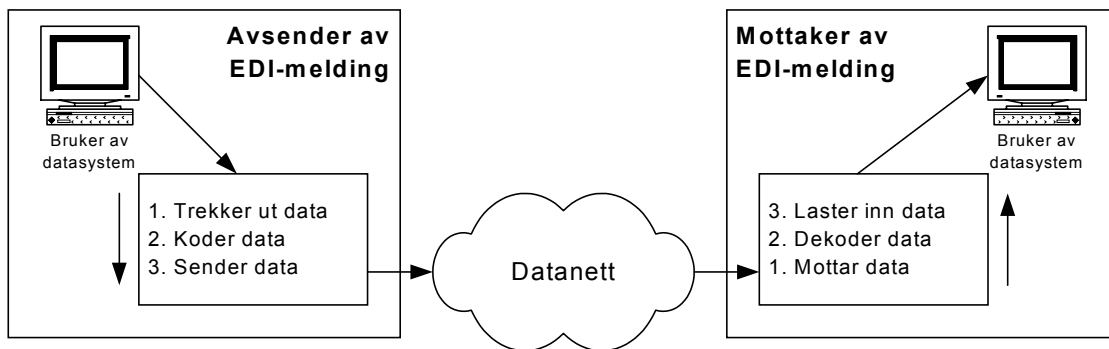
2.4.1 EDI og EDIFACT

Electronic Data Interchange (EDI) er en benevnelse for elektronisk utveksling av meldinger. EDI-meldinger har tradisjonelt vært benyttet mellom virksomheter med data- og informasjonsflyt mellom hverandre (for eksempel elektronisk handel). Hensikten med formatet har vært å effektivisere meldingsutveksling, og tilhørende forretningsprosesser, mellom etablerte forretningsforbindelser [NR01].

Grunnideen bak EDIFACT er at ulike behov og krav skal dekkes gjennom en global standard. Standarden ble opprinnelig utviklet av United Nations EDI For

Administration, Commerce and Transport (UN/EDIFACT). Dette arbeidet startet på midten av 1980-tallet [AA00]. ISO betegner standarden som ISO 9735 [NR01]. Det eksisterer også en amerikansk standard for EDI. Standarden blir kalt ANSI X12 (X12). Det er utarbeidet flere versjoner av X12 og EDIFACT, og det er visse problemer med kompatibilitet mellom disse [RMH01].

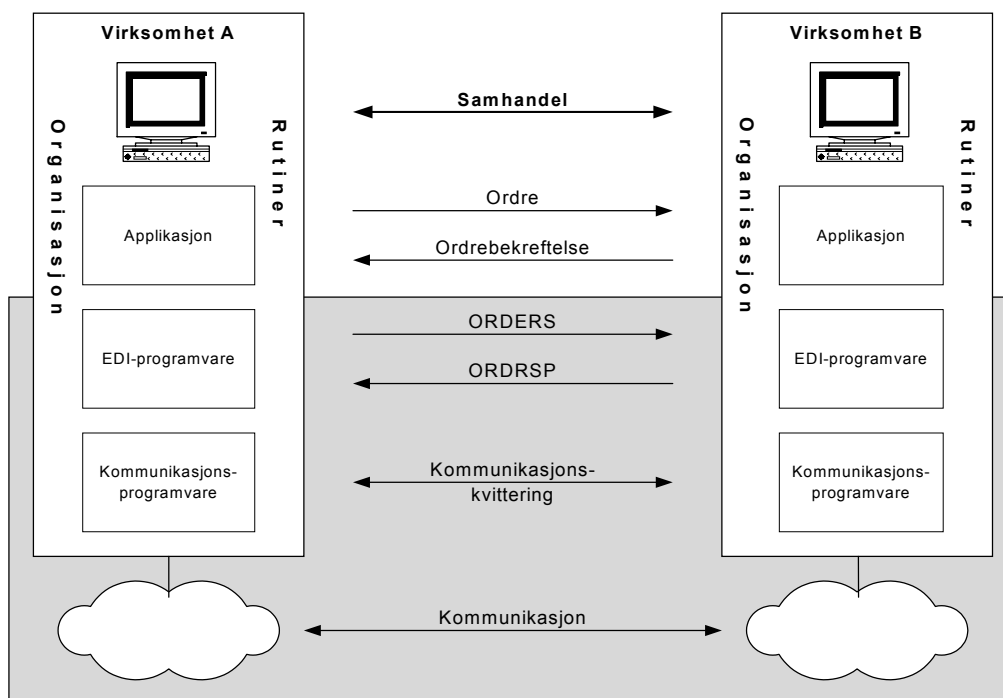
Figur 2-6 skisserer hvordan virksomheter kan utveksle EDI-meldinger.



Figur 2-6 – Eksempel på utveksling av EDI-meldinger mellom organisasjoner [NR01].

Tradisjonell EDI blir betraktet som en-til-en relasjon. Ved utvidet bruk av Internett, som gir mulighet for flere former for relasjoner (en-til-mange eller mange-til-mange), gir dette klare begrensninger. Om to parter ønsker å utveksle EDI-meldinger, etter UN/EDIFACT-standard, må begge parter avtale ytterligere presisering av standarden. Dette arbeidet er tidkrevende og kostnadsdrivende for begge parter. I tillegg er applikasjoner for produksjon og tolkning av EDIFACT-meldinger kostbare å implementere og vedlikeholde.

Figur 2-7 viser hvilken plass EDI-programvare har i infrastruktur for en bedrift.



Figur 2-7 – Komponentene i en EDI-løsning [SIN01].

Formatet for EDIFACT er bygd opp med en hierarkisk struktur av data. Strukturen er delt i sju nivå. Simple Data Element er på laveste nivå, mens Interchange er på høyeste nivå i strukturen (se tabell 2-5).

Tabell 2-5 – Oppbygging av EDI EDIFACT meldingsformat [EP01].

Element	Beskrivelse
Interchanges (høyeste nivå)	Interchange elementet er det komplette EDI-dokument. Det består av mange deler, og inkluderer blant annet dato og tid for utveksling og informasjon om adresse til mottaker.
Functional Groups	Elementet Functional Group blir vanligvis ikke brukt. Det fleste selskaper foretrekker at overføringer kun inneholder en melding.
Message	Message er bygd opp på bakgrunn av flere segmentgrupper og individuelle grupper. En overføring kan bestå av flere meldinger, men da bør alle meldingene være av samme type (for eksempel bare ordrer). Om meldingene består av ulike typer skal vi benytte Functional Groups.
Segment Groups	Segmenter kan grupperes til å forme en blokk med relaterte data. For eksempel navn- og adressesegmentet kan bli gruppert inn i et segment med andre segmenter. Segmentet Group er enten obligatorisk eller betinget i meldingen.
Segments	<p>Et EDI segment består av en blanding av enkel og sammensatte elementer. Følgende linje er et eksempel på Segments:</p> <p>NAD+II+1234:160:ZZZ+To:eProsper Ltd.:AshHouse, Ashleigh Way:Plymouth,Devon:UK+++++GB'</p> <p>Alle segmenter har en identifikator på tre karakterer. Eksemplet ovenfor har segmentnavn NAD og representerer Name og ADress segment. NAD er et av mange segmenter som kan bli satt sammen i en melding. Etter NAD segmentet følger en attributt (II), og er alltid det første segment etter NAD-segmentet. Attributten angir at data i segmentet er navn og adresse til utsteder av faktura.</p> <p>Segmentet består av flere karakterer og er brukt for skille innholdet i segmentet. Apostrof (') angir at segmentet avsluttes, mens pluss (+) separerer dataelementer.</p> <p>Segmentet er enten obligatorisk eller betinget i meldingen.</p>
Composite Elements	<p>Composite Element er en gruppe beslektede enkle elementer. For eksempel en brukers adresse består av et visst antall linjer: "eProsper Ltd.", "Ash House, Ashleigh Way", "Plymouth", "Devon", "UK". Dette er fem enkle elementer med en sammenheng. I EDI-meldinger blir elementene skilt med " : " .</p> <p>Composite Element kan bestå av fem enkle elementer.</p> <p>Composite Elements er enten obligatorisk eller betinget i meldingen.</p>
Simple Data Element (laveste nivå)	<p>Simple Data Element er en type element definert av standardens kropp. For eksempel en postkode i en adresse. Data i elementet kan ikke brytes opp i mindre elementer.</p> <p>Simple Data Element er enten obligatorisk eller betinget i meldingen.</p>

Nedenfor følger to eksempler på EDI/EDIFACT-meldinger. Første eksempel er formatert med linjeskift (se figur 2-8). I eksemplet er linjeskift brukt ved segment separator (') [EP01].

```

UNB+UNOA:1+00987654:ZZZ+123456:ZZZ+990831:1617+000244++INVOICE'
UNH+063297+INVOIC:D:94B:UN'
BGM+380:I2::F2272210+083838158'
DTM+9:19990831161743:203'
DTM+3:19990831:102'
DTM+4:19990825:102'
FTX+IIN+4++Not Insured'
FTX+CUR+4++PL 13 SAP'
FTX+PMT+4++3 Months'
RFF+ACL:0740'
RFF+AAJ:DL1235225'
RFF+AOB:234232747'
RFF+SM:60'
RFF+ON:223400162'
NAD+IV+1221231:160:ZZZ+To:A Place:Somewhere::1123343++23 A
Street+CityName++Z12 12Z+GB'
RFF+VA:GB123456789'
NAD+UD+00012318:160:ZZZ++++++GB
NAD+II+00123221:160:ZZZ++++++GB
CTA+IC+AS-12:Nicholson'
COM+01752/313224:TE'
COM+01752/313218:FX'
PAT+22+See Date:134::5082'
DTM+13:19991125:102'

```

Figur 2-8 – Eksempel på EDI-melding (opprinnelig eksempel er lengre) [EP01].

EDI-meldinger blir vanligvis sendt som en lang streng (se figur 2-9).

```

UNA:+.? 'UNB+UNOA:3+12345678:ZZ+00213421:ZZ+000629:
1556+001393++ANORDER' UNH+077284+ORDERS:001:911:UN'
BGM+105:::A12345+2000/00123'DTM+4:20000628:102'DTM
+9:20000628154718:203'DTM+10:20000831:102'FTX+DEL+
4++Deliver to the Green Door:Floor?:Room 4:Atten
tion:Ian Crawford+GB'FTX+SUR+4++PO 5 SAP EDI 3. Ru
n. Delivery Prorität=HOCH+DE'FTX+SPH+4++Y:F'RFF+AE
H:Quick'RFF+CP:A'NAD+SU+00231283:160:ZZZ++++++DE'
NAD+IU+23425222:160:ZZZ++++++GB'NAD+DP++eProsper
Ltd.:Ash House, Ashleigh Way:Plymouth:Devon:UK'NAD
+UD+002+Ian Crawford:EDI Services'UNS+D'LIN+10++10
74059001:SA'QTY+21:10'LIN+20++0023045002:SA'QTY+21
:1'LIN+30++9980012001:SA'QTY+21:129'UNS+S'CNT+2:10
'UNT+23+077284'UNZ+1+001393'

```

Figur 2-9 – Eksempel på en uformatert EDIFACT-melding [EP01].

EDI er implementert i mange bransjer. Eksempler på slike bransjer er: handel, produksjon, transport, fortolling, forsikring, administrasjon, betalingsformidling, reiseliv, fiskeri, helse, trygd, undervisning, statistikk osv [TD00, NR01].

Vekst og utbredelse av tradisjonell EDI har ikke blitt slik mange hadde forventet. Økende utbredelse og bruk av Internett blir sett på som en mulighet for å løse utestående problemer med EDI og EDIFACT. Det blir påpekt at EDI og Internett ikke skaper konflikter for hverandre, men er teknologi som utfyller hverandre [AA00].

Ved vurdering av løsninger for e-handel bør vi vurdere behovet for støtte av EDI/EDIFACT.

2.4.2 XML

Extensible Markup Language (XML) er en spesifikasjon for lagring og utveksling av data over Internett. Spesifikasjonen ble utarbeidet på oppdrag fra World Wide Web Consortium (W3C) i 1996. W3C definerer XML som en underliggende del av Standard Generalized Markup Language (SGML). SGML er et standard markeringsspråk (markup language) for dokumenter [NR01, XM01, WI01].

XML ligner Hypertext Markup Language (HTML), men det er klare forskjeller mellom språkene. XML bruker tagger og attributter til å definere innhold (data) i dokumenter. I HTML bruker vi tagger til å definere formateringer. Om vi bruker HTML må vi forholde oss til et fast sett med tagger, mens vi med XML selv kan definere ønskede antall tagger med tilhørende innhold. Taggene i XML gir semantisk informasjon for elementene de er tilknyttet. Dette gir oss informasjon om hva elementet inneholder, og en applikasjon har dermed mulighet for å forstå data i XML-dokumenter [NR01, WIN01].

Eksempel på syntaks i XML er vist i figur 2-10. Figuren viser en periodisk tabell for to elementer (hydrogen og helium).

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xml" href="14-2.xsl"?>
<PERIODIC_TABLE>
  <ATOM STATE="GAS">
    <NAME>Hydrogen</NAME>
    <SYMBOL>H</SYMBOL>
    <ATOMIC_NUMBER>1</ATOMIC_NUMBER>
    <ATOMIC_WEIGHT>1.00794</ATOMIC_WEIGHT>
    <BOILING_POINT UNITS="Kelvin">20.28</BOILING_POINT>
    <MELTING_POINT UNITS="Kelvin">13.81</MELTING_POINT>
    <DENSITY UNITS="grams/cubic centimeter"><!-- At 300K -->
      0.0899
    </DENSITY>
  </ATOM>
  <ATOM STATE="GAS">
    <NAME>Helium</NAME>
    <SYMBOL>He</SYMBOL>
    <ATOMIC_NUMBER>2</ATOMIC_NUMBER>
    <ATOMIC_WEIGHT>4.0026</ATOMIC_WEIGHT>
    <BOILING_POINT UNITS="Kelvin">4.216</BOILING_POINT>
    <MELTING_POINT UNITS="Kelvin">0.95</MELTING_POINT>
    <DENSITY UNITS="grams/cubic centimeter"><!-- At 300K -->
      0.1785
    </DENSITY>
  </ATOM>
</PERIODIC_TABLE>
```

Figur 2-10 – Eksempel på XML-syntaks [ERH99].

Strukturen i XML-dokumenter kan være komplekse. Vi kan oppfatte XML-dokumenter som kompleks fordi vi bruker tagger til å definere elementer som igjen kan ha andre tagger og elementer. Dette gir mulighet for å definere datastrukturer som i et programmeringsspråk.

Vi definerer struktur (dokumenttype) i XML-dokumenter ved hjelp av Document Type Definition (DTD). I DTD-en definerer vi hvilke tagger vi kan bruke og hvilken grammatikk dokumentet skal følge.

Vi kan definere egne *markeringsspråk* for ulike fagområder (for eksempel Mathematical Markup Language, MathML og Chemical Markup Language). Språkene forenkler utvekslingen av data og påvirker utviklingen av mer standardiserte dokumenter [NR01].

XML skiller klart mellom *innhold/struktur* og *layout/presentasjon*. XML lagrer innholdet for seg, og er strukturert og semantisk merket (vha tagger). Layout/presentasjon blir definert separat. Vi kan for eksempel bruke eXtensible Style Language (XSL) til å spesifisere hvordan XML-dokumenter skal bli presentert. XSL er et stilspråk, som har samme syntaks som XML, og er basert på en separat standard med navn The Dataset Sublanguage (DSSL).

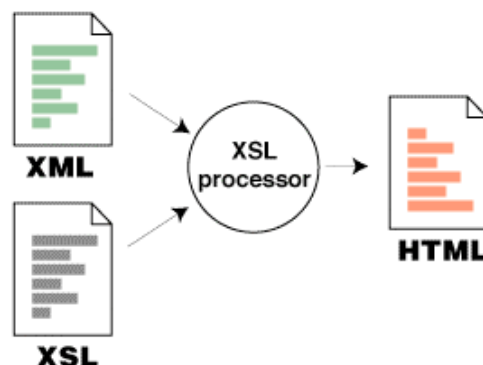
Norsk Regnesentral [NR01] fremhever at XML og XSL alene ikke er nok for produksjon av dokumenter på Internett. W3C arbeider også med *relaterte standarder* for metadata, dokumenttyper, formatering, lenker og lignende.

Figur 2-11 viser syntaks for XSL for periodisk tabell (se figur 2-10).

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="PERIODIC_TABLE">
    <html>
      <xsl:apply-templates/>
    </html>
  </xsl:template>
  <xsl:template match="ATOM">
    <P>
      <xsl:apply-templates/>
    </P>
  </xsl:template>
</xsl:stylesheet>
```

Figur 2-11 – Eksempel på syntaks i XSL for periodisk tabell [ERH99].

Figur 2-12 skisserer en løsning over hvordan vi kan koble XML og XSL for å produsere en presentasjon i HTML.



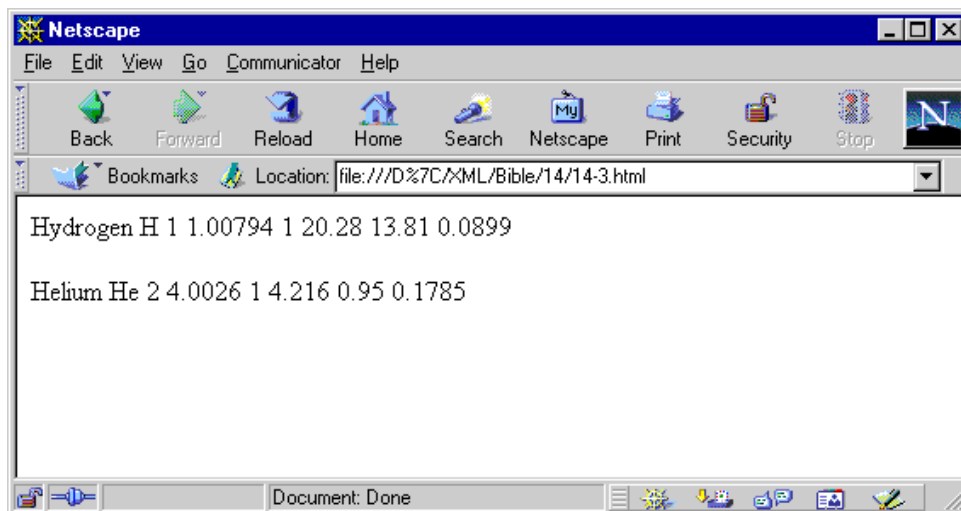
Figur 2-12 – Sammenkobling av XML og XSL gir mulighet for visning med HTML [NR01].

Koblingen av XML og XSL, gjennom en for eksempel XSL-prosessor, gir altså mulighet til å produsere HTML-dokumenter for Web. Ved å koble XML-syntaks i figur 2-10 og XSL-syntaks i figur 2-11 får vi HTML-syntaks som vist i figur 2-13.

```
<html>
  <P>
    Hydrogen
    H
    1
    1.00794
    20.28
    13.81
    0.0899
  </P>
  <P>
    Helium
    He
    2
    4.0026
    4.216
    0.95
    0.1785
  </P>
</html>
```

Figur 2-13 – HTML-kode produsert ved hjelp av XML- og XSL-kode [ERH99].

Om vi overfører koden til en nettleser får vi et skjermbilde som vist i figur 2-14.



Figur 2-14 – Web-side (i HTML) produsert ved hjelp av XML og XSL [ERH99].

Det er en rivende internasjonal utvikling av XML. Og på samme måte som for EDI, eksisterer det en rekke forskjellige standarder og versjoner av XML.

Det er minst fem XML-standarder under utvikling (fra juni 1999) [RMH01]:

- Commerce Net – utvikler ECO. Commerce Net er et konsortium med flere selskaper.
- Rosetta Net – utvikler en versjon av XML for produktkataloger. Rosetta Net er også et konsortium.
- Commerce One - utviklet et felles bibliotek for bedrifter. Utviklingen er gjort med støtte fra US National Institute for Standards and Technology (NIST).
- Ariba – samlet flere selskaper omkring commerce XML (cXML). cXML er et forslag til en standard for kataloger og ordrebestilling.
- Microsoft - har gruppert flere av disse teknologiene under konseptet BizTalk.

XML har sterk fokus på innhold. Språket er derfor godt egnet i løsninger for e-handel. Dette gjør XML meget interessant i utviklingen av fremtidige systemer for e-handel.

2.4.3 Integrasjon EDI og XML

Internett og XML har redusert barrierene for e-handel med hensyn på både kostnader og kompleksitet. Vi må ikke se fremgangen for XML som slutten på EDI. XML erstatter ikke EDI, men gir oss mulighet til å bruke systemer for e-handel i nye segmenter. Nye segmenter kan være små og mellomstore bedrifter. XML komplimenterer EDI, og på den måten kan vi revitalisere visjonen for EDI [RMH01].

Ricker, Munro and Hopeman [RMH01] fremhever at EDI og XML er forskjellig. Utviklerne av EDI fokuserte mye på størrelsen til meldingene. Dette fordi båndbredde i EDI-nettverk kostet mye, og gjør det fortsatt i dag. Metadata i meldingene er tatt vekk for å redusere størrelsen. Dette gjør EDI-meldinger vanskelig å lese og feilrette (se figur 2-8 og figur 2-9).

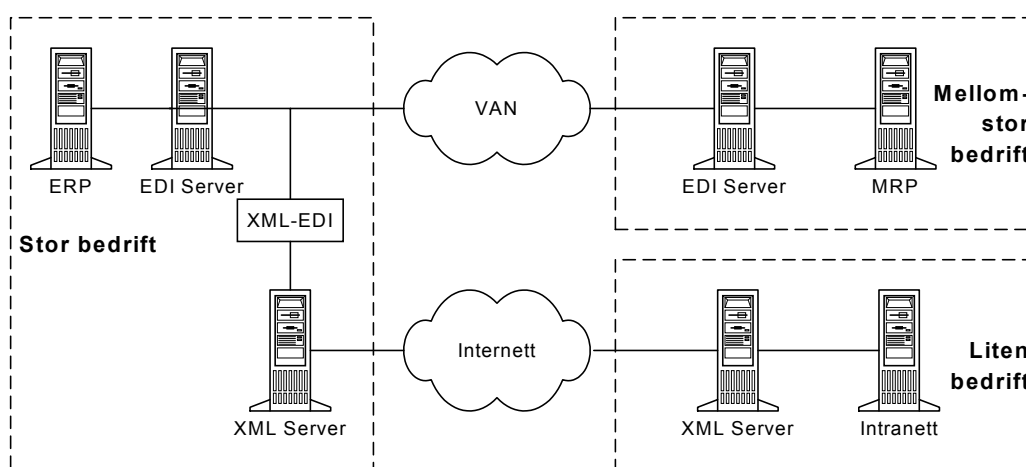
XML er rik på metadata, og det gjør at meldingene er lettere å lese og feilrette (se figur 2-10). Det er lettere å lære og forstå XML i forhold til EDI. Det blir dermed lettere å beholde og utvikle kunnskapen som igjen gjør at vi lettere kan utvikle og vedlikeholde XML-applikasjoner.

Tabell 2-6 – Sammenligning av løsninger for e-handel basert på XML eller EDI [RMH01].

Løsninger for e-handel basert på XML	Løsninger for e-handel basert på EDI
Optimalisert for enkel programmering	Optimalisert for komprimerte meldinger
Krever web-server - kostnad mellom \$0 til \$5000.	Krever dedikert server – kostnad mellom \$10.000 til \$100.000.
Bruker eksisterende Internett-tilknytning	Bruker eget nettverk (VAN) som koster mellom \$1 og \$20 per melding
Bruker format(er) som er lett å lære	Bruker format(er) som er krevende å lære
Krever script-programmerere innen JavaScript, Visual Basic, Python eller Perl script	Krever erfarne programmerere innen C++

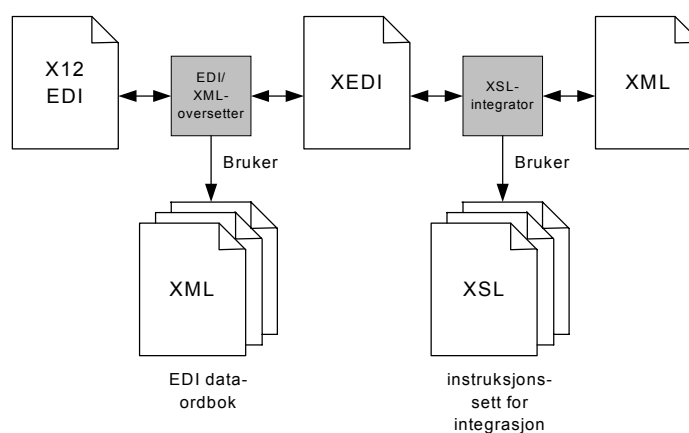
Tabell 2-6 gir en oversikt over flere variabler som påvirker utviklingen av systemer som er basert på XML eller EDI. Dette er variabler som gir en indikasjon på hvorfor EDI har hatt manglende utbredelse og hvorfor XML blir fremhevet som et godt alternativ.

Virksomheter kan vha EDI og XML kommunisere med små og mellomstore bedrifter. Store bedrifter, som i dag bruker EDI, kan utvide måten de kommuniserer på, ved å integrere en oversetter mellom XML-EDI i infrastrukturen. Denne integrasjonen gjør det mulig å samhandle med bedrifter som bruker XML-løsninger. Dette gir store fordeler for virksomheter som allerede har investert mye i EDI (se figur 2-15).



Figur 2-15 – Eksempel på handelssystem med EDI og XML [RMH01]³.

Figur 2-15 viser at store, mellomstore og små bedrifter kan kommunisere meldinger vha rene EDI-meldinger eller integrerte EDI-XML-meldinger. Om vi ønsker å kommunisere integrerte EDI-XML-meldinger må vi bruke en XEDI-oversetter. Konverteringen skjer i en prosess med flere sammenhengende aktiviteter. Eksempel på slike aktiviteter er vist i figur 2-16.



Figur 2-16 – Oversetting mellom EDI og XML [RMH01].

³ VAN står for Value-added Network

Ved konvertering fra EDI til XML bruker vi en oversetter som støtter seg til et bibliotek (X12). Når EDI-meldingen er representert i XML kan vi legge til utseende for meldingen ved hjelp av XSL. Når dette er gjort kan meldingen presenteres på web (HTML) eller vi kan overføre XML-dokumentet direkte til en annen applikasjon.

Ved konvertering fra XML til EDI blir X12 biblioteket nytt til både å konvertere og forsikre at XML-dokumenter blir komplette og presise EDI-meldinger.

Vi som arbeider med løsninger for e-handel kommer heller ikke uten om vurderingen av standarder og formater innen EDI og XML. Muligheten for å integrere formatene gir mulighet for store gevinster.

2.4.4 OBI

Open Buying on the Internet (OBI) er en standard for B2B. Standarden er basert på kunnskap om handel og kunnskap om at det er forskjellige behov for løsninger basert på B2B og B2C [OBI01].

Standarden består av en arkitektur, teknisk spesifisering og retningslinjer, og informasjon om implementasjon. Bedrifter eller privatpersoner kan nytte OBI til å utarbeide produkt, tjeneste eller løsning.

Standarden er utarbeidet av Internet Purchasing Roundtable (Roundtable), og første versjon var ferdig i 1997. Ved utarbeidelsen av standarden tok Roundtable hensyn til interesser fra kjøpende organisasjoner, selgende organisasjoner, teknologiselskaper, finansielle institusjoner og tjenestetilbydere.

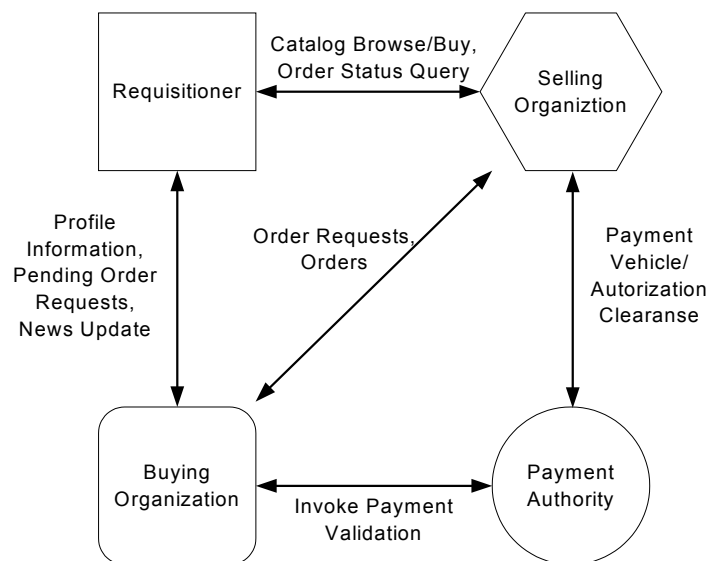
Følgende prinsipper ble lagt til grunn for standarden:

- Common Business Vision - standarden er basert på deltakernes oppfatning og forventning til B2B over Internett.
- Vendor Neutrality - standarden skal være nøytral for selger. Hensikten er å motivere for å tilby konkurrerende produkter fra flere selskaper.
- Interoperability - standarden skal motivere for implementasjon av interoperabilitet i løsninger for selger. For å oppnå dette er protokoll- og meldingsformater nøyaktig spesifisert.
- Value-Added services - standarden gir mulighet for å legge til andre produkter og tjenester. Dette gir rom for for eksempel selgere, tjenestetilbydere og andre til å tilby ulike tjenester og på den måten differensiere seg selv fra andre konkurrenter.
- Cost-Effectiveness - det skal være enkelt og billig å integrere OBI-løsninger i eksisterende infrastruktur. Kostnader i forbindelse med vedlikehold skal være rimelig.
- Robust Infrastructure - standarden motiverer for å utvikle robust infrastruktur med kapasitet for å støtte mange brukere både pålitelig og sikkert.

- Flexibility - standarden er fleksibel, og gir rom for å tilpasse seg varierende behov ved implementasjon av OBI-løsninger. Dette gjelder for både kjøpende og selgende organisasjoner.

Konseptet for OBI er basert på at virksomheter kan og bør ha egne katalogdata. Dette fordi virksomheten selv er best på å administrere innholdet når for eksempel nye produkter blir introdusert, når varer skal fases ut eller når varer skal selges ifm ulike kampanjer.

OBI-modellen (se figur 2-17) tar utgangspunkt i å benytte eksisterende investeringer i kataloger, teknologi og integrasjon til deres ordre-/kundesystem.



Figur 2-17 – Modell/arkitektur for OBI [OBI01].

Modellen har følgende komponenter med tilhørende funksjoner:

- Requisitioner (bestiller) - representerer den som initierer en bestilling eller ordre. Requisitioner er tilsluttet en kjøpende organisasjon, og er tilkoblet web og har utstedt digitale sertifikat.
- Buying Organization (kjøpende organisasjon) - representerer kjøpende enhet og informasjonssystemet som understøtter enheten. Systemet består av blant annet OBI server for mottak og bekreftelse av OBI-ordrer.
- Selling Organization (selgende organisasjon) - vedlikeholder elektroniske kataloger med produkter og priser. Informasjon om produkt og pris er styrt gjennom kontrakt med kjøpende organisasjon. Katalogen for Selling Organization må være tett integrert med inventardata, ordreadministrasjon og OBI-servere for sende og motta ordrer.
- Payment Authority - enheten skaffer til veie autorisasjon for bestiller til betalingsystemet. Payment Authorities må tilby betaling til selgende organisasjon og til rett tid betale eller utbetale til kjøpende organisasjon.

I prosessen med å velge løsning for e-handel kan vi sammenligne OBI med XML. Om vi gjør det finner vi flere ulikheter (se tabell 2-7).

Tabell 2-7 – Forskjell mellom OBI- og XML-formatet [BKS01].

	OBI	XML
Bestille (Purchase Order)	Standard OBI 850	Properitært format. Hvert utviklingsmiljø kan utvikle og støtte egen XML syntaks og dokumentformat.
Ordrebekreftelse (Order Acknowledgment)	Standard OBI 855	Properitært format. Hvert utviklingsmiljø kan utvikle og støtte egen XML syntaks og dokumentformat.
Endre ordre (Change Order)	Standard OBI 860	Properitært format. Hvert utviklingsmiljø kan utvikle og støtte egen XML syntaks og dokumentformat.
Faktura (Invoice)	Properitært format. Hvert utviklingsmiljø kan utvikle og støtte eget EDI-lignende dokumentformat.	Properitært format. Hvert utviklingsmiljø kan utvikle og støtte egen XML syntaks og dokumentformat.
Oppdatere kataloger (Catalog Updates)	Ikke påkrevet (om ikke annet er spesifisert)	Properitært format. Hvert utviklingsmiljø kan utvikle og støtte egen XML syntaks og dokumentformat.
Ordreforespørsel (Order Request - pulling data from supplier's shopping cart into a B2BE requisition)	Standard OBI Order Request	Properitært format. Hvert utviklingsmiljø kan utvikle og støtte egen XML syntaks og dokumentformat.

B2B Knowledge Source [BKS01] fremhever at OBI-standarden er enkel og industrien støtter standarden i dag. B2B Knowledge Source mener at XML ikke har standardiserte løsninger for e-handel. Dvs de som benytter XML, for e-handel i dag, har egne og properitære løsninger.

2.4.5 UDDI

Ledende aktører i markedet ser for seg en visjon som underbygger ønske om å gjøre web-tjenester allment tilgjengelig for alle salgs applikasjoner [DIG01]. Visjonen er en del av satsningen til .Net fra Microsoft, og den kan også spores tilbake til "nettsentrert databehandling" fra IBM (1995).

Digi.no [DIG01] fremhever at det er tre grunnleggende protokoller som er viktige skritt i denne retningen:

- Universal Description, Discovery and Integration (UDDI) - gjør det mulig å finne fram til hensiktsmessig kode eller tjenester som kan integreres i andre applikasjoner.
- Simple Object Access Protocol (SOAP) - beskriver hvordan applikasjoner skal kommunisere med hverandre.
- Web Services Description Language (WSDL) - forteller hvordan webtjenester skal beskrives med tanke på gjenbruk.

UDDI, SOAP og WSDL bygger på XML.

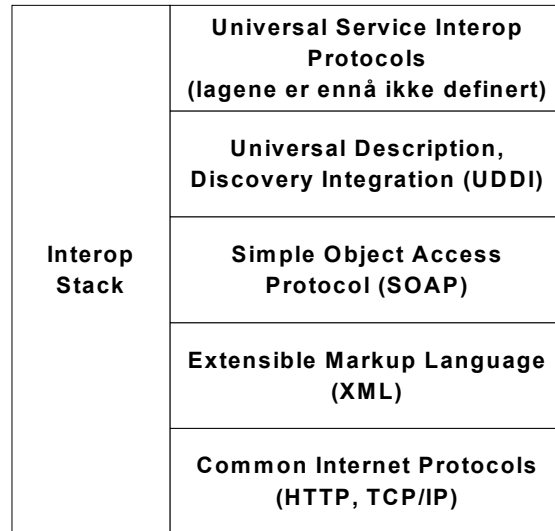
I relasjon til e-handel er UDDI, SOAP og WSDL interessante. For å gi et innblikk i en av spesifikasjonene vil jeg omtale UDDI.

UDDI er en spesifikasjon som tar utgangspunkt i å tenke nytt om e-handel. Vi bør bevege oss i retning av at bedrifter kan muliggjøre kontakt ved å oppdage og nå hverandre på en ny måte. I denne sammenheng bør vi lære om hvilke muligheter potensielle handelspartnere har, kontinuerlig oppdage nye handelspartnere, og forstå deres muligheter, og etter hvert gjennomføre handel (e-handel) med disse [UD01].

Kjernekomponenten i UDDI er registrering av virksomheter. Vi bruker komponenten for å beskrive bedriften og dens tjenester på web. Beskrivelsen blir beskrevet i en XML-fil, og består av tre komponenter:

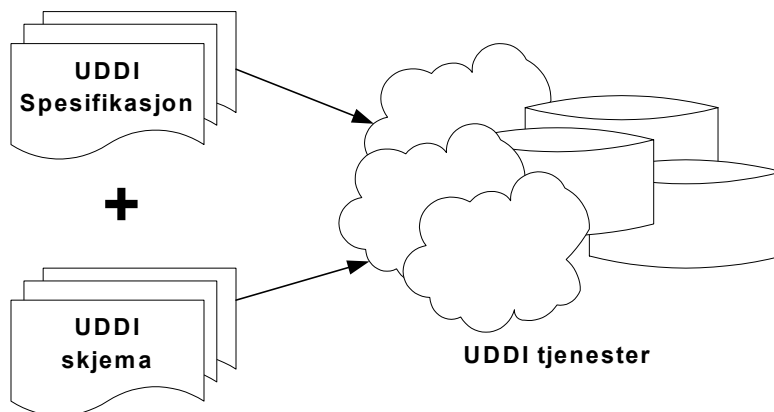
- White Pages - består av adresse, kontakt og kjente identifikatorer.
- Yellow Pages - inkluderer industriell kategorisering.
- Green Pages - teknisk informasjon om tjenester som er tilgjengelige.

Data om virksomheter kan bli brukt, av andre virksomheter, for å vurdere om en virksomhet har ønskede produkter og tjenester på web. Vi kan også bruke nevnte data for å finne selskaper innen definerte fagområder, med tilhørende tjenester, og til å lokalisere nødvendige tekniske data som er nødvendig for å kunne kommunisere med virksomheten.



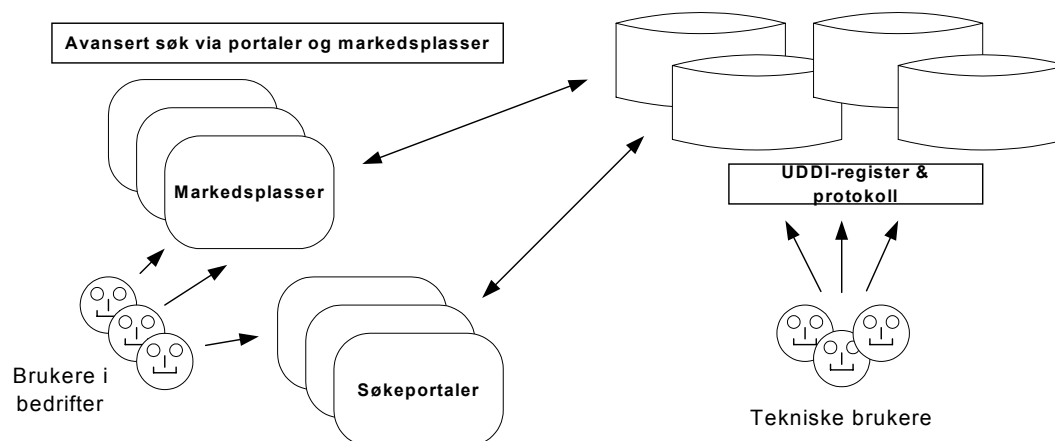
Figur 2-18 – UDDI i relasjon til andre standarder og spesifikasjoner [UD01].

UDDI tar utgangspunkt i standarder fra World Wide Web Consortium (W3C) og Internet Engineering Task Force (IETF). Eksempler på slike standarder er TCP/IP, HTTP, XML, DNS og SOAP. I relasjon til disse spesifikasjonene og standardene er UDDI plassert i øvre del av hierarkiet (se figur 2-18).



Figur 2-19 – Sammenheng mellom spesifikasjon, skjema og tjenester i UDDI [UD01].

UDDI-spesifikasjonen beskriver en konseptuel sky av web-tjenester. I tillegg beskriver spesifikasjonen et grensesnitt, med et enkelt rammeverk, for å beskrive alle typer web-tjenester. Spesifikasjonen består av flere relaterte dokumenter. Den består også av et XML-skjema, som består av en SOAP-basert protokoll, for registrering og søk etter web-tjenester. Figur 2-19 viser relasjonen mellom spesifikasjonen, skjema og UDDI-tjenester. Tjenestene gir mulighet for å "registrere deg en gang, for deretter å publisere hvor som helst".



Figur 2-20 – Bruk av UDDI [UD01].

Figur 2-20 gir en oversikt over hvordan vi kan bruke UDDI. Forretningsbrukere søker etter virksomheter gjennom markedsplasser og søkeportaler. Tekniske brukere aksesserer databaser og protokoller gjennom et eget grensesnitt.

2.5 Systemer for e-handel

Det eksisterer mange produkter for e-handel på markedet. Vi kan sette sammen ulike applikasjoner med tilhørende datakilder og få disse til å spille sammen. Fremgangsmåten krever noe tilpasning [TD01].

Figur 2-3 skisserer en løsning basert på B2C. Figuren viser at et system for e-handel kan bestå av flere underliggende systemer (se tabell 2-8). Vi har mulighet til å implementere ulike funksjonalitet i løsningen for e-handel. Dette gjelder uansett kategori – B2C, B2B eller B2P.

Tabell 2-8 – Underliggende systemer i en B2C-løsningen for e-handel [TD00].

System	Funksjonalitet
Butikksystem (Commerce Server)	Håndtere ordre og handlevogn, kommunisere med produktkatalog, kunderegister og rapportering
Innholdshåndtering (Content Manager)	Produsere og publisere innhold
Personalisering (Personalization Server)	Tilby personalisert innhold til kunden
Kundeservice (Support, Custom Relationship Management [CRM])	Håndtere e-post og kommunikasjon med kunden.
Annonsesystemer (Advertising Server)	Håndtere reklame på nettet
Dynamisk prissetting (Auction Server)	Tilby ulike former for auksjoner
Betaling (Payment Server)	Tilby løsninger for sikker betaling
Distribusjon/logistikk (Fulfillment/Shipping)	Håndtere utsending av varer

Vi kan implementere ferdige pakkøløsninger for e-handel. Pakkene inkluderer da de underliggende systemene vi har behov for. Det er i dag tilgjengelig flere slike pakkøløsninger (se tabell 2-9).

Tabell 2-9 – Leverandører av systemer for e-handel.

Leverandør	System for e-handel
Commerce One [CO01]	Commerce One Market Site / Commerce One Enterprise Solution
IBM [IBM01a]	IBM WebSphere Commerce Suite
Microsoft [MS01]	Microsoft Commerce Server
Oracle [ORA01]	Oracle E-Business Suite
SAP [SAP01a]	mySAP E-Procurement

2.6 Oppsummering

Det eksisterer et vell av *begreper* for e-handel. Begrepene kan være like mye forvirrende som oppklarende. Vær oppmerksom på likhetene og forskjellene mellom for eksempel e-handel, e-commerce og e-business.

Ved utvikling av løsninger for e-handel bør vi være bevisst på hvilke *egenskaper* mediet (Internett) har. Kunnskapen er sentral ved utvikling og bruk av strategier som direkte eller indirekte påvirker løsningen vi velger for e-handel.

Det eksisterer flere *kategorier* innen e-handel. Vi bruker ofte begrepene B2C om handel mellom kunde og bedrift. Og vi bruker B2B om handel mellom bedrifter og B2P om handel mellom det offentlige og bedrifter. Ved utvikling av systemer for e-handel må vi ha klart hvilken kategori eller kategorier vi skal støtte.

Ved utforming av løsninger for e-handel bør vi være bevisst på økonomiske, juridiske, tekniske og spesielt *sikkerhetsmessige* forhold. Vi kan benytte ulike standarder, protokoller, mekanismer eller applikasjoner for å forbedre sikkerhet ved kjøp og salg over Internett.

I relasjon til infrastruktur har jeg spesielt omtalt flere *standarder* og *spesifikasjoner* for e-handel. Ved egenutvikling av systemer for e-handel bør vi være meget bevisst på veivalg innen dette området. Ferdige systemer for e-handel (pakker) støtter trolig flere av nevnte standarder og spesifikasjoner.

3 Om katalogsystemer og katalogtjenester

Kapitlet introduserer begrepene infrastruktur, katalogtjenester og katalogsystemer. Kapitlet presenterer også arkitektur, standarder og spesifikasjoner og sikkerhet for katalogsystemer.

3.1 Innledning

Oppfatningen av begrepet infrastruktur er vidt sammensatt, og blir ofte definert i et samfunnsmessig perspektiv. Nettutgaven for Caplex Leksikon definerer infrastruktur slik:

infrastruktur, eg. "underliggende struktur", flyplasser, havner, veier, jernbaner, telekommunikasjonsmidler o.a. vitale serviceanlegg i et samfunn *mil.*, bet. på alle de faste anlegg som en væpnet styrke trenger under krigsoperasjon [CAP00].

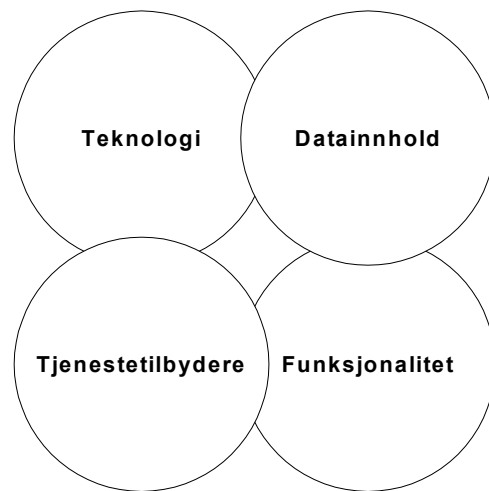
Det er mange meninger om infrastruktur for e-handel. Norsk EDIPRO har gjennom prosjekt "Infrastruktur for elektronisk handel" [NED00] definert infrastruktur, og analysert hvilke oppfatninger markedet har om begrepet.

Prosjektet definerer infrastruktur for e-handel slik:

infrastruktur for elektronisk handel - summen av de elementene som er nødvendige for å bygge ut og utvikle elektronisk handel [NED00].

Selv om det er mange meninger om begrepet, kom det likevel frem at det er en generell enighet om at infrastruktur må inneholde elementer av [NED00]:

- teknologi - i form av maskinvare, fremføringsnett (dvs Internett), kommunikasjonsprotokoller og -løsninger, sikkerhetsløsninger og dataformater
- datainnhold - i form av meldingsdefinisjoner (fortrinnsvis en syntaksnøytral beskrivelse av datainnholdet i en melding)
- tjenestetilbydere - i form av markedsplasser, portaler, formidlingsentraler, TTP-er, ol
- funksjonalitet - i form av definerte forretningsprosesser



Figur 3-1 – Komponenter i infrastruktur for e-handel [NED00].

Rapporten fremhever, på et overordnet nivå, at fremtidig åpen infrastruktur for elektronisk handel må fokusere på datainnhold og funksjonalitet [NED00].

3.1.1 Begreper om katalogsystemer og katalogtjenester

Katalogsystemer og katalogtjenester eksisterer i enhver organisasjon eller bedrift. Data i slike systemer kan være lagret både manuelt og elektronisk. Felles for systemene, uavhengig av lagringsmetode, er at omgivelsene har behov for å nytte disse data.

Det er en prinsipiell forskjell mellom katalogsystemer og katalogtjenester. System er "helhet av enkelte deler som hører sammen" [UNV98] og tjeneste er "hjelp, bistand" [UNV98].

Jeg velger å definere katalogsystem som nødvendig element for å fremskaffe katalogtjenester.

Det eksisterer flere definisjoner for katalogsystemer og spesielt katalogtjenester. I de neste avsnittene vil jeg presentere noen definisjoner for katalogtjenester og samtidig knytte noen kommentarer til disse.

Directory service provides the means to hierarchically organize and manage information, and to retrieve the information by name association [SS00].

Sheresh & Sheresh [SS00] tar utgangspunkt i en mer allmenn oppfatning av katalogtjenester. Forfatterne fremhever at vi kan definere katalogtjenester som; alt som frembringer en assosiasjon mellom to eller flere deler av informasjon om en spesifikk person eller ressurs. For denne definisjonen velger jeg å fremheve følgende stikkord; organisering og administrasjon av data.

A directory service is a service that provides information about people and resources to client requesting information. Information may be a name, telephone number, e-mail address, application preferences, etc. The client may be a person or an application (Netscape 1998) [RA00].

Archie Reeds [RA00] tar utgangspunkt i et annet perspektiv. Katalogtjenester er tjenester som tilbyr informasjon om mennesker og ressurser. Presentasjonen blir fremført på bakgrunn av en forespørsel fra en klient. Stikkord for denne definisjonen; informasjon, mennesker og ressurser.

The directory is a collection of open systems which cooperate to hold a logical database of information about a set of objects in the real world (according to the ISO recommendation X.500 "The directory service") [RA00].

International standard organization (ISO) har også skissert en definisjon for X.500 The directory service [RA00]. I forhold til tidligere definisjoner er denne mer orientert mot data og objekter. Stikkord her er: data og objekter.

A directory is an information storage location that uses a systematic scheme to organize the information [SC00a].

Curt Simmons [SC00a] tar utgangspunkt i at katalogsystem lagrer informasjon på en systematisk måte. Denne definisjonen er noe begrenset i forhold til tidligere definisjoner. Stikkord for denne definisjonen er; informasjon og organisering.

Nevnte definisjoner er beskrevet på ulike tidspunkt og bakgrunn av ulike versjoner av standarder (for eksempel X.500). Uavhengig av bakgrunn for hver definisjon er følgende stikkord gjennomgående for katalogtjenester:

- data og informasjon
- mennesker og ressurser.

Kombinasjonen av nevnte stikkord gjør katalogtjenester, uavhengig av om tjenestene er koblet til elektronisk handel, meget interessant. Dette fordi vi blant annet kan koble stikkordene til informasjons- og kunnskapsforvaltning i organisasjoner.

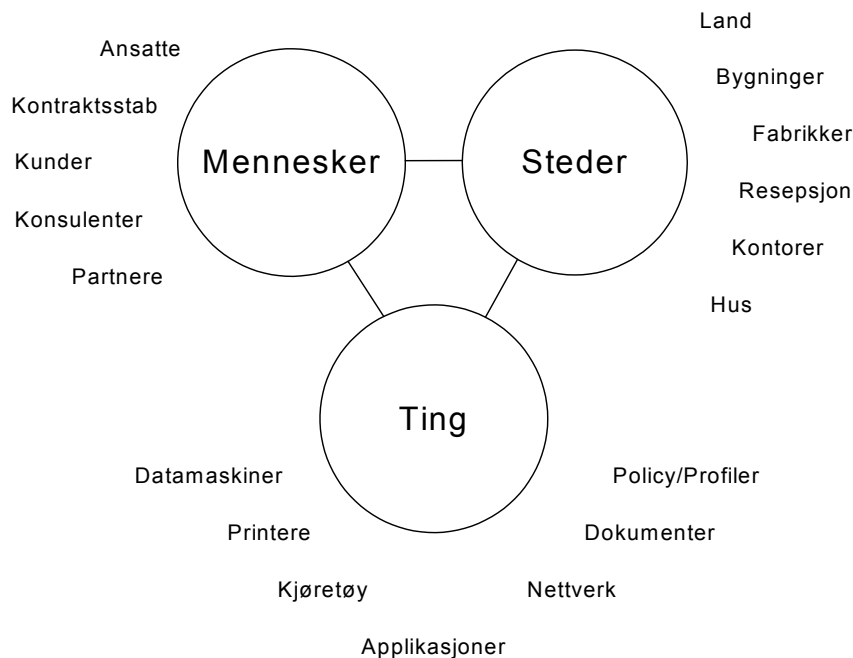
Slik jeg tolker van der Spek mfl [SBS97] er informasjons- og kunnskapsforvaltning av stor betydning for bedrifter og organisasjoner i dag, og begrepene står meget sentralt i utviklingen av vår hverdag.

3.2 Arkitektur

I dag har bedrifter og organisasjoner en eller annen form for data i en strukturert form. De er lagret i en struktur og gjerne tilknyttet en eller annen form for tjenester. Systemene og tjenestene er satt sammen på ulike måter, og eierne kan for eksempel få tilgang til innholdet manuelt og/eller automatisk.

De fleste eksisterende kataloger er tatt i bruk på bakgrunn av et spesielt behov. Generelt kan vi si at katalogene inneholder nok data til å dekke behovet for operasjoner eller applikasjoner [RA00].

Archie Reed [RA00] fremhever at katalogtjenester støtter tre kjerneelementer. Elementene er mennesker, steder og ting (se figur 3-2). Figuren viser nevnte elementer med tilhørende attributter.



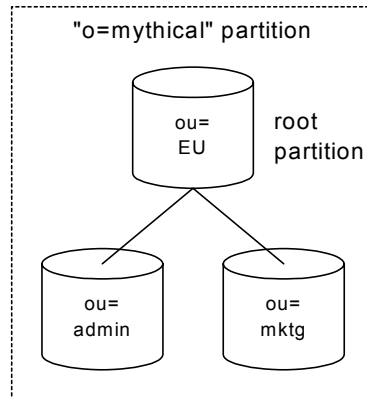
Figur 3-2 – Kjernen i katalogsystemer [RA00].

Nevnte elementer er bakgrunn for videre presentasjon av arkitektur for katalogsystemer.

3.2.1 Design og struktur

Sheresh & Sheresh [SS00] definerer en katalog som en informasjonskilde. I tillegg presiserer forfatterne at måten informasjon blir lagret og administrert på er meget viktig med hensyn på funksjonalitet i katalogene. Hvor effektivt informasjon blir lagret, gjenfunnet og distribuert kan direkte påvirke skalerbarhet, ytelse og pålitelighet av katalogen.

En katalog består normalt av flere objekter i et hierarki. Objektet som er plassert øverst i et tre eller subtre kalles rotpartisjon eller *partition root*. Eksempel på objekter og rotpartisjon er skissert i figur 3-3.



Figur 3-3 – Eksempel på katalogtre [SS00].

Informasjon i kataloger er lagret i Directory Information Base (DIB). DIB inneholder blant annet objekter som representerer for eksempel brukere, nettverksressurser, applikasjoner og administrative data. Administrative data er nødvendig for å administrere og kontrollere aksess til objektene [SS00].

For å beskrive oppbygging av kataloger vil jeg presentere og drøfte flere strukturelle forhold for kataloger:

- organisering (flat eller hierarkisk struktur)
- navnemodell (fysisk eller logisk modell)
- lagring av data (sentralisert eller distribuert katalog).

Organisering

Struktur (overordnet) er et sentralt område ved design av katalogsystemer. Sheresh & Sheresh [SS00] fremhever at vi kan skille mellom flate eller hierarkiske kataloger. I en flat katalog er alle ressurser lagret i en gruppe, mens i en hierarkisk katalog er ressursene lagret i en trestruktur. Den hierarkiske katalogen av data blir kalt *Directory Information Tree* (DIT). DIT reflekterer ofte strukturen i organisasjoner.

Ved design av DIT må vi være bevisst på primæroppgaven for katalogsystemet: vi skal designe DIT for optimal søk og oppslag. Archie Reed[RA00] fremhever også at design av strukturer i DIT er påvirket av flere forhold:

- geografisk distribusjon
- funksjonalitet
- politikk
- administrative krav
- applikasjonskrav
- tilgjengelighet i nettverk
- krav til vedlikehold
- synkroniseringsverktøy.

Å tilfredstille alle krav til applikasjoner, tjenester og systemer, som kan bruke katalogtjenester, er en meget krevende oppgave. Om vi for eksempel søker å administrere data effektivt anbefaler Archie Reed [RA00] å strukturere data i DIT på samme måte som i en database. På denne måten kan vi forbedre effektiviteten ved å bruke Standard Query Language (SQL).

Om en organisasjon er spredt på flere globale lokasjoner, og knyttet sammen vha Wide Area Network (WAN), er det vanlig at toppnivå i DIT er koblet til geografiske grenser (for eksempel EU og USA). Dette blir blant annet gjort for å forenkle administrasjon og kontroll av nettverkstrafikk [SS00].

Vi skal være bevisst på at det er fordeler og ulemper med både flate og hierarkiske strukturer. Står vi overfor valget mellom flat eller hierarkisk struktur bør vi vurdere å se valget i sammenheng med hvordan verden er i dag.

Om jeg skal gjøre et valg innen modellering av DIT vil jeg, i det fleste tilfeller, velge hierarkisk modellering.

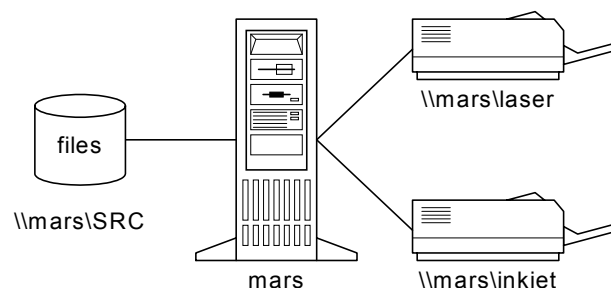
Dette fordi økende globalisering av økonomi gjennom for eksempel e-handel og sammenslåing av bedrifter, på tvers av landegrenser, krever fleksible teknologiske løsninger i bedrifter. Dette gjelder også for strukturer i katalogsystemer for virksomheter.

Navnemodell

Vi kan definere ressurser i nettverk ved å bruke *fysiske navnemodeller* eller *logiske navnemodeller* [SS00]. Egenskapene for modellene er beskrevet i kapittel 3.2.2 Egenskaper - Defineret navnerom.

Fysiske navnemodeller bruker fysiske navn for å angi ressurser i nettverk.

Eksempel på slike navn er navn på filserver og printere. Figur 3-4 skisserer en fysisk navnemodell. Den bruker aktuelle navn (for eksempel `\\mars\laser`) for å identifisere ressurser i et nettverk. Modellen er basert på Universal Naming Convention (UNC) og blir blant annet brukt av Microsoft Windows [MS01].



Figur 3-4 – Fysisk navnemodell [SS00].

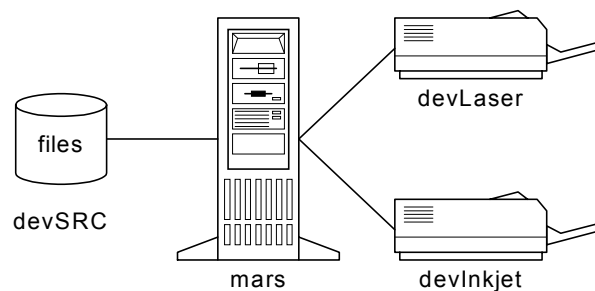
I nettverk, som nytter fysiske navnemodeller, må brukeren angi fysisk navn på ressurs før brukeren kan nytte den. Denne måten å angi en ressurs påfører brukeren et krav om å lære mer om egenskapene for ressursene. I mindre nettverk er dette ikke noe problem. Dette fordi organisasjonen har kunnskap, gjennom for

eksempel enkle skjema og hukommelse hos ansatte, om egenskapene for ressursene. Fysiske navnemodeller medfører at en endring i ressursen, for eksempel endret lokasjon av en skriver, kan føre til merarbeid for administrator og samtidig forvirre bruker. Når en nettverksressurs endrer navn, pga at den flyttes, blir det kalt for et *lokasjonsuavhengig navn*.

Håndteringen av navn kan være enkel om rammeverket for en katalog kun er fysiske elementer og brukere i nettverk. Men om rammen for kataloger utvides til å håndtere flere sett med objekter (ressurser), bør vi definere navnene på en annen måte.

Navnemodeller, for mer avanserte katalogtjenester, bør ivareta følgende:

- navn må være koblet fra fysiske objekter slik at abstrakte objekter og informasjon kan være representert.
- navn må unikt identifisere objekter i store kataloger (med navnerom).
- katalogtjenesten må kunne støtte bruk av flere navn for samme objekt.



Figur 3-5 – Logisk navnemodell [SS00].

Navnerom som nytter *logiske navnemodeller* gjør det mulig å lage relasjoner, mellom fysiske nettverksressurser, og symbolske navn som representerer ressursen. Brukere i nettverket kan deretter aksessere ressursen ved å nytte det symbolske navnet, isteden for å nytte det fysiske navnet. Figur 3-5 skisserer et nettverk hvor ressursene er tildelt logiske navn. Navnene (for eksempel devLaser og dev InkJet) er ikke koblet til det fysiske navnet for ressursen. *Katalogen* vedlikeholder koblingen mellom fysiske og logiske navn for ressurser.

En av fordelene ved å bruke abstrakte navn er at navnet kan forbli uforandret selv om det fysiske navnet blir endret. Når et navn til et nettverksobjekt forblir konsistent, selv om for eksempel den fysiske lokasjonen blir endret, kaller vi navnemodellen for en *lokasjonsuavhengig modell*.

Alle objekter, i en katalog, må tildeles et navn i henhold til en spesifisert navnemodell. Ved lokalisering av et objekt må angitt navn korresponderes med definert navnekonvensjon. Bruk av navnekonvensjoner kan være nøkkelen til hvor effektivt en bruker kan identifisere og finne ressurser.

Det eksisterer flere former for navnekonvensjoner. Noen konvensjoner har en enkel struktur, mens andre har en mer komplisert struktur. Eksempler på slike konvensjoner er [SS00]:

- X.500
- LDAP
- LDAP URL
- NDS
- UNC
- RFC 822 Email
- DNS.

Navnekonvensjoner, som er merket kursiv, er omtalt senere i avhandlingen.

Ved innføring av katalogsystemer er valg av navnekonvensjoner en meget sentral oppgave. I innføringsprosessen bør vi være bevisst på at navn ikke bare skal representere informasjon om mennesker, men også informasjon om steder og ting [RA00].

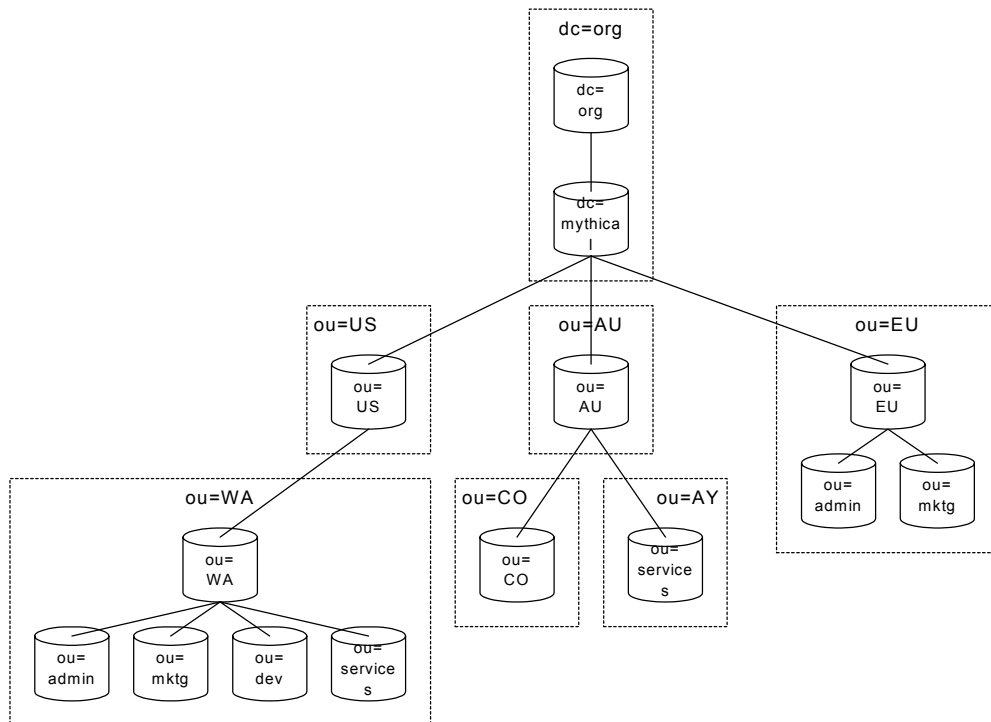
Om vi retter søkelys mot e-handel og katalogtjenester står valg og bruk av navnekonvensjoner meget sentralt. Dette fordi måten kunder eller ansatte (for eksempel selger) aksesserer objekter eller ressurser på, gjennom for eksempel web, kan være avgjørende for hvilke data som blir fremskaffet. Riktige data, om produkter og tjenester, er avgjørende for markedsføring og salg.

Lagring

En annen forskjell mellom kataloger er måten data blir lagret på i systemene. Dvs om data er lagret på en enkel server eller spredt på flere servere i et nettverk. Om data er lagret på en sentralisert server definerer vi katalogtjenesten som sentralisert. Om data er lagret på flere servere, som er sammenkoblet, kaller vi katalogtjenesten distribuert [SS00].

I distribuerte katalogsystemer, basert X.500, blir data lagret i Directory Information Base (DIB) og blir spredt på flere servere. Om data blir spredt over flere servere sier vi at katalogen blir *partisjonert*. En partisjon er en fysisk del av en database (katalog) som er lagret på og administrert av en katalogserver [SS00].

Selv om vi kan dele et katalogtre i flere partisjoner vil katalogen representere ett tre for brukere. Det betyr at selv om DIB er fysisk fordelt på flere separate servere, må data i hver katalog være koblet sammen på en eller annen måte. Katalogsystemet skal ha kontroll over alle data som er lagret og sette disse sammen til et (virtuelt) tre. Figur 3-6 viser hvordan en katalog kan partisjoneres og hvordan det er satt sammen [SS00].



Figur 3-6 – Partisjonering av en global organisasjon (the mythical) [SS00].

Partisjonering og oppdeling av kataloger kan være en fordel for systemer som skal være autonome. Det vil si at systemene skal fungere uavhengig av om for eksempel den sentrale katalogen er tilgjengelig eller ikke.

Katalogsystemer kan understøtte behovet for autonome systemer. Dette forholdet er viktig for organisasjoner som har virksomhetskritiske systemer, og bør være av stor betydning for organisasjoner som ivaretar nasjonale og internasjonale interesser (for eksempel Forsvaret).

3.2.2 Egenskaper

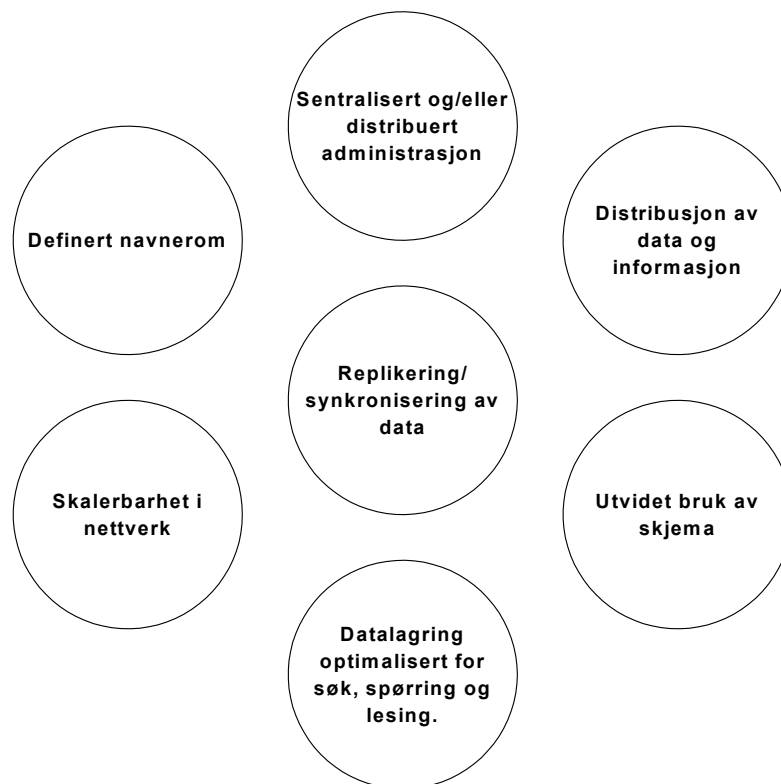
Sheresh & Sheresh [SS00] fremhever at vi har brukt kataloger, i en enkel form (for eksempel telefonkatalogen), i mange år. Om vi ser utviklingen frem til i dag er det flere fellestrekk innen design og funksjonalitet mellom katalogtjenester og generelle tjenester innen nettverk. For kunne skille mellom generelle tjenester (applikasjoner) og katalogsystemer mener forfatterne at vi kan gjøre dette ved å vurdere egenskaper:

- definert navnerom
- utvidet søkemulighet
- skalering fra små til store nettverk
- distribuert informasjonssystem
- replikere data
- datalager optimalisert for lesing
- utvidet bruk av skjema.

Andre forfattere bruker andre ord om egenskaper for katalogtjenester. Archie Reed [RA00] definerer følgende for katalogtjenester:

- sentralisert og/eller distribuert administrasjon
- enkel distribusjon
- enkel replikering
- stor skalerbarhet
- hierarkisk- og relasjonsorientert
- optimalisert for spørring, lesing eller oppslag
- standard skjema
- skjema kan defineres globalt
- skjema kan både utvides globalt og lokalt
- enkelt å replikere skjema.

Med utgangspunkt i nevnte kjennetegn vil jeg omtale de mest sentrale egenskapene for katalogtjenester (se figur 3-7).



Figur 3-7 – Egenskaper for katalogtjenester.

Definert navnerom

En fundamental egenskap for katalogtjenester er navnerom (namespace). Navnerom sier noe om hvordan ressurser i nettverket er definert. Navnerom er normalt delt inn i *fysiske navnemodeller*, som bruker navn på ressurser i nettverk,

og *logiske navnemodeller*, som bruker symbolske navn som representerer ressurser i nettverket [SS00].

Organisering av navnemodeller er et fundamentalt tema innen katalogsystemer. Vi kan ta utgangspunkt i egne metoder og modeller ved implementasjon, eller vi kan ta utgangspunkt i X.500 eller Lightweight Directory Access Protocol (LDAP). X.500 og LDAP gir retningslinjer for hvordan vi kan standardisere og organisere navnerom.

Navnerom gir store fordeler i hvordan vi kan organisere katalogsystemer og katalogtjenester. Dette er meget sentralt i store globale nettverk, og dermed også av stor betydning for e-handel. Utvidet bruk av navnerom gir også fordeler for fleksibilitet, brukervennlighet, reduksjon i administrasjon og stabilitet for ressurser i nettverk.

Sentralisert og/eller distribuert administrasjon

På bakgrunn av arkitekturen i katalogsystemer, som er påkrevd for å støtte tilhørende tjenester, må vi kunne administrere kataloger uavhengig av lokasjon [RA00].

Vi kan administrere katalogsystemer fra et sentralt driftssenter eller gjennom flere distribuerte miljø. Om en organisasjon er spredt over flere land og kontinenter, med ulike tidssoner, kan distribuert administrasjon av kataloger være en fordel.

Det kan være fornuftig å segmentere administrasjon slik at deler er administrert fra et sentralt punkt, mens andre elementer er administrert uavhengig av lokasjon. Distribuert administrasjon gir mulighet for bredere deltakelse i administrasjon og vedlikehold av systemer.

Distribusjon av data og informasjon

Katalogtjenester støtter mulighet for distribusjon av katalog(er). Dette kan vi gjøre over flere datamaskiner i et nettverk. Partisjonering av en katalog medfører en logisk oppdeling av DIB, som igjen gir adgang til delegering av kontroll og aksesstid til data i katalog [SS00].

Partisjoner av en katalog, i forskjellige deler (subsett), gir oss adgang til å lagre disse på forskjellige fysiske datamaskiner. Denne form for partisjonering kan være begrunnet i ønske om bedret sikkerhet og ytelse.

Denne form for segmentering av kataloger gir flere muligheter og begrensninger. I globale organisasjoner vil partisjonering gi fordeler ifm lagring, søk og oppslag i kataloger. Ulempen kan være høyere kostnader ifm utvikling og vedlikehold av distribuerte katalogsystemer.

Skalerbarhet i nettverk

Et nøkkelement innen skalerbarhet er effektiv distribusjon av katalogdata over flere servere i et nettverk. Sheresh & Sheresh [SS00] mener at en robust katalog skal kunne håndtere skalering fra små nettverk, med en katalog med noen hundre objekter, til store globale nett, med mange distribuerte kataloger med flere millioner objekter.

Økende globalisering av virksomheter, gjennom oppkjøp, fusjon og posisjonering i markedet, gjør skalerbarhet i katalogsystemer til et viktig element.

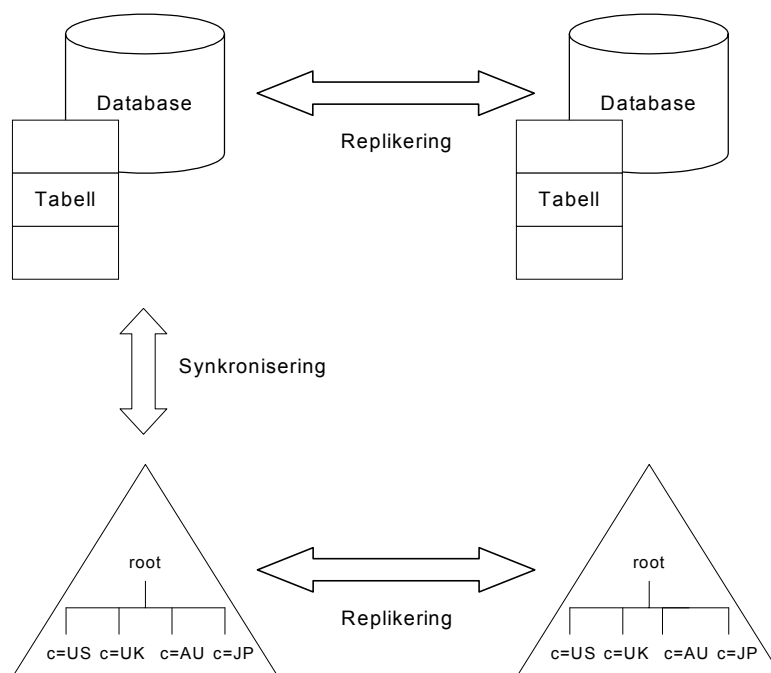
Replikering/synkronisering av data

For å gjøre nettverk eller informasjonssystemer mer resistente mot feil, og lettere tilgjengelig for bruker, støtter kataloger replikering [SS00].

Replikering av data er normalt kopiering av data mellom to *like* databaser, og gir mulighet for eksakt replika av data mellom to systemer. Dette forutsetter blant annet at databasene er definert på samme måte. Kataloger kan gjennomføre en replikering ved å kopiere data fra en database til en annen – uten transformasjon eller mutasjon mellom databasene [RA00].

Synkronisering er lik replikering, men er definert som mer komplekse relasjoner mellom *ulike* databaser. Når vi nytter synkronisering er det en forandring som initierer en annen forandring. Å si at to databaser er synkronisert betyr ikke at de har eksakt samme replika av hverandre.

Forholdet mellom replikering og synkronisering er skissert i figur 3-8.



Figur 3-8 – Forholdet mellom replikering og synkronisering [RA00].

Replikering og/eller synkronisering mellom distribuerte kataloger gir blant annet mulighet for at en enhet i Norge kan fungere autonomt uavhengig av et kontor i annet land. Eller på en annen måte: om kommunikasjon mellom enheter ikke fungerer tilfredstillende kan enheter likevel arbeide tilnærmet normalt. Når linken mellom enhetene er tilgjengelig kan data som er endret oppdateres vha replikering eller synkronisering.

Datalagring optimalisert for søk, spørring og lesing

Mekanismen for datalagring er oftest designet for å støtte et stort antall lesinger. I forbindelse med produktdesign av katalogtjenester er det vanlig å anta at 99% av operasjonene, som aksesserer DIB, vil være lesing (opplag og søk). Til sammenligning vil antall skrivinger være relativt få (endringer og tillegg).

Forholdet mellom lesing og skriving er et viktig element ved katalogsystemer. Dette fordi de fleste kataloger er optimalisert for å støtte mange oppslag gjennom operasjoner for spørring og lesing [RA00, SS00].

Forholdet oppslag/søk og endringer/tillegg må ses i relasjon til hva katalogsystemene skal benyttes til. Dette er et viktig forhold om vi skal bruke katalogsystemer for å understøtte systemer for e-handel.

Utvidet bruk av skjema

Skjema inneholder ett sett med regler som definerer hvilke type data som kan bli lagret i kataloger. Reglene definerer selve objektet og dets egenskaper. Bruk av skjema reduserer innhold og struktur gjennom definering av objekter [SS00].

Archie Reed [RA00] sier at skjema er en metode som regulerer hva katalogen skal lagre av syntakser og semantikk. Utvidet bruk av standard skjema forsterker dataintegritet i kataloger.

Novell [NOV01] (Administration Guide for NDS eDirectory) fremhever at skjema definerer objekttyper i et katalogtre (som brukere, printere og grupper), og hvilke data som er nødvendig eller valgfrie på det tidspunkt objektet blir opprettet.

Skjema er et sentralt element i katalogsystemer. Skjema gjør blant annet replikering og distribusjon av data enklere [RA00]. De fleste katalogtjenester støtter utvidet bruk av katalogskjema. Dette gir utviklere og administratorer mulighet til å utvide sett med katalogobjekter, og implementere nye og kundespesifiserte katalogobjekter [SS00].

Bruk av skjema har klare fordeler ved elektroniske transaksjoner mellom to eller flere parter. Og når elektroniske transaksjoner er selve grunnlaget for e-handel vil bruk av skjema være meget sentralt i infrastruktur for e-handel.

3.2.3 Klassifisering av kataloger

Vi kan klassifisere katalogsystemer etter *innhold* (scope of content) og *funksjonalitet* for klient og administrasjon (range of services). Sheresh & Sheresh [SS00] mener vi kan klassifisere katalogtjenester i WAN/LAN i inntil fem klasser:

- nettverksfokusede kataloger (networking-focused directories) – er designet for å støtte funksjonalitet i nettoperativsystem (NOS). Eksempel er brukerkonti, sikkerhet og administrasjon av ressurser i nettverket.
- applikasjonskataloger (applications directories) – lagrer informasjon om brukere av en spesifikk applikasjon, type operasjoner applikasjonen kan utføre og konfigurasjonsdata . Eksempel på slike applikasjoner er e-post- og gruppevare.
- begrensede kataloger (limited-use directories) – er designet for å lagre informasjon for et formål. Eksempel på slike kataloger er offentlig informasjon tilgjengelig Internett eller katalog for Domain Name System (DNS).
- generelle kataloger (general purpose directories) – støtter normalt et bredt utvalg av tjenester og funksjonalitet. Katalogene kan derfor nyttes for å støtte flere behov i virksomheter. Nåværende kataloger, av denne typen, støtter normal standarden for X.500.
- metakataloger (meta-directories) – er kataloger som støtter en fremgangsmåte for å administrere og integrere data lagret i ulike kataloger. Metakataloger bruker agenter til å samle informasjon fra diverse nettverks- og applikasjonskataloger.

Archie Reed [RA00] definerer følgende typer kataloger:

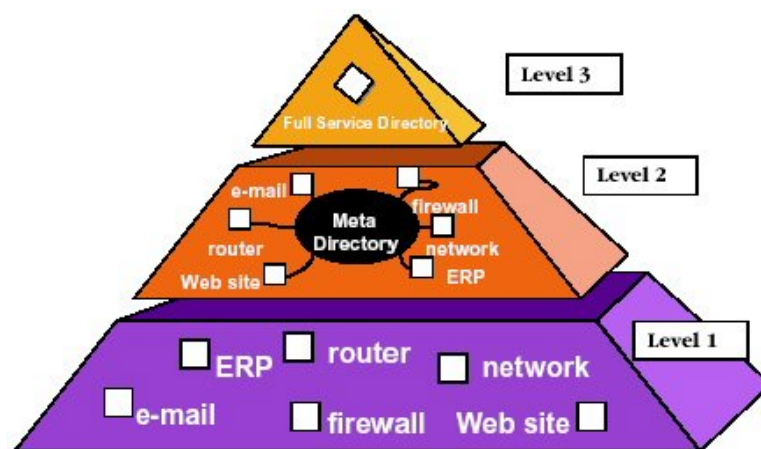
- applikasjonskataloger (applications-specific directories) – hver applikasjonskatalog dekker et spesifikt behov og er normalt optimalisert for å dekke dette behovet. Det eksisterer flere typer generiske kataloger som inngår i begrepet applikasjonskataloger. Eksempler på dette er filsystem, kataloger i NOS, kataloger for e-post, kataloger for støtteapplikasjoner og sikkerhetstjenester.
- foretakskatalog (enterprise directories) – denne type kataloger er vanligvis basert på åpne standarder. Katalogene er fleksible i design og struktur, og blir ofte brukt for å støtte flere typer applikasjoner. Katalogene må også kunne brukes for å støtte andre typer tjenester i organisasjonen.
- metakataloger (metadirectories) – vi kan definere metakataloger som en samling av alle kataloger i en organisasjon. Samlingen er en mekanisme hvor metakataloger setter sammen objekter fra flere datakilder.

Aberdeen Group [AGI99] klassifiserer katalogsystemer på *bakgrunn* av hvilke elementer systemer fokuserer på:

- physical objects
- users
- applications
- systems
- network services
- network.

Aberdeen Group definerer kataloger i tre nivå (se figur 3-9):

- nivå (level) 1 – de fleste organisasjoner har allerede kataloger av denne typen. Katalogene lagrer egne data i egne kataloger eller i isolerte systemer.
- nivå (level) 2 – dette nivå representerer en katalogstruktur der vi bruker *en katalog* (metakatalog) for å integrerer flere kataloger i eksisterende (isolerte) systemer.
- nivå (level) 3 – denne type katalog konsoliderer data på samme måte som kataloger på nivå 2. Det er to forskjeller mellom nivå 2 og nivå 3; skalering og service. Nivå 3 støtter utvidet skalering og bedre service i forhold til kataloger på nivå 2.

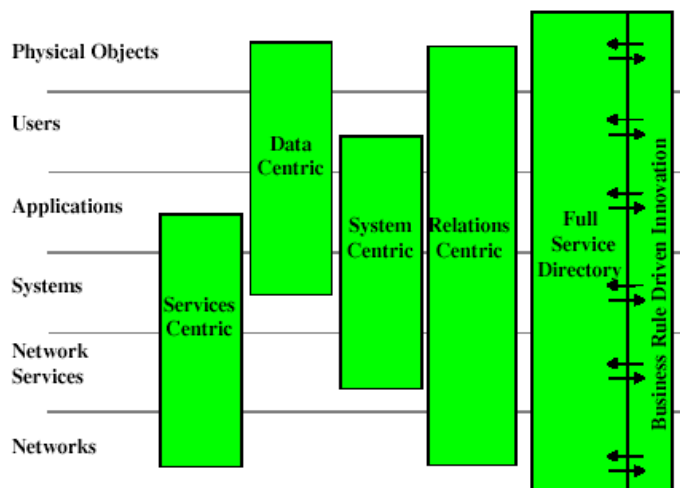


Figur 3-9 – Nivå av katalogsystemer [AGI99].

De ulike aktørene i markedet, som for eksempel IBM [IBM01b], Netscape [NET01], Oracle [ORA01], Microsoft [MS01], Novell [NOV01], har forskjellig tilnærming i utviklingen av katalogsystemer. På bakgrunn av dette har Aberdeen Group klassifisert katalogsystemer slik:

- Service-centric – konsolidering av bruker- og systemdata sett i sammenheng med data om nettverk. En sentral teknologi i denne kategorien er sikkerhet i kataloger.
- Data-centric – kataloger av denne typen har fokus på bruker og data for applikasjoner.
- System-centric – kataloger, i denne klassen, er designet for konsolidering av data innen bruker-, system-, applikasjons- og nettverksdata.

- Relations-centric – har egenskaper til å samle alle relaterte data i en katalog. Dette gir mulighet for å se relasjoner mellom datakilder, beskytte kildene og gjøre verdiene, i kildene, tilgjengelig ved behov.
- full directory services – støtter interaksjon av alle typer elementer i et nettverk (objekter, brukere, applikasjoner ect). Innholdet i elementene er nært koblet til forretningsprosessene i virksomheten.



Figur 3-10 – Klassifisering av katalogsystemer [AGI99].

Figur 3-10 skisserer hvordan Aberdeen Group klassifiserer katalogsystemer. Figuren viser sammenhengen mellom elementer og klasser i kataloger.

3.2.4 Sikkerhet

Sikkerhet er av stor betydning for systemer som bearbeider og lagrer data. Det er nødvendig å beskytte innholdet i slike systemer. Det gjelder uansett om vi bruker katalog, database eller filsystem. Og tema sikkerhet blir ikke mindre viktig når en hel verden skal kunne aksessere data, om produkter og tjenester, som er lagret i intranett- og internettløsningen for virksomheter.

Virksomheter, som har flere systemer, har med stor sannsynlighet ulike modeller for sikkerhet. Dette skaper en kompleks sammenstilling av mekanismer for sikkerhet. I større nettverk er det både kompetanse- og ressurskrevende å vedlikeholde og administrere flere sikkerhetsmodeller.

På et overordnet nivå bør virksomheter ha retningslinjer og iverksatt tiltak for [RA00]:

- fysisk sikring
- sikring av data
- sikkerhetsovervåkning
- nettverkssikkerhet.

Vi kan forbedre mekanismer for sikkerhet i katalogsystemer på flere måter. Vi kan implementere flere sikkerhetsstandarder og teknologier som støtter skalerbar sikkerhet og aksesskontroll i nettverk. Kataloger kan også integrere

støtte for industrisikkerhet og krypteringsstandarder som public key certificates og the Kerberos. I tillegg kan vi nytte sikkerhetsmekanismer integrert i NOS [SS00].

I de neste avsnittene vil jeg presentere *noen* metoder, mekanismer og standarder for autentisering⁴ og kryptering⁵. Jeg vil gi en kort presentasjon av:

- Public Key Certificates (PKI)
- Simple Authentication and Security Layer (SASL).
- Kerberos
- Secure Socket Layer (SSL)
- Transport Layer Security (TLS)
- Private Communicating Technology (PCT).

Leverandører kan normalt integrere disse elementene i NOS eller katalogsystemer.

Public key infrastructure (PKI)

PKI er system for administrasjon av sertifikater. Sertifikatene tilbyr en sikker mekanisme for autentisering og støtte for kryptering. Denne type teknologi løser problemer ved autentisering av mennesker og systemer som ønsker å gjøre forretninger med kunder [RA00].

Det er tilgjengelig en rekke mekanismer for autentisering med tilhørende forskjellige egenskaper. De viktigste egenskapene er [LØ00]:

- tilgjengelighet for produkter og tjenester
- potensiell god sikkerhet
- eneste metode for å få til åpne systemer for elektronisk signatur.

Telenor ASA og Siemens Information and Communications [TES00] har sammenfallende meninger om PKI som Lie og Ølnes [LØ00]. Selskapene mener at PKI er den beste måten å bringe stringente (bindende) elementer av sikkerhet ut i et stort heterogent område. Telenor har satset mye på denne type teknologi.

Simple authentication and security layer (SASL)

SASL er et utvidet protokoll-uavhengig rammeverk for gjennomføring av autentisering og forhandling av parameter for sikkerhet. I SASL kan en klient spesifisere hvilken protokoll den ønsker å bruke ved autentisering. Om en server støtter protokollen, gjennomfører server og klient en avtale for kommunikasjon [IP01].

En klient kan for eksempel spesifisere at den ønsker å autentisere vha protokollen Kerberos. Om serveren støtter Kerberos vil den indikere dette overfor klienten. Klienten sender deretter en "servicebillett" for LDAP-tjenesten. Serveren verifiserer "servicebilletten" og returnerer et gjensidig tegn

⁴ Autentisering betyr å bevise at en oppgitt identitet er korrekt [LØ00].

⁵ Kryptere betyr å behandle (data) slik at det blir uleselig uten bestemt utstyr [BG86].

for autentisering til klienten. Når autentiseringen er ferdig sender serveren en kode til klienten.

Lightweight Directory Access Protocol (LDAP) versjon 3 (v3) støtter bruk av SASL. Dette gir LDAP mulighet til å integrere nye metoder for autentisering i LDAP uten revisjon av protokollen. Eksempel på nye metoder er bruk av smartkort eller biometrisk autentisering.

I dag er det tilgjengelig flere mekanismer for SASL [SUN01]:

- Anonymous (RFC 2245) [IET01]
- CRAM-MD5 (RFC 2195) [IET01]
- Digest-MD5 (RFC 2831) [IET01]
- External (RFC 2222) [IET01]
- Kerberos V4 (RFC 2222) [IET01]
- Kerberos V5 (RFC 2222) [IET01]
- SecurID (RFC 2808) [IET01]
- Secure Remote Password
- S/Key (RFC 2222) [IET01]
- X.509 [ISO01, ITU01, SS00].

Nedenfor følger en utvidet omtale av Kerberos.

Kerberos

Protokollen er en hemmelig *metode* for autentisering. Her er både bruker, og tjenesten for sikkerhet, kjent med det hemmelige passordet til bruker. Kerberos krypterer passordet for bruker. På bakgrunn av krypteringen blir det etablert en *billett*. Billetten blir senere brukt til å autentisere bruker ved aksess til ressurser i nettverket [SS00].

Lie og Ølnes [LØ00] uttrykker at Kerberos er en spesifikasjon for en tjeneste for autentisering. Tjenesten er basert på symmetrisk kryptografi og passord. Kerberos er populær, og er blant annet integrert i Microsoft Windows 2000 [MS01].

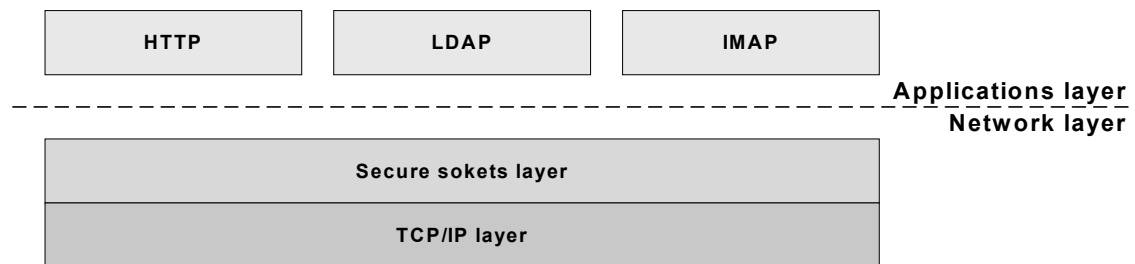
Å tilby Kerberos i kataloger gir flere fordeler. Det gir blant annet raskere autentisering mot server, delegering av brukerautentisering og transitive sikkerhetsrelasjoner [SS00].

Ulemper med Kerberos er at slike systemer virker godt innad i en organisasjon, men har svakere samvirke mellom forskjellige organisasjoner. For å rette på dette kan vi integrere Kerberos internt i virksomheten og PKI-løsninger mot eksterne organisasjoner [LØ00].

Secure socket layer (SSL)

SSL er en protokoll. Den er blitt akseptert som en universal protokoll for autentisering og kryptering av kommunikasjon mellom klient og server. Protokollen ble opprinnelig utarbeidet av Netscape [NET01].

SSL er tilknyttet mekanismer for sikkerhet i transportlaget. Dvs protokollen er forholdsvis nært tilknyttet til TCP/IP (se figur 3-11).



Figur 3-11 – SSL i relasjon til TCP/IP [NET01].

Slik beskriver Netscape [NET01] prosessen for autentisering og kryptering vha SSL over Internett og nett basert på TCP/IP:

- autentisering av SSL-server tillater en klient å få bekreftet identiteten til en server. At en klient kan bekrefte identiteten for en server er viktig. Eksempel på dette er når en klient oversender et kredittkortnummer ved handel.
- en server kan autentisere en klient for å få bekreftet identiteten for bruker. Autentiseringen skjer på samme måte som når en klient får bekreftet identiteten for en server. En slik bekreftelse er også viktig for en server. Eksempel på dette er når en bank sender konfidensielle finansielle data til en kunde.
- en kryptert SSL-forbindelse krever at alle data mellom server og klient er kryptert. Data blir kryptert i applikasjon hos avsender og dekryptert i applikasjon hos mottaker. Dette gir høy grad av konfidensialitet (fortrolighet). Konfidensialitet er viktig for begge parter uansett transaksjon.

SSL bruker TCP/IP i kommunikasjonsprosessen. Dette blir gjort på vegne av protokoller på et høyre nivå. Prosessen tillater altså en server, med SSL, å autentisere seg selv mot en klient med SSL, tillater en klient å autentisere seg mot en server og tillater begge maskiner å etablere en kryptert forbindelse.

SSL kan være nyttig ved kommunikasjon til og fra katalogsystemer. SSL er av stor betydning der høy grad av konfidensialitet er nødvendig.

Private communicating technology (PCT)

PCT er en protokoll som er utviklet av Microsoft [MS01] og Visa [VIS01]. Meldingsformatet er forholdsvis lik SSL. Den er utformet slik at en server kan kommunisere med klienter som støtter både SSL og PCT [SI99B, BLS95].

Det er noe forskjell mellom PCT og SSL. Forskjellen er i hovedsak i design av meldingsutveksling (handshake) [SI99B, BLS95]:

- PCT bruker færre meldinger mellom server og klient, og meldingene er kortere.
- PCT har flere muligheter i forhandlingsprosessen for algoritmer og dataformater.
- PCT bruker flere forskjellige nøkler ved autentisering og kryptering av meldinger. SSL bruker samme nøkkel til begge prosesser.
- i prosessen for autentisering er respons fra klient avhengig av krypteringsalgoritmen som er brukt ved utvekslingen av meldinger. SSL har ikke denne muligheten, og det er fremhevet som en svakhet for SSL.
- i fasen for meldingsutveksling (handshake) er det lagt til et felt for "innledende verifisering". Dette er gjort for at server og klient skal kunne sjekke om for eksempel kodetype i forhandlingene ikke er tuklet med.

Transport layer security (TLS).

Lien og Ølnes [LØ00] fremhever at for sikkerhet i transportlaget er SSL og Transport Layer Security (TLS) mest aktuelt. TLS er beskrevet i RFC 2246 [IET01].

Det er liten forskjell på SSL versjon 3 (v.3) og TLS versjon 1 (v.1). Protokollene er kompatibel med hverandre. SSL er en proprietær spesifisering, men det er TLS som vil bli standard for Internett. Igor Slavinsky [SI99B] omtaler TLS som etterfølger for både SSL og PCT.

Avsnittene ovenfor viser at tema for sikkerhet, i og rundt katalogsystemer, er meget omfattende og komplisert. Ved design av katalogsystemer bør vi vie stor oppmerksomhet på dette området.
--

3.3 Standarder og spesifikasjoner

Dette avsnittet omtaler standarder og protokoller tilknyttet eller nært tilknyttet katalogsystemer. Jeg vil spesielt fokusere på standardene X.500, LDAP og DNS.

3.3.1 X.500

X.500 er en samling standarder (se tabell 3-1) som kollektivt spesifiserer distribuerte katalogsystemer. Standardene definerer en global katalogtjeneste som inneholder og håndterer et logisk sett av data som er distribuert innen et bestemt område [SS00].

Standardene er utviklet av International Standards Organization (ISO) [ISO01] og International Telecommunication Union (ITU) [ITU01]. Første versjon av X.500 ble introdusert i 1988, og revidert i 1993 og 1997.

Modellen for X.500 definerer en hierarkisk katalog. Katalogen opererer med et enhetlig navnerom og lagrer data i en distribuert DIB. X.500 nytter en strukturert metode for å tildele navn for objekter. Metoden gjør at et objekt, uten problemer, kan bli identifisert og aksessert. Objektene er representert i DIT og representerer elementer av det informasjonssystem som katalogene er opprettet for.

Tabell 3-1 – Standarder for X.500 [SS00].

Dokumentnr	Tittel	Introduisert
X.500	The Directory: Overview of Concept, Models and Services	1988
X.501	Models	1988
X.509	Authentication Framework	1988
X.511	Abstract Service Definition	1988
X.518	Procedures for Distributed Operation	1988
X.519	Protocol Specifications	1988
X.520	Selected Attribute Types	1988
X.521	Selected Object Classes	1988
X.525	Replication	1993
X.530	Use of Systems Management for Administration of the Directory	1997

Sheresh & Sheresh [SS00] fremhever også at X.500-standardene definerer at vi skal bruke en database (DIB) for å lagre katalogdata. Strukturen i databasen gir oss mulighet til å lagre og bruke databasen på flere enheter. Vi kan altså spre DIB på flere servere. Dette gir oss mulighet for å redusere problemer ved oppslag og forbedre feiltoleranse i DIB.

Agenter og protokoller

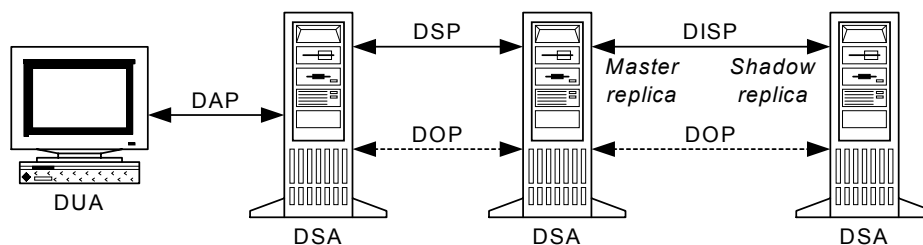
Katalogsystemer, som er basert på X.500, bruker agenter ved kommunikasjon mellom klient og server. Agentene blir brukt for å understøtte operasjoner tilknyttet katalogene. Applikasjonen, som har rollen som agent for server, blir kalt Directory System Agents. (DSA). For klient blir applikasjonen kalt Directory User Agents (DUA).

X.500-standardene spesifiserer også andre nettverksprotokoller. Protokollene blir blant annet brukt for å operere server- og klientelementer i katalogsystemet [SS00]:

- Directory Access Protocol (DAP) - er en protokoll som klient nytter for å kommunisere med katalogsystem.
- Directory System Protocol (DSP) - støtter interaksjon mellom DSA-er ved distribuerte katalogoperasjoner.
- Directory Operational Binding Shadowin Protocol (DOP) - et par DSA-er bruker DOP til å etablere "bindende avtaler" for bruk i distribuerte operasjoner.

- Directory Information Shadowing Protocol (DISP) - er brukt av en DSA til å replikere en partisjon til en annen DSA.

Forholdet mellom agenter og protokoller er skissert i figur 3-12.



Figur 3-12 – Forholdet mellom protokoller og agenter i X.500 [SS00].

Modeller

X.500-standarden bruker flere modeller for å beskrive ulike deler av katalogsystemet. Modellene gir oss mulighet til å skissere datastrukturer, operasjoner, administrasjon og sikkerhetsmekanismer i katalogsystemet. Modellene gir også mulighet for å få et sammensatt og bedre bilde at funksjonaliteten i systemet.

Per i dag opererer X.500 med følgende modeller [SS00]:

- Directory Functional Model - overordnet bilde over hvordan katalogen fungerer.
- User Information Model - viser hvordan katalogen ser ut for bruker.
- Operational and Administrative Information Model - viser hvordan katalogen ser ut for administrator.
- DSA Information Model - beskriver hvordan DSA-er opererer for å tilby aksess til kataloger.
- Directory Distribution Model - beskriver regler for distribusjon av informasjon mellom DSA-er.
- Directory Administrative Authority Model - definerer hvordan katalog(er) blir administrert.
- Security Model - definerer rammeverk for autentifikasjons-/aksesskontroll.

Entiteter og objekter

Objekter i kataloger blir ofte forbundet med de som representerer trestruktur eller entiteter i nettverk. Men slike objekter er ikke de eneste i kataloger. X.500 definerer tre typer objektklasser.

Klassene er oppdelt etter hvilke funksjoner objekter har i kataloger [SS00]:

- **Abstract** – klassen er viktig for kataloger, men bruk av klassen er svært begrenset. Per i dag er det definert en abstrakt klasse i X.500, og den blir kalt *top*. Klassen tilbyr et sett med basis egenskaper som blir brukt av hvert objekt i katalogen.
- **Auxiliary** – objekter i denne klassen blir ofte kalt "*non-effective-objects*" fordi de ikke er en del av selve treet i katalogen. Objekter i denne klassen blir brukt for å skape objekter i selve katalogtreet.
- **Structural** – er en objektklasse, for objekter, som representerer selve innholdet i katalogen. Objektet kan bli manipulert av nettverksbruker og administrator. Objekter i denne klassen blir også kalt "*effective-objects*". I henhold til X.500 kan vi dele denne klassen inn i tre underliggende typer:
 - **Container Objects** – objekter av denne typen blir brukt for å lage og strukturere hierarkiet i treet.
 - **Leaf Objects** – representerer en entitet i katalogen.
 - **Alias Objects** – sekundær representasjon av eksisterende objekter. De fungerer som pekere til originalt objekt.

Navnemodell

X.500-standarden har en navnemodell. Hvert objekt i DIT har et *Distinguished Name* (DN). DN er bygd opp av stien fra root til objektet og objektet i seg selv. Navnet på selve objektet blir også kalt *Relative Distinguished Name* (RDN) [RA00, SS00].

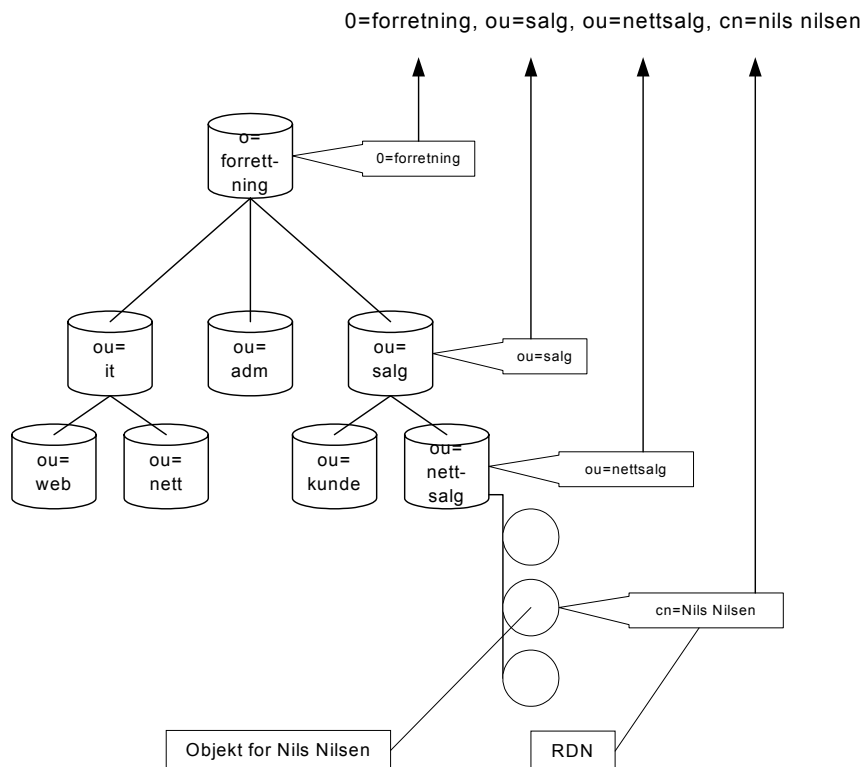
Hver navneattributt i katalogobjekter blir definert med egne bokstavforkortelser:

- C = Country
- L = Locality
- O = Organization
- OU = Organizational Unit
- CN = Common Name

Vi kan for eksempel definere et objekt i en katalog med følgende DN:

C=nor , O=forretning, OU=salg, OU=nettsalg, CN= Nils Nilsen

RDN, i dette eksemplet, er altså selve objektet eller "CN= Nils Nilsen".



Figur 3-13 – Forholdet mellom DN og RDN i DIT [SS00].

Objektet og dets lokasjon er unikt definert i DIT. Forholdet mellom DN og RDN er skissert i figur 3-13.

Operasjoner

Når vi skal kommunisere med en X.500-katalog er det nødvendig å bruke flere typer operasjoner. Slike operasjoner er også definert i X.500-standardene. Kjerneoperasjoner i X.500 er autentisering, lokasjon av objekter, manipulering av objekter og søk i kataloger.

Disse operasjonene er nært tilknyttet DSA, og er initiert etter forespørsel fra klient (DUA) eller andre DSA-er. Nedenfor følger en oversikt over noen kategorier operasjoner [SS00]:

- Binding and Authentication Operations – for at DUA skal kunne snakke med DSA må "partene" først bli enige om de kan, og evt hvordan de skal snakke sammen. DAP bruker Directory Bind for å etablere kontakt mellom DUA og DSA, og Directory Unbind for å stoppe sesjonen. Operasjonen Compare tilhører også denne gruppen. Den blir spesielt brukt ved autentisering ved å sammenligne lagret verdi, for en attributt, mot en verdi oppgitt av klient.
- Object Name Resolution – hver operasjon må starte med å lokalisere objekt eller objekter i selve operasjonen. Om vi bruker distribuerte kataloger snakker de ulike DSA-ene sammen for å finne lokasjonen for objekter. Prosessen for lokasjon blir ofte kalt *name resolution*.

- Access Operations – vi kan bruke ett sett med operasjoner på katalogobjekter, som inkluderer søk, legge til, slette, modifisere etc (se tabell 3-2).

Tabell 3-2 – Operasjoner i X.500 [SS00].

Operasjon	Beskrivelse
Read	Leser data fra katalogpost, og gir mulighet for å spesifisere hvilke attributter som skal returnere verdi.
List	Lister innhold i en konteiner (container).
AddEntry	Legger til et nytt katalogobjekt i spesifisert konteiner. Objektet må tilfredstille spesifikasjonene i skjema.
ModifyEntry	Endrer verdi i en eller flere attributter i en katalogpost. Verdier av attributter kan bli lagt til, endret eller erstattet med funksjonen.
RemoveEntry	Fjerner et enkelt objekt.
ModifyRDN	Endrer RDN for et objekt.
Search	Søker på innhold i en enkel katalogpost, enkel konteiner og dets innhold eller søke i hele treet (subtre).
Abandon	Anmoder DSA om å stoppe utestående forespørsel.

Sikkerhet

X.500 har egne spesifikasjoner for sikkerhetsmodeller og operasjoner. Spesifikasjonene er blant annet definert i X.500 og X.509 [RA00, SS00].

Design av sikkerhet er spesiell i X.500. Dette fordi X.500 både er tilbyder og bruker av sikkerhetstjenester. Som tilbyder tilbyr katalogen tjenester for autentisering og aksesskontroll til applikasjoner og andre nettverkstjenester. Som bruker av sikkerhetstjenester blir autentisering og aksesskontroll brukt for objekter i katalogen.

Kataloger bruker en egen prosess ved autentisering. Prosessen består av ulike operasjoner for å verifisere identifikator for bruker. X.509 dokumenterer et rammeverk for autentisering, og definerer samtidig tre sikkerhetstjenester:

- Simple Authentication – katalogen bruker klar tekst for brukernavn og passord for å verifisere identiteten til bruker. Agentene DSA og DUA blir brukt ved denne form for autentisering.
- Strong Authentication – bruker *public-key* (PK)-kryptografi for å kontrollere aksess til katalogen. Denne type autentisering blir betraktet som mer sikker ved overføring av sensitive data. Agentene DSA og DUA blir også brukt i denne prosessen.
- Digital Signature – mekanisme for å verifisere at overført data er mottatt, og bekrefte identiteten for avsender.

Mekanismer for aksesskontroll tilbyr ulike aksessnivå til objekter og operasjoner i kataloger. X.500-spesifikasjonen beskriver to modeller for aksesskontroll [SS00]:

- Basic Access Control (BAC) – modellen beskriver mekanismer for aksesskontroll til innholdet i katalogen.
- Simplified Access Control – modellen er et subset av skjema for basic access control.

X.500 består av mange underliggende spesifikasjoner. Jeg oppfatter spesifikasjonene som meget omfattende og kompleks.

3.3.2 LDAP

Når vi skal aksessere kataloger bruker vi egne protokoller. X.500-standaren definerer en egen aksessprotokoll. Den blir kalt X.500 *Directory Access Protocol* (DAP). Protokollen blir brukt til lese og modifisere innholdet i kataloger [SS00, JBH98].

Tabell 3-3 – RFC-dokumenter for LDAP versjon 3.

RFC nummer	Navn
2251	Lightweight Access Protocol (v.3)
2252	Lightweight Access Protocol (v.3): Attribute Syntax Definition
2253	Lightweight Access Protocol (v.3): UTF-8 String Representation of Distinguished Name
2254	The String Representation and LDAP search Filters
2255	The LDAP URL format
2256	A summary of X.500 (96) User Schema for use with LDAP v.3

LDAP er en annen aksessprotokoll. Den ble utviklet for å imøtekomme svakhetene i DAP. LDAP er ikke en del av X.500-standaren, men en åpen industristandard. Spesifikasjonen for LDAP er gjengitt i flere RFC-dokumenter (se tabell 3-3). Versjon 3 (v.3) er siste versjon av LDAP.

LDAP er mer utbredt enn DAP. Årsaken til dette er at LDAP har egenskaper som gjør protokollen enklere å implementere og bruke. Nedenfor er noen av disse egenskapene for LDAP [SS00, JBH98]:

- X.500 spesifiserer at DAP skal kommunisere gjennom hele stakken for Open System Interconnect (OSI). Dette er ressurskrevende og påvirker effektiviteten i systemet protokollen opererer i. LDAP kommuniserer gjennom TCP/IP-protokollen. LDAP kobler egne meldinger direkte til TCP i protokollen. Dette gjør LDAP mer effektiv i forhold til DAP.
- LDAP har en enklere funksjonsmodell enn X.500. LDAP omgår flere spesielle og kompliserte funksjoner. Det gjør LDAP enklere å forstå og implementere.

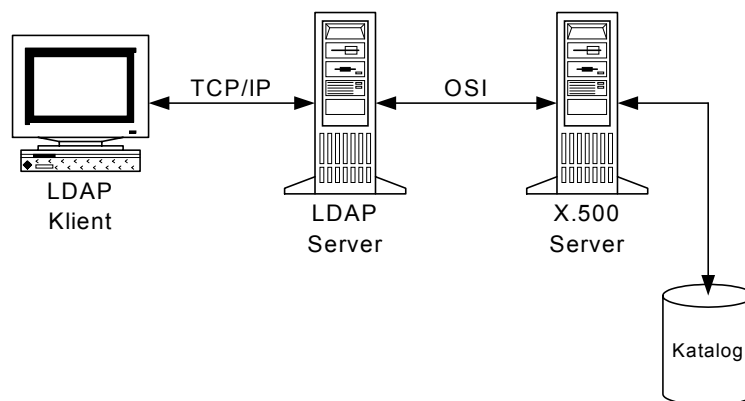
- LDAP bruker strenger til å representere data, mens X.500 bruker kompliserte strukturerte syntakser som ASN.1 (Abstract Syntax Notation One).

De neste avsnittene vil omtale flere nøkkelementer i LDAP.

Protokoll eller katalog

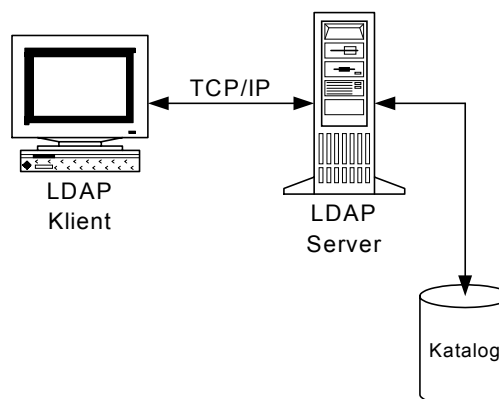
X.500-kataloger forstår i utgangspunktet ikke LDAP-meldinger. Dette fordi LDAP-klienter og X.500-servere bruker ulike protokoller for kommunikasjon (TCP/IP vs OSI).

Om vi nytter LDAP som aksessprotokoll, og ønsker å kommunisere med en X.500-server, må vi bruke en gateway. Gateway-en blir kalt LDAP-server, og har som oppgave å viderefremde meldinger mellom LDAP-klient og X.500-server. LDAP-serveren må kunne kommunisere med både TCP/IP og OSI (se figur 3-14).



Figur 3-14 – LDAP-server som gateway til X.500-server [JBH98].

LDAP kan kommuniserer direkte med en katalog (se figur 3-15). En slik katalog (LDAP-server) støtter ikke X.500. Den støtter kun egenskaper påkrevd av LDAP.



Figur 3-15 – LDAP-klient og LDAP-server [JBH98].

Ser vi disse løsningene fra en brukers perspektiv så vil han eller hun oppfatte en server, som bruker LDAP-protokoll, som en LDAP-server. Brukeren vil ha samme oppfatning, om serveren, om vi bruker en gateway mot en X.500-katalog.

Modeller

Vi kan forstå og beskrive protokollen ved å modellere LDAP i flere perspektiv. Modellene beskriver blant annet operasjoner og strukturer i protokollen.

Sheresh & Sheresh [SS00] beskriver LDAP med to modeller:

- datamodell – modellen inneholder kjerneelementer av X.500-informasjonsmodeller. Elementene beskriver struktur og innhold i LDAP-kataloger.
- protokollmodell – modellen beskriver spørringer og operasjoner for oppdateringer. Modellen spesifiserer blant annet operasjoner for katalogfunksjoner som inkluderer lage, slette, modifisere, endre og spørre.

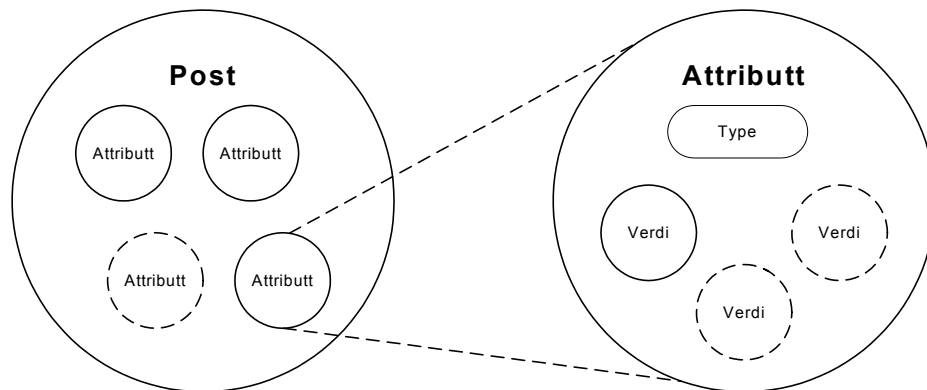
Johner, Brown, Hinner m fl [JBH98] bruker fire modeller for å beskrive LDAP:

- informasjonsmodell – beskriver struktur over hvordan data er lagret i LDAP-kataloger.
- navnemodell – beskriver hvordan data er organisert og identifisert.
- funksjonsmodell – beskriver hvilke operasjoner vi kan bruke på innhold i kataloger.
- sikkerhetsmodell – beskriver hvordan data i LDAP-kataloger kan bli beskyttet fra uautorisert aksess.

Både Sheresh & Sheresh [SS00] og Johner, Brown, Hinner m fl [JBH98] fremhever at nevnte LDAP-modeller har sitt utspring fra DAP i X.500.
--

Informasjonsmodell

I informasjonsmodellen er poster det grunnleggende element i kataloger. En post representerer objekter i den virkelige verden – som mennesker, servere, organisasjoner etc. Hver post består av flere attributter. Attributtene beskriver egenskaper til objektet. Hver attributt har en definert type (syntaks) og en eller flere verdier (se figur 3-16).



Figur 3-16 – Poster, attributter og verdier [JBH98].

Navnemodell

Navnemodellen definerer hvordan poster blir identifisert og organisert. Poster er lagret i DIT og er ordnet i treet etter navnet (DN) på objektet. Navnemodellen i LDAP er som navnemodellen i X.500. Det er noe forskjell mellom modellene, og navnehåndtering i LDAP skiller seg fra X.500 på to områder [MET00]:

1. LDAP bruker invertert trestruktur ved hierarkisk representasjon av treet i kataloger.
2. LDAP er mindre rigid ved defineringen av trestruktur i forhold til X.500.

Eksempel på representasjon av DN i X.500 er:

```
C=nor , O=forretning, OU=salg, OU=nettsalg, CN= Nils Nilsen
```

Samme representasjon av DN i LDAP:

```
CN=Nils Nilsen, OU=nettsalg, OU=salg, O=forretning, C=nor
```

Funksjonsmodell

Funksjonsmodellen definerer operasjoner for å aksessere eller modifisere poster i kataloger. Vi kan dele LDAP-operasjoner i tre kategorier:

- spørre (Query) – operasjoner for søk og sammenligning.
- oppdatere (Update) – operasjoner for å legge til, slette og modifisere.
- autentisere (Authentication) – operasjoner for å koble til og koble fra kataloger.

Operasjonene har stort sett samme funksjon og navn som i DAP (se tabell 3-4). De ni første operasjonene er basisoperasjoner for LDAP [MET00].

Tabell 3-4 – Operasjoner i LDAP [SS00, MET00].

Operasjon	Beskrivelse
Search	Søker og gjenfinner data i katalog (som Search i DAP).
Compare	Sammenligner verdier (som Compare i DAP).
Add	Legger til en ny post i katalog (som AddEntry i DAP).
Delete	Fjerner post i katalog (som RemoveEntry i DAP).
ModifyDN	Endrer RDN i et objekt eller flytter objektet/subtre til en annen lokasjon i katalogtreet.
Modify	Oppdaterer eksisterende poster i katalog (som ModifyEntry i DAP).
Bind	Oppretter en sesjon (kobling) mellom klient og tjener (som DirectoryBind i DAP).
Unbind	Avslutter sesjon mellom klient og server (som Directory Unbind i DAP).
Abandon	Avbryter tidligere initiert operasjon (som Abandon i DAP).
Unsolicited notification	Meldinger fra server til klient om tilstand på server som klient bør være oppmerksom på.
Extended operation	Utvidet operasjon for å legge til nye operasjoner til LDAP-protokollen. Disse operasjonene er lagd for en spesiell hensikt eller definert i RFC.

Sikkerhetsmodell

Sikkerhetsmodellen i LDAP er basert på et rammeverk for å beskytte data i kataloger mot uautorisert bruk [IP01]. Sikkerhetsmekanismene i LDAP har nær sammenheng med generelle sikkerhetsmekanismer for katalogsystemer (se 3.2.4 Sikkerhet).

Sikkerhetsmodellen er basert på bind-operasjoner, og LDAP støtter flere typer bind-operasjoner. Antall og variasjoner av slike operasjoner er avhengig av hvilken teknologi eller protokoll vi bruker i LDAP [MET00].

LDAP v.3 støtter rammeverket SASL, og kan dermed bruke operasjoner (bind) og protokoller som for eksempel Kerberos, SSL og TLS. Implementering av nevnte mekanisme/teknologi avhenger av hvilken versjon av LDAP vi bruker [JBH98, MET00].

3.3.3 DNS

Tidligere i avhandlingen har jeg introdusert og presentert bakgrunn for navnemodell(er). DNS (eller Domain Name System) er et eksempel på en slik modell. Jeg vil nå presentere dette systemet mer inngående.

DNS er et system for hierarkisk gruppering av tjenere (hosts) tilkoblet Internett. Hierarkiet følger en egen struktur som kan skille seg fra grupperinger av IP-adresser eller en geografisk gruppering av tjenere. Systemet har en tjeneste som kobler et navn (på tjener) i DNS til en IP-adresse [HIO01].

Om vi skal kommuniserer med en maskin på et IP-nett (for eksempel Intranett eller Internett) må vi kjenne til IP-adressen for enheten. En IP-adresse består av 4 byte (IP versjon 4), og kan være vanskelig å huske. Adressen sier ingenting om hvem som eier maskinen og hva den inneholder.

For å forenkle bruk av IP-adresser ble det opprettet et system, som setter opp alle verter på Internett i et hierarki, og en tilhørende katalogtjeneste. Hierarkiet blir kalt *Domain Name Space* og tjenesten blir kalt *Domain Name Service*. Både hierarki og tjeneste blir forkortet DNS [HIO01].

DNS blir karakterisert som *begrensede katalog* (specific-use directory). Dette fordi katalogen er dedikert til en eller få bestemte oppgave(r). Sheresh og Sheresh [SS00] fremhever også, på overordnet nivå innen design og funksjonalitet, at DNS og X.500 er like på mange områder.

DNS er spesifisert i flere RFC-dokumenter. Oversikt over dokumenter som helt eller delvis påvirker utformingen DNS er gjengitt i tabell 3-5.

Tabell 3-5 – RFC-dokumenter for DNS [IET01, UIT01].

RFC-nummer	Navn
1034	Domain names - concepts and facilities
1035	Domain names - implementation and specification
1023	Requirements for Internet Hosts
2181	Clarifications to the DNS Specification
1033	Domain administrators operations guide
1535	A Security Problem and Proposed Correction With Widely Deployed DNS Software
1536	Common DNS Implementation Errors and Suggested Fixes
1537	Common DNS Data File Configuration Errors

Hierarki

Hierarkiet i DNS er delt opp i flere rot-domener (se tabell 3-6). Disse domenenene har flere underliggende domener. USA har egne typer rot-domener. Disse domenenene blir brukt av flere organisasjoner. Resten av verden bruker rot-domener med en kode for et land.

Tabell 3-6 – Rot-domener i USA og resten av verden [HIO01].

Rot-domener i USA						Landkoder - fra Afghanistan til Norge og Zimbabwe					
com	edu	mil	org	gov	net	af	al	no	zw

Under hvert rot-domene er det sub-domener. Det kan eksistere mange nivå med sub-domener.

Når vi skal gjøre oppslag i DNS skriver vi navnet på verter slik:

host-navn . sub-domene(r) . rot-domene
--

Eksempel på et oppslag (navn) i DNS er:

test . idi . ntnu . no

Her er *test* host-navn, *idi* sub-sub-domene, *ntnu* sub-domene og *no* rot-domene.

Det er vert å merke seg at det foregår en utvikling av rot-domener på Internett. I tillegg til eksisterende top-domener (se tabell 3-6), har the International Ad Hoc committee (IAHC) [IAHC01] godkjent sju nye top-domener (se

Tabell 3-7 – Nye top-domener [IAHC01].

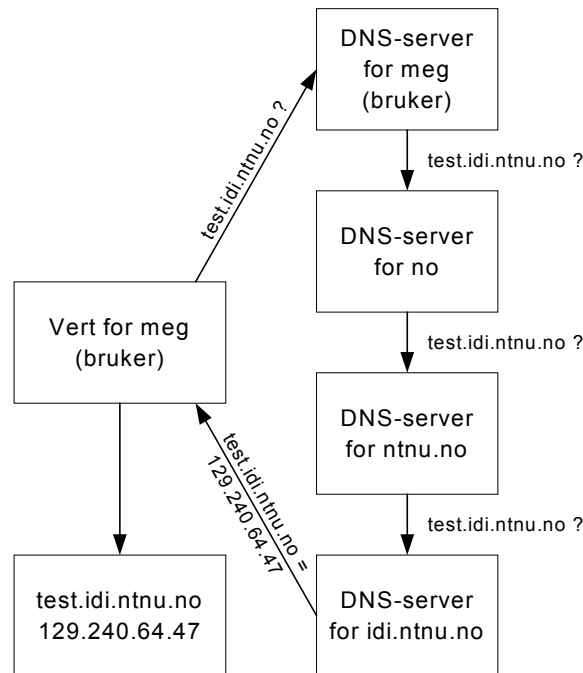
Nye topp-domener						
web	store	firm	info	arts	rec	nom

Tjeneste

Katalogtjenesten i DNS gir oss altså muligheten til å finne riktig IP-adresse for en ressurs, slik at vi kan for eksempel benytte tjenesten på maskinen. Eksemplet nedenfor viser hvordan et navneoppslag for test.idi.ntnu.no blir gjennomført [HIO01]. Eksemplet er arrangert og gjenspeiler ikke DNS-strukturen ved NTNU (se figur 3-17):

1. Jeg bruker en maskin med en applikasjon. Applikasjonen ønsker å sende en melding til test.idi.ntnu.no. Min maskin kjenner ikke IP-adressen til test.
2. Applikasjonen sender en forespørsel til DNS-serveren som betjener min maskin. DNS-serveren kjenner ikke IP-adressen til test.idi.ntnu.no. Forespørselen blir derfor videresendt til DNS-serveren for rot-domenet .no.
3. DNS-serveren for .no kjenner ikke alle verter i alle sub-domener. Men serveren vet hvor alle DNS-servere, til alle sub-domener i rot-domenet .no, befinner seg. Dvs den har IP-adressen til alle DNS-servere. Forespørselen blir deretter sendt til DNS-serveren for ntnu.no.
4. Samme prosedyre blir kjørt som for rot-domenet no: DNS-serveren for ntnu.no kjenner ikke verten test.idi.ntnu.no, men kjenner DNS-serveren for idi.ntnu.no. Forespørselen blir dermed sendt til denne serveren.
5. DNS-serveren for idi.ntnu.no mottar forespørselen, finner IP-adressen for test.idi.ntnu.no og sender den til meg. Dvs applikasjonen på min maskin som startet forespørsel etter IP-adresse.

Resultatet av oppslaget, dvs adressen og navnet for test, blir lagret hos min DNS-server for en kortere periode. Dette gjør at nye forespørsler ikke trenger å "gå samme vei" som tidligere forespørsel.



Figur 3-17 – Oppslag i DNS [HIO01].

Prosessen med å lokalisere enheter på denne måten blir kalt for *name resolution* [SS00].

Agenter

DNS er også basert på en klient-/server-modell. Dvs dedikerte agenter, eller applikasjoner på klient og server, utfører operasjoner for klient og server. I denne sammenheng har DNS to sentrale komponenter [SS00]:

- Name Servers – tilbyr DNS-funksjonalitet fra server.
- Resolver – komponent for klient.

DNS-server (name server) utfører oppslag på bakgrunn av forespørsel fra DNS-klient (resolver). Det er disse komponentene som gjennomfører *name resolution* i DNS.

DNS-servere har også en annen oppgave. DNS skal holde oversikt over hvilken vert som skal ta i mot e-post (e-post server). Dvs når e-post blir sendt må applikasjonen foreta et oppslag i DNS for å lokalisere hvilken vert som skal motta e-posten [HIO01].

3.4 Systemer for katalogtjenester

Katalogsystemer har vært tilgjengelig i flere år, og er ofte levert som del av eller et tillegg til NOS. Eksempler på slike systemer er NDS for Novell NetWare og Active directory [SC00a, SC00b] for Windows Server (se tabell 3-8).

Flere andre miljø er også aktive på dette fagområdet. Utviklingsmiljø, som tradisjonelt sett ikke har utviklet og levert katalogsystemer for "egne" NOS, har nå etablert seg i dette segmentet. Nå leverer de egne løsninger for

katalogtjenester. Eksempler på dette er Oracle med Oracle Internett Directory [ORA01] og alliansen SUN-Netscape med iPlanet Directory Server [IP01].

Tabell 3-8 – Leverandører og produkter av katalogsystemer.

Leverandør	Katalogsystem
Computer Associates [CA01]	eTrust Directory
IBM [IBM01a, IBM01b]	IBM SecureWay Directory
iPlanet [IP01]	iPlanet Directory Server / iPlanet Meta-Directory
MaXware [MAX01]	The MaXware Directory Explorer
Microsoft [MS01]	Microsoft Active Directory
Nexor [NEX01]	NEXOR Directory
Novell [NOV01]	Novell Directory Services (NDS) / Novell eDir / Novell DirXML
Oracle [ORA01]	Oracle Internet Directory

Flere produsenter av katalogsystemer leverer ulike typer kataloger. Dvs leverandørene kan levere kataloger som støtter ulike kategorier katalogsystemer. Tabell 3-8 gir en oversikt over noen aktører med tilhørende produkter for katalogsystemer.

Ved valg av katalogsystemer bør vi være bevisst på alternative katalogsystemer og tilhørende leverandører.

3.5 Oppsummering

Vi har tilgang til mange definisjoner om katalogsystemer. Begrepene er formulert på en måte som gjøre det naturlig å trekke paralleller til data og informasjon, og mennesker og ressurser. Disse definisjonene gir et klart uttrykk for at vi kan bruke katalogsystemer til mer enn bare e-handel.

Det er mulig å beskrive egenskapene med katalogsystemer gjennom *strukturelle* forhold som organisering, navnemodell og lagring i katalogsystemer. Vi kan også skille mellom applikasjoner og katalogsystemer gjennom *egenskaper* som for eksempel navnerom, skalerbarhet, replikering/synkronisering og distribusjon.

Det er mulig å *klassifisere* katalogsystemer etter egenskaper, og vi kan skille mellom: nettverksfokuserede kataloger, foretaks-, applikasjons-, begrensede, generelle-, og metakataloger. Ved innføring og bruk av katalogsystemer bør vi være meget bevisst på egenskaper for katalogsystemer og hvilke kategorier katalogsystemer vi velger å bruke.

Sikkerhet er også et sentral tema for katalogsystemer. Vi kan bruke en rekke mekanismer for sikkerhet som for eksempel public key certificates (PKI), og simple authentication and security layer (SASL) [Kerberos, X.509, S/Key]. Det eksisterer en rekke muligheter og begrensinger med sikkerhetsmekanismer i katalogsystemer. Vi bør vie stor oppmerksomhet på sikkerhet ved valg av katalogsystemer.

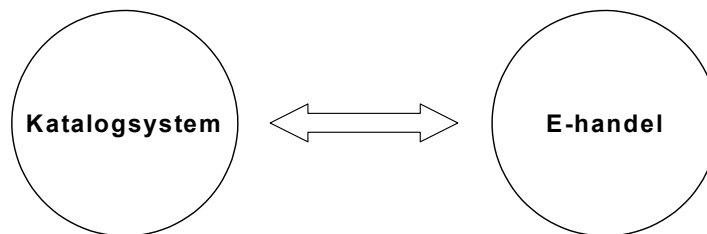
Utforming av katalogsystemer er påvirket av en rekke *standarder* og *protokoller*. X.500 er den mest kjente spesifikasjonen for katalogsystemer. DAP (X.500) og LDAP (industristandard) er navn på aksessprotokoller som vi kan bruke mot kataloger. DNS er en annen spesifikasjon som definerer håndtering av navnerom på Internett. X.500, DAP og LDAP er standarder som direkte påvirker utvikling og bruk av katalogsystemer. Om vi velger katalogsystemer som en sentral del av infrastruktur bør teknologien støtte disse standardene på en eller annen måte.

4 Om katalogtjenester i elektronisk handel

Kapitlet retter søkelys på katalogtjenester i e-handel, og omtaler spesielt metakatalog i arkitektur for e-handel. Kapitler fremhever også fordeler og ulemper med katalogsystemer i e-handel og suksesskriterier for e-handel.

4.1 Innledning

I kap 2. Om elektronisk handel og kap 3. Om katalogsystemer og katalogtjenester har jeg introdusert og omtalte flere tema omkring katalogsystemer og e-handel. Dette er gjort hver for seg. Nå vil jeg rette sterkere søkelys på integrasjon av katalogtjenester, som et sentralt element i infrastrukturen, for e-handel (se figur 4-1).



Figur 4-1 – Fokus på integrasjon av katalogsystemer i e-handel.

Den globale fokuseringen på e-handel retter øynene på forvaltning av data, informasjon og kunnskap. Og elektronisk lagrede data, om produkter og tjenester, vil ha positiv betydning ved overgang fra tradisjonell til elektronisk handel. Dette fordi vi kan gjenbruke data i eksisterende systemer med nye systemer for e-handel. I dette perspektiv fremhever flere globale aktører koblingen mellom katalogtjenester og e-handel som vesentlig [AGI99, RA00, SS00].

Vi kan bruke katalogtjenester for personalisering av kunder, forbedre sikkerhet og forbedre tilgang til distribuerte ressurser. Dette er eksempler på hvordan vi kan bruke katalogsystemer i e-handel.

4.2 Arkitektur

Tidligere i avhandlingen har jeg presentert og beskrevet flere tema innen katalogtjenester og e-handel. Jeg har konsentrert meg om arkitektur, standarder, spesifikasjoner, sikkerhet og systemer for slike løsninger.

Sentrale byggeklosser i infrastruktur for e-handel kan være EDIFACT, XML, OBI og UDDI. For katalogsystemer kan slike byggeklosser være X.500, DAP og LDAP. Valg av slike byggeklosser har stor betydning ved utforming av ny infrastruktur.

Vi kan altså kategorisere katalogsystemer etter om de er applikasjons-, foretaks- eller metakataloger. Litteraturen fremhever at en (metakatalog), eller en kombinasjon av ulike katalogtyper, er egnet for e-handel. På bakgrunn av dette vil jeg omtale egenskapene for metakatalog i de neste avsnittene.

4.2.1 Metakatalog

Kategorien metakatalog er interessant på flere måter. Dette fordi mange virksomheter har heterogene nettverk. Nettverkene vokser frem på bakgrunn av ulike arbeidsoppgaver og dermed ulike behov hos brukere. På bakgrunn av dette består nettverkene av ulike ressurser innen for eksempel NOS, datakilder, sikkerhetsløsninger og applikasjoner.

Datakilder til ressurser

Ressursene i slike nettverk er på en eller annen måte koblet til en datakilde. Kilden har opplysninger om egenskapene for ressursene: data om for eksempel hvordan vi skal konfigurere og bruke ressursene.

Om vi ser slike datakilder i sammenheng har de ofte overlappende innhold. At data om ressurser er lagret i flere mindre kataloger har noen fordeler, men listen over ulemper er lengre. Å lagre egenskaper om ressurser i flere datakilder eller kataloger gir grobunn for blant annet:

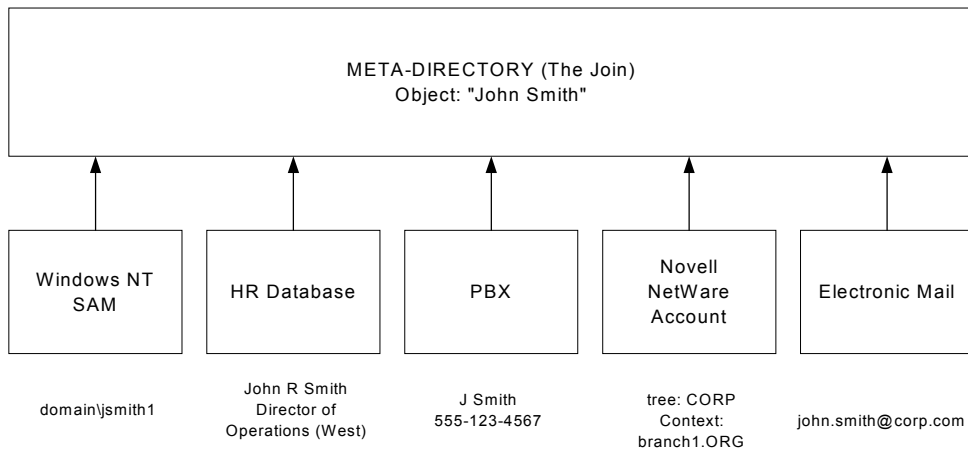
- større mulighet for at feil data blir brukt i virksomheten
- krevende administrasjon av ressurser i nettverket
- økende behov for overordnet administrasjon av virksomheten
- økende utviklings- eller tilpasningskostnader
- krevende å opprettholde sikkerhetstiltak for å beskytte data.

Funksjonalitet og egenskaper

Vi kan beskrive funksjonalitet og egenskaper for metakataloger ved et eksempel. En virksomhet har flere ressurser med tilhørende datakilder. Ressursene kan ha datakilder med data om brukere på denne måten (se figur 4-2):

- nettoperativsystem (NOS/DNS) – for eksempel navn på domene og plassering av bruker i domene.
- personelldata (HR-database) – for eksempel navn og stilling for bruker.
- telefonsentral (PBX) – for eksempel navn og telefonnummer for bruker.
- operativsystem (eller NOS) – for eksempel brukerkonto i nettverk.
- system for e-post – for eksempel e-postadresse for bruker.

Nevnte datakilder viser at egenskapene til brukere er lagret i flere datakilder. Kildene har overlappende (like) attributter og dermed overlappende data. Figuren viser flere attributter, fra flere separate kataloger, koblet sammen til en metakatalog.

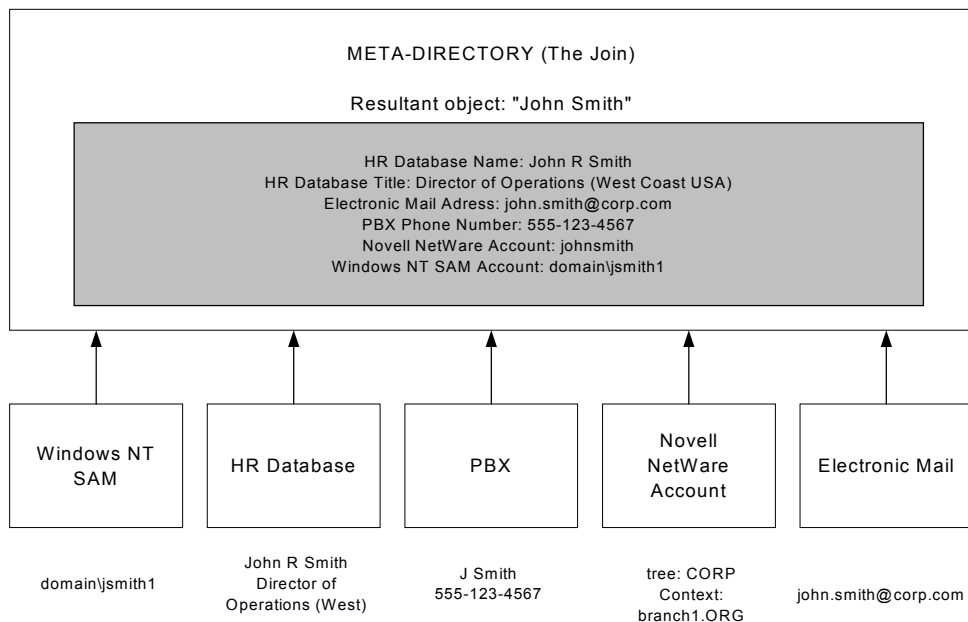


Figur 4-2 – Eksempel på sammenkobling av separate kataloger til metakatalog [RA00].

Vi kan altså beskrive en ressurs ved å sammenstille attributter fra flere datakilder. Archie Reed [RA00] og The Burton Group [TBC98] kaller prosessen for sammenstilling av data for *join*.

Join er et felles begrep for sammenstilling av data i ulike systemer (databaser, katalogsystemer etc)

Resultatet av prosessen blir et objekt med bred beskrivelse av egenskapene for brukeren. Skisse over en slik sammenkobling er presentert i figur 4-3, og viser at attributter om objektet *John Smith* er hentet fra ressurser som NOS, personalsystem, telefonsentral, operativsystem og applikasjonen for e-post.



Figur 4-3 – Eksempel på sammenkobling av attributter fra kataloger til metakatalog [RA00].

Vi bør spesifisere hvilke regler som gjelder for kommunikasjon mellom metakatalog og underliggende kataloger. Eksempel på slike regler er synkronisering/replikering mellom kataloger, navngivning, aksess og

beskyttelse av data i ressurser. Dette gjelder ikke bare mellom metakatalog og underliggende kataloger, men også på tvers av underliggende kataloger og eksterne kataloger.

Utfordringen ved bruk av *join* er å finne hvilke unike identifikatorer, i for eksempel tabeller, vi skal benytte ved sammenstillingen av datakilder.

Metakatalog i e-handel

Eksemplet ovenfor viser hvilke muligheter vi har med metakataloger. Om vi retter søkelyset mot *eksisterende* virksomheter, som ønsker å etablere seg med e-handel, kan metakatalog være en sentral komponent i infrastrukturen.

I dag har de fleste bedrifter etablert en rekke ressurser med tilhørende datakilder. Det kan være systemer som har oversikt over ansatte, økonomi, lagerbeholdning, produkter og tjenester. Innholdet i slike datakilder er av stor interesse om vi skal gjøre produkter og tjenester tilgjengelig på Internett.

Katalogsystemer kan ha en grunnleggende betydning for e-handel. Dette standpunktet blir uttrykt av flere forfattere.

Archie Reed [RA00] uttrykker at katalogsystemer gir flere fordeler i e-handel:

- konsistent identifisering av partner-, leverandør-, kunde- og brukerinformasjon.
- støtter konsistente sikkerhetsmodeller.
- støtter PKI og sikker e-post.
- støtter standarder for sikker og systematiske metoder for allmenn aksess til data i virksomheten.

Aberdeen Group [AGI99] fremhever at katalogsystemer vil ha en meget betydelig rolle i e-handel.

Sheresh & Sheresh [SS00] skildrer at metakataloger har en sentral rolle i integrasjon av ulike datakilder.

The Burton Group [TGB98] skriver at metakataloger er et viktig element i applikasjoner for e-handel.

Netscape [NET01] mener at det er to forhold som fremhever bruk av katalogsystemer i e-handel. Det ene er sikkerhet (aksesskontroll og autentisering), og det andre er muligheten integrasjon av eksisterende applikasjoner og applikasjoner for e-handel.

Løsninger

Det eksisterer flere løsninger for metakataloger i markedet. The Burton Group presenterer en knippe med leverandører som støtter denne katalogtypen [TBG01] (se tabell 4-1).

Tabell 4-1 – Leverandører og produkter av metakataloger

Leverandør	Metakatalog
MetaMerge [MET01]	MetaMerge Integrator (tidligere ArchiTech Server)
Critical Path [CP01]	Critical Path InJoin Meta-Directory
iPlanet [IP01]	iPlanet Meta-Directory
MaXware [MAX01]	MaXware Enterprise Meta Solution
Microsoft [MS01]	Microsoft Metadirectory Services
Novell [NOV01]	Novell NDS DirXML
Siemens [SIM01]	Siemens Meta Directory

Tabellen viser at det eksisterer en rekke alternativer for metakataloger. Vi har med andre ord mange valgmuligheter.

4.3 Gevinster med katalogtjenester i e-handel

Hvilke gevinster vi kan oppnå ved bruk av katalogtjenester er avhengig av hvilken type katalogtjenester (se pkt 3.2.3 Klassifisering) vi bruker og i hvilke deler av bedriften den blir benyttet.

Nedenfor følger en generell oversikt (se tabell 4-2) over fordeler ved bruk av katalogsystemer. Grunnlag for oversikten er hentet fra Archie Reed [RA00] og Sheresh & Sheresh [SS00].

Tabell 4-2 – Gevinster med katalogsystemer

Archie Reed	Sheresh & Sheresh
forenkler og reduserer kostnader ifm utvikling og administrasjon av distribuerte nettverk/systemer	forenkler håndteringen av distribuerte nettverk
forbedrer stabilitet og ytelse i nettverk	forenkler nettverksadministrasjon
støtter flere sikkerhetsløsninger og forbedrer sikkerhet i nettverk	forbedrer nettverkspålitelighet og ytelse
tilbyr kontrollert aksess og bruk av felles data	lik aksess til distribuerte ressurser
støtter personalisering av web-tjenester og applikasjoner	forbedrer sikkerhet
støtter og forbedrer aksess til distribuerte ressurser	
støtter Directory Enabled Networking (DEN)	
støtter Virtual Private Networks (VPN), Public Key Infrastructure (PKI), Single Sign ON (SSO)	
støtter tjenester for oppslag i for eksempel White Pages og Domain Name Service (DNS)	

Tabellen ovenfor viser at virksomheter kan oppnå mange gevinster ved å nytte katalogtjenester.

Om vi ser nevnte gevinster i *relasjon* til e-handel vil jeg spesielt fremheve:

- tilbyr kontrollert aksess og bruk av distribuerte ressurser.
- støtter flere sikkerhetsløsninger og forbedrer sikkerhet i nettverk.
- støtter personalisering av web-tjenester og applikasjoner.

Tilbyr kontrollert aksess og bruk av felles data

Katalogtjenester tilbyr mekanismer som forenkler lokasjon og aksess til ressurser i nettverket. Ved å integrere flere datakilder (for eksempel nettverkstjenester, systemer, kontoer etc) kan katalogtjenester minimalisere kompleksiteten ved aksess til ressursene.

Katalogtjenester gir mulighet for å benytte singel logon-tjeneste. Tjenesten gir tilgang til flere ressurser ved første gangs innlogging. Dette gir mulighet for å forenkle operasjoner og administrasjon i nettverk. Samtidig kan vi forbedre sikkerhet og stabilitet i nettverket.

Kontrollert aksess i nettverk, uansett arkitektur (B2C, B2B eller B2P), gir muligheter for bedre sikkerhet og stabilitet i systemer for e-handel.

Støtter flere sikkerhetsløsninger og forbedrer sikkerhet i nettverk

Katalogsystemer støtter flere løsninger og mekanismer for sikkerhet. Dette gir stor mulighet og fleksibilitet i hvordan vi ønsker å utforme sikkerhetsløsninger for infrastruktur i virksomheten.

Mulighet for å integrere flere mekanismer for sikkerhet gjør katalogsystemer egnet i e-handel.

Støtter personalisering av web-tjenester og applikasjoner

Kunder innen e-handel har behov for å bli gjenkjent, "kjælt med" og oppdatert gjennom for eksempel e-poster [AGI99]. Katalogtjenester kan understøtte ønske om å dekke dette behovet. Tjenester tilknyttet katalogsystemer gir mulighet for administrasjon av profiler for kunder og grupperinger av kunder.

Et eksempel på personalisering er ved generering av profiler for kunder. Profilene blir for eksempel definert og lagret ved første gangs bruk av tjenesten. Profilen blir lagret i eller tilknyttet katalogsystemet. Denne profilen kan vi bruke ved neste gangs innlogging av denne brukeren. Da kan applikasjonen for eksempel tilpasse tilbud, presentasjon av varer og tjenester, og hjelpe kunden ifm kjøp og forsendelse.

Personalisering gir positiv synergieffekt med hensyn på personlig tilrettelegging av web-tjenester og applikasjoner. Og gjennom personalisering støtter katalogsystemer sentrale strategier innen e-handel.

4.4 Ulemper med katalogtjenester i e-handel

Litteraturen fokuserer mye på fordelene med katalogtjenester, men det er også flere ulemper med slike systemer og tjenester. Tor Even Dahl [MAX00a] fremhever følgende om katalogsystemer som er utformet som metakataloger :

- krever ofte endringer i skjema
- vanskelig å lage gode spørringer og rapporter fra katalog
- krever en spesiell kompetanse
- problemer med ytelse.

Krever ofte endringer i skjema

Om vi ikke følger standard skjema, eller forhåndsdefinerte skjema, i katalogsystemet må vi definere disse selv. Dahl fremhever at det kreves ofte endringer i disse skjemaene. Det er både tid- og ressurskrevende.

Vanskelig å lage gode spørringer og rapporter fra katalog

Det er vanskelig å lage gode spørringer og rapporter fra kataloger i forhold til spørringer og rapporter fra databaser. Dette forholdet bør også vies oppmerksomhet ved bruk av kataloger.

Krever spesiell kompetanse

Kompetanse om katalogsystemer er ikke like utbredt som for databaser. Siden katalogsystemer krever "spesiell" eller "ny" kompetanse bør dette også tas hensyn til ved valg og implementasjon av metakataloger.

Problemer med ytelse

Dahl fremhever også at katalogsystemer, som er utformet som metakataloger, ikke vil kunne gi samme ytelse som databaser.

4.5 Suksesskriterier for e-handel

Innføring av systemer for e-handel blir målt og vurdert opp mot hensikten med løsningen. Slike målinger og analyser er nødvendige og viktige. Konklusjonen fra slike analyser gir oss mulighet til å forbedre eksisterende løsninger og bli bedre ved neste oppgradering eller implementering.

For å oppnå suksess på den nye markedsplassen bør vi rette fokus på å forstå kundeverdier. Vi bør arbeide for å kunne forutse nye verdier og behov, for eksisterende og nye kunder, på Internett. Nye kundeverdier og kunderelasjoner gjør det nødvendig å tenke nytt i forhold til tradisjonelle forretningsmodeller [BG99].

I presentasjonen "Suksesskriterier for e-handel" er flere forhold omkring suksess presentert og diskutert. Bryhn og Galtung [BG99] fremhever at vi må fokusere på å tilfredstille verdiene gjennom nye og innovative produkter og tjenester.

Artikkelen konkluderer med at flere kriterier gir mulighet for suksess:

- pris/kostnad - kunne tilby vare eller tjeneste til lav pris/kostnad. Dette gjelder både for selger og kjøper.
- kjernekompetanse - beholde og videreutvikle kjernekompetanse, innen for eksempel salg og distribusjon, i virksomheten.
- kundefokus - helhetlig kundefokus gjennom hele salgsprosessen. Vi kan gjøre dette ved å "outsource" funksjoner, og evt bruke leverandører i verdikjeden, slik at virksomheten kun fokuserer på salg.
- legitimitet og aksept - bruke strategiske alliansepartnere for å skape image og oppmerksomhet.
- kunde verdi - tilfredstille ulike kunde verdier gjennom kundefokus (ikke produktfokus). Dette kan vi gjøre gjennom produkttilpassing, opplæring av kunden eller forenkle kjøpsprosessen.
- betalingsformer - integrere sikker og kundetilpasset betalingsform. Pga motstand mot betaling med kredittkort gir alternative betalingsformer større frihet.
- bygging av merkenavn - aktiv markedsføring og PR for å påvirke ønsket kundesegment.
- teknologistøtte - bruke ny teknologi for å skape merverdi for kunden.
- kundekontakt - aktiv relasjonshåndtering for å skape varige relasjoner mellom kunde og selger.

For å vurdere grad av suksess er det nødvendig å definere hvilke kriterier som gjelder for suksess. Suksess i et prosjekt er ikke nødvendigvis det samme i et annet prosjekt.

4.6 Oppsummering

Vi kan bruke katalogsystemer i infrastruktur for e-handel. Om vi skal sammenstille mange datakilder gjennom katalogsystemer, for å presentere innholdet på Internett, bør vi utforme katalogen som en metakatalog.

Det er flere gevinster og ulemper med katalogsystemer. Om vi velger å bruke katalogsystemer for e-handel bør vi være bevisst på disse forholdene. Vi bør også være bevisst på hvilke strategier innen e-handel som katalogsystemer understøtter.

Ved innføring og bruk av løsninger for e-handel kan vi måle grad av suksess etter flere definerte kriterier. Det er gjort flere erfaringer også på dette området. Kunnskapen fra erfaringene bør vi ta med oss når vi skal definere egne suksesskriterier, og evt grad av oppnåelse av slike kriterier.

5 Strategi, metode og modellering

Kapitlet presenterer bakgrunn for og valg av strategi for systemutvikling, metode for innsamling og bearbeiding av data og verktøy for modellering.

5.1 Innledning

For å kunne anbefale en infrastruktur for e-handel var det nødvendig å vurdere en løsning for e-handel gjennom et forsøk. Jeg så det som gunstig å knytte forsøket til en liten eller mellomstor bedrift med en etablert infrastruktur, som består av disse komponentene:

- nettverk og server(e) med egenskaper for fillagring, database(r) og periferutstyr.
- aksess til Internett og egen hjemmeside
- mulighet for salg av produkter eller tjenester over Internett
- mulighet for bruk av intranett og katalogsystemer

Med dette utgangspunktet søkte jeg samarbeid med en bedrift. Jeg etablerte kontakt med et selskap og det vurderte forsøket som interessant.

Forsøket gir grunnlag for å diskutere alternative løsninger og til slutt anbefale infrastruktur for e-handel.

For bedriften vil forsøket være grunnlag for videre diskusjon og beslutninger i bedriften.

Bedriften er presentert i kapittel 6.1 Om samarbeidspartner og vedlegg B.

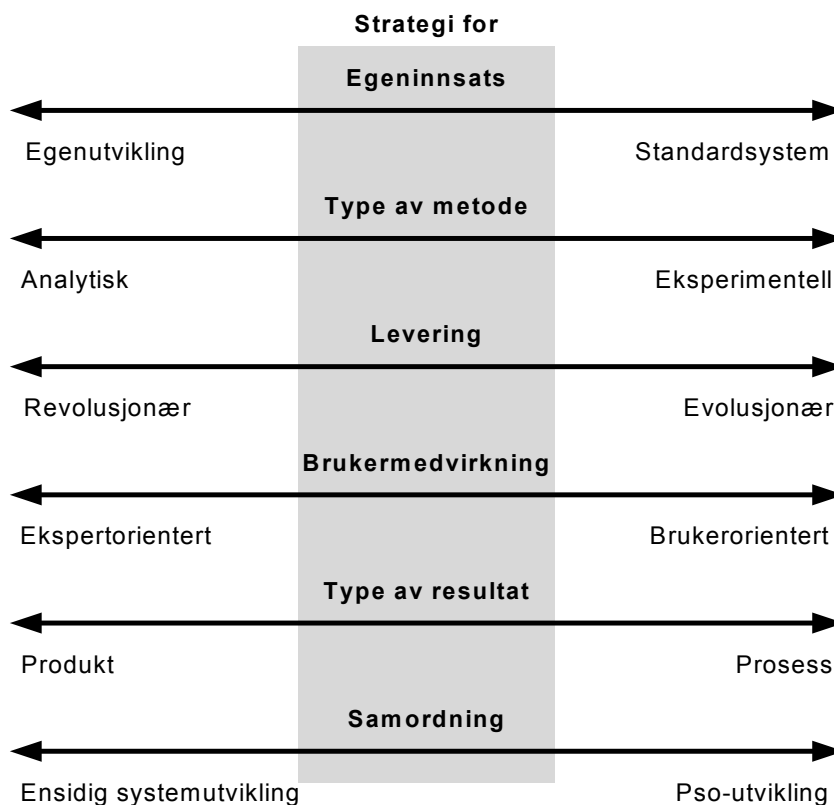
5.2 Strategi for systemutvikling

Prosjektet så det som nødvendig å definere en eller flere strategier for utvikling [AES98, PR97].

Valgt(e) strategi(er) for utvikling skal være grunnlag for videre analyse, drøfting og anbefaling av infrastruktur for e-handel.

Andersen [AES98] presenterer flere forskjellige strategier for utvikling (se figur 5-1). Han fremhever at det eksisterer mange etablerte strategier. Dette fordi det er forskjellig oppfatning av *systemutvikling*. Andersen fremhever at det eksisterer forskjellige meninger om:

- *egeninnsats* fra egen virksomhet.
- *type av metode* for å komme frem til en kravspesifikasjon.
- *type levering* av resultater (i mange små eller færre store porsjoner).
- *grad av brukermedvirkning* i utviklingen.
- *type av resultat* fra utviklingen (prosess vs produkt).
- *samordning* av annet utviklingsarbeid.



Figur 5-1 – Utviklingsstrategier [AES98].

Valg av *prosjekts* strategier for systemutvikling ble overveid i relasjon med:

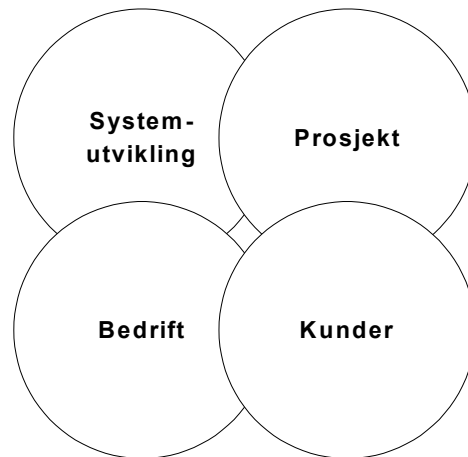
- *mål* og *problemstilling* for studiet.
- *tid* til disposisjon og *ressurser* tilgjengelig.

I tillegg ble studiets mål og problemstilling vurdert i relasjon til flere forhold i *bedriften*:

- *mål* og *strategi* for salg av tjenester.
- nåværende *infrastruktur* i bedriften.
- etablerte prosesser for *salg* og *markedsføring*.
- *relasjon* til samarbeidspartnere.
- *kultur* og *evne* til forandring.

Vi betraktet også, men i noe mindre skala, forhold omkring bedriftens *kunder*:

- relasjon til *eksisterende* kunder.
- kobling til *potensielle* kunder.
- kunders *holdning* og *kultur* til e-handel.



Figur 5-2 – Grunnlag for valg av strategier for utvikling.

Figur 5-2 gir en oversikt over hvilke kriterier som var grunnlag for valg av utviklingsstrategier. Gruppene er ikke isolert, men sett i sammenheng med hverandre.

5.2.1 Strategi for utvikling i prosjektet

Prosjektet tok utgangspunkt i Andersens [AES98] modell for utviklingsstrategi(er). Ved å koble modellen til variablene i *prosjekt*, *bedrift* og *kunder* (se figur 5-2), kunne vi definere et sett med strategier for gjennomføringen av systemutviklingen.

I denne prosessen tok vi hensyn til følgende momenter for *prosjektet*:

- hovedmål for prosjektet (mål for hovedfagsoppgaven).
- tid til disposisjon var ca 10 måneder eller ett skoleår.
- tilgjengelige ressurser hos NTNU/samarbeidspartner. Prosjektet hadde tilgang til følgende ressurser: kontorlokaler, kompetanse hos ansatte, maskiner og programvare for utvikling av prototyp.

Prosjektet vurderte følgende forhold i *bedriften* (se kapittel 6.2 Nå-situasjon i IKON):

- det eksisterte ingen dokumenterte mål og strategier for salg av tjenester hos samarbeidspartner. Salgsprosessen er ikke formelt dokumentert.
- bedriften har ingen systemer for e-handel.
- salg og markedsføring ble tidligere gjennomført av ansatte og enkel hjemmeside.
- bedriften ønsket å selge eksterne tjenester, det vil si konsulenter og rådgivere fra evt samarbeidspartnere, gjennom samme system.
- bedriften var "forholdsvis ung" og var positiv til endringer som forenklet administrative funksjoner. Bedriften hadde evne, kompetanse og ressurser til å gjennomføre nødvendige endringer av prosesser og rutiner.

Vi studerte og diskuterte også hvilke relasjoner bedriften hadde til sine *kunder*:

- bedriften har mange kunder som kjøper konsulent- og rådgivertjenester. Kundene er både i privat og offentlig sektor.
- bedriften ønsker å nå et større og mer variert marked for kunder.
- kundene er direkte eller indirekte koblet til fagområdene informasjonsteknologi (IT) og informasjonssystemer (IS). Kundene er med andre ord kjent med bruk av IT, IS og e-handel.

Vi vurderte ulike kombinasjoner av nevnte strategier for utviklingen av løsning (prototyp). Vurderingene er presentert i avsnittene nedenfor, og endelig vektlegging av strategivalg er vist figur 5-3.

Egeninnsats

Prosjektet ønsket å gjennomføre utviklingen selv. Vi søkte å bruke standard verktøy for utvikling, og eksisterende hylleware for operativsystem, katalogsystem, nettleser etc.

Type av metode

For at prosjekt, bedrift og evt kunder skulle kunne vurdere en fremtidig løsning var det nødvendig å utarbeide en kravspesifikasjon. For å gjøre det mulig valgte vi en mer eksperimentell enn analytisk metode. Ved eksperimentell utvikling brukte vi prototyp som grunnlag for å forbedre eksisterende kravspesifikasjon og for kunne vurdere den fremtidige løsningen.

Levering

Vi ønsket å levere prototypen gradvis til bedriften, gjennom flere mindre og hyppige leveranser. Denne form for levering blir kalt evolusjonær levering.

Brukermedvirkning

Utviklingen skulle skje med nær kobling til prosjekt og bedrift. Brukerne skulle bli tatt med på råd, men noe senere i utviklingsprosessen (etter levering av prototyp). Denne form for medvirkning blir definert som mer ekspertorientert enn brukerorientert.

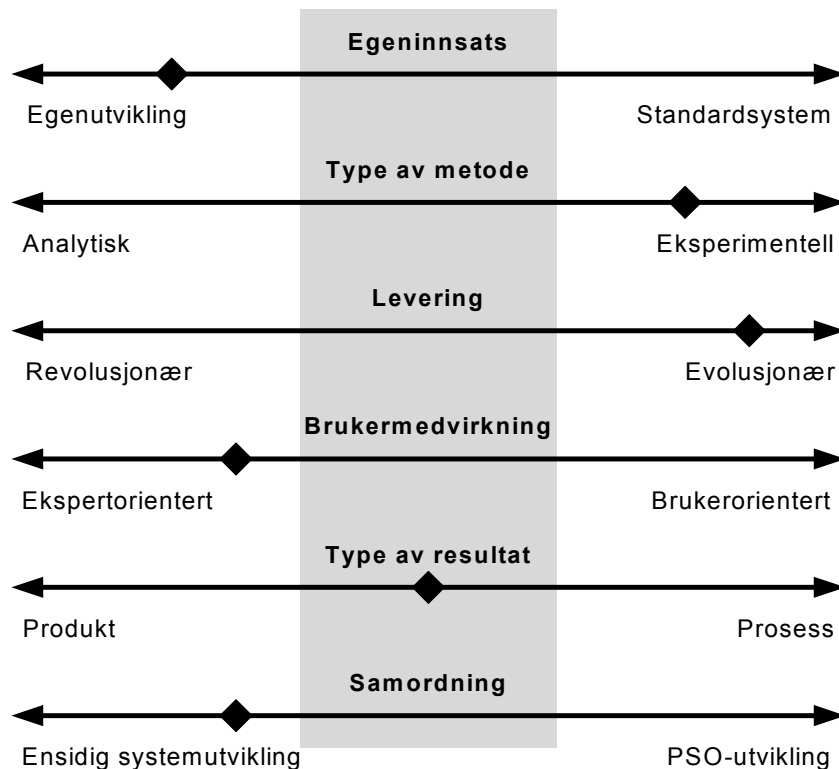
Type av resultat

Bedriften ønsket først og fremst en løsning for salg av tjenester på Internett. Den ønsket også å få tilført mer kompetanse innen utvikling av løsninger for e-handel. Denne kompetansen ønsket de senere å selge som tjenester til nye eller eksisterende kunder. Prosjektet så for seg et resultat som også skulle tilføre deltakerne og bedriften mer kompetanse på både gjennomføring av slike prosjekter og den teknologiske løsningen.

Samordning

For å understøtte problemstillingen var prosjektet avhengig av å fokusere mer på ensidig systemutvikling enn PSO (personal-, system- og

organisasjonsutvikling). Vi så for oss at etter en evt godkjenning, av videre utvikling fra prototyp, ville det bli mer fokus på PSO i bedriften.



Figur 5-3 – Valgte strategier for prosjekt.

Vi vurderte strategier innen e-handel og systemutvikling som sentrale elementer i utviklingsprosessen.

5.3 Metode for innsamling og bearbeiding av data

Ved gjennomføring av prosjekter og forskningsoppgaver er det i en eller flere faser nødvendig å samle inn og bearbeide data. I denne prosessen kan vi støtte oss til flere etablerte metoder.

Vi kan definere begrepet metode på flere måter. Halvorsen [HAL95] gjør det slik:

Metode er en systematisk måte å undersøke virkeligheten på [HAL95].

I et forskningsopplegg bør vi ta hensyn til flere variabler. Halvorsen [HAL95] mener at noen av de viktigste variablene i et forskningsopplegg er:

- type undersøkelsesenheter
- antall undersøkelsesenheter og egenskaper hos undersøkelsesenheten(e)
- type egenskaper (variabler)
- tidsfaktoren
- forskerens kontroll over systemet han undersøker
- direkte eller indirekte opplegg
- kildegrunnlaget for data
- metode for datainnsamling
- arten av data, bearbeiding, tolking, forklaringstyper og forskningsinteresse.

Vi bør velge metode(r) for datainnsamling når problemstillingen er definert. På bakgrunn av problemstillingen får vi oversikt over hvilke metoder som gir data av størst relevans. Samtidig bør vi definere formålet med undersøkelsen(e). Søker vi å få en helhetlig forståelse av spesifikke forhold, eller ønsker vi å få en representativ oversikt over generelle forhold? Før vi setter i gang med evt datainnsamling bør vi også få oversikt over egenskaper til studieobjektene, og forskerens forhold til kildene for data [HAL95].

Det eksisterer flere metoder for innsamling og bearbeiding av data:

- kvalitativ metode (observasjon i laboratorier og felt, ustrukturerte intervjuer, gruppesamtaler).
- kvantitativ metode (strukturerte intervjuer og spørreskjema).
- indirekte metoder (observere handlinger, registrere fysiske spor etter mennesker i omgivelsene)
- metodetriangulering (kombinere kvalitativ og kvantitativ metode).

5.3.1 Metode i prosjektet

I prosjektet ble valg av metode, for innsamling og bearbeiding av data, sett i relasjon til følgende behov:

- kunne beskrive bedriften og dens relasjoner til eksisterende og potensielle kunder.
- forklare, forstå og evaluere forhold mellom gammel og ny løsning for salg av tjenester.
- fremskaffe bedre grunnlag for beslutninger for aktuell bedrift, og organisasjoner som vurderer å innføre lignende løsninger for e-handel.

Prosjektet vurderte også andre forhold i forbindelse med gjennomføring av undersøkelsene:

- hvilken type data som er hensiktsmessig for prosjektet.
- validitet⁶ i datainnsamling og undersøkelser.
- egen forutsetning og tilgjengelige ressurser.
- egenskaper til objekter og/eller prosesser som blir undersøkt.
- bearbeiding og analyse av innsamlet data.

Prosjektet valgte *kvalitativ metode*⁷ for innsamling og bearbeiding av data. Vi nyttet *interjuer* (strukturert og ustrukturert) og *observasjon* som fremgangsmåte for å fremskaffe nødvendige data til prosjektet.

Prosjektet gjennomførte en spørreundersøkelse (intervju). Hensikten med undersøkelsen var å få oversikt og beskrive nå-situasjon i bedriften. Innsamlet data var grunnlag for videre diskusjon og gjennomføring av prosjektet.

Under systemutviklingen brukte vi metoden *observasjon* [NKS98]. Denne metoden er ofte brukt i sammenheng med observasjon av mennesker. Slik jeg oppfatter metoden kan vi også nytte den i prosjektet. Vi kan observere og samle inn data om egenskapene til prototypen gjennom utvikling av løsning.

Prosjektet var bevisst på at kvalitativ metode har flere fordeler og svakheter. Fordelen med kvalitativ metode og *spørreundersøkelse* er at vi kan få presise og utdypende svar. Vi er til stede under hele intervjuet, og kan registrere ulike reaksjoner hos respondentene. Vi kan også komme med nye eller oppklarende spørsmål under selve intervjuet. Den største svakheten med metoden er et vi bruker for få antall respondenter. Antall respondenter kan påvirke undersøkelsenes kvalitet og troverdighet [NKK98]. Fordelen med *observasjon* er at forskeren selv er til stedet og kan observere situasjonen. Ulempen er svakheten med *kontrolleffekten* [NKK98].

5.4 Verktøy for modellering

Vi har tilgang til mange typer verktøy for systemutvikling, og vi kan benytte flere av dem i dette arbeidet. Verktøyene gir oss blant annet bedre mulighet til å kunne beskrive, forstå, kommunisere og dokumentere eksisterende informasjonssystem så vel som nytt informasjonssystem .

Ved å *kombinere* flere verktøy kan vi bedre forstå den eller de systemer vi ønsker å endre eller skape. Dette gir oss mulighet til å gjøre utviklingen mer effektiv, og gi oss bedre grunnlag for å ta beslutninger gjennom hele utviklingsprosessen.

⁶ Validitet er kombinasjon av pålitelighet og definisjonsmessig validitet i undersøkelsen [HAL95].

⁷ Kvalitative data ikke er tallfestet, og beskriver de kvalitative egenskapene til enheten(e) som blir undersøkt [HAL95].

5.4.1 Verktøy for modellering i prosjektet

Prosjektet så det som en fordel å bruke flere verktøy i arbeidet med å beskrive og forstå eksisterende infrastruktur. Samtidig kan vi bruke slike verktøy som instrumenter for å beskrive og kommunisere forslag til ny løsning.

Vi ønsket også å bruke verktøyene gjennom i fasene for analyse, definering av krav og utforming av prototyp. På bakgrunn av dette definerte jeg følgende krav til slike verktøy:

- beskrive og dokumentere virkeligheten. Dette for å gjøre den enkel å kommunisere og forstå.
- støtte internasjonale standarder.
- være akseptert og i bruk av programvareindustrien.
- understøtte bruk av flere typer modeller (fra generisk til spesifikke modeller).

Prosjektet ønsket å bruke færrest mulige modelleringsverktøy. Å bruke flere verktøy er mer ressurskrevende med hensyn på å kjøpe, lære og bruke. Med bakgrunn i krav og behov i prosjektet valgte vi følgende hjelpemiddel:

- Unified Modeling Language (UML) som språk ved modellering av systemutviklingen (se vedlegg A – Om UML).
- standard verktøy for modellering av relasjonsdatabaser.

Unified Modeling Language

Booch mfl [BRJ99] fremhever at ved modellering av *enkle, klient/server* eller *distribuerte* systemer kan og bør vi bruke ulike kombinasjoner av diagramtyper i UML. Dette gir oss grunnlag for å belyse utviklingen i flere perspektiv.

Prosjektet karakteriserte ny løsning som et *distribuert* system. Ved utvikling av slike systemer anbefaler Booch mfl [BRJ99] å vise utviklingen i flere perspektiv (se tabell 5-1).

Tabell 5-1 – Type visninger (View) og diagrammer for distribuerte systemer i UML

View	Diagram
Use Case View	Use Case diagrams
Design View	Activity diagrams
	Class diagrams
	Interaction diagrams
Process View	Statechart diagrams
	Class diagrams
Implementation View	Interactions diagrams
	Component diagrams
Deployment View	Deployment diagrams

Tabell 5-1 gir en oversikt over ulike "view" i UML og hvilke tilhørende typer diagrammer vi kan bruke sammen med angitte "view". Tabellen viser flere uthevede diagrammer. Disse diagrammene ble brukt av prosjektet.

Modellering av relasjonsdatabaser

For å modellere datastrukturer for relasjonsdatabaser kan vi bruke konseptuelle modeller. McFadden & Hoffer [MH91] bruker E-R-diagrammer for å skape slike modeller. Dette er modeller som viser entiteter (E) med tilhørende relasjoner (R). Denne form for modellering er mye brukt ifm modellering av relasjonsdatabaser.

Prosjektet ønsket å bruke et verktøy for modellering av datastrukturer. Det vi tok i bruk var Modelator [MET98]. Vi brukte verktøyet for å modellere datastrukturer som vi senere skulle implementere i prototypen.

5.5 Oppsummering

Før å gjøre systemutvikling mer effektiv og kvalitativt bedre kan vi støtte oss til flere strategier for systemutvikling, metoder for innsamling og bearbeiding av data og verktøy for modellering

Prosjektet valgte å bruke flere strategier for systemutvikling. Vi definerte *egeninnsats* fra egen virksomhet, *type av metode* for å komme frem til en kravspesifikasjon, *type levering* av resultater, *grad av brukermedvirkning* i utviklingen, *type av resultat* fra utviklingen og *samordning* av annet utviklingsarbeid. Til innsamling og bearbeiding av data valgte vi kvalitativ metode (intervju) og observasjon. For modellering av nåværende, og forslag til ny løsning, valgte vi å modellere ved hjelp av UML og modellere ved hjelp av verktøy for relasjonsdatabaser.

6 Systemutvikling

Kapitlet omtaler samarbeidspartner, nå-situasjon hos samarbeidspartner, utvikling av prototyp og selve prototypen for e-handel.

6.1 Om samarbeidspartner

Prosjektet ble gjennomført i samarbeid med IT-Konsulentene i Oppland AS (IKON AS) [IK01].



Figur 6-1 – Logo for IKON AS [IK01].

Selskapet har hovedkontor på Lillehammer, og per 2001-09-01 er det 22 deltids-/heltidsansatte. Selskapet ble stiftet i mars 1997. IKON har kunder over hele Østlandet og spesielt i Oslo (se vedlegg B).

6.2 Nå-situasjon i IKON

For å få oversikt over nå-situasjonen i selskapet gjennomførte vi en spørreundersøkelse. Undersøkelsen ble gjennomført med støtte i en intervjuguide (se vedlegg C). Hensikten med guiden var å gjøre intervjuprosessen enklere og mer oversiktlig, og gi oss bedre grunnlag for å bearbeide innsamlede data.

Intervjuguiden består av to deler. Første del introduserer undersøkelsen og sier noe om prosjekt, hvordan undersøkelsen er forberedt, hvordan vi skal gjennomføre undersøkelsen, og hvordan vi skal bearbeide innsamlede data. Andre del består av kolonner med spørsmål og plass for å notere svar.

Vi intervjuet to personer i bedriften: en tilknyttet *salgprosessen* og en annen tilknyttet *infrastrukturen* i selskapet. Intervjuene ble gjennomført i november/desember 2000.

Hensikten med spørreundersøkelsen var å få *helhetsoversikt* over salgsprosessen og hvordan infrastrukturen i bedriften understøttet denne prosessen.

Resultatet fra undersøkelsen er presentert i de neste avsnittene. Avsnittene fremstiller et sammendrag av de svar jeg fikk fra intervjuene.

6.2.1 Om bedrift

IKON AS fokuserer på å ha et godt arbeidsmiljø, gode kunderelasjoner, minimale kostnader og at alle ansatte produserer. At alle produserer betyr at alle arbeider for kunder. Dette gjør at ansatte ikke treffer hverandre så ofte. Derfor er det lagt stor vekt på å et godt arbeidsmiljø gjennom hyppige møter (måned- og kvartalsmøter) og flere sosiale samlinger, med familie, hvert år.

Selskapet bruker et bedriftsinternt nett (intranett). Internett er kommunikasjonsbærer for tjenesten. Intranettet er et viktig element for å holde ansatte informert og oppdatert om hva som skjer i selskapet. Tjenesten består blant annet av nyheter, oversikt over ansatte (CV), oppgavelister for ansatte, møteoversikt, rutinebeskrivelser og retningslinjer for ansatte (personalveiviseren). Data i intranettet er gjenbrukt i bedriftens Internett-løsning [IK01].

6.2.2 Om salg og markedsføring

Alle ansatte er selgere. Dvs alle har ansvar for "å selge seg selv" til eksisterende eller nye kunder. Daglig leder og avdelingsleder har et spesielt ansvar i denne prosessen. At alle er selgere, og samtidig arbeider, er krevende. Fordelen med denne form for salg er at bedriften ikke bruker mye ressurser på markedsføring og salg.

Bedriften selger ca 70% nye eller oppdaterte kontrakter per år. Dvs de fleste ansatte blir berørt direkte eller indirekte av salgsprosessen hvert år. Selskapet nytter ingen spesielle verktøy for markedsføring og salg. Noen verktøy er prøvd, men de fleste av disse støtter kun bedrifter som selger produkter. IKON selger kun tjenester og dermed passer ikke verktøyene for selskapet.

Bedriften har brukt noe ressurser på å analysere og forstå markedet de opererer i. Det er utarbeidet en markedsplan som selskapet bruker aktivt. Selskapet har positive erfaringer med å bruke en markedsplan.

Salgsprosessen i IKON er ikke formelt dokumentert. Den har blitt til mens bedriften har utviklet seg. Selskapet erkjenner at det kan bli bedre på salg av tjenester. Bedriften ser potensialet for forbedringer på flere områder:

- forbedre oversikt over kontrakter mellom kunde og IKON. Herunder forhold som:
 - tid (start/slutt)
 - timepris
 - evt indeksregulering av kontrakter
 - CV (fokus på hva vi er gode på, ikke på hva vi har gjort).
- bedre oppfølging av kunder, og forbedre kontrakter.
- bedre kobling mellom IKONs Internett- og Intranettsløsning.

IKON fremhever at de søker å følge de lover og forskrifter som er pålagt for selskapet.

IKON definerer følgende som *suksesskriterier* ved salg av tjenester:

- strukturert tilbuds- og kontraktsprosess.
- god oppfølging av kontrakter.
- raskt tilbud til kunden - med riktige data om kandidat.
- mer attraktiv hjemmeside.
- komme "nærmere" nye og eksisterende kunder.
- videreutvikle personlige relasjoner til kunder.

På bakgrunn av dataanalyse fra intervjuene modellerte jeg "dagens salgsprosess" vha Activity og Interaction Diagram i UML. Diagrammene er skissert i vedlegg D.

6.2.3 Om infrastruktur

Følgende komponenter i infrastrukturen er brukt ved *salg* av tjenester:

- system for e-post.
- system for møteplanlegging.
- database med CV-er.
- database/filsystem med oversikt over prosjekt/kontrakter ("prosjektdatabasen").
- telefon/telefaks.

Nevnte systemer er installert på en felles plattform (unntatt telefon/ telefaks).

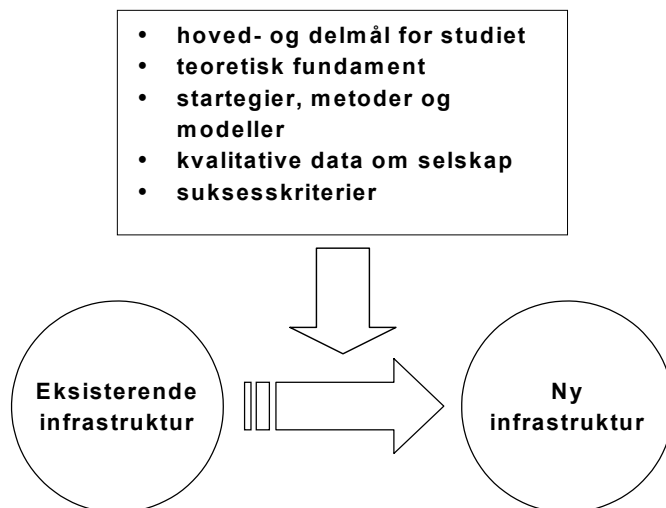
Infrastrukturen er bygd opp slik: serverkomponenter er installert på server ved hovedkontor i Lillehammer og hos ASP-leverandør i Oslo. Komponenter for klienter er installert på stasjonære PC-er i Lillehammer eller på bærbare PC-er som de ansatte disponerer. IKON bruker databaseapplikasjonen MySQL [MYS01]. Den er brukt til lagre CV-er og data om prosjekter (prosjektdatabase). MS Exchange/Outlook [MS01] er brukt som applikasjon for e-post og møteplanlegging. På bakgrunn av noe varierende bruk og administrasjon av databasen har enkelte opprettet egne datakilder for lagring av samme type data. Eksempel på dette er data, om ansatte og deres kundeforhold, som er lagret i flate filer (Microsoft Excel).

Dagens infrastruktur har ikke katalogsystemer, men eksisterende infrastruktur (hardware, operativsystemer, protokoller for kommunikasjon etc) er klargjort for å bruke slike systemer.

Overordnet infrastruktur, *før* gjennomføring av dette prosjektet, er modellert ved hjelp av Deployment Diagram i UML. Diagrammet er også skissert i vedlegg D.

6.3 Krav til ny løsning

Ny løsning for e-handel (prototyp) er utviklet på bakgrunn av hoved- og delmål for studiet, teoretisk fundament, valgt(e) strategier, utviklingsmetoder, modeller, kvalitative data om selskapet (nå-situasjon), og selskapets suksesskriterier for e-handel (se figur 6-2).



Figur 6-2 – Variabler som påvirket utviklingen av ny infrastruktur.

På bakgrunn av nevnte variabler (se figur 6-2) ble krav til ny løsning utformet. Kravene ivaretar både funksjonalitet for løsning og egenskaper for infrastrukturen.

6.3.1 Krav til funksjonalitet

Ny infrastruktur for IKON skal ivareta handel av tjenester over web (Internett). Potensielle kunder er virksomheter (privat og offentlig) som ønsker å kjøpe konsulent- og rådgivningstjenester av IKON.

Løsningen skal ivareta følgende *funksjonelle krav*:

- potensielle eller eksisterende kunder skal kunne aksessere hjemmeside for IKON AS over Internett. Kunder skal enkelt kunne navigere seg frem til tjeneste for "salg av rådgivere/konsulenter".
- kunder skal kunne søke på ønsket profil for konsulenter/rådgivere (for eksempel kompetanse, pris og tidsrom).
- avdelingsleder skal få melding om kunde(r) ikke finner tilgjengelige kandidat(er).
- om kandidater er tilgjengelig, og kunde er interessert, skal kunde kunne sette opp møte med avdelingsleder i IKON.

Forslag til ny løsning er modellert ved hjelp av Activity og Interaction Diagram (se vedlegg E).

6.3.2 Krav til infrastruktur

Ny infrastruktur skal bygge på flest mulige komponenter fra eksisterende infrastruktur. Grunnlagsdata (metadata) for tjenesten skal vedlikeholdes gjennom Intranettet i bedriften. Hver ansatt har ansvaret for å administrere data om seg selv (for eksempel personlige data og CV). Data om selve bedriften, som er tilgjengelig på IKONs hjemmeside, skal administreres som i før.

Ved etablering av katalogsystem skal katalogen fungere som metakatalog. Prototypen skal bestå av katalogsystem, kobling mellom web/katalogsystem og katalogsystem/datakilder.

Prototypen skal ikke understøtte hele salgsprosessen, men den skal inneholde tilstrekkelige funksjoner slik at vi kan vurdere om katalogsystemer kan nyttes i infrastruktur for e-handel.

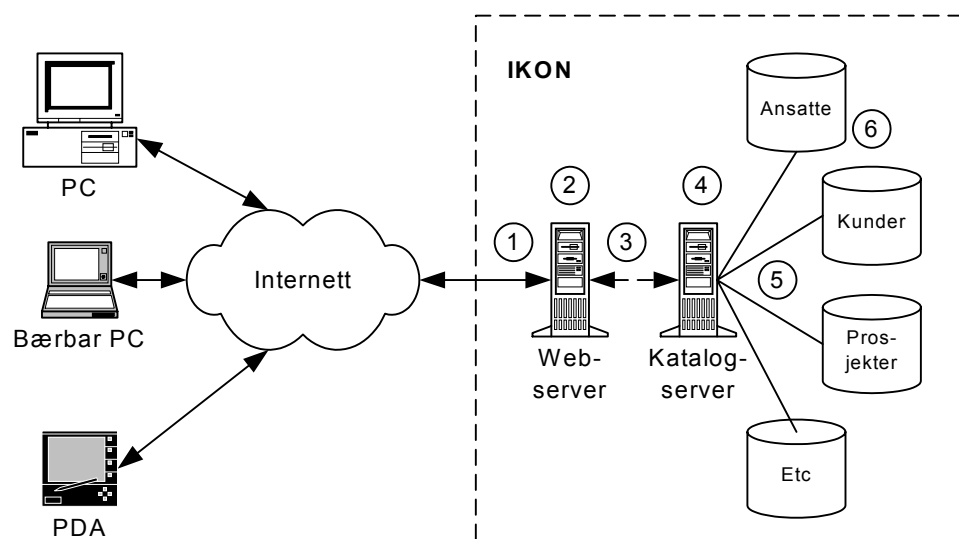
Prototypen skal være en separat komponent. Prototypen skal ikke være integrert med produksjonssystemet, men bygd opp som en enhet som simulerer koblingen mellom eksisterende og ny infrastruktur.

Forslag til ny overordnet infrastruktur er vist i Deployment Diagram (se vedlegg E).

6.4 Utvikling av ny løsning

Eksisterende infrastruktur og krav til ny løsning var utgangspunkt for design av ny løsning. Eksisterende infrastruktur bestod av (se figur 6-3):

- ① - Internett-tilkøpling.
- ② - Webserver.
- ⑥ - Datakilder.



Figur 6-3 – Ny infrastruktur

Figur 6-3 er utarbeidet med bakgrunn i studiets problemstilling (se figur 1-1) forslag til ny infrastruktur (Deployment Diagram).

Studiet, frem til dette stadiet, viste at det var behov for flere nye komponenter i ny infrastruktur. Nye komponenter med egenskaper som katalogsystem og system(er) for integrasjon mellom web, katalogsystem og datakilder.

På bakgrunn av dette valgte jeg nye komponenter etter følgende kriterier:

- nye komponenter skal fungere i eksisterende infrastruktur og være leverandøruavhengig.
- katalogsystem skal kunne utformes og brukes som metakatalog, og skal støtte etablerte standarder og spesifikasjoner.
- systemer for integrasjon skal kunne kommunisere med katalogsystem.
- enkel å lære, utvikle og tilpasse.

De fleste leverandører av katalogsystemer har, foruten katalogsystemer, også egne verktøy for integrasjon mellom web, katalogsystem og datakilder.

Det eksisterer også flere uavhengige leverandører av slike verktøy. Verktøyene er normalt utformet slik at de fungerer sammen med flere ulike katalogsystemer.

Jeg hadde altså mulighet for å velge flere ulike komponenter til ny infrastruktur. Noen alternativer er skissert i tabell 6-1.

Jeg valgte tre nye komponenter til eksisterende infrastruktur (se figur 6-3):

- ③ - Verktøy (applikasjon) for integrasjon mellom web/katalogsystem.
- ④ - Katalogsystem/katalogserver.
- ⑤ - Verktøy (applikasjon) for integrasjon mellom katalogsystem og eksisterende kataloger/datakilder.

De nye komponentene ble satt sammen, med eksisterende infrastruktur, for å fungere slik:

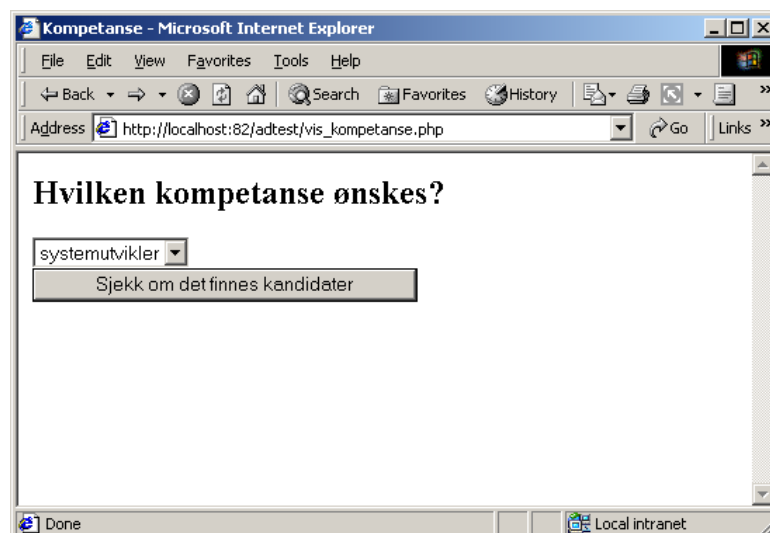
- kunder, ansatte eller partnere skal kunne aksessere hjemmesiden for IKON vha nettleser fra PC-er og PDA-er (se pkt ① i figur 6-3).
- kommunikasjon mellom klient og web-server (se pkt ② i figur 6-3) skal støtte standard protokoller for Internett. Web-server skal bruke operativsystemet MS Windows Server og Internet Information Server.
- katalogdata skal gjøres tilgjengelig på web vha script/program i PHP [PHP01] (se pkt ③ i figur 6-3). Scriptene/programmene skal bruke LDAP-operasjoner mot katalogsystem. Katalogsystem skal være iPlanet Directory Server (se pkt ④ i figur 6-3).
- koblingen mellom katalogserver og underliggende datakilder skal være gjennomført med MaXware DSE [MAX00b] (se pkt ⑤ i figur 6-3). DSE skal synkronisere data begge veier (fra datakilder til katalog og fra katalog til datakilder).

Tabell 6-1 – Valg av nye komponenter til infrastruktur for e-handel.

Komponent	Alternativer komponenter	Valg
③ - Integrasjon web/katalogsystem	C C++ Java Perl <i>PHP</i> VBScript	Vi valgte PHP [PHP01] som verktøy for integrasjon web/katalogsystem. PHP er et verktøy (script-språk) som gir mulighet for å lage dynamiske web-sider. Verktøyet er gratis å bruke, og vi kan bruke det på flere plattformer. Vi brukte verktøyet til å lage nødvendige operasjoner (LDAP) mot katalogserver.
④ - Katalogserver	Active Directory eTrust Directory <i>iPlanet Directory Server</i> <i>Novel NDS eDir/DirXML</i> Oracle Internet Directory	Vi valgte iPlanet Directory Server som katalogsystem. IPlanet Directory Server (IDS) er en katalogsystem utviklet av alliansen SUN-Netscape [IP01]. Vi brukte systemet som metakatalog sammen med PHP og MaXware DSE.
⑤ - Integrasjon katalogsystem /datakilder	<i>MaXware DSE</i> Novel NDS eDir/DirXML DC-MetaLink Active Directory Connectors (ADC) Import/export (LDIF ⁸) til/fra katalog	Vi valgte MaXware DSE [MAX01] som verktøy for synkronisering mellom datakilder. MaXware DSE er et verktøy for konvertering og sammenføring av data. Vi brukte verktøyet for synkronisering av data mellom datakilder (tabeller) og katalogsystem.

6.5 Dokumentasjon av ny løsning

Ny løsning (prototyp) er utformet med bakgrunn i krav til ny løsning og valgte komponenter.

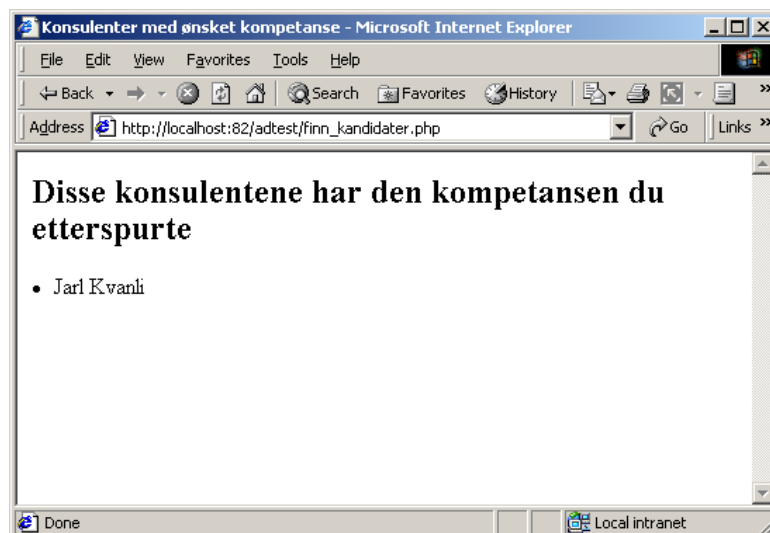


Figur 6-4 – Skjerm bilde for søk etter ønsket kompetanse.

⁸ LDAP Data Interchange Format (LDIF) – et standard tekstformat som representerer data ved import til eller eksport fra katalogsystem [IP01].

Prototypen er konstruert for å virke slik:

- potensielle kunder åpner skjermbilde "Hvilken kompetanse ønskes?" (se figur 6-4).
- i skjermbildet definerer kunder ønsket profil (kompetanse) på konsulent/rådgiver. Når søket er bekreftet blir spørring kjørt mot katalogsystem.
- på bakgrunn av profilen i søket får potensielle kunder tilbakemelding om bedriften har angitt kompetanse (se figur 6-5).



Figur 6-5 – Skjermbilde for tilbakemelding på søk etter kompetanse.

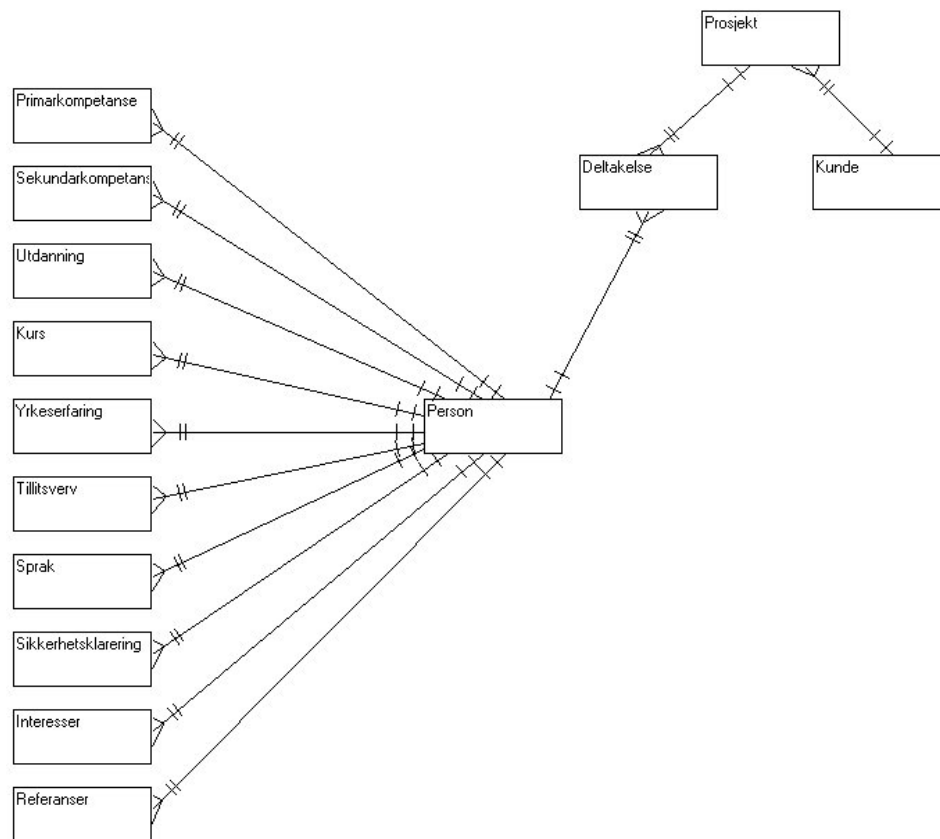
6.5.1 Klargjøring og konfigurasjon av server, datakilder og katalogsystem

Før vi startet arbeidet med å klargjøre server, datakilder og katalogsystem modellerte vi datastrukturen for prototyp. Modelleringen ble gjennomført som for modellering av relasjonsdatabaser (se figur 6-6).

Ferdig datamodell ble overført til katalogsystem. Dvs strukturen i katalogsystemet, som objekter, skjema etc, er definert som for relasjonsdatabaser. Dette er egentlig ikke nødvendig, men vi gjorde det for å holde bedre oversikt over integrasjon og utviklingsprosessen.

Datamodellen viser data om ansatte (CV) og hvilke prosjekter de er koblet til (kan være flere). I tillegg gir modellen oversikt over hvilke kunder prosjektene er koblet til.

Erfaringene fra modellen er gode, men det er fortsatt rom for forbedringer. Vi kan for eksempel utvide modellene slik at den nytter flere entiteter for å beskrive en person (CV). Dette for blant annet å unngå dobbellagring og redundans i datastrukturen.



Figur 6-6 – Modell over datastrukturer.

Modellen ga oss grunnlag for å klargjøre og konfigurere server, datakilder og katalogsystem.

Server, datakilder og katalogsystem er installert og klargjort på en server. Prototypen er i sin helhet utviklet på denne serveren.

Fremgangsmåte og resultat fra arbeidet er beskrevet i vedlegg F.

6.5.2 Integrasjon datakilder og katalogsystem

Etter klargjøring og konfigurasjon av server, datakilder og katalogsystem startet arbeidet med gjensidig integrasjon mellom datakilder og katalog. Vi valgte Data Synchronization Engine (DSE) fra MaXware [MAX01] for denne oppgaven. DSE ble installert på samme server som katalogsystem.

Gjennom ansatte i IKON, som hadde kunnskap om DSE, fikk vi tilgang til både produkt og kunnskap. Vi kunne valgt andre verktøy, men vi gikk for DSE fordi det er "enkelt å lære, utvikle og tilpasse".

DSE ble brukt til å synkronisere data mellom datakilder og katalogsystem. DSE ble konfigurert til å bruke DSA direkte.

Integrasjon mellom datakilder og katalogsystem er nærmere beskrevet i vedlegg G.

6.5.3 Integrasjon web og katalogsystem

Etter integrasjon mellom datakilder og katalogsystem, startet vi arbeidet med å integrere web og katalog. Vi valgte script-språket PHP til denne oppgaven. Valget ble begrunnet med at språket har innebygd støtte for LDAP, det er enkelt å lage web-sider i det (i forhold til andre språk) og det er velkjent. I tillegg hadde flere ansatte i IKON erfaringer med PHP. Gjennom de ansatte fikk vi tilgang til og kunnskap om språket.

PHP ble brukt til å lage spørringer (LDAP) mellom web (kunde) og katalogsystem.

Integrasjon web og katalogsystem er beskrevet i vedlegg H.

6.6 Oppsummering

Studiet ble gjennomført i samarbeid med selskapet IKON AS. Før prototypen, ble utarbeidet, kartla og dokumenterte vi nå-situasjonen i selskapet. Data fra nå-situasjonen ble innhentet og bearbeidet fra en spørreundersøkelse.

På bakgrunn av nå-situasjon ble krav til ny løsning definert. Kravene ble utformet på bakgrunn av studiets mål, teoretisk fundament, valgt(e) strategier, metoder, modeller, og selskapets suksesskriterier for e-handel.

På bakgrunn av krav til ny løsning ble tre komponenter integrert med eksisterende infrastruktur. Vi valgte følgende komponenter: katalogsystem (iPlanet Directory Server), verktøy for integrasjon datakilder/katalogsystem (MaXware DSE) og verktøy for integrasjon web/katalogsystem (PHP).

Prototypen ble utformet med utgangspunkt i datamodellen for ny løsning. Datamodellen ble overført til katalogsystem, og DSE ble konfigurert til å synkronisere data mellom datakilder og katalogsystem. Og PHP ble brukt til å lage LDAP-spørringer mellom web og katalogsystem.

7 Diskusjon og anbefaling

Kapitlet diskuterer alternative løsninger for e-handel, og presenterer endelig anbefaling for studiet. Kapitlet omtaler også mulige alternativer for både utviklingsprosess og løsning.

7.1 Alternative valg for utvikling og løsning

Studiet har gitt meg mange erfaringer, og gjennom disse ser jeg mulighet for flere alternative valg for både utvikling og løsning. De neste avsnittene vil omtale alternative valg for strategi, metoder og modeller, systemer for e-handel og teknologisk infrastruktur.

7.1.1 Strategi, metoder og modeller

Erfaringer fra prosjektet viser at det er en stor fordel å bruke definert(e) strategi(er) og metode(r) ved utvikling. IKON er i bransjen for IT/IS. For denne bedriften er slike strategier og metoder for systemutvikling allment kjent. Men for bedrifter i andre bransjer er dette trolig mindre kjent. Om vi skal gjennomføre tilsvarende prosjekter for virksomheter, som kan tradisjonell handel, er budskapet om prosessen frem til løsning viktig.

Vi kan bruke andre *utviklingsstrategier* som understøtter utvikling av tilsvarende løsninger. Andre prosjekter bør for eksempel vurdere annen vektlegging av *egeninnsats*, *brukermedvirkning* og *samordning*. Prosjekter som arbeider for bedrifter i andre bransjer, bør velge en annen tilnærming enn den vi har brukt. Det er trolig en fordel med større vektlegging på standardsystemer enn egenutvikling, mer brukermedvirkning enn ekspertorientert utvikling og mer PSO-utvikling enn ensidig systemutvikling. Dette vil gi mulighet for større involvering fra dem det nye systemet vil angå.

Det eksisterer flere *metoder* for innsamling og bearbeiding av data. Vi har benyttet metode for *kvalitativ* datainnsamling og bearbeiding. I prosjekter, som involverer for eksempel mange ansatte, kan det være en fordel å bruke andre metoder. Vi kan for eksempel bruke *kvantitativ* metode eller en kombinasjon av kvalitativ og kvantitativ metode (*metodetriangulering*) ved datainnsamling og databearbeiding. Dette gir mulighet for å få flere og bedre tilbakemeldinger fra fremtidige brukere.

Vi kan bruke flere *strategier* og *metoder* ved systemutvikling, og vi har erfart at det er en fordel å bruke definerte strategier og metoder for å underbygge målsettingen for tilsvarende prosjekter.

Valg av verktøy for *modellering*, som UML og verktøy for modellering av relasjonsdatabaser, er underordnet. Prosessen i forbindelse med definering av mål, delmål, strategier og metoder gir klare indikasjoner på hvilke verktøy vi bør velge. Erfaringene fra prosjektet viser at bruk av slike verktøy gir mange fordeler. Verktøyene gjør det mulig å gjennomføre utviklingen raskere og mer effektivt.

7.1.2 Systemer for e-handel

Avhandlingen har tidligere omtalt *suksesskriterier* fra Bryhn og Galtung [BG99], og hva IKONs selv definerer som suksesskriterier for e-handel. Teorien [BG99] trekker frem pris/kostnad, kjernekompetanse, legitimitet og aksept, bygging av merkenavn og kundekontrakt som suksesskriterier for e-handel. IKON definerer suksesskriterier for e-handel slik: strukturert tilbuds- og kontraktsprosess, god oppfølging av kontrakter, raskt tilbud til kunden - med riktige data om kandidat, mer attraktiv hjemmeside, komme "nærmere" nye og eksisterende kunder og videreutvikle personlige relasjoner til kunder.

Det er noe forskjell mellom det Bryhn og Galtung [BG99] fremhever og det IKON selve definerer som suksess. Vi bør se forskjellene og likhetene i sammenheng med at teorien har sterkere fokus på e-handel, mens IKONs har fokus på handel, der e-handel er et supplement til tradisjonell handel. Vi bør være bevisst på dette perspektivet ved utviklingen av tilsvarende løsninger for andre miljø.

IKON kan leve uten en *løsning* for e-handel. Men å leve uten en slik løsning bør selskapet vurdere nøye. Skal vi oppfylle selskapets kriterier for suksess er det nødvendig med en eller annen løsning for handel over web. Når det er sagt kan vi velge mellom mange alternative løsninger. *Egen løsning* eller *ferdige systemer* fra leverandør? Løsninger basert på *åpne standarder* eller *properitære* pakker? Skal systemet *støtte hele* eller *deler* av *salg-* og *markedsføringsprosessen* i virksomheten? Mange av disse spørsmålene er interessante. Ikke fordi de er komplekse, men for å kunne svare på de må du ha kommet langt i hvordan en evt ny eller endret forretningsplan skal være for virksomheten.

Ved utvikling av løsninger for e-handel bør vi ikke overdrive fokus på ny infrastruktur. Vi bør se slike løsninger i sammenheng med hva bedriften ønsker med e-handel.

Vi bør ha klare formeninger om hvilke *strategier* vi ønsker å bruke for å understøtte en forretningsplan. Ved å implementere gode strategier for salg og markedsføring kan vi oppnå god avkastning på våre investeringer. Erfaringer fra prosjektet er at vi bør bruke flere strategier og vi bør være bevisst på hvilke vi velger.

I utviklingsprosessen kommer vi før eller siden inn på spørsmål om hva en løsning skal bestå av. Skal løsningen støtte mange funksjoner; butikkssystemer, innholdshåndtering, personalisering, kundeservice, annonsesystemer etc (se kap 2.5. Systemer for e-handel), eller skal systemet kun støtte deler av tilgjengelige funksjoner? Skal løsningen ta i bruk alle funksjoner fra første dag? Disse forholdene må ses i sammenheng med hva bedriften selv ønsker.

Prosjektet har vist at vi kan sammenstille og gjøre data raskt tilgjengelig på web ved hjelp av katalogsystemer. Men for å ivareta alle kriteriene for suksess, som selskapet selv definerer, består jobben i noe mer enn å få data fra datakilder til

web. For å tilfredstille alle kriterier for suksess må endelig løsning støtte flere funksjoner enn det prototypen viser.

Vi kan velge flere teknologiske løsninger for e-handel. Vi valgte å ta utgangspunkt i eksisterende infrastruktur og tilpasse den for e-handel. Vårt forslag til ny infrastruktur passer ikke for alle virksomheter.

Ved utvikling av løsninger for e-handel bør vi være bevisst på hensikt og mål med løsning, kriterier for suksess og hvilke utviklingsstrategier vi ønsker å benytte.

7.1.3 Teknologisk infrastruktur

Prosjektet tok utgangspunkt i eksisterende infrastruktur i en bedrift. Til infrastrukturen integrerte vi tre nye verktøy: katalogsystem, verktøy for integrasjon datakilder/katalogsystem og web/katalogsystem.

Prosjektet vurderte flere katalogsystemer, og brukte forholdsvis mye tid på å velge katalogsystem. Først installerte vi Active Directory fra Microsoft, deretter NDS Dir XML fra Novell og til slutt iPlanet Directory Server fra SUN-Netscape. Alle systemene tilfredstilte våre behov. Det som skilte dem var kravet om "enkel å lære, utvikle og tilpasse". Løsningen fra iPlanet var enklere å ta i bruk og lære. Dette forholdet gjorde utviklingen mer effektiv og raskere.

Erfaringer fra studiet viser at vi kan bruke hvilke som helst katalogsystem. Men katalogsystemene må kunne utformes som metakatalog og støtte standarder og spesifikasjoner som X.500 og LDAP. Med dette som utgangspunkt bør valg av katalogsystemer ses i sammenheng med hvilke andre oppgaver systemet skal støtte i virksomheten.

Det eksisterer flere alternativer til katalogsystemer for e-handel. Jeg vil spesielt trekke frem *database-teknologi*. I relasjon til studiets problemstilling kan vi erstatte katalogsystemet med databaser, og vi kan bruke samme verktøy, som vi har brukt, for integrasjon mellom web og datakilder. Dvs nytte DSE for overføre data mellom datakilder og database og bruke PHP for å lage et grensesnitt mellom web og database. Vi kan også bruke *andre verktøy for integrasjon* mellom web, database og datakilder.

Database-systemer er et reelt alternativ. Dette fordi det er mer tilgjengelig kunnskap om databasesystemer i forhold til katalogsystemer. Det finnes i tillegg flere databasesystemer enn katalogsystemer, og slike systemer er trolig billigere å kjøpe, administrere og vedlikeholde. Nevnte forhold bør også ses i relasjon til teknologivalg for infrastrukturen.

Om valg av teknologi blir gjort med vekt på egenskaper i teknologien bør vi være oppmerksom flere kvaliteter. Vi bør ha en formening om kvaliteter som *distribuert administrasjon, sikkerhet, skalerbarhet, distribusjon, replikering/synkronisering og lagring av data*. Disse egenskapene skiller databasesystemer og katalogsystemer klarere fra hverandre, og er av stor betydning hvis datakilder er lokalisert på flere plasser (for eksempel i flere land). For *globale virksomheter* bør slike egenskaper stå meget sentralt ved valg av infrastruktur for e-handel.

Løsninger for e-handel, som er spredt over store geografiske områder, vil trolig se katalogsystemer som et bedre alternativ enn databasesystemer. Om datakilder, katalogsystem og web-løsning er samlokalisert i for eksempel *små- og mellomstore bedrifter* vil disse egenskapene trolig være underordnet. Vi skal heller ikke utelukke de. Det gjelder spesielt virksomheter som har en hardt belastet infrastruktur, og der det er behov for hyppige endringer eller utvidelser.

Det er tilgjengelig flere alternative verktøy for integrasjon mellom datakilder og web. Flere leverandører, av katalogsystemer, har også verktøy for slike oppgaver. Verktøyene er integrert i katalogsystem eller vi kan integrere de som en separat enhet til katalogsystem. Å bruke tradisjonelle programmeringsspråk, for å lage løsninger for integrasjon, vil erfaringsmessig medføre mer koding og forbruk av ressurser.

Vi valgte DSE for integrasjon mellom datakilder/katalogsystem og PHP for integrasjon mellom web/katalogsystem. Vi har positive erfaringer med å bruke verktøyene.

DSE gir oss mulighet for å integrere og vaske data fra ulike kilder. Verktøyet er meget fleksibelt, og det er forholdsvis lett å koble sammen mange typer datakilder. Verktøyet gjorde systemutviklingen raskere og det ga oss samtidig mulighet for bedre synkronisering mellom datakilder og katalogsystem. Å arbeide med og utvikle i PHP er relativt enkelt. Men vi skal være bevisst at det er forskjeller å konstruere LDAP-spørringer mot katalog og SQL-spørringer mot databaser i PHP (se vedlegg I).

Våre erfaringer tilsier at det er enklere å lage SQL-spørringer enn LDAP-spørringer i PHP.

Ved valg av teknologi (katalogsystem vs databasesystem) bør vi være oppmerksom på dette forholdet.

Ved integrasjon av data i katalogsystemer kan vi også bruke egenskapene for import og eksport i katalogsystemer. Vi kan importere fra og eksportere til katalogsystem ved hjelp av LDIF. Tidlig i studiet prøvde vi å bruke formatet ved integrasjon. Resultatet fra disse testene viste at vi kan bruke formatet ved integrasjon mellom datakilder og katalogsystem.

Ved valg av verktøy for integrasjon bør vi være oppmerksom på behovet for kunnskap. Det vil normalt være mer ressurskrevende å utvikle løsninger med utradisjonelle verktøy. Utviklingstiden kan også bli påvirket av hvilke veivalg vi gjøre på dette området.

7.2 Resultat og anbefaling

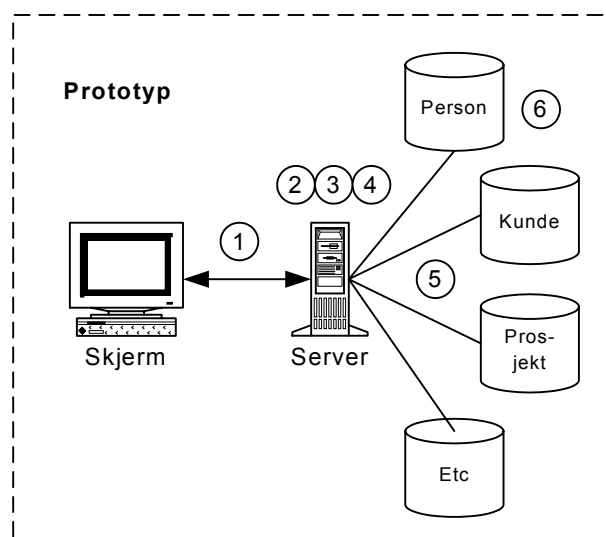
Studiet har gitt følgende resultat: en prototyp for e-handel og et dokument (avhandling) med anbefaling av infrastruktur for e-handel. Prosjektets resultat (prototyp) og anbefaling er omtalt i de neste avsnittene.

7.2.1 Resultat

Konstruksjon av prototyp ble gjort med utgangspunkt i krav til ny løsning (se figur 6-3), og prototyp ble satt sammen med følgende komponenter (se figur 7-1):

- ① - Internett-tilkopling (gjennom datanettet i bedriften).
- ② - Web-server (med Microsoft Windows 2000 Server og Microsoft Internet Information Server).
- ③ - Verktøy for integrasjon mellom web/katalogsystem (PHP).
- ④ - Katalogsystem/katalogserver (iPlanet Directory Server).
- ⑤ - Verktøy for integrasjon mellom katalogsystem/datakilder (MaXware DSE).
- ⑥ - Datakilder (filer i Microsoft Excel).

Alle komponenter (se pkt ① - ⑥) i prototypen ble installert på *en* server.



Figur 7-1 – Prosjektets resultat - prototyp.

Prototypen viser at vi kan synkronisere data mellom datakilder og katalogsystem på katalogserver. Løsningen gir potensielle kunder mulighet for å søke etter ønsket kompetanse i IKON. Potensielle kunder definerer søket i et enkelt web-grensesnitt.

I prototypen ble katalogsystem utformet som en metakatalog, og datastruktur i katalogen ble designet som en relasjonsdatabase. Vi brukte LDAP (versjon 2 og 3) som aksessprotokoll. Det ble ikke brukt noen spesielle mekanismer for sikkerhet (autentisering og kryptering). Prototypen bruker DSE til å synkronisere data mellom datakilder og katalog, og PHP-script for å utføre spørringer mellom web og katalog.

7.2.2 Anbefaling

På bakgrunn av prosjektets hovedmål anbefaler jeg følgende infrastruktur for e-handel i små og mellomstore bedrifter:

Infrastruktur for e-handel kan bestå av datamaskin(er) med egenskaper som server, web-server og katalogserver. Katalogsystem bør være utformet som metakatalog, og støtte etablerte standarder og spesifikasjoner for katalogtjenester, kommunikasjon og sikkerhet. Foruten katalogsystem bør infrastrukturen bestå av verktøy for integrasjon av data mellom web, katalogsystem og datakilder.

Ved utvikling av slike løsninger bør vi bruke definerte strategier, etablerte metoder for systemutvikling og verktøy for modellering.

8 Evaluering

Kapitlet inneholder en evaluering av prosjektet/studiet. Det omtaler vurdering av måloppnåelse, evaluering av prosjektarbeid, valg av samarbeidspartner og forslag til videre arbeid.

8.1 Evaluering av måloppnåelse, systemutvikling og suksesskriterier

De neste avsnittene vil presentere en evaluering av prosjektets måloppnåelse. Jeg vil først og fremst fokusere på om vår anbefaling tilfredstiller hovedmålet for prosjektet. Deretter vil jeg gjennomgå en vurdering av alle delmål. Jeg vil i tillegg omtale utviklingsprosessen frem til endelig anbefaling.

8.1.1 Oppnåelse av hoved- og delmål

Før evaluering av måloppnåelse vil jeg ta utgangspunkt i hoved- og delmål for studiet (ref pkt 1.1 Mål).

Hovedmål for studiet var:

Anbefale en infrastruktur, med katalogtjenester, for e-handel i små og mellomstore bedrifter.

Hovedmålet ble understøttet med følgende delmål:

Introdusere og beskrive tema e-handel, katalogtjenester og katalogsystemer i relasjon til infrastruktur for virksomheter.
Dokumentere fremgangsmåte for utvikling og løsning (prototyp) for e-handel.
Analysere og evaluere løsning for e-handel.

Hovedmål

På bakgrunn i teori innen e-handel og katalogsystemer, valgte strategier, metoder og modeller, og prototyp har prosjektet presentert en anbefaling for infrastruktur for e-handel. Jeg mener anbefalingen understøtter studiets hovedmål.

Studiets hovedmål er oppnådd.

Med dagens erfaringer ser jeg at vi kunne gjennomført studiet på flere måter. Sett i relasjon til hovedmålet kunne vi lagt mer vekt på å utnytte flere av egenskapene i valgt katalogsystem. På denne måten kunne vi trolig redusere antall eksterne verktøy som igjen kunne redusere både utviklingstid og ressursforbruk.

Delmål

Første del av avhandlingen (ref første delmål og kapitel 1, 2 og 3) har presentert flere sentrale tema innen e-handel, katalogsystemer/katalogtjenester og katalogtjenester i e-handel. Avhandlingen omtaler blant annet begreper innen e-handel og katalogsystemer. Den setter også søkelys på standarder og spesifikasjoner og alternativ arkitektur for systemer innen e-handel og katalogtjenester. Jeg har også presentert flere sentrale mekanismer og løsninger for sikkerhet.

Andre del av avhandlingen (ref andre delmål og kapitel 5 og 6) beskriver hvordan og hva vi har gjort frem til ferdig prototyp. Denne delen av dokumentet bruker første del av avhandlingen som utgangspunkt.

Siste del av avhandlingen (ref tredje delmål og kapitel 7 og 8) presenterer alternative strategier for systemutvikling, metoder for innsamling og bearbeiding av data og verktøy for modellering. Jeg har også kommentert vår løsning i relasjon til andre systemer for e-handel og alternative valg for infrastruktur.

Med utgangspunkt i avhandlingens første, andre og tredje del mener jeg alle delmål er tilfredstilt.

Studiets delmål er tilfredstilt.

Vi kunne gjennomført prosjektet med annen vektlegging, og jeg vil bruke de neste avsnittene til å kommentere dette.

I første del av avhandlingen (ref første delmål) kunne vi fokusert mer på ferdige pakkelsninger for e-handel. Men i prosjektperioden kunne jeg ikke registrere noen tilgjengelighet for pakkelsninger, med katalogsystemer, for e-handel.

Med bakgrunn av andre del av avhandlingen (ref andre delmål) kunne vi brukt mer ressurser på utforming av selve prototypen. For eksempel kunne vi gjøre prototypen mer fullstendig i forhold til grensesnitt (dialog med potensielle kunder), og vi kunne brukt database(r) som datakilde(r) i stedet for filer i Microsoft Excel. Vi kunne også hatt en tettere integrasjon mellom prototyp og produksjonssystemet hos samarbeidspartner.

I relasjon til tredje del av avhandlingen (ref tredje delmål) kunne prosjektet hatt en bredere diskusjon på alternative løsninger for samarbeidspartner.

Avhandlingen kunne for eksempel presentert en diskusjon om det i hele tatt er hensiktsmessig å innføre en løsning for e-handel i selskapet.

8.1.2 Systemutvikling

Det er ikke konkretisert noen mål for selve utviklingsprosessen. Jeg velger likevel å bruke noen ord på prosessen. Dette fordi den har gitt meg verdifull erfaring, og den har vært med på å underbygge min evne til å oppnå studiets mål.

Først og fremst vil jeg fremheve at selv om vi har rask og enkel tilgang til datakilder, med data om produkter og tjenester, bør prosessen for utvikling følge en definert fremgangsmåte. Vi valgte en fremgangsmåte som ble underbygget med flere underliggende strategier. Bruk av utviklingsstrategier ga oss mulighet for å utvikle prototyp mer effektivt og til et bedre produkt.

Vi brukte tid på å diskutere hvorledes eksisterende data og datastrukturer hadde verdi i ny løsning. Vi tok utgangspunkt i eksisterende datastrukturer. Nye datastrukturer ble definert på bakgrunn av diskusjoner med ansatte i selskapet. I ettertid kan jeg si at disse diskusjonene har vært meget nyttig.

Systemutviklingen (ikke studiet) ble forsinket i ca 3 måneder. Dette fordi vi fikk problemer med å aksessere et katalogsystem. Da vi oppdaget og rettet feilen ble forsinkelsen begrunnet med tekniske problemer. I ettertid skal jeg være åpen for kritikk på dette punktet. Det var riktig at det var et teknisk problem, men årsaken til det var trolig mangel på kunnskap om autentisering mot katalogsystemer. Erkjennelsen er et eksempel på hvordan det er å arbeide med systemer og utviklingsspråk vi ikke har erfaringer med.

8.1.3 Oppnåelse av suksesskriterier

IKON spesifiserte et sett med suksesskriterier for en evt fremtidige løsning for e-handel. Jeg vil i de neste avsnittene kommentere om vi har oppnådd disse. Mine kommentarer tar utgangspunkt i prototypen og sier noe om hva vi evt kan forvente av en ferdig løsning.

Prototypen tilfredstiller ingen av nevnte kriterier. Men en ferdig utviklet løsning, basert på prototyp, vil tilfredstille de fleste suksesskriterier.

Tabell 8-1 – Oppnåelse av suksesskriterier for IKON.

Suksesskriterier	Tilfredstilt?	Kommentar
Strukturert tilbuds- og kontraktsprosess	Ja	Kriteriet blir tilfredstilt om endelig løsning blir basert på prototyp og modell for "ny løsning for salg av tjenester i IKON".
God oppfølging av kontrakter	Nei	Endelig løsning støtter ikke kriteriet.
Raskt tilbud til kunden - med riktige data om kandidat	Ja	Kriteriet blir tilfredstilt om endelig løsning blir basert på prototyp og modell for "ny løsning for salg av tjenester i IKON".
Mer attraktiv hjemmeside	Ja	Kriteriet blir tilfredstilt om endelig løsning blir basert på prototyp og modell for "ny løsning for salg av tjenester i IKON".
Komme "nærmere" nye kunder og eksisterende kunder	Ja	Kriteriet blir tilfredstilt om endelig løsning blir basert på prototyp og modell for "ny løsning for salg av tjenester i IKON".
Videreutvikle personlige relasjoner til kunder.	Nei	Kriteriet blir delvis tilfredstilt. Dette er avhengig av hvor tett en eksisterende kunde blir integrert i fremtidig løsning for e-handel.

Tabell 8-1 viser hvilke kriterier IKONspesifiserte. For hvert kriterium har jeg angitt om det ble tilfredstilt og tilhørende kommentar.

8.2 Evaluering av prosjektarbeid

Arbeidet med hovedfagsstudiet startet våren 2000. Arbeidet med selve hovedfagsoppgaven startet høsten 2000. Frem til 1 oktober 2001 brukte jeg totalt 15 studiemåneder på hele studiet.

Jeg er meget godt fornøyd med at hovedfagsstudiet er gjennomført på normert tid.

Gjennomføringen av prosjektet har vært svært interessant. Jeg har tilegnet meg ny kompetanse om systemutvikling, strategier for e-handel, løsninger for e-handel og teknologi vi kan benytte i e-handel.

Prosjektet brukte et verktøy for prosjektadministrasjon. Verktøyet ble brukt til å lage en prosjektplan, og ble brukt gjennom hele studieperioden. Planen ble etablert ved oppstart av studiet, og ble "frosset" da arbeidet startet. Under prosjektet kunne jeg se evt hvor og når avvik oppstod i forhold til "frosset" plan. Gjennom studiet brukte jeg planen flittig for å holde oversikt over arbeidsoppgaver, milepæler og kritisk linje. Planen ble gjennomgått på hvert veiledermøte. Dette hjalp både veileder og meg selv til holde oversikt i over utviklingen. Jeg kunne brukt mer tid på å følge opp planen med hensyn på forbruk av ressurser og timer. I fremtidige prosjekter vil jeg søke å bruke slike prosjektverktøy med utvidet støtte for å holde oversikt over ressursforbruk.

Jeg har også gjort meg andre erfaringer med prosjektarbeid. I denne sammenheng kunne det vært en fordel å starte arbeidet med implementeringen av prototyp på et tidligere tidspunkt. Jeg kunne også vært mer kritisk til ulike veivalg i prosjektet. Dette gjelder for eksempel teknologivalg for prototyp.

8.3 Valg av samarbeidspartner

Prosjektet ble gjennomført i samarbeid med selskapet IKON AS. Før prosjektet startet kjente jeg noen ansatte i selskapet. Sett i ettertid var disse relasjonene av stor betydning for gjennomføringen av systemutviklingen frem til prototyp. Samarbeidet har vært meget nyttig, og jeg vil spesielt fremheve:

- tilgang til kompetanse om operativsystemer og verktøy for integrasjon.
- ansatte kjenner metoder for modellering og utvikling.
- ansatte har bidratt i utviklingen av prototyp.
- ansatte var interessert i resultatet fra prosjektet.

Jeg har vært bevisst på at relasjonen med selskapet ikke skulle redusere kvaliteten på resultatet av studiet. Dvs at utviklingen av løsningen ikke skulle gå på bekostning av hva vi kan forvente av forskningsresultat fra et slikt prosjekt.

8.4 Forslag til videre arbeid

I løpet av prosjektet har jeg avdekket flere områder som jeg gjerne skulle viet mer tid til. De mest nærliggende områdene er:

- det er behov for nærmere vurdering av *kostnad/nytte* ved innføring og bruk av katalogsystemer. Resultat fra en slik analyse bør vise om katalogsystemer er lønnsomt i infrastruktur for e-handel.
- erfaringene fra utviklingen av prototyp tilsier at det er behov for mer kunnskap om hvilken *teknologi* (katalogsystemer vs databasesystemer) som er best egnet for e-handel for små, mellomstore og store virksomheter.
- *strategier for utvikling* er meget interessant. Sett i relasjon til dette prosjektet er det behov for mer kunnskap om hvordan vi raskt og effektivt kan innføre løsninger for e-handel i etablerte virksomheter.
- det er en rivende utvikling i arbeidet med å definere ulike *standarder* og *spesifikasjoner* som, helt eller delvis, påvirker utviklingen av e-handel. Jeg skulle ønske jeg hadde bedre oversikt over utviklingen, og hvordan den evt har påvirket eller vil påvirke bruk av katalogsystemer i infrastruktur for e-handel.

9 Referanser

- [AA00] Arbeids- og administrasjonsdepartementet (2000), *Programbeskrivelse*, Statens forvaltningstjeneste, Statskjøpavdelingen, Norge. URL: <http://www.ehandel.dep.no/>.
- [AA99a] Arbeids- og administrasjonsdepartementet (1999), *E-handel i det offentlige*, Norge. URL: <http://www.ehandel.dep.no/>.
- [AA99b] Arbeids- og administrasjonsdepartementet (1999), *Innføring av elektronisk handel i offentlige innkjøp - Del III : Forslag til program 1999 - 2003*, Statens forvaltningstjeneste, Statskjøpavdelingen, Norge. URL: <http://www.ehandel.dep.no/>.
- [AES98] Erling S. Andersen (1998), *Systemutvikling*, NKI-forlaget, Norge. ISBN: 82-562-3266-8.
- [AGI99] Aberdeen Group, Inc. (1999), *The Business Case for Directory-Guided IT*, Boston, USA. URL: <http://www.aberdeen.com>.
- [ALT01] AltaVista (per 2001). URL: <http://www.altavista.com>.
- [AN99] Ulf Andersen mfl (1999), *Elektronisk handel*, Cappelen Akademiske Forlag, Oslo, Norge. ISBN: 82-456-0756-8.
- [BG86] Bjarne Berulfsen og Dag Gundersen (2001), *Fremmedordbok*, Kunnskapsforlaget Aschehoug - Gyldendal, Norge. ISBN: 82-573-0276-7.
- [BG99] Hans Erik Bryhn og Lars Ivar Galtung (1999), *Suksesskriterier i elektronisk handel*, Norges Handelshøyskole, Bergen, Norge.
- [BKS01] B2B Knowledge Source (2001), *B2B Knowledge Source - Focused Industry Information*. URL: <http://www.b2bknowledge.com/>
- [BLS95] Josh Benaloh, Butler Lampson, Daniel Simon, Terence Spies og Bennet Yee (1995), *Internet Draft - document version 1.00*, Microsoft Corporation, Washington, USA. URL: <http://activex.adsp.or.jp/english/specs/pct.htm>.
- [BRJ99] Grady Booch, James Rumbaugh, Ivar Jacobsen (1999), *The Unified Modeling Language User Guide*, Addison-Wesley Longman Inc, Massachusetts, USA. ISBN: 0-201-57168-4.
- [CA01] Computer Associates (per 2001). URL: <http://www.ca.com>.
- [CAP00] Cappelen Caplex (per 2000). URL: <http://www.caplex.net>.
- [CO01] Commerce One (per 2001). URL: <http://www.commerceone.com>.
- [CP01] Critical Path (per 2001). URL: <http://www.cp.net>.
- [CT00] Charles Trepper (2000), *E-Commerce Strategies*, Microsoft Press, Redmond, Washington, USA. ISBN: 0-7356-0723-0.
- [DIG01] digi.no (per 2001). URL: <http://www.digi.no>.

-
- [EP01] eProsper (per 2001). URL: <http://www.eprosper.co.uk>.
- [ERH99] Elliotte Rusty Harold (1999), *XML bible*, Foster City, California, USA. ISBN: 0-7645-3236-7.
- [FAS01] Fast Search (per 2001). URL: <http://www.fastsearch.com>.
- [HAL95] Knut Halvorsen (1995), *Å forske på samfunnet – en innføring i samfunnsvitenskapelig metode*, Bedriftsøkonomenes Forlag A/S, Oslo. ISBN: 82-7037-794-5.
- [HIO01] Høgskolen i Østfold – Avdeling for Informatikk og Automatisering (2001). URL: <http://olaf.hiof.no/~trondl/>.
- [IAHC01] The Internet International Ad Hoc Committee (2001). URL: <http://www.iahc.org>.
- [IBM01a] IBM Norge (per 2001). URL: <http://www.ibm.no>.
- [IBM01b] IBM International (per 2001). URL: <http://www.ibm.com>.
- [IET01] The Internet Engineering Task Force – IETF (2001). URL: <http://www.ietf.org>.
- [IK00] IKON AS (per 2000-12), *Kort presentasjon av IKON AS – IT-Konsulentene i Oppland*, Lillehammer, Norge.
- [IK01] IKON AS (per 2001). URL: <http://www.ikon.as>.
- [IP01] iPlanet – A SUN-Netscape Alliance (per 2001). URL: <http://www.iplanet.com>.
- [ISO01] International Standards Organization – ISO (per 2001). URL: <http://www.iso.ch>
- [ITU01] International Telecommunication Union – ITU (per 2001). URL: <http://www.itu.org>.
- [JBH98] Heinz Johner, Larry Brown, Franz-Stefan Hinner, Wolfgang Reis, Johan Westman (1998), *Understanding LDAP*, IBM Redbooks , New Yoork, USA. URL: <http://www.redbooks.ibm.com/>.
- [KUN01] Kunnskapsforlaget (1999), *Bokskap 4.0 – Norsk Språk*, Kunnskapsforlaget, Oslo, Norge. ISBN: 82-573-0916-8.
- [KVA01] Kvasir (per 2001). URL: <http://www.kvasir.no>.
- [LAN01] The Landfield Group (per 2001), *Internet Security Standards (web/commerce/security)*. URL: <http://www.landfield.com/isn/mail-archive/1998/Mar/0066.html>.
- [LØ00] Anund Lie og Jon Ølnes (2000), *FEIDE: Felles elektronisk ID for UoH-sektoren - Norsk Regnesentral rapport nr. 963*, Norsk Regnesentral, Norge. ISBN: 82-539-0467-3.
- [MAS01] MasterCard International (per 2001). URL: <http://www.mastercard.com>.
-

- [MAX00a] Tor Even Dahl (2000), *Metaløsninger for Telenor - notat*, MaXware International, Trondheim, Norge.
- [MAX00b] MaXware International (2001), *MaXware Synchronization Engine – User Guide*, MaXware International, Trondheim, Norge.
- [MAX01] MaXware International (per 2001). URL: <http://www.maxware.com>.
- [MET98] MetodeData AS (1998), *Innføring i Modelator 4.0*, MetodeData AS, Kolsås, Norge. ISBN: 82-91915-40-7.
- [MET00] Metadirectory (per 2000). URL: <http://www.metadirectory.co.uk>.
- [MET01] MetaMerge (per 2001). URL: <http://www.metamerge.com>.
- [MH91] Fred R McFadden og Jeffery A Hoffer (1991), *Data base management*, The Benjamin/Cummings Publishing Company, Redwood City, USA. ISBN: 0-8053-6040-9.
- [MM00] Constante H. McLaren og Bruce J. McLaren (2000), *E-commerce – Business on the Internet*, South-Western Educational Publishing, Cincinnati, Ohio, USA. ISBN: 0-538-68918-8.
- [MS01] Microsoft Home Page (per 2001). URL: <http://www.microsoft.com>.
- [MT00] Mai-Ian Tomsen (april 2000), *Killer Content – Strategies for Web Content and E-Commerce*, Massachusetts, USA. ISBN: 0201657864.
- [MYS01] MySQL (per 2001). URL: <http://www.mysql.com>.
- [NED00] Norsk EDIPRO (22 august 2000), *Infrastruktur for elektronisk handel – Prosjektdirektiv spesifikasjonsfasen v 0.9*, EdiSys AS, Norge.
- [NET01] Netscape (per 2001). URL: <http://www.netscape.com>.
- [NEX01] Nexor (per 2001). URL: <http://www.nexor.com>.
- [NHD01] Nærings- og handelsdepartementet (2001), *Digitale signaturer gir tillit til elektronisk kommunikasjon*, Norge. URL: <http://kimen.dep.no/nhd/publ/esign/esign-08.html>.
- [NHO00] Næringslivets Hovedorganisasjon (per 2000). URL: <http://www.nho.no>.
- [NKK98] Sverre M Nyberg, Dagfinn Kristoffersen og Jarl Kvanli (1998), *Informasjonsstrategi for Forsvarets kompetansesenter for kontor-, arkiv- og meldingstjeneste i Forsvarets informasjonssystem*, Høgskolen i Hedmark, Rena, Norge.
- [NOV01] Novell World Wide (per 2001). URL: <http://www.novell.com>.
- [NR01] Norsk Regnesentral (per 2001), *Norsk Regnesentral*, Oslo, Norge. URL: <http://www.nr.no>.
- [OBI01] Open Buying on the Internet (per 2001). URL: <http://www.openbuy.org>.

-
- [ORA01] Oracle Corporation (per 2000). URL: <http://www.oracle.com>.
- [PHP01] PHP: Hypertext Preprocessor (per 2001). URL: <http://www.php.net>.
- [PR97] Roger S Pressman (1997), *Software Engineering – A practitioners approach – Fourth Edition*, McGraw-Hill, USA. ISBN: 0-07-709411-5.
- [RA00] Archie Reed (2000). *Implementing directory service*. McGraw-Hill, New York, USA. ISBN: 0-07-134408-X.
- [RMH01] Jeffrey Ricker, Drew Munro and Doug Hopeman (2001), *XML and EDI:Peaceful Co-Existence*, XMLSolutions Corporation. URL: <http://www.xedi.org>.
- [SAP01a] SAP (per 2001). URL: <http://www.sap.com>.
- [SAP01b] Central Michigan University (2001), *University Alliance Program*. URL: <http://sap.mis.cmich.edu>.
- [SBS97] B.R. van der Spek and A.L. Spijkervet (1997), *Knowledge Management: Dealing intelligently with knowledge*, Kenniscentrum CIBIT, Utrecht, Nederland.
- [SC00a] Curt Simmons (2000), *Creating Active Directory Infrastructures*, Prentice Hall PTR, New Jersey, USA. ISBN: 0-13-087631-1.
- [SC00b] Curt Simmons (2000), *Master Active Directory Visually*, IDG Books Worldwide Inc., California, USA. ISBN: 0-7645-3425-4.
- [SET01] SET Secure Electronic Transaction (per 2001). URL: <http://www.setco.org>.
- [SI99a] Solheim, Ivar (1999), *Strategier for elektronisk handel i offentlig sektor. Drøfting av erfaringer fra det svenske "Prosjekt Elektronisk handel"*, Norsk regnesentral, Oslo, Norge. ISBN: 82-539-0418-5.
- [SI99b] Igor Slavinsky (1999), *Secure communications using Transport Layer protocols*. URL: <http://cactus.eas.asu.edu/Partha/Teaching/Crypto.1999S/slavinsky.html>.
- [SIM01] Siemens (per 2001). URL: <http://www.siemenscomms.co.uk>.
- [SIN01] Sintef Oslo (per 2001). URL: <http://www.oslo.sintef.no>.
- [SOL00] Dag L Solhaug (per 2000), *Katalogtjenester - Organisasjonsmodellering i Active Directory*, Høgskolen i Agder, Grimstad, Norge. URL: http://www.siving.hia.no/ikt00/ikt6400/dlsolhau/Rapporten.htm#_Toc488143265
- [SS00] Beth Sheresh og Doug Sheresh (2000), *Understanding Directory Services*, New Riders Publishing, USA. ISBN: 0-7357-0910-6.
- [STA96] Rapport fra Statssekretærutvalget for IT (1996), *Den norske IT-veien. Bit for bit*, Statens forvaltningstjeneste, Norge. URL: <http://odin.dep.no/>.

- [SUN01] Sun Microsystems (per 2001). URL: <http://www.sun.com>.
- [TBG01] The Burton Group (per 2001). URL: <http://www.tbg.com>.
- [TBG98] The Burton Group (1998), *Network strategy overview – Enterprise Directory Infrastructure: Meta-Directory Concepts and Functions v2*, USA. URL: <http://www.tbg.com>.
- [TD00] Camilla AC Teffers og Claude Marie Davidsen (august 2000), *Elektronisk handel – fra buzz til biz*, Tapir Akademiske Forlag, Trondheim, Norge. ISBN: 82-519-1606-2.
- [TEL01] Telenor ASA (per 2001). URL: <http://www.telenor.no>.
- [TES00] Telenor ASA og Siemens Information and Communications (2000), *Tilbud – Elektronisk markeds plass for det offentlige*, Oslo, Norge. URL: <http://www.ehandel.dep.no>.
- [UD01] UDDI.org (2001), *Universal Description, Discovery and Integration of Business for the Web*. URL: <http://www.uddi.org>.
- [UIT01] Universitet i Tromsø – Nettseksjon (per 2001). URL: <http://www.cc.uit.no/nett/>.
- [UNI00] UNISYS Norge (per 2000). URL: <http://www.unisys.no>.
- [UNV98] Universitetsforlaget (1988), *Bokmålsordboka*, Norbok a.s., Oslo, Norge. ISBN: 82-00-07667-9.
- [VIS01] Visa International (per 2001). URL: <http://www.visa.com>.
- [WIN01] Windows 2000 Magazine (per 2001). URL: <http://www.win2000mag.com>.
- [XM01] XML.com (per 2001). URL: <http://www.xml.com>.
- [YAH01] Yahoo (per 2001). URL: <http://www.yahoo.com>.
- [ØM00] Østbø, Morten (1999), *Anvendelse av UML til dokumentasjon av generiske systemer*, Høgskolen i Stavanger, Stavanger, Norge.

Om Unified Modeling Language

Vedlegget beskriver Unified Modeling Language (UML), og følgende er omtalt:

- introduksjon
- konseptuell modell
- arkitektur
- verktøy.

Følgende kilder er grunnlag for beskrivelsen:

- The Unified Modeling Language – User Guide [BRJ99].
- Anvendelse av UML til dokumentering av generiske systemer [ØM00].

1 Introduksjon til UML

UML er et standardisert språk for grafisk modellering. Vi kan bruke språket til å modellere flere systemer. Det gjelder fra store informasjonssystemer til mindre distribuerte web-applikasjoner.

Språket er meget uttrykksfullt, og gjør det mulig å visualisere utvikling og implementasjon av applikasjoner og systemer [BRJ99].

Vi kan bruke språket til flere formål [BRJ99]:

Visualisere - vi kan bruke UML til å kommunisere konseptuelle modeller til for eksempel deltakere i et utviklingsprosjekt. Vi kan også bruke språket til å visualisere komplekse, og fremme kommunikasjon av, problemstillinger i applikasjon(er).

Spesifisere - vi kan nytte UML til å konstruere presise, entydige og komplette modeller. Vi kan i denne sammenheng nytte språket som grunnlag for beslutninger innen analyse, design og implementasjon av informasjonssystemer.

Konstruere - vi kan overføre modeller i UML direkte til et programmeringsspråk, tabeller i relasjonsdatabaser eller til persistente lager for objektorienterte databaser.

Dokumentere - vi kan bruke UML til å dokumentere alle elementer i en systemarkitektur. Vi kan også nytte språket til å definere kravspesifikasjon og tester. I tillegg kan vi bruke UML til å modellere aktiviteter i prosjekter og administrasjon av utgivelser (av applikasjoner).

UML er i hovedsak beregnet for applikasjonsorienterte systemer.

Språket har blitt brukt i forbindelse med modellering av:

- store informasjonssystemer
- tjenester innen bank og finans
- telekommunikasjon
- transport
- forsvar/romfartsindustri
- detaljhandel
- teknologi for medisin
- vitenskap
- distribuerte web-baserte tjenester.

2 Konseptuel modell

Litteraturen fremhever at for å forstå UML er det behov for å skape en konseptuel modell av språket, og det krever at vi forstår tre sentrale elementer i det [BRJ99]:

- basis byggesteiner i UML.
- regler for hvordan byggeklossene kan settes sammen.
- noen felles mekanismer som gjelder for UML.

2.1 Basis byggesteiner i UML

UML består av tre typer byggesteiner:

1. Ting.
2. Relasjoner.
3. Diagrammer.

Vi har fire typer *ting* i UML; strukturelle, atferdsorienterte, grupperende og spesielle ting. Tingene er basiselementer i UML, og vi bør bruke de aktivt for å lage gode modeller.

I UML finner vi fire typer *relasjoner*:

1. Avhengighet - er en semantisk relasjon mellom to ting hvor en endring i en ting (avhengig ting) muligens vil påvirke semantikken i en annen ting (uavhengig ting).
 2. Assosiasjoner - er en strukturell relasjon som beskriver et sett med forbindelser. En forbindelse er en kobling mellom objekter.
 3. Generalisering - er en spesialisert/generalisert kobling hvor objekt av det spesialiserte element (barn) er stedfortreder for objekt av det generaliserte element (forelder).
 4. Realisering - er en semantisk relasjon mellom klasser hvor en klasse spesifiserer en kontrakt hvor en annen klasse garanterer for gjennomføring.
-

Nevnte relasjoner er basis for hvordan vi kan bruke språket. Vi kan også bruke variasjoner av relasjonene.

Diagram er grafisk presentasjon av et sett med elementer. Vi kan bruke diagram for å visualisere systemer fra forskjellige perspektiv.

Ting, relasjoner og diagrammer er kjerneelementer i UML; *relasjoner* kobler *ting* sammen og *diagrammer* grupperer interessante samlinger med *ting*.

2.2 Regler for bruk byggeklosser

UML har, som alle andre språk, et sett med regler, og vi bør bruke reglene til å lage gode modeller. En god modell er en modell som er semantisk konsistent og i harmoni med alle relaterte modeller.

UML har semantiske regler for:

- navn – det vi kan kalle ting, relasjoner og diagrammer.
- ramme – kontekst som gir en spesiell mening til et navn.
- synlighet – hvordan navn kan bli sett og brukt av andre.
- integritet – hvordan ting på en ordentlig og konsistent måte relaterer til hverandre.
- utførelse – hva det betyr å kjøre eller simulere dynamiske modeller.

Reglene skal motivere, ikke tvinge bruker, til å rette fokus på viktige spørsmål omkring analyse, design og implementasjon.

2.3 Noen felles mekanismer som gjelder for UML

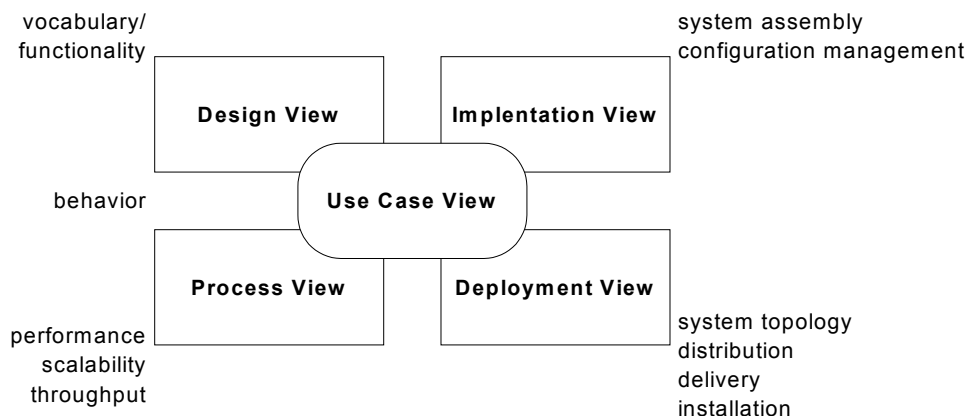
Vi kan bruke flere mekanismer for å gjøre språket bedre og mer konsekvent. UML støtter fire former for mekanismer:

1. Spesifikasjoner – støtter et semantisk redskap, som inneholder alle deler av en modell i et system, hvor hver del er relatert til hverandre i en konsistent form.
 2. Utseende – de fleste elementer i UML har en unik og direkte grafisk notasjon. Den gir en visuell representasjon av hvert element.
 3. Felles deler – i objektorienterte systemer kan vi "dele verden opp" i flere deler. Med deler menes for eksempel klasse/objekt og grensesnitt/implementasjon. UML bruker felles deler eller symboler i notasjon for eksempel klasse/objekt, og skiller de fra hverandre ved å bruke forskjellige skrifttyper.
 4. Utvidbare mekanismer – UML er et standard språk for modellering av applikasjoner, men UML kan også utvide språket på en kontrollert måte. Språket tilbyr flere typer mekaniser for utvidelse.
-

3 Arkitektur

Ved visualisering, spesifisering, konstruering og dokumentering av systemer må vi kunne se systemene fra forskjellige perspektiver. Dette er viktig når vi skal kommunisere med forskjellige mennesker - brukere, utviklere, testere, systemadministratorer, prosjektdeltagere osv.

Booch [BRJ99] mfl mener at arkitektur i systemer kan best bli beskrevet vha fem integrerte elementer (se figur A-1). Hvert element gir ulik eksponering over organisering og struktur i et system.



Figur A-1 – Modellering av systemarkitektur [BRJ99].

Use Case View - omfatter beskrivelser av systemer slik brukere, analysepersonell og testere ser det. Dette perspektivet spesifiserer elementer som former systemets arkitektur. I UML blir statiske elementer, i dette perspektivet, beskrevet vha Use Case Diagram. Dynamiske elementer blir beskrevet vha Interaction (se figur A-2), Statechart og Activity Diagram.

Design View - omfatter beskrivelser av klasser, grensesnitt og samarbeid som former et problem til løsning. Perspektivet fokuserer på funksjonelle krav til systemet. Kravene skal understøtte brukers ønsker og behov til tjenester i systemet. I UML blir statiske elementer, i dette perspektivet, beskrevet vha Class og Object Diagram. Dynamiske elementer blir beskrevet vha Interaction, Statechart og Activity Diagram.

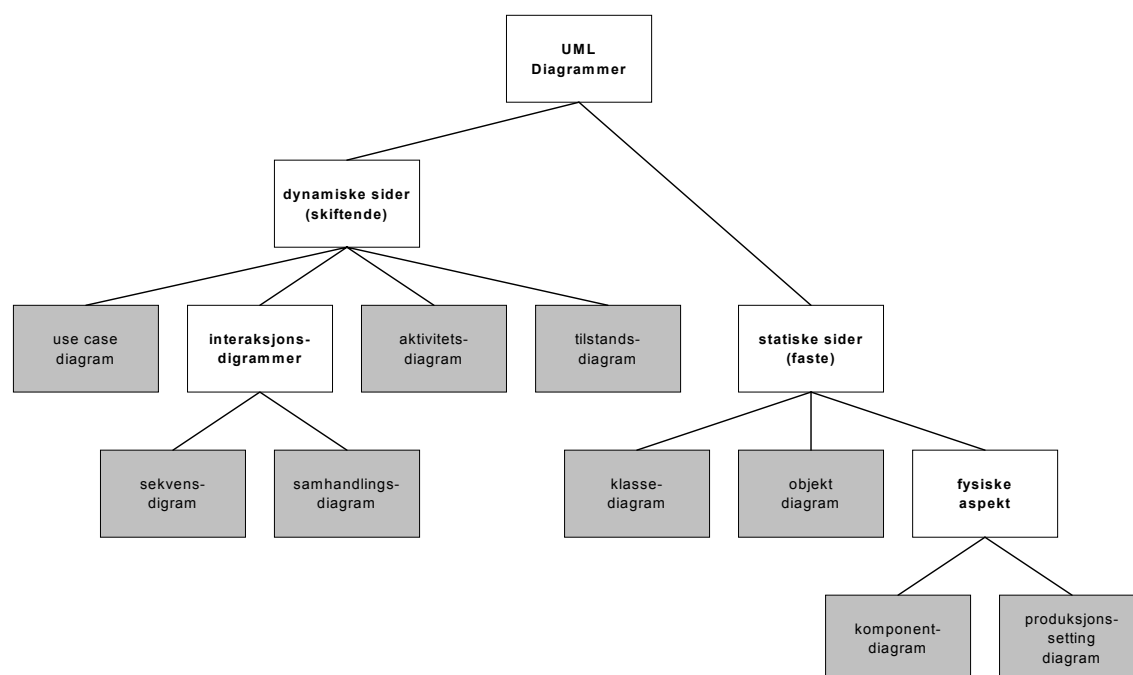
Process View - omfatter elementer og prosesser som former systemets sammenfallende og synkroniserte mekanismer. Dette perspektivet fokuserer på ytelse, skalerbarhet og gjennomstrømning (av data) i systemet. I UML bruker vi samme diagramtyper (både statisk og dynamisk) som ved *Design View*, men med mer fokus på aktive klasser.

Implementation View - omfatter komponenter og filer som er brukt for å samle og frigi det fysiske system. Perspektivet ser konfigurasjonshåndtering som viktig ved implementasjon av et system. I UML bruker vi Component Diagram for å beskrive det statiske perspektivet av systemet. I det dynamiske perspektivet bruker vi Interaction, Statechart og Activity Diagram.

Deployment View - omfatter elementer som former systemets topologi (hardware). Perspektivet fokuserer på distribusjon, levering og installasjon av deler av det fysiske system. I UML kan statiske elementer bli beskrevet vha Deployment Diagram. Dynamiske elementer kan bli beskrevet vha Interaction, Statechart og Activity Diagram.

Ut fra nevnte perspektiv kan vi altså gruppere UML slik [ØM00] (se figur A-2):

- dynamiske diagrammer - viser dynamiske (skiftende) aspekter i systemer.
- statiske diagrammer - viser statiske (faste) aspekter i systemer.



Figur A-2 – Kategorier og diagramtyper (grå farge) i UML [ØM00].

Interaksjonsdiagrammer viser en interaksjon bestående av et sett objekter og deres relasjoner, inklusive meldinger, som kan bli sendt mellom dem. UML har to typer interaksjonsdiagrammer.

Diagrammer under fysiske aspekt er komponentdiagram og diagram for produksjonssetting. Vi kan bruke diagrammene til å modellere de fysiske aspektene i objektorienterte systemer. UML har to typer fysiske diagrammer.

UML består av totalt 9 typer diagrammer [SOL00]:

1. Use Case diagram (Use Case diagram) - beskriver brukerkrav til systemet. Dvs å finne systemets aktører, brukermønstre for aktører og avgrense ansvarsområde for systemet.
2. Sekvensdiagram (Sequence diagram) - beskriver hvordan utføre en oppgave, og inneholder aktører og klasser som må til for å gjennomføre oppgaven.

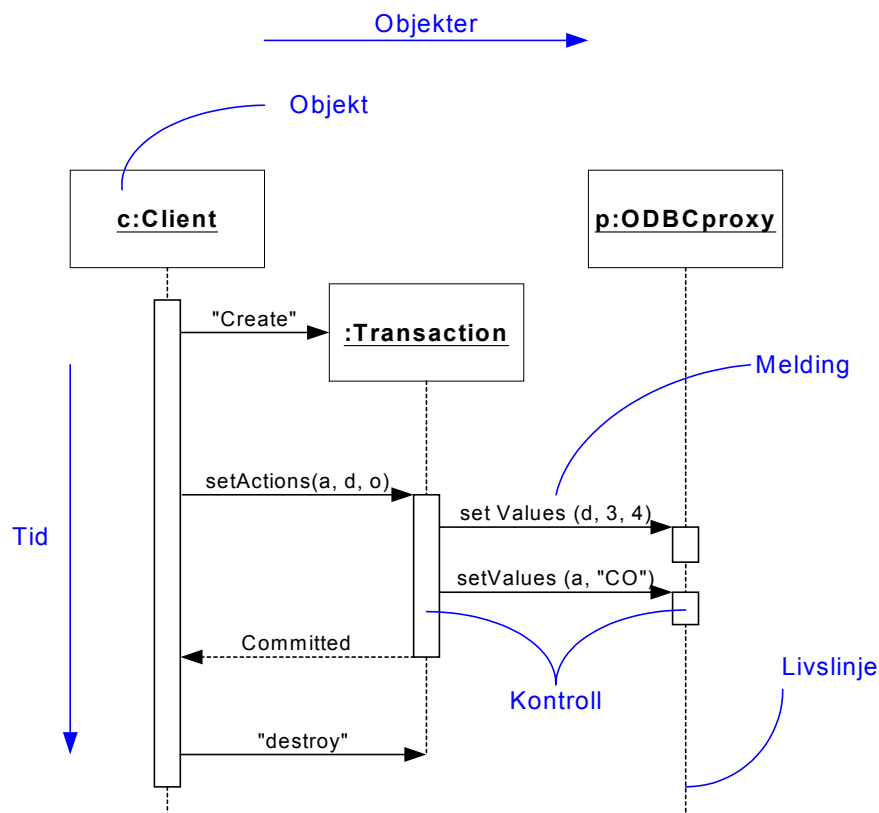
3. Samhandlingsdiagram (Collaboration diagram) - fokuserer på organiseringen av objekter som deltar i en interaksjon. Diagrammet kan også vise flyten og rekkefølgen av meldinger.
4. Aktivitetsdiagram (Activity diagram) - beskriver hvordan aktiviteter er samordnet og avhengige av hverandre (flyten mellom dem).
5. Tilstandsdiagram (Statechart diagram) - beskriver mulige tilstander til objektene i en klasse. Diagrammet kan også vise hva som skal til for at et objekt skifter tilstand.
6. Klassediagram (Class diagram) - beskriver alle klasser som systemet består av og hvilke relasjoner det er mellom dem. Diagrammet viser en statisk beskrivelse av systemets struktur.
7. Objektdiagram (Object diagram) - diagrammet viser objekter og deres relasjoner på et gitt tidspunkt.
8. Komponentdiagram (Component diagram) - beskriver realiseringen av systemet med komponenter og pakker. Diagrammet viser systemets katalog, filstruktur og avhengigheter mellom dets pakker og komponenter.
9. Produksjonssettingsdiagram (Deployment diagram) - viser konfigurasjon mellom fysiske maskiner eller prosesser.

4 Notasjon for UML

Nedenfor følger en enkel oversikt og beskrivelse av diagrammer som vi brukte i prosjektet.

4.1 Sekvensdiagram (Sequence diagram)

Sekvensdiagram er et adferdsorientert diagram. Det legger vekt på strukturell organisering av objekter som sender og mottar meldinger.

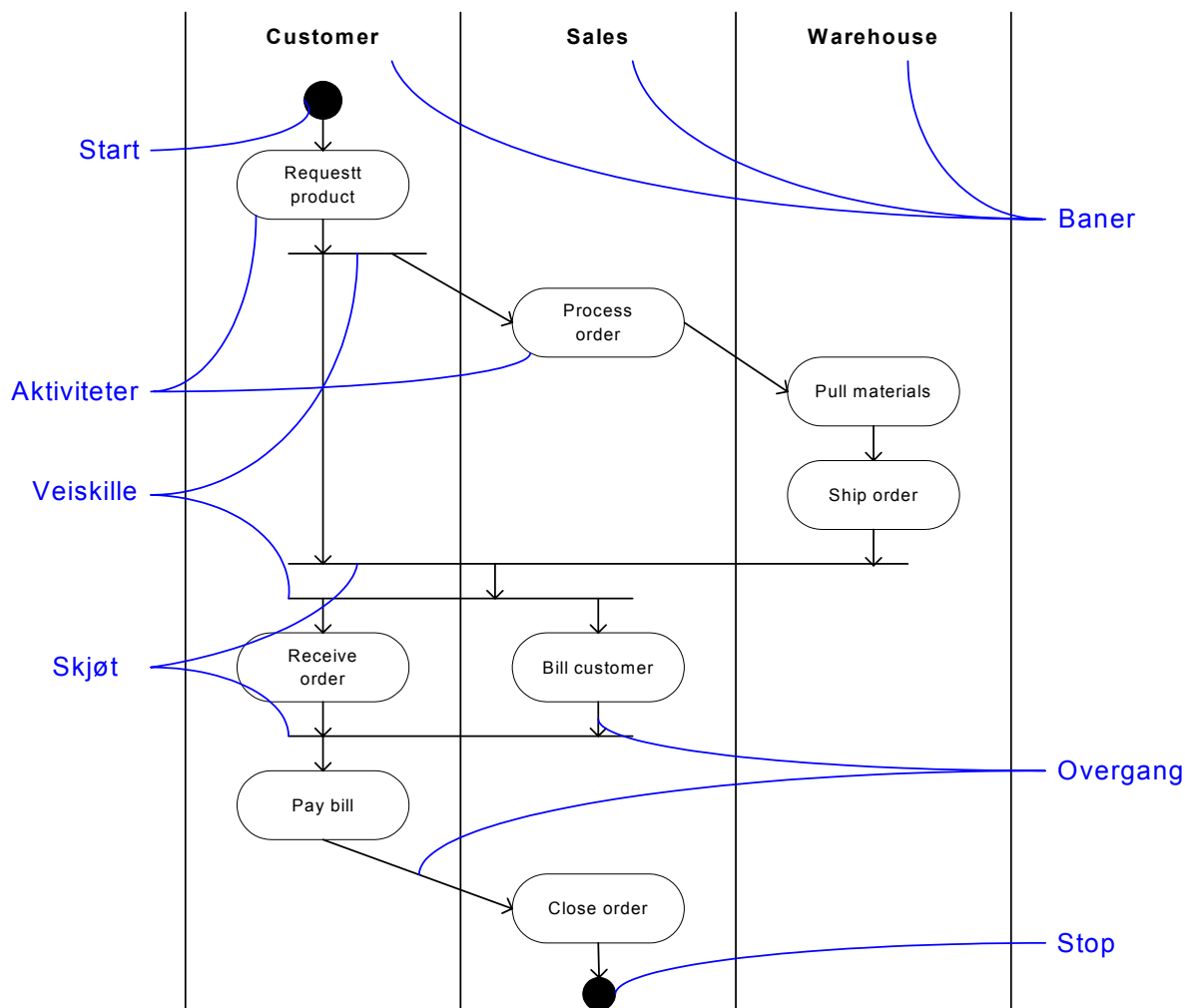


Figur A-3 – Notasjon for Sequence Diagram [BRJ99].

Figur A-3 viser et eksempel på et Sequence Diagram. Skissen viser hvordan meldinger flyter mellom objekter over en tidsperiode. Hvert objekt har en livslinje. Objektet har kontroll (arbeidsoppgave) når livslinjen er uthevet.

4.2 Aktivitetsdiagram (Activity diagram)

Aktivitetsdiagram er også et adferdsorientert diagram som viser flyt mellom aktiviteter.



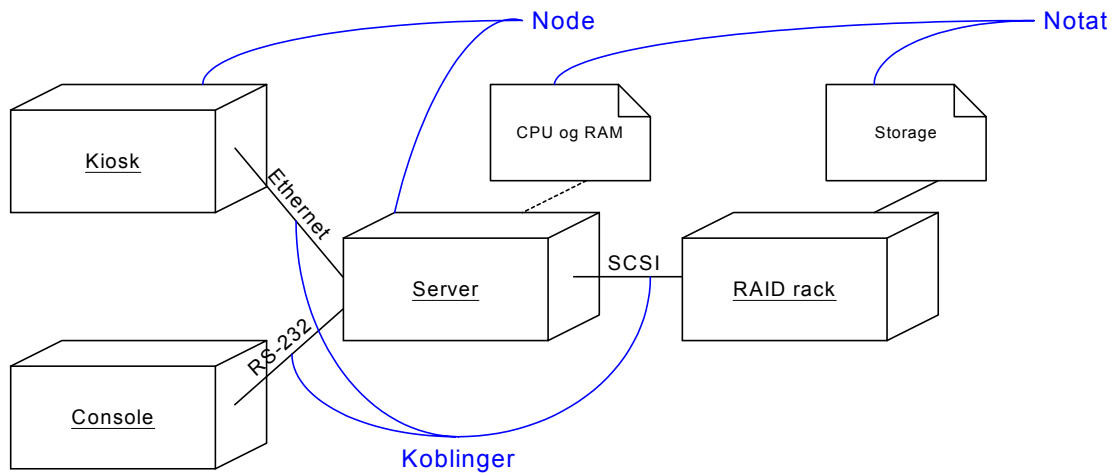
Figur A-4 – Notasjon for Activity Diagram [BRJ99].

Figur A-4 viser et eksempel på et Activity Diagram. Tegningen viser hvem og hva som blir gjort, i relasjon til kunde, salg og varehus, ved salg av varer. En serie med aktiviteter er angitt med start- og stoppunkt. Hver aktivitet er tilknyttet med overganger. Ved overgang fra en aktivitet til en annen er foregående aktivitet avsluttet. En avsluttet aktivitet kan initiere flere nye aktiviteter (se "veiskille"), eller flere aktiviteter kan sammensluttet (se "skjøt") om en aktivitet.

Det kan være tilknyttet vilkår ved overganger mellom aktiviteter. Vi kan angi overganger med alternative veier i diagrammet. Vi kan avgjøre alternativt veivalg ut fra en test (f eks Boolean-test). Slike tester blir angitt med tegnet **◆** i diagrammet.

4.3 Produksjonssettingsdiagram (Deployment diagram)

Diagram for produksjonssetting viser et sett med noder og relasjoner mellom disse.



Figur A-5 – Notasjon for Deployment Diagram [BRJ99].

Figur A-5 skisserer et eksempel på Deployment Diagram. Skissen viser sammenkobling mellom ulike noder. En node kan være et konsoll, PC, server, eksterne rack etc.

Ved modellering av Deployment Diagram gir UML mulighet til å bruke andre visuelle effekter. Ved bruk av slike effekter er det lettere å kommunisere innholdet i modellen. Figur A-5 viser at vi skal bruke en PC for noden Kiosk, et konsoll for noden Console eller et rack for noden RAID rack. Vi kan også bruk egne tegninger for telefon, faks, modem, kamera, skanner, skriver etc.

5 Applikasjoner for UML

Det eksisterer flere typer applikasjoner som understøtter modelleringprosessen i UML. Fordelene med å bruke slike verktøy er å ha oppdaterte og konsistente modeller gjennom hele utviklingsprosessen. Eksempel på slike verktøy er Rational Rose fra Rational Corporation og Select fra Princeton Softech [ØM00].

Prosjektet brukte ikke egne applikasjoner for UML. Vi brukte et ordinært tegneverktøy i Microsoft Windows. Vi vurderte verktøyet som "godt nok".

Om IKON AS

Vedlegget presenterer selskapet IKON AS, og omtaler følgende emner:

- forretningsidé, strategi og tjenester
- kjennetegn og verdigrunnlag
- nøkkeltall og utvikling.

Bakgrunn for vedlegget er IKONs hjemmeside [IK01] og egen presentasjon av selskapet [IK00].

1 Forretningsidé, strategi og tjenester

Selskapet selger konsulent- eller rådgivningstjenester innen fagområdet IT eller nært tilknyttet IT. IKON har en *forretningsidé*. Den presiserer eksistensgrunnlaget for selskapet, og er slik:

IKON skal tilby rådgivning og konsulenttjenester innenfor følgende områder:

- strategisk bruk av IKT og virksomhetsstrategi
- prosjektledelse og koordinering
- spesialisttjenester innenfor IKT.

IKON skal framstå som en konkurransedyktig og attraktiv samarbeidspartner for større offentlige og private virksomheter på nasjonalt plan.

Selskapet har valgt en *strategi* som tar utgangspunkt i premissene gitt av kunden. Strategien er utformet slik:

Utgangspunktet for IKONs bidrag i utviklingen av IT-løsninger bygger på kundenes forretningsstrategier, enten det handler om å skape kostnadseffektivitet, effektiv informasjonsflyt eller konkurransefortrinn.

IT strategien skal således understøtte forretningsdriftens overordnede målsettinger.

Teknologien skal være et middel, ikke et mål i seg selv.

IKON tilbyr et variert sett med *tjenester* innen IT og IS. Satsningsområdene for IKON er slik:

IKON tilbyr rådgivning- og konsulenttjenester innenfor følgende områder:

- strategisk bruk av IKT og virksomhetsstrategi
- prosjektledelse og koordinering
- spesialisttjenester innenfor:
 - IT infrastruktur og driftskonsepter
 - kommunikasjonsløsninger
 - informasjonsforvaltning
 - systeminnføring
 - web-løsninger

2 Kjennetegn og verdigrunnlag

Selskapet fremhever følgende om seg selv [IK00]:

- utadvendt virksomhet, selvstendige konsulenter og team.
- kunden er vårt kontor.
- alle ansatte produserer for IKON.
- minimal intern infrastruktur, men moderne mobile løsninger.
- godt humør og positive holdninger.
- den enkelte medarbeider er vår viktigste ressurs og aktivum.

I presentasjonen av selskapet blir følgende sagt om *verdigrunnlaget*:

- den enkeltes holdning, *kompetanse* og erfaring.
- relasjoner / nettverk - kundefokus.
- sosialt og faglig miljø.
- sunn økonomi med fokus på resultater.
- "kunden er vår beste selger".

Slik jeg oppfatter verdigrunnlaget retter det fokus på kunder og ansatte. Selskapet søker å skape en verdiutvikling gjennom et godt arbeidsmiljø og sunn bedriftskultur.

3 Nøkkeltall og utvikling

Nøkkeltall for selskapene (IKON AS og IKON Øst AS) er gjengitt i tabellene nedenfor. Tallene for IKON AS er fra oppstarten våren 1997, og tallene for IKON Øst AS er fra 1998. Per 2000-05 er IKON AS og IKON Øst AS slått sammen i IKON AS.

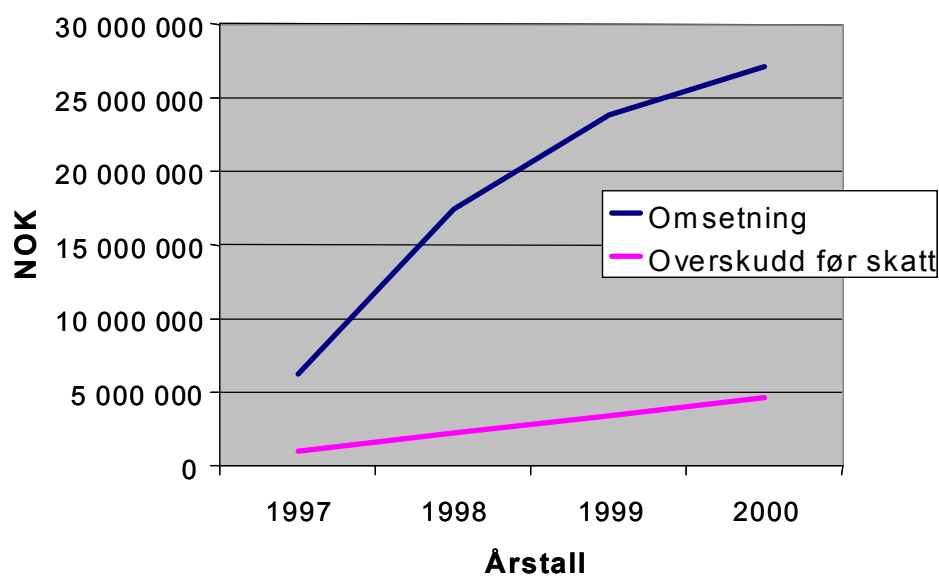
Tabell B-1 – Nøkkeltall for IKON AS (tall i mill kr).

Regnskapsavslutning	31.12.1999	31.12.1998	31.12.1997	1998-1999	1997-1998
Ordinært resultat før skatt	3.089	2.294	1.042	35,05 %	120,15 %

Tabell B-2 – Nøkkeltall for IKON Øst AS (mill kr).

Regnskapsavslutning	31.12.1999	31.12.1998	1998-1999
Ordinært resultat før skatt	1.317	0,199	561,81 %

Figur B-1 viser utvikling av resultat og omsetning i perioden 1997-2000.



Figur B-1 – Omsetning og resultatutvikling for IKON [IK01].

Intervjuguide for datainnsamling av nå-situasjon i IKON

Intervjuguiden [HAL95, NKK98] er utarbeidet for å ivareta nødvendig kvalitet i forberedelse med datainnsamling, bearbeidelse og analyse av data. Guiden skal sikre at intervjuene blir gjennomført under like forhold, og ivaretagelse av alminnelige forskningsetiske prinsipper.

Undersøkelsen er utarbeidet, gjennomført og analysert i henhold til følgende retningslinjer:

1. Sette seg grundig inn i relevant bakgrunnsinformasjon før hvert intervju.
2. Når det er foretatt nødvendige avklaringer omkring problem, metodevalg og nødvendige avgrensninger skal vi gjøre avtaler med respondenter for gjennomføring av selve intervjuet.
3. Start intervjuet med å skape en atmosfære som bidrar til størst mulig åpenhet. Om det er flere deltagere (intervjuer og respondenter) på intervjuet skal gruppen bli enige om hvem som fører ordet. Andre deltagere kan supplere ved behov under presentasjonsfasen.

Følgende elementer inngår i presentasjonen:

- presentasjon av prosjekt (med problemstilling) og herunder hensikt i relasjon til eget studie.
 - nytteverdi for bedriften.
 - forankring ved NTNU.
 - kort om metode og rammefaktorer.
 - diskresjon og behandling av "rådata".
 - øvrige respondenter.
 - forventet ferdigdato.
 - kort om fremgangsmåten for selve intervjuet (inkluderer presentasjon av tema).
 - tema for undersøkelsen: personalia, om bedriften, om infrastruktur og om salg av tjenester.
4. Intervjuguiden består av spørsmål med tilhørende felter for notering. Om det er flere som gjennomfører intervjuet kan deltakerne fordele arbeidet. Under selve intervjuet *bør* en person ha ansvar for å stille spørsmålene, mens en annen har ansvar for å notere og registrere. Evt rollefordeling *må* være avklart på forhånd. Om det er flere som gjennomfører ulike intervju, *bør* de foregå mest mulig likt.

Øvrige forhold: de innledende spørsmålene (1 - 4) er benyttet for å få respondentene "på gli". Svarene vil derfor være av mindre betydning ved analyse av data.

Personalia	Navn og alder?	
1		
2	Hvor lenge har du arbeidet i bedriften?	
3	Kan du gi en kort beskrivelse av din stilling? (fagområde, ansvar, samarbeidspartnere, ledelse etc)	
4	Hva er de viktigste interne (i bedriften) arbeidsoppgaver for din stilling?	
Om bedriften	Når ble bedriften etablert, og hvor mange arbeider i bedriften i dag?	
5		
6	Hva er visjon, strategi og mål for bedriften?	
7	Hva er oppnådde resultater? (nøkkeltall)	

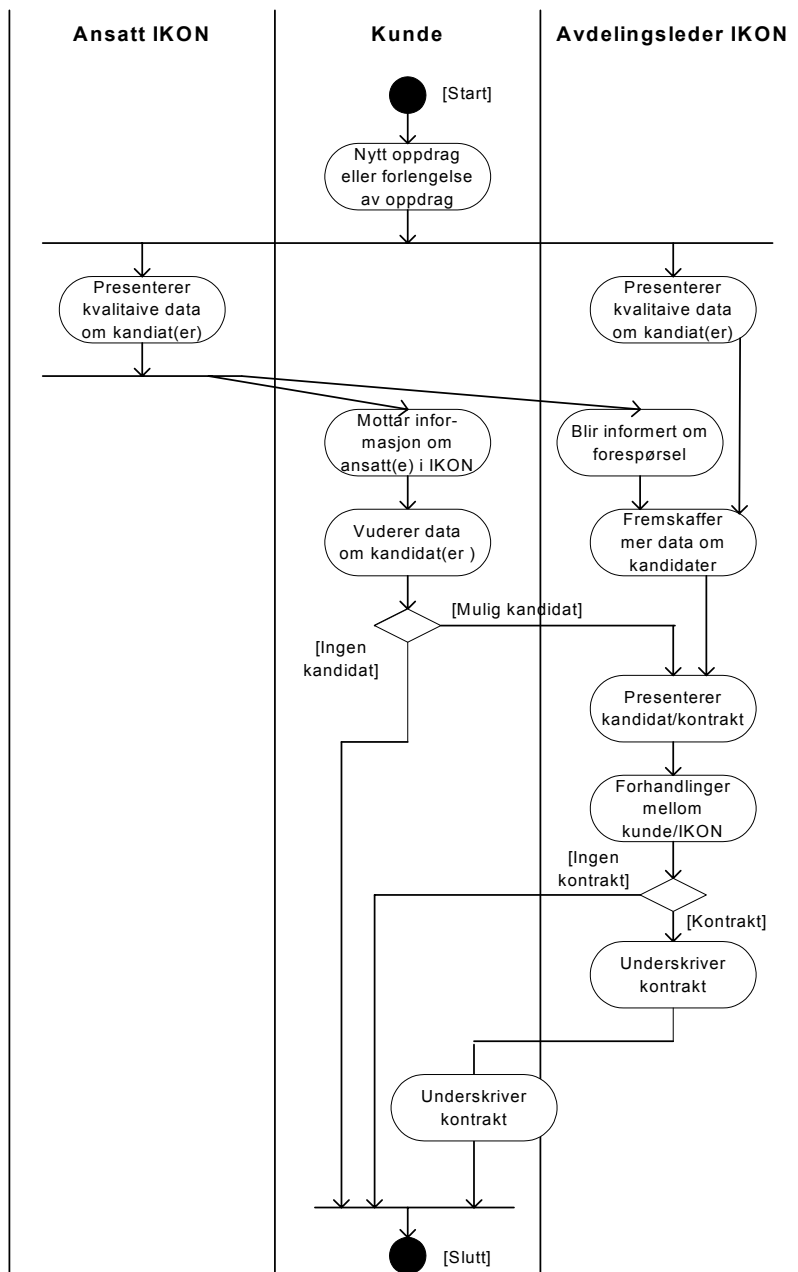
8	Hvor kjent er IKON med kundenes forhold til handel over Internett?	
9	Hva definerer du som suksesskriterier for e-handel i IKON?	
Om infrastruktur 10	Hvem har ansvaret for den teknologiske infrastrukturen?	
11	Hva støtter dagens infrastruktur i bedriften, og hvordan blir dette gjort? (eks salg, e-post, register over ansatte og kunder, kunder, administrasjon etc)	
12	Hva består infrastrukturen av i dag? (HW, SW, databaser, katalogsystemer) (skisse) (hvilke deler av infrastrukturen nyttes til hvilke oppgaver?) (hvilke investeringer er gjort?)	
13	Hvordan blir infrastrukturen etablert eller endret? (på bakgrunn av hvilke innspill, hvem gjør hva og hva gjøres)	
14	I hvilken periode er dagens infrastruktur innført?	

15	Hva bør bedriften gjøre for å forbedre infrastrukturen? (ift salg etc)	
16	Hvordan vurderer du dagens infrastruktur/teknisk løsning i bedriften?	
17	Hvordan understøtter dagens infrastruktur salgsprosessen i bedriften?	
Om salg av tjenester 18	Hvem har ansvar for salg av tjenester i bedriften?	
19	Hvor mange konsulent-/rådgiversalg gjennomføres per år?	
20	Hvordan gjennomføres salgsprosessen for tjenester? (hvem, hva, hvorledes, når ift kontakter, infrastruktur, ansatte, kunder etc)	
21	Hvordan endres salgsprosessen for tjenester? (hvem, hva, hvorledes, når ift kontakter, infrastruktur, ansatte, kunder etc)	

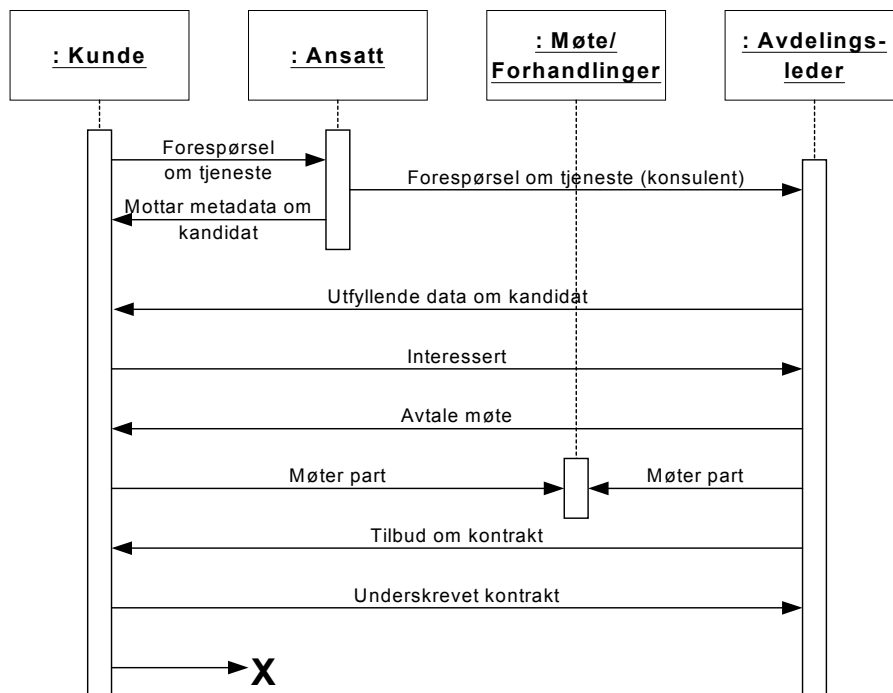
22	Hvordan er infrastrukturen for salg av tjenester?	
23	Hvordan rangerer du infrastrukturen for salg av tjenester?	
24	Hvordan rangerer du salgsprosessen i dag?	

Modellering av nå-situasjon for salg av tjenester i IKON

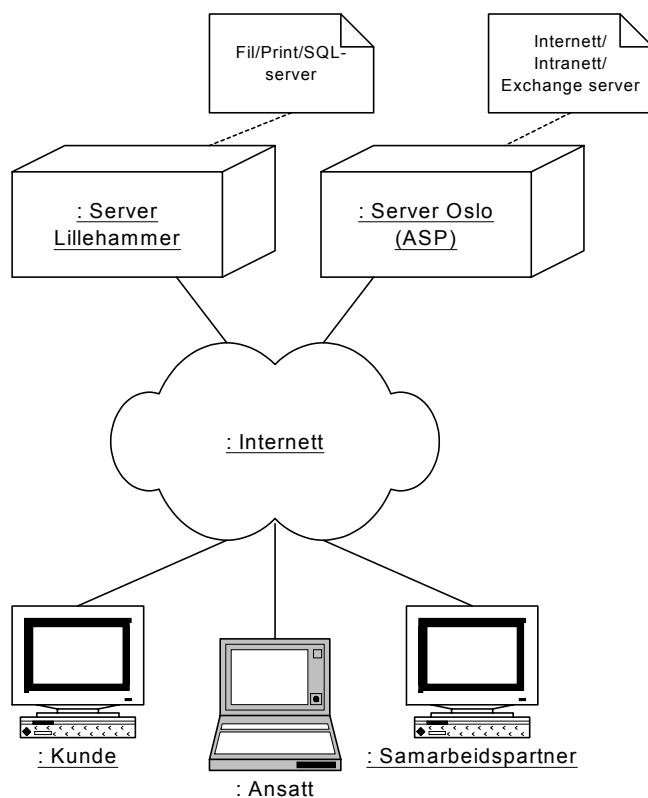
Vedlegget skisserer tre UML-skjema (Activity, Interaction og Deployment Diagram). Skjemaene viser nå-situasjonen for salg av tjenester i IKON.



Figur D-1 – Nå-situasjon for salg i IKON AS (Activity Diagram).



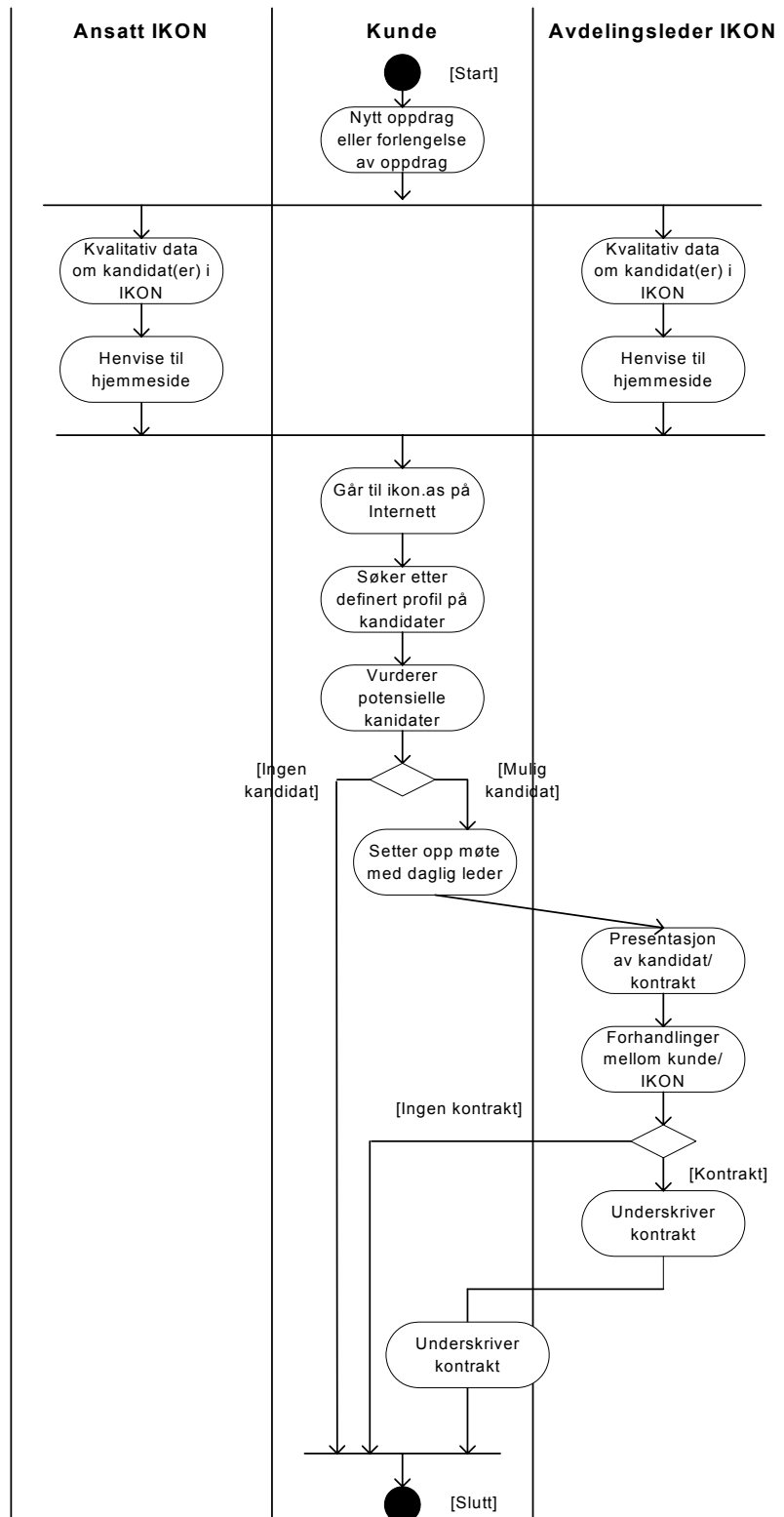
Figur D-2 – Nå-situasjon for salg i IKON AS (Sequence Diagram [Interaction Diagram]).



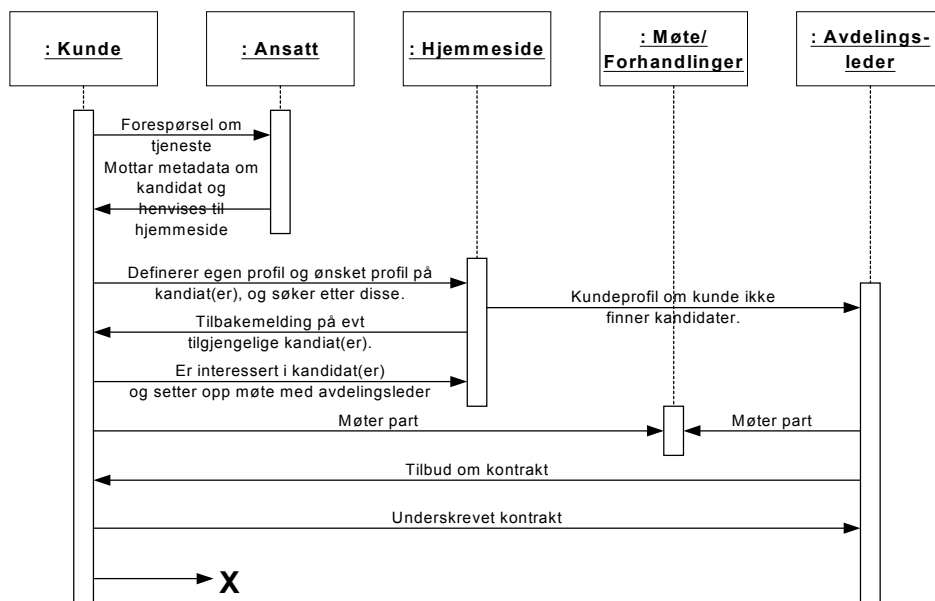
Figur D-3 – Nå-situasjon for salg i IKON AS (Deployment Diagram).

Modellering av ny løsning for salg av tjenester i IKON

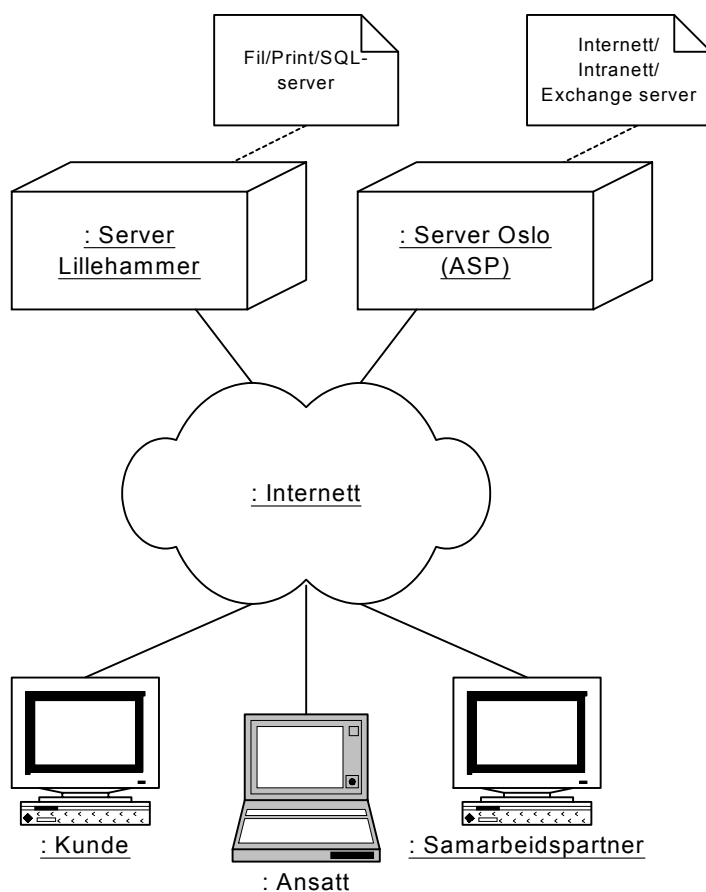
Vedlegget skisserer tre UML-skjema (Activity, Interaction og Deployment Diagram). Skjemaene viser forslag til ny løsning for salg av tjenester i IKON.



Figur E-1 – Ny løsning for salg i IKON AS (Activity Diagram).



Figur E-2 – Ny løsning for salg i IKON AS (Sequence Diagram [Interaction Diagram]).



Figur E-3 – Ny løsning for salg i IKON AS (Deployment Diagram).

Klargjøring og konfigurasjon av server, datakilder og katalogsystem

Vedlegget beskriver hvordan og hva som ble gjort i forbindelse med klargjøring og konfigurasjon av server, datakilder og katalogsystem.

1 Server

Server ble satt opp og konfigurert noe *under* spesifikasjon. I henhold til spesifikasjon anbefalte iPlanet å bruke 256 MB RAM for katalogsystem (iPlanet Directory Server). Vi brukte 164 RAM og 2 GB virtuelt minne (harddisk). Øvrige krav til maskinvare og operativsystem/applikasjoner ble ivaretatt. Tabell F-1 viser maskinvare, operativsystem og applikasjoner for server.

Tabell F-1 – Maskinvare og operativsystem/applikasjoner for server.

Maskinvare	Operativsystem/Aplikasjoner
Intel Pentium 200 MHz	Windows 2000 Server SP 2
164 MB RAM	iPlanet Directory Server 5.0
22 GB HD	MaXware DSE 5.3.17
CD-ROM/Diskettstasjon	PHP 4.0.5 – dev.

Serveren ble integrert i IKONs datanettverk. Serveren var ikke en del av produksjonssystemet for IKON. Datanettverket var tilkoblet Internett.

2 Datakilder

Datakilder i prototypen var tabeller (i en simulert database). Tabellene ble definert på bakgrunn av entiteter i datamodellen. I prototypen ble hver tabell definert og lagret som en fil i Microsoft Excel 5.0. Sammenhengen mellom tabeller og filer i Excel er skissert i tabell F-2.

IKON bruker databasen MySQL. Prosjektet kunne brukt databasen, i stedet for Excel-filer, ved implementasjon av datamodellen. Vi valgte å bruke Excel-filer som tabeller. Dette for å redusere utviklingstiden.

Tabell F-2 – Sammenheng mellom tabeller og datafiler i Microsoft Excel.

Entiteter/tabeller	Filer i Excel
Kunde	Kunde_tabell.xls
Prosjekt	Prosjekt_tabell.xls
Deltakelse	Deltakelse_tabell.xls
Person	Person_tabell.xls
Primærkompetanse	Primarkomp_tabell.xls
Sekundærkompetanse	Sekundarkomp_tabell.xls
Utdanning	Utdanning_tabell.xls
Kurs	Kurs_tabell.xls
Yrkeserfaring	Yrkeserfaring_tabell.xls
Tillitsverv	Tillitsverv_tabell.xls
Språk	Sprak_tabell.xls
Sikkerhetsklarering	Sikkerhet_tabell.xls
Interesser	Interesser_tabell.xls
Referanser	Referanse_tabell.xls

Tabell F-3 og F-4 viser to tabeller i prototypen. Tabellene inneholder attributter med tilhørende innhold. Data i tabellene er ikke relevant for IKON.

Tabell F-3 –Excel-fil med tabellen Prosjekt.

ProsjektID	KundeID	Prosjektnavn	ProsjektStartDato	ProsjektSluttDato	Kommentar	Merknad
PRO01	FOR111	FISBASIS	1999-12-13	2001-01-01	Ingen	Ingen
PRO02	TEL222	ADSL	1999-12-13	2001-01-01	Ingen	Ingen
PRO03	SIM333	GIGASET	1999-12-13	2001-01-01	Ingen	Ingen

Tabell F-4 –Excel-fil med tabellen Primærkompetanse.

PrimarID	PersonID	Kompetanse	Kommentar	Merknad
PRI01	FAA	prosjektleder	Ingen	Ingen
PRI02	SAA	prosjektleder	Ingen	Ingen
PRI03	JKV	systemutvikler	Ingen	Ingen
PRI04	JKV	programmerer	Ingen	Ingen
PRI05	GKV	drifter	Ingen	Ingen

3 Katalogsystem

Vi installerte og klargjorde Windows 2000 Server, og katalogsystemene Active Directory (AD), NDS DirXML (NDS) og iPlanet Directory Server (IDS) på samme server.

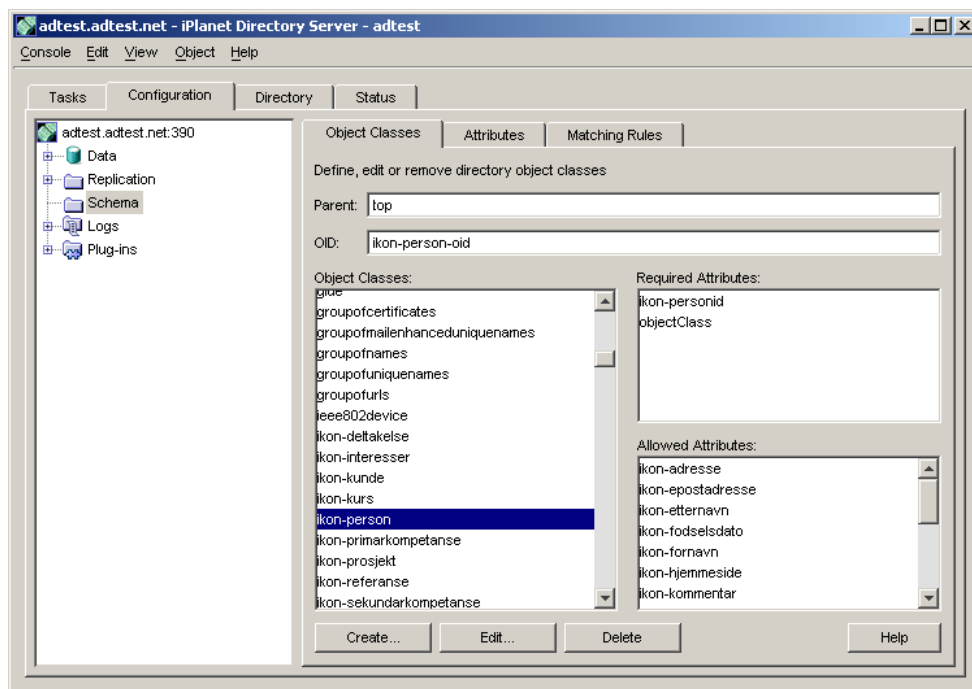
Vi brukte noe tid på å bli kjent med alle katalogsystemene, og hvordan systemene var integrert med operativsystemet. På bakgrunn av våre krav til katalogsystem valgte vi IDS. Selv om IDS var mer ressurskrevende enn f eks AD var det enklere å arbeide med. NDS fra Novell er et større og mer

komplekst katalogsystem. NDS ga oss inntrykk av å være for overdimensjonert i forhold til målsettingen for prosjektet.

Ved installasjon av katalogsystemet IDS ble systemet satt opp med parameter som innloggingsnavn for administrator (admin), administrator for katalog (directory manager), navn på katalog, IP-adresse, port-nummer etc.

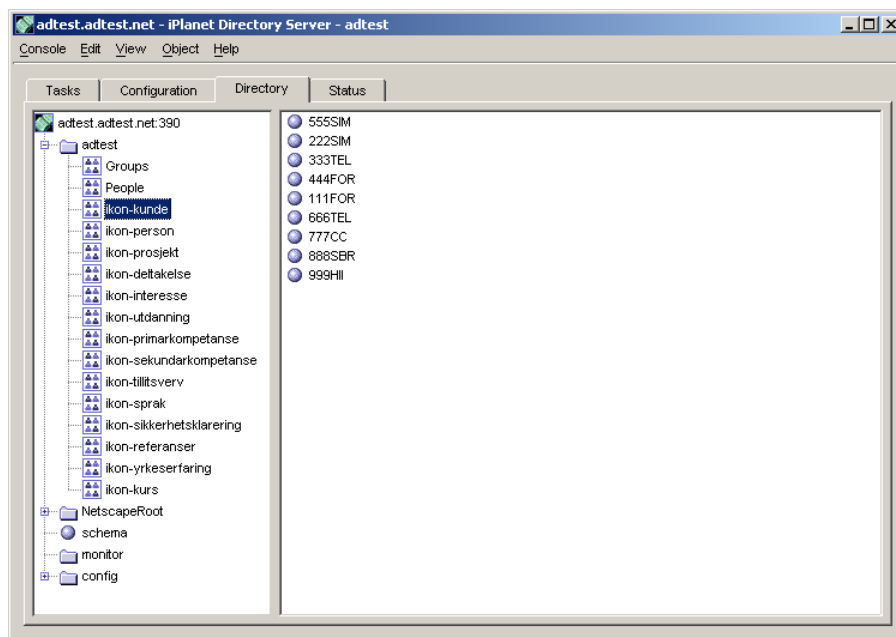
Ved installasjon ble IDS konfigurert til å starte ved oppstart av server. Ved oppstart ble DSA aktivert og gjorde katalogsystem klar til å kommunisere med LDAP-klient (og verktøy for integrasjon).

Etter at server og katalogsystem var installert og klargjort begynte vi arbeidet med å definere nødvendige antall skjema med tilhørende attributter. I prototypen tilsvarer hver tabell/excel-fil et skjema i katalogsystemet. Eksempel på skjema, med tilhørende attributter, er skissert i figur F-5 (se Objectclass, Required attributes og Allowed Attributes).



Figur F-5 – Definisjon av skjema for "ikon-person".

Hvert skjema ble deretter definert som egne grupper i katalogsystemet. Eksempel på slike grupper er angitt i figur F-6.



Figur F-6 – Objektgrupper med innhold for katalog "adtest".

Når alle skjema var definert, og koblet til ønsket katalog, var systemet klar for å lagre, slette eller endre data mot ønsket katalog. Vi brukte disse operasjonene direkte mot katalog og vha eksterne verktøy (MaXware DSE) gjennom DSA.

Tabell F-5 – Sammenheng mellom tabeller, datafiler og skjema.

Entiteter/tabeller	Datafiler i Microsoft Excel	Skjema i IDS
Kunde	Kunde_tabell.xls	Ikon-kunde
Prosjekt	Prosjekt_tabell.xls	Ikon-prosjekt
Deltakelse	Deltakelse_tabell.xls	Ikon-deltakelse
Person	Person_tabell.xls	Ikon-person
Primærkompetanse	Primarkomp_tabell.xls	Ikon-primarkompetanse
Sekundærkompetanse	Sekundarkomp_tabell.xls	Ikon-sekundarkompetanse
Utdanning	Utdanning_tabell.xls	Ikon-utdanning
Kurs	Kurs_tabell.xls	Ikon-kurs
Yrkeserfaring	Yrkeserfaring_tabell.xls	Ikon-yrkeserfaring
Tillitsverv	Tillitsverv_tabell.xls	Ikon-tillitsverv
Språk	Sprak_tabell.xls	Ikon-sprak
Sikkerhetsklarering	Sikkerhet_tabell.xls	Ikon-sikkerhet
Interesser	Interesser_tabell.xls	Ikon-interesser
Referanser	Referanse_tabell.xls	Ikon-referanse

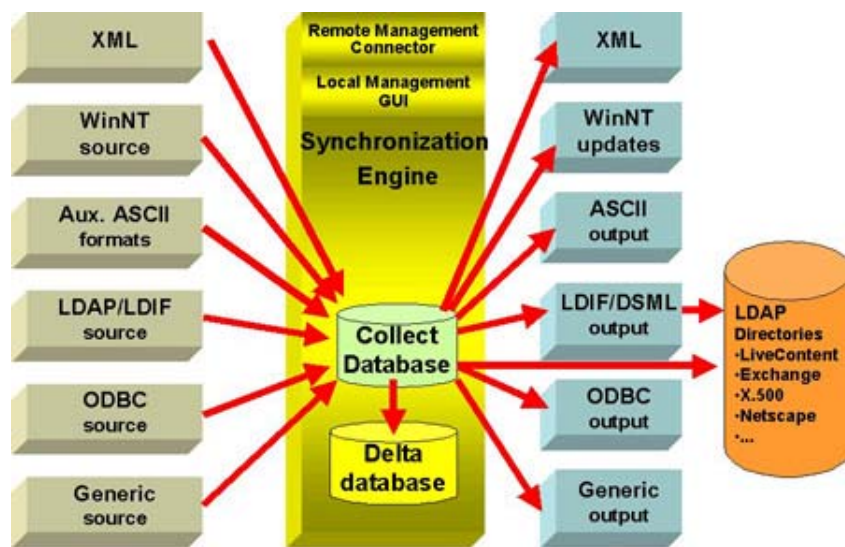
I forsøket søkte vi å skape et "en til en" forhold mellom entiteter/tabeller, datafiler og skjema. Sammenhengen mellom disse er skissert i tabell F-5. Dette forholdet gjorde utviklingsprosessen enklere, og det ble også lettere å vurdere endelig løsning.

Integrasjon datakilder og katalogsystem

Vedlegget beskriver hvordan og hva som ble utført ved koblingen av datakilder og katalogsystem.

1 Introduksjon til MaXware DSE

DSE [MAX01b] er et verktøy som gjør det mulig å synkronisere kataloger (X.500 eller LDAP) med andre datakilder. DSE har et eget verktøy for å lage oppgaver for synkronisering. Ferdige oppgaver kan vi senere kjøre periodisk. Vi kan også bruke DSE til å synkronisere data i filer og databaser med ulike formater (se figur G-1).



Figur G-1 – Dataformater DSE kan synkronisere [MAX01a].

DSE synkroniserer data ved å dele oppgaven (jobb) opp i flere mindre oppgaver. Hver jobb er definert med flere såkalte pass.

2 Sammenkobling datakilder og katalog

Ved synkronisering av data mellom Excel-filer (tabeller) og katalog definerte vi to jobber. Jobb 1 var:

ODBC-source -> Collect Database (med delta) -> LDAP Directory

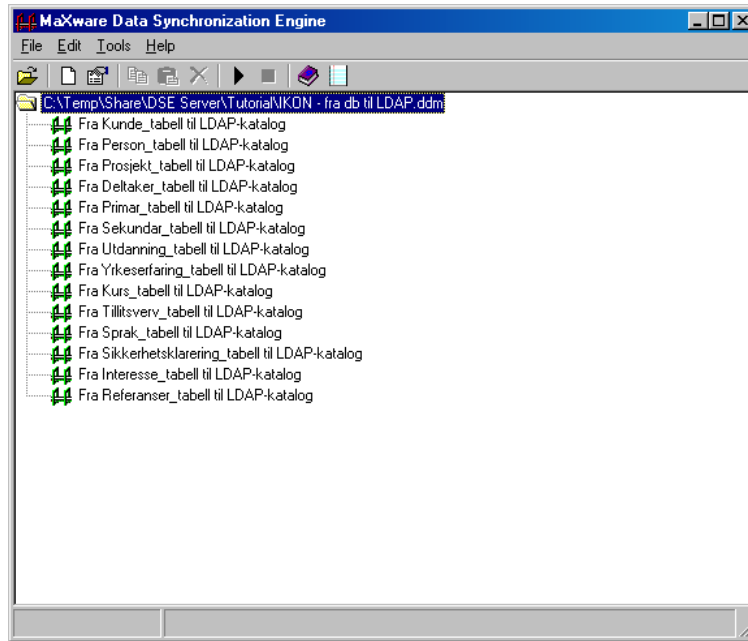
Jobb 2 var:

LDAP Directory -> Collect Database (med delta)-> ODBC-source

2.1 Definisjon av jobb 1

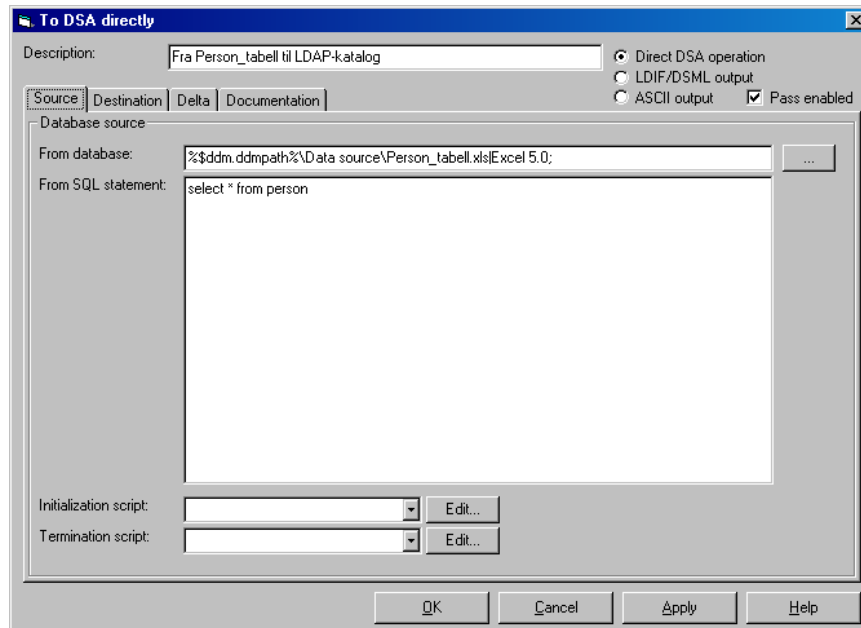
Ved overføring av data fra datakilder til katalog definerte vi like mange pass som tabeller og skjema i katalog. Det er ikke et krav å gjøre det på denne måten, men en mulighet vi benyttet.

Jobb 1 bestod av i alt pass 14 pass (se figur G-2).



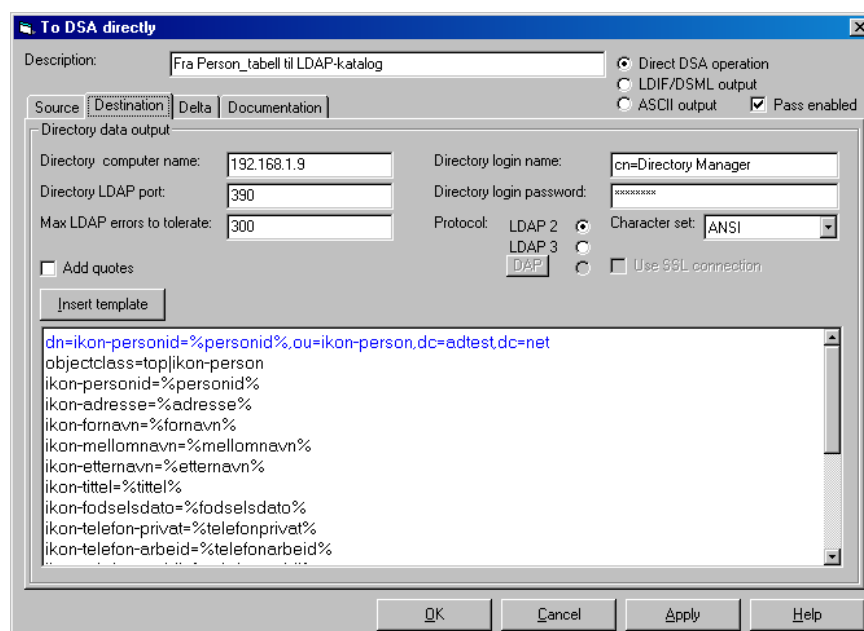
Figur G-2 – Jobb 1 i DSE (fra ODBC-kilde direkte til LDAP-katalog).

Vi konfigurerte alle pass (fra LDAP-katalog til ODBC-kilde) gjennom dialogboks To DSA Directly. Et pass består av flere arkfaner (Source, Destination og Delta). I arkfanen Source definerte vi database med SQL-spørring (se figur G-3). Legg merke til at From database (eller tabell) er en flat fil (Microsoft Excel).



Figur G-3 – Pass fra Person_tabell til LDAP-katalog (arkfane Source).

I arkfane Destination (se figur G-4) definerte vi hvor vi ønsket å flytte data til. Her satte vi opp katalognavn (IP-adresse for katalogserver), LDAP-port, login/passord navn for administrator (katalog) og versjon av LDAP-protokoll.



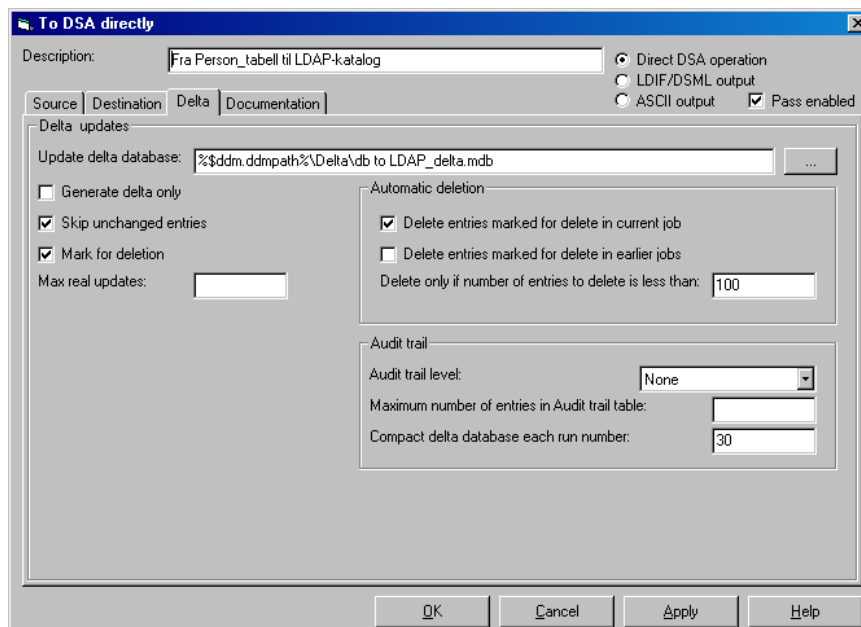
Figur G-4 – Pass fra Person_tabell til LDAP-katalog (arkfane Destination).

I tillegg ble variabler i datakilde koblet til skjema-variabler i katalog (se figur G-4). Figur G-5 gir full oversikt over alle koblinger mellom variablene i datakilde og katalog.

```
dn=ikon-personid=%personid%,ou=ikon-person,dc=adtest,dc=net
objectclass=top|ikon-person
ikon-personid=%personid%
ikon-adresse=%adresse%
ikon-fornavn=%fornavn%
ikon-mellomnavn=%mellomnavn%
ikon-etternavn=%etternavn%
ikon-tittel=%tittel%
ikon-fodselsdato=%fodselsdato%
ikon-telefon-privat=%telefonprivat%
ikon-telefon-arbeid=%telefonarbeid%
ikon-telefon-mobil=%telefonmobil%
ikon-selskap=%selskap%
ikon-merknad=%merknad%
ikon-postnr=%postnr%
ikon-poststed=%poststed%
ikon-telefaks=%telefaks%
ikon-epostadresse=%epostadresse%
ikon-hjemmeside=%hjemmeside%
ikon-kommentar=%kommentar%
```

Figur G-5 – Kobling mellom variabler i datakilde og katalog (fra arkfane Destination).

I arkfane Delta (se figur G-6) definerte vi deltaspesifikasjonen for passet. Deltafunksjonen gjør det mulig å synkronisere kun endringer fra siste synkronisering.



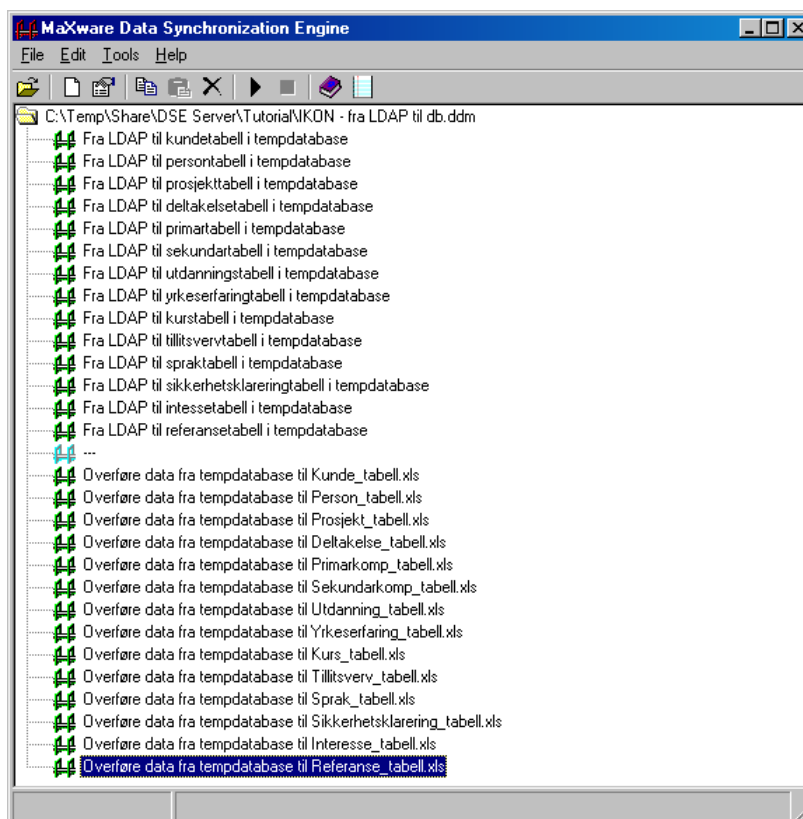
Figur G-6 – Pass fra Person_tabell til LDAP-katalog (arkfane Delta).

Deltafunksjonen bruker en egen intern database. Vi brukte denne databasen for alle pass i "Jobb 1".

2.2 Definisjon av jobb 2

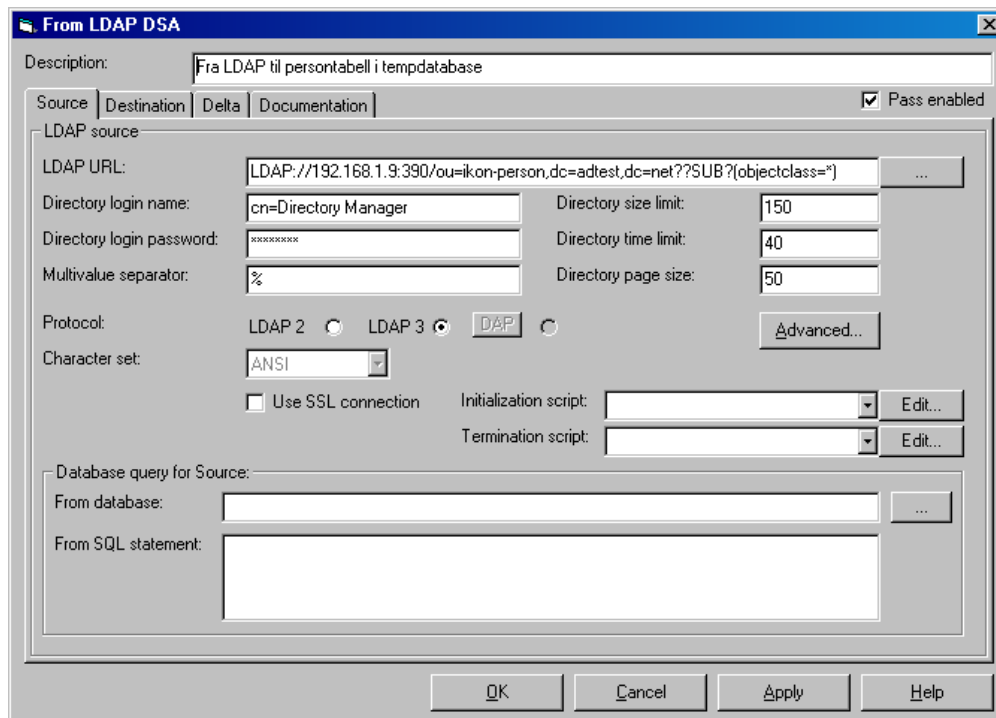
Jobb 2 bestod av i alt 28 pass (se figur G-7). Jobben ble delt i to grupper. Første gruppe overførte data fra katalog til temporær database, mens den andre gruppen overførte data fra temporær database til datakilder (Excel-filer).

Vi brukte typen From LDAP DSA for første gruppe pass og To ODBC database for andre gruppe pass.



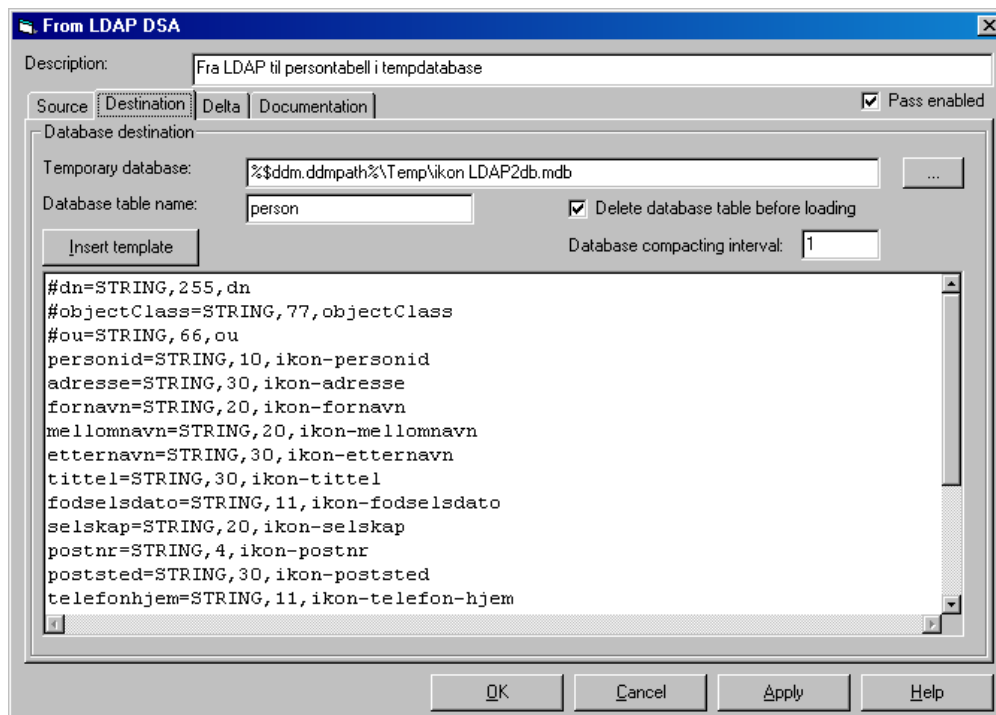
Figur G-7 – Jobb 2 i DSE (fra LDAP-katalog til ODBC-kilde).

Pass av typen From LDAP DSE ble også definert vha dialogboks med flere arkfaner (se figur G-8). I arkfanen for Source ble kilden for LDAP definert. LDAP URL er forskjellig skjema til skjema i katalog.



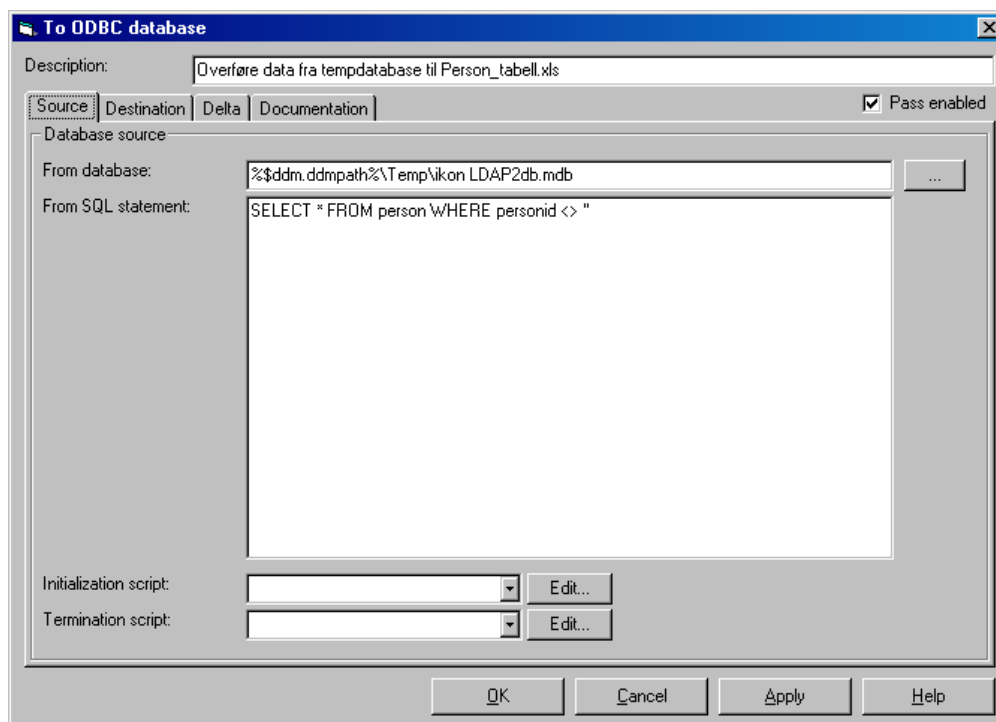
Figur G-8 – Pass From LDAP DSA (arkfane Source).

I arkfane Source definerte vi LDAP URL med tilhørende login-navn og login-passord. I arkfane Destination (se figur G-9) spesifiserte temporær database. I tillegg ble kobling mellom variabler i katalogskjema og tabeller i database spesifisert.



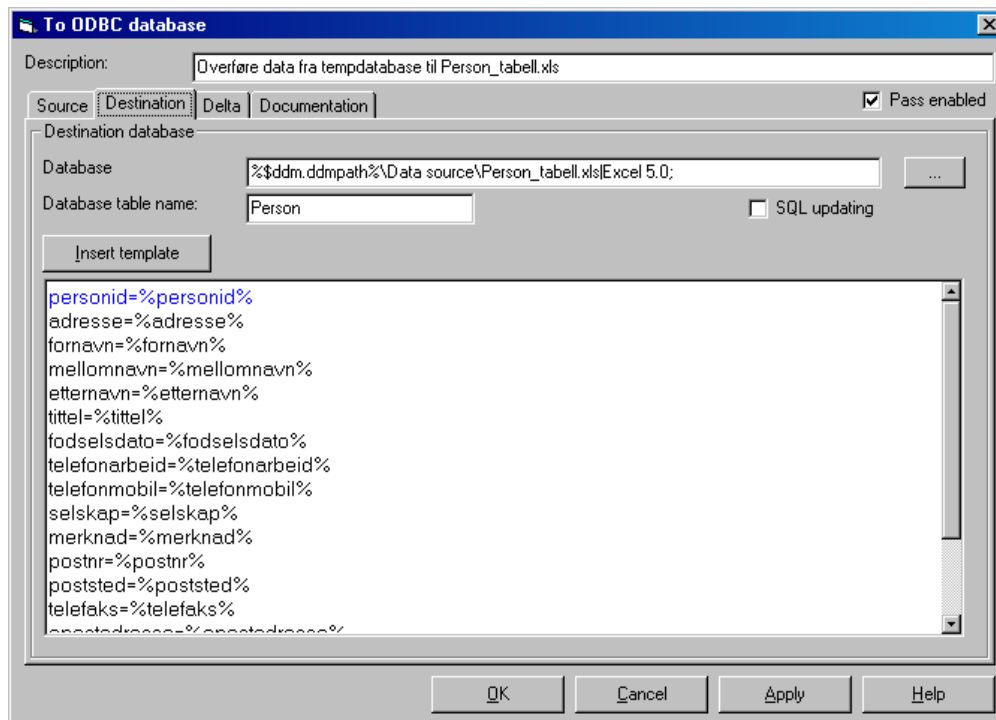
Figur G-9 – Pass From LDAP DSA (arkfane Destination).

Pass av typen To ODBC database ble også definert vha dialogboks med flere arkfaner (se figur G-10). I arkfanen for Source ble temporær database definert med tilhørende SQL-spørring.



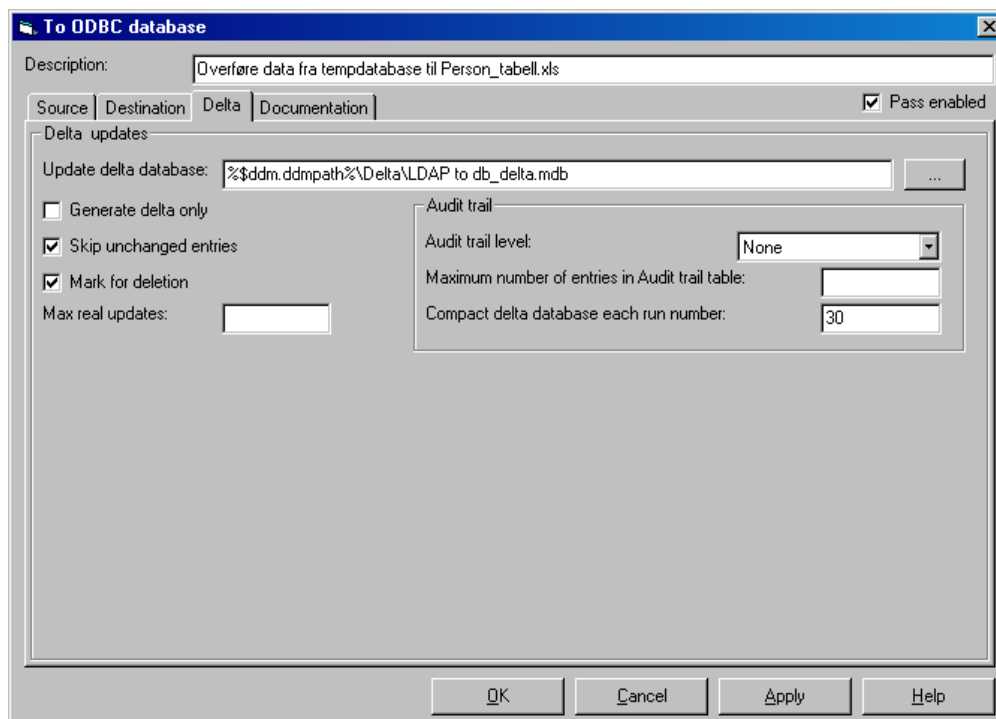
Figur G-10 – Pass To ODBC database (arkfane Source).

I arkfanen for Destination ble opprinnelig datakilde (Microsoft Excel-filer) definert. Arkfanen spesifiserer database, tabell og kobling mellom felter i temporær database og felter i datakilde.



Figur G-11 – Pass To ODBC database (arkfane Destination).

Vi brukte også deltafunksjonen i jobb 2. Deltafunksjonen ble brukt i passene for To ODBC database (se figur G-12), og ikke passene for From LDAP DSA.



Figur G-12 – Pass To ODBC database (arkfane Delta).

3 Automatisk kjøring av jobber

Vi kan konfigurere DSE slik at jobb 1 og 2 blir kjørt automatisk. Dette kan vi gjøre ved å koble kjørbare filer (bat-filer), som DSE genererer, til rutinejobber som blir styrt av operativsystemet.

Eksempel på en slik kommando, som DSE genererte fra vår prototyp, er:

```
"C:\Program files\DSE\MDDM32.exe" C:\Program files\DSE\Tutorial\IKON - fra db  
til LDAP.ddm
```

Prototypen ble ikke konfigurert til å kjøre jobbene automatisk. I prototypen ble alle jobbene kjørt manuelt.

Integrasjon web og katalogsystem

Vedlegget beskriver hvordan og hva som ble utført ved integrasjon web og katalogsystem.

1 Introduksjon til PHP

PHP er et verktøy som gjør det mulig å lage dynamiske web-sider. PHP er et tekstbasert språk, og språket har fokus på utvikling av HTML-baserte løsninger. PHP er forholdsvis lik språket Perl.

Vi installerte og klargjorde PHP på samme server som datakilder, katalogsystem og verktøy for integrasjon datakilder og katalog (MaXware DSE). Under installasjonen detekterte PHP vår server som web-server, og koblet seg til denne. Koblingen gjorde det blant annet mulig å kjøre PHP-script direkte i nettleser.

2 Sammenkobling web og katalog

Vi tok utgangspunkt i modelleringen av ny løsning for salg av tjenester i IKON (se vedlegg E). Scriptene støttet ikke modellene fullt ut, men de viste oss at interaksjonen fungerte.

Vi lagde to script for spørring etter gitt primær-kompetanse hos konsulenter/rådgivere i selskapet. Det første scriptet angir hva vi skal spørre etter (se figur H-1 og H-2).

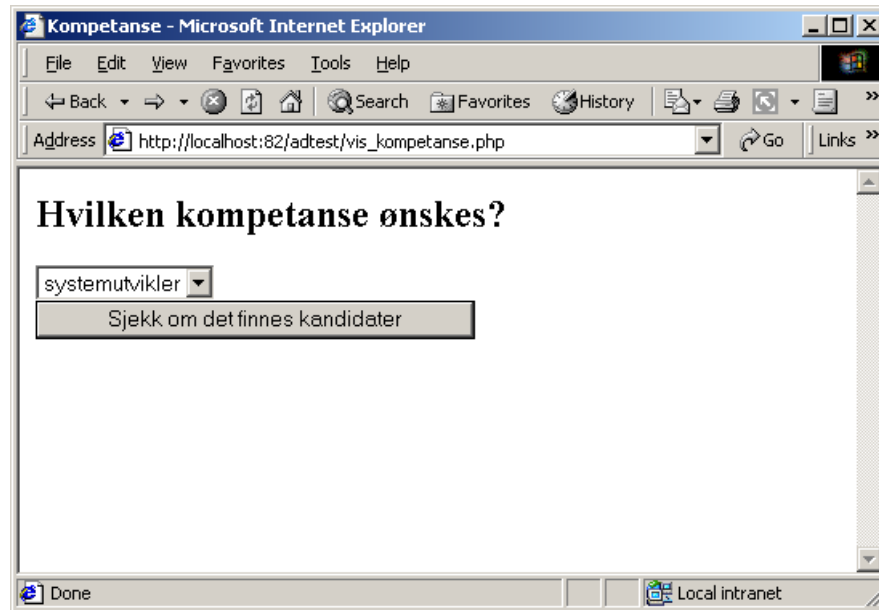
```
<?php
?>
<title>Kompetanse</title>
<h2>Hvilken kompetanse ønskes?</h2>
<?
$basedn=$dn;
$basedn="ou=ikon-primarkompetanse,dc=adtest,dc=net";
$filter="ikon-primarid=*";
$ds=ldap_connect("localhost",390); // must be a valid LDAP server!
if ($ds)
{
    $r=ldap_bind($ds);
    $sr = ldap_search($ds,$basedn,$filter);
    if (!$sr)
    {
        printf("LDAP-Errno: %s\n", ldap_errno($ds));
        printf("LDAP-Error: %s\n", ldap_error($ds));
        die("Argh!\n");
    }
}
```

```
$info = ldap_get_entries($ds, $sr);
# echo "Data for " . $info["count"] . " items returned:<br>";
echo "<form action='finn_kandidater.php' method=post>\n";
echo "<select name=\"ikon-personid\">\n";

for ($k=0; $k<$info["count"]; $k++)
{
    $basedn=$info[$k]["dn"];
    $filter="ikon-primarid=*";
    $sr2 = ldap_search($ds,$basedn,$filter);
    $entry = ldap_first_entry($ds,$sr2);
    for ($i=0; $i<ldap_count_entries($ds,$sr2); $i++)
    {
        $attrs = ldap_get_attributes($ds, $entry);
        echo "<option value=\"\". $attrs["ikon-personid"][0] . "\"
>";
        echo $attrs["ikon-kompetanse"][0] . "</option>\n";
        $entry = ldap_next_entry($ds,$sr2);
    }
}
echo "</select>\n<br>\n";
echo "<input type=submit value=\"Sjekk om det finnes kandidater\">";
echo "</form>";
ldap_close($ds);
}
else
{
    echo "Unable to connect to LDAP server";
}
?>
```

Figur H-1 - Kildekode for scriptet vis_kompetanse.php.

Kildekoden i figur H-1 gir skjermbildet i figur H-2. Her angir potensiell kunde hvilken primærkompetanse han/hun søker.



Figur H-2 - Skjermbilde for scriptet vis_kompetanse.php i nettleser.

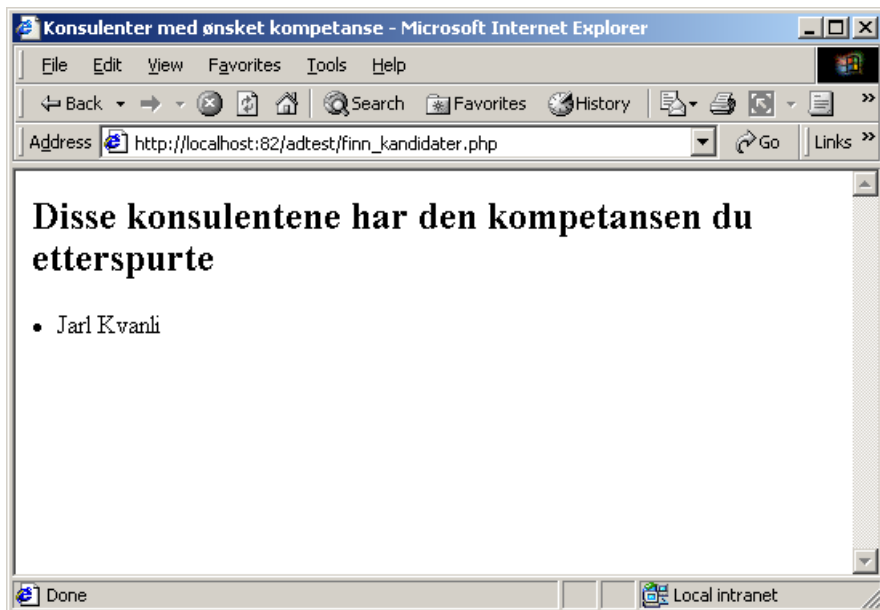
Den andre scriptet (se figur H-3 og H-4) skriver ut verdier på bakgrunn av hva potensiell kunde søker etter.

```
<?php
/*
phpinfo();
*/
?>
<title>Konsulenter med ønsket kompetanse</title>
<h2>Disse konsulentene har den kompetansen du etterspurte</h2>
<?
$basedn="ou=ikon-person,dc=adtest,dc=net";
$filter = "ikon-personid=" . $_HTTP_POST_VARS["ikon-personid"];
$ds=ldap_connect("localhost",390); // must be a valid LDAP server!
if ($ds)
{
    $r=ldap_bind($ds);
    $sr = ldap_search($ds,$basedn,$filter);
    if (!$sr)
    {
        printf("LDAP-Errno: %s\n", ldap_errno($ds));
        printf("LDAP-Error: %s\n", ldap_error($ds));
        die("Argh!\n");
    }
    $info = ldap_get_entries($ds, $sr);
    $entry = ldap_first_entry($ds,$sr);
```

```
for ($i=0; $i<ldap_count_entries($ds,$sr); $i++)
{
    $attrs = ldap_get_attributes($ds, $entry);
    echo "<li>" . $attrs["ikon-fornavn"][0] . " " . $attrs["ikon-
etternavn"][0] . "</li\n>";
    $entry = ldap_next_entry($ds,$sr);
}
ldap_close($ds);
}
else
{
    echo "Unable to connect to LDAP server";
}
?>
```

Figur H-3 - Kildekode for scriptet finn_kandidater.php.

Ved søk etter "systemutvikler" ble resultatet som vist i figur H-4.



Figur H-4 – Skjermbilde for scriptet finn_kandidater.php i nettleser (eksempel).

Vi brukte ingen spesielle mekanismer for autentisering og sikkerhet ved oppslag i katalog.

Utvikling av LDAP- og SQL-operasjoner i PHP

Vedlegget presenterer noen forskjeller ved utvikling av LDAP- og SQL-operasjoner i PHP.

1 Innledning

PHP gir oss mulighet til å utvikle LDAP- og SQL-spørringer mot henholdsvis katalog- og databasesystem. Ved konstruksjon av slike spørringer må vi bruke ulik syntaks i kildekoden. Forskjellene mellom LDAP- og SQL-operasjoner påvirker også hvilken funksjonalitet vi kan integrere i kildekoden.

Vedlegget bør ses i sammenheng med vedlegg H og spesielt kildekode for LDAP-spørringer mot katalogsystem.

2 SQL-spørring mot database

Nedenfor følger et eksempel (se figur I-1) på PHP-kode. Koden spesifiserer SQL-spørring mot en database. Oppgitt database er en tenkt database i MySQL. Databasen er normalisert slik at vi kan ha mange forekomster i IKON-person med samme kompetanse. Det vil si strukturen i databasen er noe annerledes i forhold til strukturen i vår prototyp.

```
<?php
// Oppkobling mot database
$link = mysql_connect("mysql_host","user","passwd");

//Valg av database
mysql_select_db("database");

//Vi ønsker å hente ut personopplysninger, sammen med kompetanseopplysninger,
//fra en valgt kompetanseid for alle personer som har kompetansen.
$query = "SELECT IP.FIRSTNAME, IP.LASTNAME, IK1.KOMPETANSENAVN ";
$query .= "FROM IKON_PERSON IP, IKON_KOMPETANSE IK1";
$query .= "WHERE IK.PK_ID = '" . $kompetanseid . "' ";
$query .= "AND IP.FK_KOMPETANSE_ID = IK1.PK_ID";

$result = mysql_query($query);
print "<table>\n";
while($line = mysql_fetch_array($result))
{
    print "\t<tr>\n";
    while(list($col_name, $col_value) = each($line))
    {
        print "\t\t<td>$col_value</td>\n";
    }
    print "\t</tr>\n";
}
print "</table>\n";
mysql_close($link);
?>
```

Figur I-1 – Eksempel på SQL-spørring i PHP.

Koden forutsetter at vi har valgt en gitt kompetanse, og den skal liste ut alle personer som har valgt kompetanse.

Kildekoden i figur I-1 er bygd opp slik:

- først blir program koblet til database, og så velger vi ønsket database (tabeller).
- deretter blir søket bygd opp
- så blir søket gjennomført og skrevet ut.

Eksemplet viser at det er forholdsvis enkelt å lage SQL-spørring med PHP (se spesielt syntaks for Query).

3 Erfaring

SQL gir oss bedre mulighet til å trekke ut data fra flere tabeller samtidig. Dette ser vi på som den største forskjellene mellom SQL og LDAP. Om vi søker å få samme data fra katalogsystemer med LDAP, som fra databasesystemer med SQL, må vi konstruere flere uttrekk og løkker. Dette kan virke både unødvendig og ressurskrevende.

Databaser har innbygd støtte for håndtering av ulike typer data (for eksempel dato). Utrekk av data fra alle rader mellom to datoer vil være enkelt fra databaser. Vi er noe usikker på hvordan dette kan best bli ivaretatt med LDAP, men det virker mer tungvint enn med SQL.

I relasjon til prosjektets prototyp kan vi på en *enkel* måte liste ut personer fra en *database*, med angitt primærkompetanse, som ikke er med i noe prosjekt i en angitt periode. I tillegg kan vi spesifisere at personen skal ha sikkerhetsklarering "hemmelig". Vi tror en slik spørring vil, med forbehold, være mer krevende å utvikle og teste med LDAP.
