

Erik Vold

## Åpen Kanal

Ytringsrett og distribuert samhandling

Trondheim, April 2006

Norges teknisk-naturvitenskapelige universitet  
Fakultet for Informasjonsteknologi, matematikk og  
elektroteknikk  
Institutt for telematikk

Masteroppgave

Studieprogram: Master i kommunikasjonsteknologi

Hovedveileder: Leif Arne Rønningen, ITEM

Biveileder(e): Andresen Sigmund, Fjernsynsteknikk (TVF),  
Høgskolen i Lillehammer



## MASTEROPPGAVE

Kandidatens navn:	Erik Vold
Emne:	Kommunikasjonsteknologi: Menneske, samfunn og media
Oppgavens tittel:	<b>Åpen kanal – Ytringsrett og distribuert samhandling</b>
Oppgavens tekst:	<p>Oppgaven skal ta for seg juridiske så vel som tekniske utfordringer med en Åpen kanal i dag. Som en modell for å sikre ytringsrett i det norske fjernsynsmediet er det ønskelig å benytte tyske åpne kanaler. Her er deltakerne produsenter under eget redaktøransvar og deres praksis kan danne grunnlag for en juridisk utredning forenelig med Stortingets intensjoner om kanalen i Norge.</p> <p>Realiseringen av vår nasjons første public access TV kanal, Åpen kanal, har vært motivasjonen for å skrive om nett-kontribusjon tidligere (prosjekt i 9.semester). Systemet som ble utviklet har nå blitt testet over lengre tid og det viste seg at filoplasting var et hinder for mange av deltakerne. Det vil derfor bli utviklet en enkel opplastingsklient der brukeren ikke trenger forholde seg til bakenforliggende prosesser.</p> <p>I samarbeid med Høgskolen i Lillehammer og Gjøvik ønsker jeg også å drøfte muligheter for live kontribusjon med distribuert samhandling der målet er at deltakere og seere vil oppleve en nærhet til andre deltakere. Her vil TiDE-SVR prosjektet ved HiL samt egne eksperimenter danne grunnlag for redegjørelsen.</p> <p>Fra et teknologisk synspunkt vil jeg redegjøre for objektbaserte produksjonsteknikker og hvordan dette kan gi mer fleksible distribusjonsmuligheter samt overvinne begrensninger i kontribusjonsnettene. Det vil også bli drøftet hvordan teknikken kan utnyttes i den innholdsmessige utformingen der man med enkle midler kan realisere ikke-sann og sann 3D med god perseptuell kvalitet. 3D-videokonferanse vil bli realisert under TiDE-SVR som en pilot for videre utvikling av telepresence.</p>
Besvarelsen leveres innen:	10. april 2006
Besvarelsen levert:	10. april 2006
Utført ved:	Institutt for telematikk
Veileder:	Leif Arne Rønningen
	Trondheim, 10. april 2006

Leif Arne Rønningen, faglærer

## Forord

Denne oppgaven er skrevet med bakgrunn i min og mange andres frivillige innsats. Det er i stor grad medlemmene av foreningen *Åpen kanal* som har lagt premissene for kanalen og med deres støtte gjennom flere år har de gjort det mulig å bygge opp relasjoner som nå endelig gjør at kanalen kan realiseres både på nett og TV. Om kanalen blir åpen også på TV avhenger kun av politisk vilje. Det er ingen tvil om at man bør være veldig takknemmelig for tålmodigheten enkelte miljøer har fremvist. For selv en så innlysende bra sak som et demokratisk redskap har vært vanskelig å få støtte til kanskje nettopp fordi ideen ikke har vært fremmet tidligere. Det finnes ingen fond for dette formålet. De som har stilt opp har dermed gjort det med frivillig innsats. Spesielt tenker jeg da på Uninett og NTNU som ubetinget har støttet opp om ideen og stilt både mannskap og utstyr til vår disposisjon. Og ikke minst enkeltpersoner som vår utvikler Roy Galaasen og primus motor samt styreleder Ola Tellesbø som har viet all sin tid. Uten denne støtten ville *Åpen kanal* vært en god idé og sannsynligvis ligget i en papirhaug hos enkelte politikere. Det at vi nå har det som trengs for å vise hva konseptet er godt for, gjør etter min mening det hele til et tidsspørsmål. Jo flere som vil ta dette i bruk, jo flere vil innse at det er en opplagt folkerett. Vi er fra før kjent med begrepet ytringsfrihet, men hva innebærer egentlig det? Jo, at man her krav på ikke å bli sensurert. Det er en negativ rett til å ytre seg. Hva med den positive retten, retten til å bli hørt - ytringsretten? Har vi egentlig rett på tilgangen til mediet? I Norge har vi ikke ytringsrett i dag. Infrastrukturen legger ikke til rette for det. Hvilke TV-kanaler er interessert i å høre hva du har å si – hvis du ikke er en kjent person eller på annen måte kan trekke mange seere? Kanalene er ikke der for at du skal nå din målgruppe, de er der for å sikre seertall, sikre omsetning. En åpen kanal trenger vi, ikke i første omgang for seeren sin skyld, men for de som har noe på hjertet. Og kanskje det etter hvert tjener seeren også. For det er som Jostein Gårder sa det: *"Man må begripe for å bli grepet"*. Vi lever i informasjonsbulimiens tidsalder. Alt skal presenteres på en effektiv måte. Alt skal være formet som overskrifter i kampen om å bli hørt. Alt skal være så langt unna virkeligheten som mulig. For den er vel ikke spennende? Men kanskje er den nettopp det, hvis man gir den tid. For hvor ble det av den enkle samtalen? Der man ser hva man er enig i og hva man strides om. Der man får tid til å drøfte og fordype seg. Der man får tid til å begripe. Jeg, og mange med meg, ser frem til å bli grepet av virkeligheten!

## Sammendrag

Foreningen Åpen kanal er Norges første og eneste initiativtaker til å sikre allmennheten en ytringsrett i fjernsynet. Ytringsrett er en reell ytringsfrihet der man blir gitt tilgang til en kanal for fri ytring. For at allmennheten skal ha en ytringsrett i TV-mediet er det ikke tilstrekkelig å kun sørge for fravær av sensur slik ytringsfriheten praktiseres i dag, det må også sørges for at TV-kanalen eksisterer og at den er åpen. En åpen kanal kan ikke ekskludere noen verken på bakgrunn av bosted, økonomi eller kompetanse.

Denne masteroppgaven er en realisering av Åpen kanal fra ende til ende. I praktiske forsøk med distribuert samhandling og telepresence vil jeg argumentere for hvordan man kan skape en nærhet mellom deltakerne til tross for geografiske avstander. Jeg vil også avdekke hvilke tiltak som er gjort og må gjøres for at organisasjoner og enkeltindivider skal komme frem med sin mening til tross for en begrenset økonomi. Jeg har realisert en sentral infrastruktur der kontribusjon og publisering av innhold kan skje med utstyr de fleste har tilgang på. Det er viktig at tjenesten blir så brukervennlig at deltakere uten IT-kompetanse kan ytre seg. Infrastrukturen er den første i sitt slag og byr på mange utfordringer både organisatorisk og teknisk. Den har i skrivende stund blitt prøvd ut av deltakerne i Åpen kanal og deltakerne fant at det som gjør Åpen kanal så unik er også det som gjør den så vanskelig i bruk. For at kostnadene skal holdes nede i en kanal der medlemstallet forventes å stige raskt og produsentene vil være spredd over det ganske land, er hele infrastrukturen nettbasert. I den første implementasjonen av systemet måtte deltakerne laste opp innslagene til den sentrale serveren med en FTP-klient. Her måtte de forholde seg til IP-adresse, brukernavn og passord samt et grensesnitt de færreste var kjent med. Dette har begrenset tilfanget av innhold til kanalen.

Det er besluttet at Åpen kanal skal sende på det nye digitale bakkenettet. I forkant av sendingen er kanalen avhengig av å ha en viss størrelse på arkivet for å sikre severdighet og kontinuitet. I samråd med foreningen (der jeg fungerer som daglig leder), ble det derfor prioritert å utvikle et brukervennlig opplastingssystem i denne oppgaven. Utviklingen har vært vellykket, men det er et stort potensiale for forbedring, spesielt med tanke på feilhåndtering og redundans. Pålitelighet og ytelse vil bli drøftet av andre master studenter ved NTNU. Det er også ønskelig med en videreutvikling av TV-avviklingssystemet som ble testet ut i denne oppgaven. Ideelt sett bør all kommunikasjon med systemet foregå via et felles Web-basert grensesnitt der brukeren ikke trenger forholde seg til bakenforliggende prosesser. Det er svært lite utvikling som gjenstår før vi har fått til det.



En større utfordring er kostnaden ved sending på TV nettet. Åpen kanal er avhengig av at distributør blir pålagt formidlingsplikt for å være åpen. Stortingets intensjoner med kanalen er ikke i overensstemmelse med departementets krav til distributøren NTV. Det er kommersielle vilkår som råder og skal KKD "legge til rette for at Åpen kanal blir en realitet" må departementet bruke de erfaringer som finnes fra andre land. Der man har realisert Public access konsepter har, man også pålagt formidlingsplikt av den enkle grunn at kanalen skal være åpen og ikke legge opp til at deltakerne må drive kommersiell virksomhet.

At departementet er avventende i forhold til realiseringen av Åpen kanal kan tilskrives at konseptet er nytt i Norge og at de frykter medieforvaltningen vil få en umulig jobb dersom alle slipper til. I samarbeid med foreningen Åpen kanal er en redaktørmodell utredet på bakgrunn av erfaringer man har gjort i de tyske åpne kanalene gjennom 20 år. En deltakerstyrt kanal der produsentene har selvstendig redaktøransvar fungerer godt der. For at vi skal få vist at dette også vil gjelde i Norge er det ønskelig med en pilotdrift av Åpen kanal. I denne oppgaven prosjekteres et år med utprøving under forutsetning av statlig støtte. Det er å håpe at Åpen kanal, gitt en slik mulighet, kan bidra til ikke bare å bevise verdien av public access men videreutvikle konseptet. Åpen kanal i Norge har en unik tilgang på ressurser for distribuert samhandling og utprøving av nyskapende videoteknikk. Med eksperimenter ved Høgskolen i Lillehammer og kunnskap ervervet gjennom studiet ved NTNU ønsker jeg her å belyse verdien av kommende teknologi i en distribuert samhandling. Forsøkene har vist at objektbasert videoproduksjon gir større fleksibilitet i komposisjonen av delte virtuelle rom samt at det kan løse utfordringer med linjekapasitet i samhandlingsnettverket.

Åpen kanal er til for deltakerne. Det er derfor viktig at man klarer å skape debattarenaer der deltakerne føler en nærhet til hverandre til tross for den geografiske avstanden i mellom dem. Teknikker for å oppnå blikkontakt blir i oppgaven prøvd ut i praksis. Hvordan miljøet ellers påvirker følelsen av nærhet blir også studert. Avslutningsvis blir teknikker for å realisere 3-dimensjonale konferanser realisert og dybdesyn drøftet i forhold til verdien av nærhet. Et permanent system for distribuert samhandling blir ikke implementert i denne oppgaven. Realiseringen av virtuelle rom i sanntid krever mye prosesseringskraft og må i dag utføres på kostbare hardware-enheter. Men man er svært nær å få dette til på konvensjonelle PC-er (Baker, 2003). Det regnes derfor som hensiktsmessig for Åpen kanal å høste konseptuelle erfaringer på lånt utstyr (v/HiL og HiG) frem til teknologien blir rimelig og tilgjengelig for alle.

## Innholdsfortegnelse

<b>FORORD</b> .....	<b>II</b>
<b>SAMMENDRAG</b> .....	<b>III</b>
<b>INNHOLDSFORTEGNELSE</b> .....	<b>V</b>
<b>FIGURLISTE</b> .....	<b>XI</b>
<b>TABELLISTE</b> .....	<b>XIV</b>
<b>FORKORTELSER</b> .....	<b>XV</b>
<b>1 INNLEDNING</b> .....	<b>16</b>
1.1 AVGRENSNINGER .....	17
1.2 OPPBYGNING .....	19
<b>2 HVA ER EN ÅPEN KANAL</b> .....	<b>21</b>
<b>3 ÅPEN KANAL I NORGE</b> .....	<b>23</b>
<b>4 POLITISK STØTTE</b> .....	<b>26</b>
<b>5 ORGANISATORISKE PREMISER</b> .....	<b>29</b>
5.1 EN OPPSKRIFT FOR FRAMTIDENS FOLKESTYRE .....	30
5.1.1 Alle skal kunne nå alle .....	31
5.1.2 Åpent og aktuelt .....	33
5.1.3 Autonomi .....	34
5.2 FORDELING OG OMFORDELING AV SENDEFLATEN .....	35
<b>6 TEKNISK STØTTE</b> .....	<b>37</b>
<b>7 PROSJEKTERING AV ÅPEN KANAL</b> .....	<b>43</b>
7.1 KONSEPTET ÅPEN KANAL .....	44
7.2 LOKALISERING .....	46
7.3 OPPGRADERING AV EKSISTERENDE SYSTEM .....	47
7.4 ETABLERING AV SIKRE RUTINER FOR REGISTRERING AV REDAKTØRER .....	49
7.5 KVALITETSSIKRING AV DRIFTEN .....	50
7.6 ETABLERING AV MEDIA LITERACY FOR BARN OG UNGDOM .....	51
7.7 FLYTTING AV PRODUKSJONSLINJEN OVER TIL SATELLITT/FJERNSYN: EN MULIGHETSSTUDIE .....	51
7.8 DISTRIBUERT SAMHANDLING/MØTEPLASSER: EN MULIGHETSSTUDIE .....	53
7.8.1 Distribuert samhandling, teknologiske forutsetninger .....	54
7.9 TESTSENDINGER PÅ BAKKENETTET: EN MULIGHETSSTUDIE .....	57
7.10 UAVHENGIG EVALUERING: EN MULIGHETSSTUDIE .....	59
7.11 AVSLUTTING AV PILOTEN .....	60
7.12 RESSURSER FOR PILOTEN BASERT PÅ JEVN BELASTNING .....	60
7.12.1 Stillingsbeskrivelser .....	61
7.13 TIDSPLANEN .....	62
7.14 INNTEKTER OG UTGIFTER FOR TILRETTELLEGGINGENS FØRSTE ÅR .....	63

<b>8</b>	<b>JURIDISKE UTFORDRINGER: IKKE BARE YTRINGSFRIHET MEN OGSÅ LOVFESTET</b>	
	<b>YTRINGSRETT. ....</b>	<b>65</b>
8.1	TAP AV STEMMERETTEN .....	68
8.2	TAP AV FØRERRETTEEN .....	69
8.3	FORSLAG TIL HÅNTERING AV YTRINGSRETTEEN OG TAP AV DENNE .....	70
8.3.1	Stenging av sendeflaten mens sending pågår .....	72
8.3.2	Stenging av sendeflaten for en bestemt aktør i påvente av dom .....	75
8.3.3	Tap av ytringsretten for en nærmere angitt tid .....	76
8.3.4	Verst tenkte tilfelle av misbruk av ytringsretten .....	77
8.4	ANDRE RETTSLIGE SIDER VED NORGES ÅPEN KANAL .....	77
8.4.1	Ikke konsesjon men <i>disposisjon</i> .....	78
8.4.2	Opptak av program, sikring av bevis .....	78
<b>9</b>	<b>TEKNISKE UTFORDRINGER .....</b>	<b>79</b>
9.1	TERMINALER .....	79
9.1.1	Maskinvare .....	80
9.1.2	Programvare .....	80
9.2	KOMMUNIKASJON .....	84
9.2.1	Stamnett .....	84
9.2.2	Aksessnett .....	85
9.3	TJENESTETILBYDER .....	87
9.3.1	Maskinvare .....	88
9.3.2	Programvare .....	89
9.4	PÅLITELIGHET .....	90
9.4.1	Tilgjengelighet .....	90
9.4.2	Funksjonssikkerhet .....	91
9.4.3	Modell for systemet .....	91
<b>10</b>	<b>BRUKERBASERT REGISTRERINGS- OG OPPLASTNINGSSYSTEM: .....</b>	<b>94</b>
10.1	VIRKEMÅTE .....	94
10.2	KRAVSPESIFIKASJON FOR PUBLISERINGSSYSTEMET: .....	94
10.2.1	Prosedyre for registrert bruker .....	94
10.2.2	Krav til sikkerhet .....	95
	INNSLAGENES GANG I SYSTEMET .....	96
<b>11</b>	<b>IMPLEMENTERING AV DEMONSTRASJONSMODELL .....</b>	<b>97</b>
11.1	INFRASTRUKTUR .....	97
11.2	DATABASE .....	99
11.2.1	Beskrivelse av tabeller i databasen .....	103
<b>12</b>	<b>FORSLAG TIL UTBEDRING AV SYSTEMET .....</b>	<b>106</b>
12.1	AUTOMATISK MULTICAST SCHEDULING .....	106
12.1.1	Kravspesifikasjon .....	107
12.2	VIDERUTVIKLING FOR LOKALE SENDINGER OG ØKT REDUNDANS .....	108
12.2.1	En landsdekkende sending i prøvedrift .....	108
12.2.2	Lokale og landsdekkende sendinger i full drift .....	109

<b>13</b>	<b>UTVIKLING I DIPLOMOPPGAVEN.....</b>	<b>110</b>
13.1	BRUKERVENNLIG OPPLASTING.....	111
13.1.1	Begrensninger i systemet.....	113
13.1.2	Prototypen og dens videre utvikling .....	114
13.1.3	Demonstrasjon for faglærer .....	114
13.2	AVVIKLING AV TV-SENDING .....	118
13.2.1	Testoppsett for avvikling av TV-sending .....	119
13.2.2	Virkemåte .....	120
13.2.3	Utfordringer i testoppsettet.....	121
13.2.4	Testoppsett for sammenligning av videoformater .....	122
13.2.5	Demonstrasjon for faglærer .....	124
13.2.6	Stereoskopisk playout.....	126
13.2.7	Utbedringer av avviklingsystemet.....	127
<b>14</b>	<b>EKSPERIMENTER MED DISTRIBUTUERT SAMHANDLING V/HIL .....</b>	<b>129</b>
14.1	SAMPRODUKSJON MELLOM HiL OG HiG.....	130
14.2	UTFORDRINGER I TV-PRODUKSJONER MED DISTRIBUTUERT SAMHANDLING.....	131
14.3	TERMINOLOGI.....	132
14.4	LABORATORIEOPPSETT .....	133
14.5	GJENNOMFØRING.....	133
14.6	REKTORINTERVJU .....	134
14.6.1	Intervju av person på distanse uten mediator .....	134
14.6.2	Intervju av person på distanse med mediator.....	135
14.7	SKUESPILLERE .....	136
14.8	INFRASTRUKTUR.....	137
14.9	MÅLSETTING .....	139
14.10	SEERUNDERSØKELSENS RELEVANS.....	139
14.11	INTERAKTIVITETSBEGREPET .....	141
14.12	INTERAKTIVITET I TIDE PROSJEKTET .....	142
14.12.1	Hastighet.....	142
14.12.2	Range.....	142
14.12.3	Mapping .....	144
14.13	DRØFTING.....	145
14.13.1	Interaktivitet og fortellerteknikk.....	145
14.13.2	En-til-mange.....	145
14.13.3	Ulike deltakererfaringer.....	146
14.13.4	Erfaringer og kunnskap .....	147
14.14	KONKLUSJON FOR DELTAKERE .....	147
14.15	PRODUKSJONSRESULTAT FOR VISNING .....	149
14.16	KONTAKT OG LIKEVERDIGHET.....	152
14.17	KONKLUSJON FOR SEERE .....	154
14.18	DRØFTING AV FORBEDRINGSPOTENSIALER .....	155
14.18.1	Fysisk .....	155
14.18.2	Teknisk .....	156
14.18.3	Fortellerteknisk.....	156
14.19	SAMPRODUKSJON MELLOM HØGSKOLEN I LILLEHAMMER OG NRK .....	158
14.19.1	Formål .....	159
14.19.2	Svakheter ved tidligere benyttet scenekomposisjon .....	159
14.19.3	Forsøk med objektbasert scenekomposisjon .....	159
14.19.4	Gjennomføring .....	160
14.19.5	Infrastruktur.....	161
14.19.6	Tekniske erfaringer.....	163
14.19.7	Fortellertekniske erfaringer .....	166
14.20	VIDERE UTVIKLING AV DET VIRTUELLE ROM .....	168
14.20.1	Bevegelig kamera i virtuelle rom .....	168
14.20.2	Bevegelig kamera og utnyttelse av visuelle parametere .....	170
14.20.3	Virtuelle studio med rimelig software .....	171
14.20.4	Utfordringer med DV og HDV-formatet i chromakeying .....	173

---

14.20.5	Semiprofesjonelle HD-kameraer egnet for chromakeying.....	176
14.20.6	Alternativer til chromakeying.....	178
<b>15</b>	<b>3D I DISTRIBUTERT SAMHANDLING .....</b>	<b>183</b>
15.1	BAKGRUNN.....	183
15.2	EKSPERIMENT MED TILGJENGELIG TEKNOLOGI.....	183
15.2.1	Oppsetting av visning og opptaksutstyr.....	183
15.2.2	3D-kommunikasjon via videokonferanseenheter .....	184
15.2.3	Forslag til oppbygging av en 3-dimensjonal mediator .....	186
15.2.4	Relatert forskning for distribuert samhandling i virtuelle rom .....	188
15.2.5	Enkel sammensetning av virtuelle rom med SMIL .....	193
15.2.6	Eksperimenter med syntetiserte avatarer .....	194
<b>16</b>	<b>KONKLUSJON .....</b>	<b>198</b>
16.1	KONKLUSJON FOR SAMFUNNSDELEN .....	198
16.2	KONKLUSJON FOR SYSTEMDELEN .....	198
<b>17</b>	<b>REFERANSER.....</b>	<b>201</b>
<b>A</b>	<b>– APPEDIKS: VEILEDNING I STEREOSKOPISK PRODUKSJON.....</b>	<b>204</b>
17.1	BAKGRUNN.....	204
17.2	AVGRENSNINGER.....	204
17.3	STEREOSKOPIske PRINSIPPER .....	206
17.4	TILGJENGELIG TEKNOLOGI .....	208
17.5	GJENNOMFØRING.....	212
17.5.1	Stereoskopisk opptak.....	212
17.5.2	Kameraoppsett.....	213
17.6	NOEN TIPS UNDER OPPTAK.....	220
17.7	STEREOSKOPISK ETTERARBEID .....	225
17.7.1	Redigering .....	225
17.7.2	Konvertering av stereovideo med graph edit (demultipleksing).....	226
17.7.3	Eksportering av video fra Adobe Premiere .....	227
17.7.4	Komposisjon.....	227
17.7.5	Sammensetning og kompresjon.....	230
17.8	OPPLASTING OG STREAMING.....	232
17.9	STEREOSKOPISK VISNING.....	233
<b>18</b>	<b>B - APPEDIKS: VEILEDNING I BRUK AV ÅPEN KANAL .....</b>	<b>235</b>
18.1	BRUK AV OPPLASTINGSSYSTEMET .....	235
18.1.1	Lage ny bruker.....	235
18.1.2	Registrere deg og din organisasjon.....	237
18.1.3	Se på videoarkiv .....	239
18.1.4	Laste opp innslag.....	241
18.1.5	Legge inn informasjon om innslag .....	242
18.1.6	Publisere innslag.....	242

**19 C-APPENDIKS: KILDEKODE MED FORKLARING FOR OPPLASTINGSKLIENT OG TJENER**

**245**

19.1	OPPLASTINGSKLIENTEN .....	245
19.1.1	Endringer som utføres i registeret .....	245
19.1.2	"shnetfile.h" .....	247
19.1.3	"shnetfile.cpp" .....	248
19.1.4	"shmain.cpp" .....	252
19.2	OPPLASTINGSTJENEREN .....	257
19.2.1	Parametere som benyttes i installasjon av opplastingstjeneren .....	257
19.2.2	"cmsusers.h" .....	258
19.2.3	"cmsdbfunc.ccp" .....	259
19.2.4	"cmsdbfunc.h" .....	262
19.2.5	"cmsmain.ccp" .....	263
19.2.6	"cmsserverclass.cpp" .....	266
19.2.7	"cmsserverclass.h" .....	270
19.2.8	"cmsusers.cpp" .....	272

**20 D - APPENDIKS: KILDEKODE MED FORKLARING AV KOMPRIMERING- OG**

**PUBLISERINGSSYSTEMET .....282**

20.1	REGISTRERINGSSYSTEMET .....	282
20.1.1	"logon.php" .....	282
20.1.2	"adduser.php" .....	285
20.1.3	"registrer.php" .....	286
20.1.4	"addred.php" .....	288
20.1.5	"addorg.php" .....	290
20.1.6	"index.php" .....	292
20.2	OPPLASTNINGSSYSTEMET .....	293
20.2.1	"browse.php" .....	293
20.2.2	"editcontent.php" .....	294
20.2.3	"editcontent_alter.php" .....	296
20.2.4	"viewfileinfo.php" .....	296
20.3	ADMINISTRASJONSFUNKSJONER .....	298
20.3.1	"administer.php" .....	298
20.3.2	"admusers.php" .....	299
20.3.3	"category.php" .....	301
20.3.4	"changeowninfo.php" .....	302
20.3.5	"fcheck.php" .....	303
20.3.6	"feedback.php" .....	304
20.3.7	"log.php" .....	305
20.3.8	"logout.php" .....	306
20.3.9	"published.php" .....	307
20.3.10	"show.php" .....	308
20.4	BIBLIOTEKER .....	310
20.4.1	"_designlib_php" .....	310
20.4.2	"_contentlib_php" .....	311
20.4.3	"_dblib_php" .....	316
20.4.4	"_htmlib_php" .....	318
20.4.5	"_browselib_php" .....	319
20.4.6	"_funklib_php" .....	321
20.4.7	"medieforv_skjema.php" .....	323
20.5	KORT OM PHP SINE SPESIELLE FUNKSJONER .....	326

<b>21</b>	<b>E - APPENDIKS: KOMPRIMERINGSSYSTEMET .....</b>	<b>330</b>
21.1	GENERERING AV SCRIPT.....	330
21.1.1	”autoscript.bat”.....	330
21.1.2	”gscript.pas”.....	331
21.1.3	”sgen.pas”.....	336
21.1.4	”gsstart.pas”.....	353
21.1.5	”gsstop.pas”.....	355
21.1.6	”gs-clean.pas”.....	356
21.1.7	”gswait.pas”.....	358
21.2	FORSLAG TIL UTBEDRINGER.....	359
<b>22</b>	<b>F - APPENDIKS: ”DRAG AND DROP” KOMPRIMERINGSSCRIPT .....</b>	<b>360</b>
22.1	KOMPRIMERING TIL WINDOWS MEDIA 9 .....	360
22.2	KOMPRIMERING TIL MPEG2 OVER MPEG2TS.....	361
22.3	KOMPRIMERING TIL H264 OVER MPEG2TS.....	361
22.4	KOMPRIMERING TIL H264 I EN STREAMBAR MPEG4 FIL .....	362
	<b>VEDLEGG 1 – SAMARBEID MED UNINETT .....</b>	<b>363</b>
	<b>VEDLEGG 2 – ÅPEN KANALS VEDTEKTER.....</b>	<b>365</b>
	<b>VEDLEGG 3 - OVERSETTING AV REGLEMENT GJELDENE FOR ÅPNE KANALER I</b>	
	<b>TYSKLAND .....</b>	<b>368</b>

## Figurliste

FIGUR 3-1: ILLUSTRERT BRUK AV ÅPEN KANAL .....	24
FIGUR 5-1: ORGANISASJONSSTRUKTUR FOR ÅPEN KANAL .....	30
FIGUR 6-1: TEKNISK INFRASTRUKTUR FOR ÅPEN KANAL .....	39
FIGUR 7-1: VIDEO- OG AVSPILLINGSSYSTEMET .....	48
FIGUR 7-2: INFRASTRUKTUR FOR KONTINUERLIG DRIFT .....	51
FIGUR 7-3: ETTER SENDING, DE FIRE TIL VENSTRE ER PÅ GJØVIK, DE FEM TIL HØYRE PÅ LILLEHAMMER .....	56
FIGUR 7-4: TIDSPAN FOR ÅPEN KANAL PILOT .....	63
FIGUR 7-5: BUDSJETT ÅPEN KANAL PILOT .....	64
FIGUR 9-1: MODELL AV EN ÅPEN KANAL PÅ INTERNET .....	79
FIGUR 9-2: BLOKKSJEMA FOR BEREGNING AV PÅLITELIGHET .....	92
FIGUR 9-3: BLOKKSJEMA AV ENKEL SERVER .....	93
FIGUR 9-4: BLOKKSJEMA AV REDUNDANT SERVER .....	93
FIGUR 10-1: TILSTANDSDIAGRAM FOR ET INNSLAG .....	96
FIGUR 11-1: INFRASTRUKTUR FOR DEMONSTRASJONSMODELLO .....	98
FIGUR 11-2: DATABASEMODEL I OPPRINNELIG KRAVSPESIFIKASJON (ÅRKIVTJENESTE) .....	99
FIGUR 12-1: WEB-GRENSESNITT FOR PROGRAMMERING AV SENDEPLAN I MULTICAST STRØM .....	106
FIGUR 12-2: INFRASTRUKTUR FOR REALISERING AV AUTOMATISK SENDEPLANSPROGRAMMERING .....	107
FIGUR 12-3: SYSTEMETS INFRASTRUKTUR I DAG .....	108
FIGUR 12-4: INFRASTRUKTUR MED MULIGHET FOR LOKALE SENDINGER OG ØKT REDUNDANS .....	109
FIGUR 13-1: FILHÅNTERINGEN OG ANDRE BAKENFORLIGGENDE PROSESSER SKJULES FOR BRUKEREN .....	112
FIGUR 13-2: TILBAKEMELDING FRA TJENERAPPLIKASJONEN .....	115
FIGUR 13-3: OPPLASTINGSKLIENTEN "SILENT HELPER" SOM STARTES FRA EN LINK PÅ ÅPEN KANAL SIN NETTSIDE .....	115
FIGUR 13-4: INTERN BROWSER MED STREAMINGLINKER OG STATUSOVERSIKT .....	117
FIGUR 13-5: PUBLISERING AV INNSLAGET UNDER EN GITT KATEGORI .....	117
FIGUR 13-6: STREAMING PÅ ÅPEN KANAL SIN OFFENTLIGE NETTSIDE .....	118
FIGUR 13-7: ÅPEN KANAL PÅ ET BORD .....	119
FIGUR 13-8: BLOKKSJEMA FOR TV-SYSTEMET (PEDERSEN, 2003) .....	120
FIGUR 13-9: OSCILLATOR / COFDM-MOD.(VENSTRE), MIKSER (MIDTEN), SET-TOP-BOKS / ANALYSATOR (HØYRE) .....	120
FIGUR 13-10: SCHEDULERT SENDING MED TIMERE .....	125
FIGUR 13-11: OPPLASTING AV VIDEO (VENSTRE), TILKNYTNING TIL KANAL (MIDTEN), AKTIVERING AV TIMER (HØYRE) .....	125
FIGUR 13-12: VENSTRE BILDE (VENSTRE), HØYRE BILDE (MIDTEN) OG STEREOSKOPIK STUDIOKAMERA (HØYRE) .....	126
FIGUR 14-1: FØRSTE FIBER DEDIKERTE FIBERSAMMENSLUTNING MELLOM HiL OG HiG. TERMINERINGSVERKTØY NØDVENDIG FOR SAMMENSLUTNING (TIL VENSTRE) OG GJENNOMSLAG AV FØRSTE BILDE (TIL HØYRE) .....	130
FIGUR 14-2: SENTRALE BIDRAGSYTERE TIL TIDE PROSJEKTET. MATS ERIXON, KTH (TIL VENSTRE), SIGMUND ANDRESEN, HiL (I MIDTEN) OG CLAUS KNUDSEN, HiG (TIL HØYRE) .....	130
FIGUR 14-3: SAMLING AV DELTAKERNE I SVR-PROSJEKTET VED HØGSKOLEN I LILLEHAMMER .....	131
FIGUR 14-4: MEDIATOREN. DET HALVGJENNOMSKINNELIGE SPEILET SØRGER FOR BLIKK-KONTAKT VED AT BILDET FRA PERSONEN MAN KOMMUNISERER MED BLIR REFLEKTERT FORAN LINSEN TIL KAMERA (SAMME PRINSIPP SOM I EN TELEPROMTER) .....	132
FIGUR 14-5: KONTROLLROM FOR SAMMENSETTING AV DET VIRTUELLE ROMMET (TIL VENSTRE). CLAUS OG MEG I STUDIO, JOBBER FOR Å OPPRETTE IP-FORBINDELSE TIL GJØVIK VIA VIDEOKONFERANSE-ENHETER (TIL HØYRE) .....	133
FIGUR 14-6: INTERVJU UTEN MEDIATOR, PRINSIPPSKISSE .....	134
FIGUR 14-7: INTERVJU VED BRUK AV MEDIATOR, PRINSIPPSKISSE .....	135
FIGUR 14-8: SKUESPILLER MED MEDIATOR, PRINSIPPSKISSE .....	136
FIGUR 14-9: SIGNALVEIER VED HHV. HØGSKOLEN I GJØVIK OG HØGSKOLEN I LILLEHAMMER. ....	137
FIGUR 14-10: VISER SKILLELINJEN MELLOM DE TO BILDENE FRA GJØVIK (TIL VENSTRE) OG LILLEHAMMER (TIL HØYRE). I OMRÅDET I MIDTEN ER DET EN "SOFT EDGE" PÅ "WIPE"-EFFEKTEN DER DET FORANLIGGENDE BILDET TIL VENSTRE BLIR GRADVIS MER TRANSPARENT MOT DET BAKENFORLIGGENDE BILDET TIL HØYRE. DETTE OMRÅDET KAN IKKE BRUKES I PRODUKSJONEN UTEN Å BRYTE ILLUSJONEN (PRØVEOPPTAK TIDE, 2005) .....	143



FIGUR 14-11: BILDET VISER ET DÅRLIG SAMMENSATT VIRTUELT ROM PÅ GRUNN AV A) ELEVEASJONSFORSKJELL, B) IKKE SAMMENFALLENDE OBJEKT, C) ULIK CHROMA/FARGE-METNING OG D) ULIK VATRING PÅ KAMERA. (PRØVEOPTAK TIDE, 2005).....	144
FIGUR 14-12: OBJEKTER SOM BEVEGER SEG I BAKGRUNNEN (HER: KAMERA) KAN VIRKE SVÆRT FORSTYRRENDE PÅ DELTAKERNES BLIKKONTAKT OG SYNLIG TEKNOLOGI KAN BRYTE ILLUSJONEN AV ET DELT ROM. (TIDE FILM, 2005).....	148
FIGUR 14-13: SCREENSHOT- SEKVENNS A, ALLE TRE AKTØRER I BILDE. KNUDSEN (TIL VENSTRE) OG MOREN (TIL HØYRE) I STUDIO LILLEHAMMER. WROLDSEN (MIDTEN) FRA STUDIO GJØVIK (TIDE FILM, 2005). ....	150
FIGUR 14-14: SCREENSHOT- SEKVENNS B, WROLDSEN AVSKÅRET MOREN. 3-SKUDD OVER SKULDER. STØRRELSSEFORHOLDET MELLOM REKTOR OG MEDIATOR GIR IKKE LIKEVERD (TIDE FILM, 2005). ....	151
FIGUR 14-15: SCREENSHOT – SEKVENNS C, DE TO BILDENE FRA STUDIO I GJØVIK OG STUDIO I LILLEHAMMER, SAMMENFØYD MED WIPE-EFFEKT. (TIDE FILM, 2005). ....	152
FIGUR 14-16: SCREENSHOT - SEKVENNS C, OVER-SKULDER MOT KVINNE. LIKEVERDIG STØRRELSSEFORHOLD MELLOM AKTØRENE (TIDE FILM, 2005).....	154
FIGUR 14-17: BILDET TIL VENSTRE VISER KONTROLLROMMET OG BILDET TIL HØYRE STUDIO VED HIL. ....	158
FIGUR 14-18: BILDET TIL VENSTRE VISER KONTROLLROMMET OG BILDET TIL HØYRE STUDIO VED HIG. ....	158
FIGUR 14-19: SIGMUND ANDRESEN I KK ..... 161	161
FIGUR 14-20: INFRASTRUKTUR FOR DISTRIBUTERT SAMHANDLING MELLOM STUDIO PÅ NRK OG STUDIO PÅ HIL. .... 162	162
FIGUR 14-21: OSCILOSCOPET (TIL VENSTRE) VISER FALLENDE LUMINANSVERDI FOR FARGENE FRA VENSTRE TIL HØYRE SOM ER HVIT, GUL, CYAN, GRØNN, MAGENTA, RØD, BLÅ OG SORT. VEKTORSKOPET (TIL HØYRE) VISER FARGEVERDIEN (HUE) SOM VINKEL (FASE) I FORHOLD TIL DE KORTE VEKTORENE SOM PEKER TIL VENSTRE OG RETT OPP (BLACKBURST HAR TO KOMPONENTER SOM ER 90 GRADER UTE AV FASE PÅ GRUNN AV "PHASE ALTERNATING LINE" TEKNIKKEN I PAL). FARGEMETNINGEN KAN LESES UT AV LENGDEN PÅ VEKTOREN. .... 163	163
FIGUR 14-22: BILDE FRA NRK (VENSTRE), FRA HIL (MIDTEN) OG KOMPOSISJON MED FELLES BAKGRUNN (HØYRE). .... 165	165
FIGUR 14-23: ORANGE-FILTRETT SPISSELYS ØVERST TIL HØYRE (VENSTRE BILDE). ALFAKANAL I KK (HØYRE BILDE). .... 166	166
FIGUR 14-24: ORANGE-FILTRETT SPISSELYS ØVERST TIL HØYRE (VENSTRE BILDE). ALFAKANAL I KK (HØYRE BILDE). .... 166	166
FIGUR 14-25: STEROSKOPISK PROJEKTOR (VENSTRE). FRONTALT KAMERA OG SILVERSCREEN, NRK (HØYRE).... 167	167
FIGUR 14-26: VALGDAGSSENDING NRK 03, SENSORHODE (435VRM), RENDRINGSENHET (VIRTUAL SCENARIO) ..... 169	169
FIGUR 14-27: "GARBAGE MASK" MED TRACKERE FØR (TIL VENSTRE) OG ETTER KAMERABEVEGELSE (TIL HØYRE). .... 170	170
FIGUR 14-28: KOMPOSISJON FØR (TIL VENSTRE) OG ETTER KAMERAFLYTNING (TIL HØYRE)..... 171	171
FIGUR 14-29: PARAMETERE FOR CHROMA-KEYING (TIL VENSTRE), CROP AV KILDE (NEDERST), ALFAMASKE (ØVERST) ..... 172	172
FIGUR 14-30: I DEBATT MED MEG SELV, ENDELIG KOMPOSISJON ("MED UPRESING") UTFØRT ETTER OPTAK. .... 174	174
FIGUR 14-31: FARGESAMPLING UTEN REDUKSJON 4:4:4 (ØVERST), FARGEREDUKSJON 4:2:2 OG 4:2:0 (NEDERST) 174	174
FIGUR 14-32: FARGEKOMPONENTER I 4:2:2 FARGESAMPLING (ØVERST) OG 4:2:0 FARGESAMPLING (NEDERST) ... 175	175
FIGUR 14-33: HDV KAMERA, CANON XL2 (TIL VENSTRE). P2 KAMERA, PANASONIC HVX200 (TIL HØYRE)..... 178	178
FIGUR 14-34: PRINSIPPSKISSE FOR DYBDEAVLESNING ..... 181	181
FIGUR 14-35: KUN DEN ANDELEN AV SCENEN SOM HAR KORREKT DYBDE BLIR NØKLET UT (ALFAKANAL TIL HØYRE)..... 182	182
FIGUR 15-1: OPPSETT AV STEROSKOPISK KAMERA (TIL VENSTRE) OG STEROSKOPISK SKJERM (TIL HØYRE)..... 185	185
FIGUR 15-2: STEROSKOPISK PRØVEBILDE (TIL VENSTRE). FØRSTE STEROSKOPISE BILDET FRA GJØVIK (TIL HØYRE)..... 185	185
FIGUR 15-3: 3D-MEDIATOR, PRINSIPPSKISSE..... 186	186
FIGUR 15-4: KONFERANSE I ET VIRTUELT ROM. KAMERAOPPSETT (VENSTRE), RENDRET DELTAKERBILDE (HØYRE). .... 188	188
FIGUR 15-5: BAKGRUNN/REF.BILDE(VENSTRE), DIFFERANSEKALKULERT OMRIS(MIDTEN), INTERPOLASJON(HØYRE) ..... 189	189
FIGUR 15-6: 3-DIMMENSJONALE AVATARER GENERERT MED IBPH (TIL VENSTRE) OG ZCAM (TIL HØYRE)..... 189	189
FIGUR 15-7: PRINSIPPSKISSE AV BBC SITT STUDIO FOR GENERERING AV VIRTUELLE ROM. .... 191	191
FIGUR 15-8: STUDIO MED REELT OPPSETT. INTERAKSJONSSKJERM(VENSTRE), DIODERING(MIDTEN), DUK(HØYRE). .... 192	192
FIGUR 15-9: RESULTATBILDER. MESH KALKULERT FRA 12 KAMERA(VENSTRE), MULTIKAM(MIDTEN), SCENE(HØYRE) ..... 192	192

FIGUR 15-10: ANALYSE AV ANSIKTSBEVEGELSER OG MAPPING TIL EN PREDEFINERT 3D-MASKE (METAXAS, 1999)	194
FIGUR 15-11: KONSTRUKSJON AV EN 3-DIMENSJONAL AVATAR MED PROGRAMMET FACEGEN	195
FIGUR 15-12: MEG I ULIKE VINKLER I DAG, OM 30 ÅR, INDER, ASIATER, SINT OG UNDER UTTALELSE AV VOKALEN O.	195
FIGUR 15-13: GJENKJENNING AV KROPPSBEVEGELSER VED HJELP AV MULTIKAMERA ANALYSE (METAXAS, 1999).	196
FIGUR 15-14: AVATAR MED KROPP. GENERERING AV ANSIKTSTEKSTUR (VENSTRE). ANIMERBAR KROPP (HØYRE)	197
FIGUR A-1: NUVIEW 3D-UTSTYR (VENSTRE), PRINSIPPSKISSE FORSATS (MIDTEN), FIELD SEKV. RESULTAT (HØYRE)	205
FIGUR A-2: MENNESKETS VISUELLE SYSTEM	207
FIGUR A-3: PROJEKSJONSPLAN OG PARALLAKSE	208
FIGUR A-4: STEREOSKOPIK DYBDEILLUSJON	208
FIGUR A-5: PRINSIPPSKISSE FARGESEPARASJON	210
FIGUR A-6: SIRKULÆR OG LINEÆR POLARISASJONSFILTRERING	210
FIGUR A-7: LENTIKULÆR MULTIVIEW, PRINSIPPSKISSE (RØNNINGEN, 2003)	211
FIGUR A-8: STEREOSKOPIK KAMERA OG MONITORRIGG (SHUTTERGLASS-BASERT)	214
FIGUR A-9: UTPRØVING AV PERISKOP FOR Å REDUSERE "ØYEAVSTANDEN"	215
FIGUR A-10: AUTOSEREOSKOPIK VISNINGSSYSTEM FRA SEEREAL TECHNOLOGIES.	217
FIGUR A-11: STEREOSKOPIK VISNING. PROJEKTORER MED POLAFILTER (VENSTRE), SILVERSCREEN-LERRET (HØYRE)	217
FIGUR A-12: HØY KAMERAPLASSERING FOR REALISERING AV OVERVINKLET BILDE MED NØDVENDIG TELE	219
FIGUR A-13: PORTABELT STEREOSKOPIK VISNINGSSYSTEM	220
FIGUR A-14: LYSRIGG MED CHIMERA OG SPOTLYS	221
FIGUR A-15: TRÅDLØS MIKROFON MED SENDER OG MOTTAKE (VENSTRE), RETNINGSSTYRTE MIKROFONER (HØYRE).	222
FIGUR A-16: FARGESEPARERT STEREOSKOPIK BILDE SOM VISER 0-PARALLAKSEN (I ETT MED LERRETPLANET)	223
FIGUR A-17: STEREOSKOPIK REDIGERING I ADOBE PREMIERE	226
FIGUR A-18: FILTERSTRUKTUR I GRAPH EDIT	227
FIGUR A-19: STEREOSKOPIK KOMPOSISJON I AFTER EFFECTS.	228
FIGUR A-20: STEREOSKOPIK KOMPOSISJON I STEREO MOVIE MAKER	229
FIGUR A-21: DUBBING I VIRTUAL DUB	231
FIGUR A-22: KOMPRIMERING MED WINDOWS MEDIA ENCODER	232
FIGUR A-23: VISNING MED STEREOSCOPIC PLAYER	234
FIGUR A-24: STEREOSKOPIK VISNING MED POLARISASJONSFILTRE, PRINSIPPSKISSE	234
FIGUR B-1: GRENSESNIITT FOR PÅLOGGING	235
FIGUR B-2: GRENSESNIITT FOR REGISTRERING AV NY BRUKER	236
FIGUR B-3: GRENSESNIITT FOR PÅLOGGING AV BRUKER ETTER REGISTRERING	236
FIGUR B-4: GRENSESNIITT FOR REGISTRERING AV REDAKTØR	237
FIGUR B-5: GRENSESNIITT FOR REGISTRERING AV ORGANISASJON	237
FIGUR B-6: GRENSESNIITT PÅ SYSTEMETS VELKOMSTSIDE	238
FIGUR B-7: SKJEMA TIL STATENS MEDIEFORVALTNING	239
FIGUR B-8: MENYVALG FOR TILGANG TIL VIDEOARKIV	239
FIGUR B-9: INTERNT VIDEOARKIV	240
FIGUR B-10: GRENSESNIITT FOR OPPLASTING AV FILER	241
FIGUR B-11: INNMATING AV INFORMASJON OM INNSLAG	242
FIGUR B-12: OFFENTLIG TILGJENGELIG VIDEOARKIV	243
FIGUR B-13: VIDEOLENKER FOR MULTICAST SENDING	244
FIGUR C-1: OPPLASTINGSKLIENTEN AKSESSERT FRA ÅPEN KANAL SIN NETTSIDE	245
FIGUR C-2: KVITTERING FRA OPPLASTINGSTJENER	257
FIGUR D-1: LOGON.PHP	282
FIGUR D-2: ADDUSER.PHP	285
FIGUR D-3: REGISTER.PHP	287
FIGUR D-4: BROWSE.PHP	293
FIGUR D-5: VIEWFILEINFO.PHP	297
FIGUR D-6: MEDIEFORV_SKJEMA.PHP	323

## **Tabelliste**

TABELL 9-1: SAMMENLIGNING MELLOM VIDEOFORMATER.....	82
TABELL 9-2: ADSL TILBUD FRA NOEN STORE TELELEVERANDØRER AUGUST 2003 .....	86
TABELL 9-3: ADSL TILBUD FRA NOEN STORE TELELEVERANDØRER APRIL 2006.....	86

## Forkortelser

<b>ASI</b>	Asynchronous Serial Interface Grensesnittstandard med enten fiber- eller koaksialkabel tilkobling og overføringsrate opp til 270Mbit/s.
<b>CCD</b>	Charge Coupled Device Brikken i kameraet som gjør lyset om til elektriske signaler
<b>COFDM</b>	Coded Orthogonal Frequency Division Multiplexing Digital modulasjonsform som brukes i DVB-T nettverk.
<b>DV</b>	.Forkortelse for Digital Video. Et heldigitalt videoformat
<b>DVB</b>	Digital Video Broadcasting Et sett med standarder som definerer kringkasting av digital TV ved bruk av eksisterende satellitt, kabel og terrestriell infrastruktur
<b>DVB-T</b>	DVB – Terrestrial, standard for kringkasting over terrestrielt nett
<b>FTP</b>	File Transfer Protocol. Protokoll som ligger over TCP. Inneholder kommandoer for overføring av filer, fillister, etc.
<b>IP</b>	Internet Protocol Forbindelsesløs, pakkeorientert, best-effort protokoll som ruter trafikken gjennom internett.
<b>KKD</b>	Kultur- og kirke departementet. Håndterer konsesjonssaker og er sentral i behandlingen av formidlingsplikten for Åpen kanal.
<b>MHP</b>	Multimedia Home Platform Standard som definerer et generelt grensesnitt mellom interaktive applikasjoner og terminalene som disse applikasjonene skal kjøre på.
<b>MPEG</b>	Moving Picture Expert Group En gruppe av standarder for koding A/V-innhold til et digitalt komprimert format.
<b>PHP</b>	Hypertext PreProcessor, Scriptingspråk for å lage avansert funksjonalitet på nettsider.
<b>RTP</b>	Real Time transfer Protocol, brukes til sanntidsoverføring av video.
<b>SDI</b>	Serial Digital Interface En standard for overføring av digital ukomprimert video.
<b>TCP</b>	Transmission Control Protocol En pålitelig, pakkeorientert protokoll som ligger over IP. Etablerer en forbindelse mellom to parter og tar hånd om at ingen pakker går tapt.

## 1 Innledning

En Public Access kanal skal realiseres i Norge og foreningen *Åpen kanal* har gått i spissen for denne utviklingen. Foreningen er representert med stor bredde blant ikke-kommersielle videoprodusenter og jobber for at ikke bare disse, men samtlige individer med behov for å formidle et budskap skal sikres tilgang til TV-mediet.

Begrepet ytringsfrihet er noe medienorge lenge har kjempet hardt for å sikre. Men hva sikrer det for allmennheten? Det betyr ikke noe annet enn at en svært liten andel av befolkningen, journalister og mediemoguler, kan ytre seg og føle seg trygge på ikke å bli sensurert. Ytringsfrihet betyr ikke at folket har krav på å bli hørt eller sett. Det betyr ikke at folket har krav på tilgang til de samme kanalene som de som kan nyte ytringsfriheten. Allmennheten er i dag ekskludert fra det offentlige rom i den forstand at de ikke kan ytre sine meninger i massekommunikasjon. Med mindre journalister eller produsenter finner deres innhold interessant da. Og i hvilke tilfeller er det? Jo, stort sett når journalisten eller produsenten har noe å tjene på det. Skal allmennheten sikres denne tilgangen må folket innvilges en sterkere rett, en ytringsrett.

Foreningen *Åpen kanal* jobber for å sikre ytringsrett i TV-mediet både juridisk og praktisk med konkrete forslag til lovendring og utvikling av systemer for distribuert samhandling.

Denne oppgaven tar for seg samfunnsmessige og tekniske aspekter ved etableringen av en åpen kanal. Utfordringer er det mange av, men det er på høy tid at vi gjør hva vi kan for å få opp en kanal til fri formidling. Behovet for en slik kanal tiltar mens det øvrige mediebildet gradvis overstyres av store kommersielle aktører. Samtidig prøver de å overbevise oss at kommersialiseringen er som en ufravikelig "naturlov" hvor den mest kapitalsterke vinner i kampen om å bli hørt eller sett. Det er grunner til å betvile gyldigheten av dette. Demokrati handler om å ta valg – og de viktige valgene vil alltid måtte koste, hvis ikke var det et enkelt valg.

Vi lever i et politisk medielandskap som snarere favoriserer underholdning enn holdninger, meninger og engasjement. *Åpen kanal* vil ikke bare være en motkultur til dette, den vil gjeninnføre i TV-mediet den naturlige dialogen blant folket, og la allmennheten gjenerobre det offentlige rommet. Kommunikasjonen befolkningen mellom må tilrettelegges i en form som gjør at vi oppnår innsikt. Det handler ikke om protest og opprør mot dagens kultur, men å nyttegjøre iboende egenskaper for å komme tilbake til det som er naturlig, nemlig den enkle dialogen.

Edu-tainment, info-tainment og enter-tainment kan ikke være malen for all massekommunikasjon. Vi trenger en arena hvor folk kan snakke ut. Vi må gi folk sjansen til å reflektere, skape bredde og dybde i resonnementene. Vi må kunne ha samtaler i studio uten fiffige lede- og herskerteknikker ut fra seertall og underholdning. Vi må kunne ha råd til smale dybdeprogrammer som tar folk på alvor og ikke primært ser et program som en bro over til neste reklameinnslag. Det er på høy tid at vi får et supplement til dagens medier, men hvor skal man begynne? Første steg må være å sikre tilgangen til mediet. Det er dette jeg tar for meg i denne oppgaven. Hvilke forutsetninger kan man bygge videre på, og hvilke utfordringer bør vi ta tak i nå? I oppgaven vil jeg drøfte hva som må til for å skape en ÅPEN kanal. Jeg vil også vise med eksempel hvordan innsamling og publisering kan fungere på nett med et enkelt Public Access system. Det Web-baserte systemet er utviklet i tidligere prosjektoppgave men vil nå implementeres fra ende til ende med en enkel og brukervennlig opplastingsapplikasjon samt et system for TV-sending. Det vil også bli utført forsøk på live distribuert samhandling. Dette for å studere forutsetningene for deltakernes følelse av nærhet og for å finne tekniske løsninger på debattproduksjoner i et nettverk der deltakerne er spredt over hele landet. Løsningene er ikke nødvendigvis basert på teknologi som finnes i dag. Oppgaven tar sikte på å demonstrere konseptuelle modeller som et grunnlag for å vurdere nytteverdien i kommende teknologi. Spesielt interessant er teknologi som bygger opp under forutsetningene for en naturlig kommunikasjon. Det blir i den sammenheng utviklet en prototyp som sikrer blikkontakt mellom aktørene i en distribuert samhandling. Nyten av dybdesyn blir også studert med realiseringen av stereoskopiske videokonferanser. For at seeren skal ha en følelse av deltakernes nærhet til tross for den fysiske avstanden i mellom dem blir TV-produksjonen bygget opp som et delt virtuelt rom. I første forsøk med enkle midler, senere med objektbaserte teknikker og en drøfting rundt videreutviklingen av disse.

### *1.1 Avgrensninger*

*Åpen kanals* skal være en åpen TV-kanal. I prosjektoppgaven ble sentrale deler av nett-teknologi for å realisere åpenheten implementert.. Kvaliteten på det som samles inn og arkiveres i systemet skulle være godt nok for en fremtidig TV-sending. Implementasjonen ble dermed den første byggesteinen i et endelig system for realisering av Public Access TV i Norge. Systemet er også det første i sitt slag, som en 100% nettbasert TV kanal med kontribusjon, registrering og

publisering på nett. Det siste trinnet, TV-distribusjon, ble ikke realisert i prosjektoppgaven. Dette fordi det ikke var tilgang på nødvendig utstyr. I denne diplomoppgaven blir et system for TV-avvikling realisert med den distribusjonsteknikken som vil bli benyttet i det digitale bakkenettet. Dette fordi det i konsesjonsbetingelsene for utbyggingen av dette nettet ble reservert en plass til en åpen kanal. Hvorvidt denne plassen kan benyttes avhenger av om deltakerne i Åpen kanal kan få sende uten å måtte betale vederlag til distributør. I motsatt tilfelle er det ingen åpen kanal ettersom kun de med penger vil få tilgang til å ytre seg. En ytringsrett må inkludere alle. Hvordan ytringsretten kan sikres og hva det innebærer i juridisk forstand vil bli drøftet i denne oppgaven. Kravene som stilles til distributør er ikke i tråd med Stortingets intensjoner med Åpen kanal så her gjenstår det mye arbeid før kanalen er en realitet. Til dette arbeidet trenger kulturdepartementet en rettleiding fra foreningen Åpen kanal. Jeg har derfor i samfunnsdelen av denne oppgaven prioritert å skrive om ytringsrett i samarbeid med styreleder Ola Tellesbø som er gjesteforsker ved institutt for medieforvaltning på Universitetet i Oslo. Det blir også utarbeidet et forslag til en pilot av konseptet, i samarbeid med styret i Åpen kanal, der forutsetningen er statlig økonomisk støtte.

Det er ønskelig at denne oppgaven kan fungere som et grunnlag for videre utvikling av kanalen både teknisk og organisatorisk. I den forbindelse har jeg påbegynt utvikling av applikasjoner som er kritsikk for at kanalen skal fremstå åpen for folk flest. Det viste seg gjennom deltakernes bruk av det Web-baserte publiseringssystemet at opplastingen som er det første trinnet i infrastrukturen ikke var brukervennlig nok. For at Åpen kanal i en eventuell pilotavvikling skal ha nok innhold til å kunne kjøre daglige sendinger må denne biten være på plass snarest mulig. Systemer for oppbygging av arkivet med en brukervennlig opplastingsapplikasjon og tilhørende tjener har derfor blitt prioritert fremfor et mer brukervennlig grensesnitt for avvikling av TV-sending. Den siste biten er noe som kan utvikles nærmere en eventuell pilotsending.

I forbindelse med piloten kan det være ønskelig for Åpen kanal sine deltakere å prøve ut distribuert samhandling. Systemer for live kontribusjon blir ikke utviklet i denne oppgaven ettersom det ikke er en forutsetning for å kunne si at kanalen er åpen. Forhåndsprodusert materiale fra en distribuert samhandling kan lastes opp men ikke sendes live. Dette gir klare begrensninger med hensyn på interaktivitet mot seerne som vi ønsker å aktivisere. TV-avviklingsystemet har støtte for livesending over bakkenettet, men dette forutsetter en investering i et MPEG2 encoder kort i playout-enheten som må tilkobles en sentral

streamingplayer (PC med SDI ut). Slike investeringer var det ikke rom for i oppgaven. Videreutviklingen er likevel enkel og kan utføres kort tid før piloten dersom Åpen kanal blir tildelt midler. I denne oppgaven har jeg valgt å begrense meg til konseptuelle skisser for distribuert samhandling med demonstrasjoner fremfor en implementering av et brukervennlig system. Teknologien er under rask utvikling og Åpen kanal kan tjene på å vente til rimelige kommersielle løsninger blir tilgjengelig. En studie av kommende teknologi der prinsipper blir demonstrert i praksis regnes derfor som mer nyttig for kanalen. Med forsøkene utført i denne oppgaven kan Åpen kanal få kjennskap til menneskelige aspekter og forutsetninger for å gjenvinne den nære, naturlige og åpne samtalen i TV-mediet. Distribuert samhandling kan danne grunnlag for en ny debattarena der deltakerne enkelt kan være tilstede med hverandre for seeren og selv føle en nærhet til tross for geografiske avstander.

## *1.2 Oppbygning*

Med emnebeskrivelsen menneske, samfunn og media har jeg tillatt meg å skrive nok så inngående om organisatoriske og juridiske aspekter. Likevel har jeg ikke kunnet la det gå på bekostning av den tekniske realiseringen. Det at medlemmene i *Åpen kanal* har fått tilgang til tjenesten og gitt tilbakemelding på hva de ønsker siden implementasjonen i prosjektet, har vært en forutsetning for å kunne utbedre systemet. Dessverre har ikke prøvetiden blitt like lang som forventet ettersom utstyret systemet er bygget på ble hacket og serverne måtte sendes på service. Utstyret kom ikke tilbake før diplomen. Denne oppgaven har derfor i stor grad bestått i å få opp systemet igjen på alternative enheter. Software som ble utviklet i prosjektet er likevel i prinsippet det samme med kun mindre endringer for at koden skal hente ressurser fra nytt utstyr (Koden har også blitt utbedret for å unngå omkodning ved nye forflytninger). Det som er nytt i denne oppgaven er utviklingen av kode som måtte prioriteres i henhold deltakernes tilbakemelding fra prøveperioden. Som tidligere nevnt ble det en brukervennlig opplastingsklient. Om det siste leddet, avvikling til TV-sending, har vi ingen erfaringer fra brukerne men antar at også dette bør gjøres mer brukervennlig. Foreløpig er det kun realisert med hyllevareprogrammer som er installert på en server (installasjonen måtte gjøres fra bunnen av ettersom også denne serveren var hacket). Dokumentasjonen av de to endepunktene i systemet forutsetter kunnskap om det sentrale systemet som ble utviklet i prosjektet. Beskrivelsen av disse har derfor blitt plassert etter



dokumentasjonen av det opprinnelige systemet. Etter anbefaling fra faglærer ble det bestemt at diplomoppgaven skulle være komplett og også inkludere deler som har blitt realisert tidligere.

Prosjekteringen av Åpen kanal er plassert etter en kort beskrivelse av konseptet åpen kanal. Her er det ikke en forutsetning med inngående kjennskap til systemet ettersom dette kapittelet omhandler mer organisatoriske aspekter ved en pilotrealisering. De juridiske utfordringene ved en åpen kanal både med tanke på formidlingsplikt og håndtering av redaktørrollen er nytt for medieforvaltningen. Her vil foreningen Åpen kanal måtte veilede kirke og kultur departementet etter beste evne. Denne delen er svært relevant for at en åpen kanal skal kunne bli åpen, og både KKD og foreningen har søkt kunnskap og erfaringer i de tyske åpne kanalene der konseptet har blitt prøvd ut gjennom 20 år. Reglementet som gjelder for disse er vedlagt med en oversetting (etter beste evne) til norsk.

For videre utvikling av konseptet Åpen kanal har jeg prioritert å gjøre studier av produksjoner med distribuert samhandling. Dette er ikke erfaringer vi kan hente fra andre åpne kanaler, men det finnes et miljø for dette ved Høgskolen i Lillehammer og Høgskolen i Gjøvik. Claus Knudsen som er dr.techn fra KTH har tatt med seg sine medieteknikk-studenter fra HiG til Lillehammer i et prosjekt der telepresence blir realisert med hjelp fra fjernsynsteknikk-studentene der. Jeg har inneværende semester studert fjernsynsteknikk ved HiL samtidig med skrivingen av masteroppgaven. Dette har jeg utnyttet til å dokumentere erfaringene i samarbeid med studentene ved HiG. Med min bakgrunn fra NTNU kunne jeg også bidra til dette prosjektet ved å foreslå alternative teknikker basert på mer avansert videoteknologi. Objektbaserte teknikker og 3-dimensjonale realiseringer av telepresence er prøvd ut i samarbeid med NRK som er vegg i vegg med Høgskolen i Lillehammer. Forsøkene har vist at slike teknikker kan være egnet til å utvikle mer brukervennlige løsninger for distribuert samhandling, samt styrke deltakernes følelse av nærhet. For realiseringen av den 3-dimensjonale avbildningen har jeg nyttegjort egne erfaringer i produksjon av en stereoskopisk opplæringsfilm for legestudenter ved NTNU. Dette har også dannet grunnlaget for en veiledning i stereoskopisk produksjon som er vedlagt til bruk for Åpen kanal sine deltakere. Avansert videoteknikk er ikke avgjørende for realiseringen av Åpen kanal, men med Uninett og Midgard Media Lab som samarbeidspartnere samt tilgang på ressurser ved HiL vil Åpen kanal kunne være en pilot for videreutvikling av andre åpne kanaler, og forhåpentligvis gi noe tilbake til de som har hjulpet oss med realiseringen av public access i Norge.

## 2 Hva er en åpen kanal

Public Service-kanalers offisielle mål er å opplyse folket. Dette er i hovedsak en enveiskommunikasjon der man setter et klart skille mellom konsument og produsent av innhold. En seer forventes med andre ord ikke å ha interesse av å produsere innhold for å få vist dette på TV-kanalen. Seeren er kun en passiv mottaker og konsument av innhold. Dette står i kontrast til Public Access TV, eller åpne TV-kanaler, der seeren på enkelt vis kan spille en aktiv rolle i utforming av kanalen. Seeren, eller mer allment beskrivende brukeren, av en åpen kanal kan selv påvirke debatter ved å sende inn innlegg eller selv produsere og sende inn innslag de ønsker å få vist.

Nettfora er eksempler på Public Access-tjeneste. Behovet for å kommunisere er allment, men det har ikke alltid vært like enkelt å formidle innhold som i dag. For litt over to tiår siden var nettfora kun tilgjengelig for allmennheten som BBS (Bulletin Board System) der man abonnerte på et tema og fikk inn relevante meldinger som ren tekst i tilnærmet "lesehastighet". Selv om tjenestene kunne vært interessante for mange, ettersom de hadde et nokså begrenset tema, medførte grensesnittet at kun datakyndige fikk glede av dem. Fora man finner på nett i dag er langt mer brukervennlige og gjerne multimediale, men prinsippet er det samme,– man mottar informasjon man er interessert i. Med økende brukervennlighet har interessen for disse fora økt i så stor grad at de har blitt essensielle arenaer for svært mange temaer. Stadig flere må, for å holde seg oppdatert, debattere eller konsultere seg med andre på disse, noe som igjen gjør dem mer interessante og viktige.

Det er rimelig å forvente en liknende utvikling for videotjenester. I dag er det allerede enkelt å produsere videoinnslag og for noen enkelt å utveksle dem. Men arenaene for dette er ikke godt nok organisert og kan heller ikke tilby et enkelt grensesnitt som gjør at alle kan slutte seg opp om "videofora" og dermed danne essensielle formidlingskanaler.

I dag har vi de teknologiske forutsetningene som skal til for å videreutvikle dagens TV-nett i denne retningen, men det er likevel store utfordringer med hensyn til folks brukervaner. Det kan være naturlig å orientere seg i erfaringsmaterieell fra etableringen av eksisterende TV-kanaler i Norge, men her ligger det helt andre prinsipper til grunn. Det nærmeste man kommer av eksisterende tjenester er nettfora som står i kontrast til Public Service-kanaler der det som blir

formidlet snarere er laget etter tildelt oppdrag enn etter genuint engasjement. Våre riksdekkende kanaler NRK1, NRK2 og TV2 er eksempel på dette.

For å forsvare lisensavgiften, eller for at kommersielle kanaler skal nå ut til alle kundenes målgrupper, må innholdet i dagens TV-kanaler helst treffe så mange som mulig. Programmene får en generell profil, og ikke blir man ved et tema for ettertanke og refleksjon. Unntaksvis kan man holde seg til et tema over lengre tid dersom majoriteten av seerne tillater det. I så fall velger produsentene en tabloid profil for å vinne frem i kampen om seerne og den beste sendetiden. Dessverre handler en tabloid profil mer om å skape interessante nyheter enn å få frem faktiske forhold. Profilen fokuserer kraftig på kjente personer framfor sak, og smale grupper av samfunnet faller utenfor søkelyset. Det er det sistnevnte som er det viktigste å få gjort noe med. En åpen kanal bør være en beredskapskanal der grupper som vanligvis ikke syntes i mediebildet, enten fordi de ikke passer inn i en bred profil eller fordi de ikke har majoritetens interesse, likevel har en mulighet til å ytre sine meninger eller fremme sine saker. I denne sammenhengen er det verdt å trekke på et eksempel. I Göteborg 1998 brøt det ut en brann i et diskotek der de fleste pårørende var fremmedkulturelle. Det viste seg at nettopp *Öppna kanal Göteborg* kunne gå på luften umiddelbart med opplysninger om hva som ble gjort, hvor man kunne henvende seg, hva som burde gjøres m.v. av den enkle grunnen at der fantes alt som skulle til for fremmedspråklige sendinger.

Åpne kanaler er allerede vanlig i de fleste landene vi sammenligner oss med. England, Irland, Tyskland, Nederland, Danmark, USA og Canada har Public Access TV. I USA begynte fenomenet tidlig med Paper Tiger Television. Dette var et kollektiv av radikale "videokunstnere", med kontor i Lafayette Street på nedre Manhattan. Med svært lite midler har de imponert med pågangsmot, oppfinnsomhet og humor (Gripsrud, 2003), Slagordet deres lyder "Don't Just Watch TV, Make It!" De begynte med ukentlige sendinger i New Yorks kabelnett, i den såkalte Public Access-kanalen som alle kabeloperatører i USA har vært lovpålagt å stille til disposisjon for allmennheten. De mediekritiske, halvtimelange programmene var produksjonsmessig stort sett meget enkle, men gode på sitt vis. Det var her den venstrevridde medieforskeren Herb Schiller skarpt og vittig kalte New York Times "700 Pages of Waste". Rulletekstene var skrevet med tusj på grått innpakkingspapir og ble sveivet frem for hånd. Ikke noe fiksfakseri - tv kan lages av hvem som helst! Det dristigste Paper Tiger dro i gang, var imidlertid det såkalte "Deep Dish" prosjektet: De oppdaget at prisen for korttidsleie av satellittidistribusjon var overkommelig, og

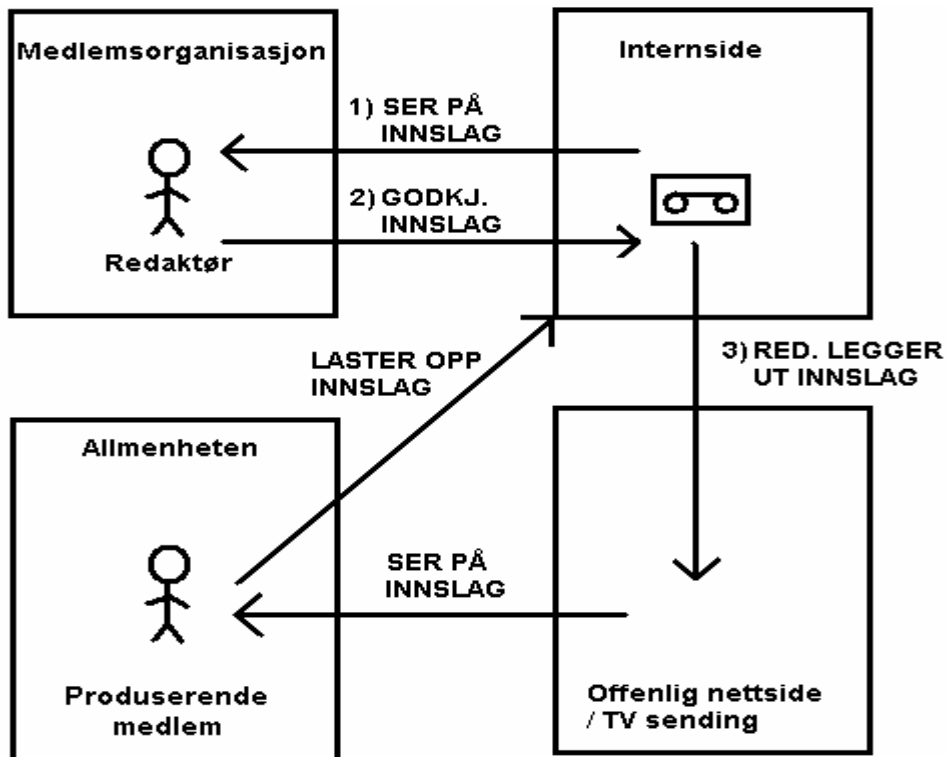
brukte dette til en primitiv nasjonal "networking" av samfunnskritiske Public Access-produsenter. Papirtigrene var nesten rørende begeistret over slik å ta innersvingen på "Makten" ved hjelp av "Maktens teknologi" (Gripsrud, 2003).

I dag får åpne kanaler gratis kringkasting under forutsetning at de ikke tar reklameinntekter, det vil si tapper det økonomiske grunnlaget til kommersielle kanaler. Det er også blitt vanlig med statlig støtte eller regulering som sikrer drift og den fysiske tilgjengeligheten.

### **3 *Åpen kanal* i Norge**

Arbeidet med å opprette en Public Access TV i Norge ble påbegynt i det små høsten 2000. Den 24. februar 2002 dannet åtte ikke-kommersielle produsenter og støttespillere foreningen *Åpen kanal*, Norges ikke-kommersielle fjernsynsprodusenters interesseforening for å sikre en rettferdig fordeling av felles sendeflate, samt bevirke til at infrastruktur for videoinnsamling og kringkasting finnes.

*Åpen kanal* er med andre ord ingen ny produsent av innhold. Ansvar for produksjonen ligger hos produsentene. Gjennom sitt medlemskap i *Åpen kanal* vil de også være representert i styret. Produsenten får tilgang til infrastruktur for kontribusjon (innsamling av innhold), scheduling (tildeling av sendetid) og distribuering (formidling) av innslag på nett og etter hvert digitale bakkesendinger. Figuren nedenfor er en konseptuel skisse av trinnene en produsent og en redaktør må gjennom for å få formidlet innhold på *Åpen kanal*:



Figur 3-1: Illustrert bruk av Åpen kanal

Arbeidet i styret har tatt form som pedagogisk aksjonsforskning, hvor fortløpende forhandlinger, samtaler og overveielser (deliberasjon) har avgjort formen og den videre utviklingen. For å bygge opp under foreningens mål om å videreutvikle demokratiet forutsettes også en gjennomført demokratisk fremgangsmåte internt i organisasjonen. Erfaringene så langt er at *det virker*. Det er karakteriserende at nettopp de produserende medlemmene selv har drevet fram arbeidet med Norges åpne fjernsynskanal.

Ved utgangen av 2003 hadde tolv nye produserende medlemmer sluttet seg til *Åpen kanal*.

Medlemstallet er nå 20, spredt over Midt-Norge og Øst- og Vestlandet med Trondheim, Oslo, Lillehammer og Bergen som samlingspunkter.

Medlemmene er i skrivende stund Amnesty Norge, Attac, Bergen senter for Elektronisk Kunst, Elektronisk Forpost Norge, Fokus-TV, Gateavisa, Indymedia, Medieverkstedet i Bergen, Mediefabrikken i Akershus, NORMAL, Noregs Mållag, Norsk Filmforbund, Norsk folkemusikk- og danselag, Norges fredsråd, PROION, Pro Cam Video, Student-TV Bergen, Student-TV Trondheim, SKA-TV og Utviklingsfondet.

Støttespillerne har vært fagfolk innen jus, økonomi, teknologi, journalistikk, pedagogikk, filosofi og ikke minst produsentenes selvoppnevnte råd/ankeinstans bestående av Ada Haug Grythe, Eva Joly, Rosemarie Köhn, Kaare R. Norum, Arne Næss, Kirsten Osen, Mosse Jørgensen og Leif Arne Rønningen. Rådet tilser at *Åpen kanal* virker etter intensjonene, det vil si at kanalen er åpen og gir allmennheten en reell (positiv rett til) ytringsfrihet.

For *Åpen kanal* gjelder følgende:

- Alle skal kunne nå ut til alle med egne synspunkter via fjernsynsmediet.
- *Åpen kanal* er ingen ny fjernsynsprodusent men innsamling og kringkasting av ideell film- og fjernsynsproduksjon i Norge.
- Mens NRK lager noe for alle, lager i *Åpen kanal* alle program for noen.
- Engasjerte når engasjerte med innhold av høy verdi for seeren i målgruppen.
- Selvstendig redaktøransvar hos den enkelte produsent sikrer ytringsfriheten.
- I *Åpen kanal* er det ingen som eier kanalen annet enn alle.
- Engasjement, ikke konsum
- Gratis kringkasting

## 4 Politisk støtte

Det er lagt et grunnlag for en åpen kanal i Stortingets behandling av mediemeldingen. Arbeidet som førte til at *Åpen kanal* ble prioritert i det framtidens digitalt, bakkebundet fjernsynsnett, var initiert av styrelederen Ola Tellesbø. Sommeren 1999 etablerte Tellesbø et initiativ sammen med Arne Næss og Erik Dammann for å utnytte ny og billig fjernsynsteknologi for å fremme ytringsfrihet, den åpne samtalen og refleksjon over dagens samfunn. Etter studier av det europeiske åpen kanal-konseptet og grundig forarbeid, ble den norske *Åpen kanal* etablert av seks ulike typer produsenter som samtidig la fram konseptet for Stortinget. Som et ledd i behandlingen var kulturkomiteen på befaring hos *Öppna kanaler* i Sverige og fikk med det en god forståelse av Public Access TV og dens nytte. 25. april 2002 ble det fastslått at Norge skal ha en slik fjernsynskanal. Det var særlig i forhold til distribusjonsmuligheter at Stortinget ga tiltaket en uvurderlig støtte. Timingen for *Åpen kanal* var svært heldig ettersom kulturkomiteen med dette fikk anledning til å knytte konseptet til utbyggingen av det nye distribusjonsnett for TV. Utskifting av det analoge, jordbunnede nettet, som sørger for at vi kan motta TV med antenne, var på dagsorden. I innstilling nr. 142 til Stortinget (2001-2002) drøftes muligheter som blir gitt ved at hver analoge kanal kan erstattes av fem digitale, samt hvilke prinsipper som skal ligge til grunn for en ny infrastruktur. *Åpen kanal* gis mye omtale, det står:

*”Et flertall av familie-, kultur- og administrasjonskomiteen, alle unntatt medlemmene fra Fremskrittspartiet, ville også peke på at man ut fra ønsket om en bredt anlagt offentlig debatt, må se på den faktiske tilgjengelighet allmennheten har til TV-mediet. I mange land har man ønsket å sikre denne tilgangen gjennom egen lovgivning som forplikter allmennkringkastere eller distributører å holde av en del av sendekapasiteten til livssynskanaler, ideelle organisasjoner og lignende. I Norden betegnes dette gjerne som åpne kanaler. Flertallet mente det var et interessant område sett i et ytringsfrihetsperspektiv. De oppfordret derfor staten til å legge til rette for dette når man ved innføring av nye distribusjonskanaler skal utarbeide regelverk og gi konsesjoner.*

*Komiteens medlemmer fra Arbeiderpartiet, Sosialistisk Venstreparti og Senterpartiet mente at en Public Access TV kanal i Norge (*Åpen kanal*) må være landsdekkende, ha formidlingsplikt og baseres på at det ikke skal sendes reklame eller drives eksponering mot sponsorer.*

*Et annet flertall, alle unntatt medlemmene fra Fremskrittspartiet og Sosialistisk Venstreparti, ville særlig understreke at følgende sentrale problemstillinger blir ivaretatt i forbindelse med utbyggingen av det digitale nettet: full dekning i Norge, muligheter for vesentlig flere kanaler, ivaretagelse av lokalfjernsyn og åpen kanal, samt garanti for kontroll over den fremtidige elektroniske programguiden. ”*

I Stortingets budsjettinntilling nr.2 (2002-2003), avsnitt 6.14, blir kulturdepartementet purret:

*”Komiteens flertall, alle unntatt medlemmene fra Fremskrittspartiet, vil peke på lokalradio, lokal-TV og åpen kanal som viktige instanser der ungdom kan få muligheten til å prøve seg på mediarbeid. Ikke minst er dette et aspekt ved åpen kanal som i vesentlig grad vil være rettet inn mot studentkanaler og formidling av ideelle organisasjoners arbeid.*

*Flertallet mener det er viktig at planene om en åpen kanal kan realiseres, og ber departementet bidra til at dette skjer.”*

Når formidling er ivaretatt vil det kun være behov for en mindre driftsenhet for å sikre oppetid, tilgjengelighet, teknisk støtte og veiledning. Grunnen til at driften blir så enkel, er det tette samarbeidet mellom *Åpen kanal*, Uninett, NTNU (Midgard Media Lab) og forsknings- og universitetsmiljøet for øvrig, og at produksjonsansvaret ligger hos produsentene i *Åpen kanal*.

Dessverre har ikke departementet fulgt opp med hensyn til finansiering av prosjektet eller annen form for tilrettelegging/realisering. Foreningen *Åpen kanal* kan derfor ikke garantere for sitt mål om å sørge for en infrastruktur til bidrag av innhold. For allmenn nytte er systemet som vi har utviklet avhengig av et stabilt driftsmiljø som krever finansiering.

*Åpen kanal* tar i mellomtiden sikte på å sikre yringsretten gjennom endring av yringsfrihetsparagrafen, §100. Når yringsretten er sikret vil formidlingsplikt være en selvfølge og en finansiering av drift en forutsetning for at loven skal gjelde. *Åpen kanal* vil dermed stå mye sterkere enn med de politiske intensjonene som så langt har blitt ytret.



I Stortingsmelding<sup>1</sup> nummer 26 (2003-2004) finner vi følgende tilråding til endring av Grl. § 100 fra Justis- og politidepartementet (godkjent<sup>2</sup> av statsråd 19.mars):

*”Tilgangen til mediene som ytringskanaler vil normalt være begrenset, i alle fall for dem som ikke er blant medienes foretrukne kilder. Å kjøpe seg tilgang til mediene gjennom annonser er uaktuelt for de fleste, først og fremst fordi det er svært kostbart. Ulike ordninger på medieområdet bidrar til å opprettholde et mangfold av ytringskanaler. Det er flere ordninger som er særlig innrettet for å sikre ytringsmulighetene for smale grupper i samfunnet. Etablering av såkalte «åpne kanaler», der ulike organisasjoner mv. redigerer sine egne programmer, kan slippe flere til i mediene. Slike ordninger er kjent fra utlandet. I Norge bygger foreningen Åpen kanal på et slikt konsept. Også andre former for ikke-kommersiell kringkasting kan bidra til at flere slipper til. I konsesjonen for et digitalt bakkenett for fjernsyn er det en forutsetning at det skal settes av kapasitet på rimelige vilkår til en åpen kanal. Konsesjonen til en slik kanal vil måtte tildeles etter åpen utlysning, jf. Stortingets behandling av St.meld. nr. 44 (2002-2003) Om digitalt bakkenett for fjernsyn.”*

I kapittel 7 går departementet inn for å grunnlovsfeste<sup>3</sup> infrastrukturkravet, det vil si kravet om at staten aktivt skal medvirke til at individer og grupper har en faktisk ytringsmulighet. Statens oppgave utvides her fra passivt å avstå fra inngrep til aktivt å sikre borgernes ytringsmulighet. Infrastrukturkravet innebærer en forpliktelse til å legge til rette for kanaler og institusjoner og for en åpen og opplyst offentlig samtale, kort sagt et overordnet statlig ansvar for oppbyggingen av et offentlig rom. En slik forpliktelse er direkte knyttet til hovedargumentene for ytringsfriheten: sannhetsargumentet, demokratiargumentet og autonomiargumentet.

---

<sup>1</sup> Fra kap. 7: <http://odin.dep.no/jd/norsk/publ/stmeld/012001-040019/index-hov008-b-n-a.html>

<sup>2</sup> <http://odin.dep.no/jd/norsk/publ/stmeld/012001-040019/index-dok000-b-n-a.html>

<sup>3</sup> Fra sammendraget: <http://odin.dep.no/jd/norsk/publ/stmeld/012001-040019/index-hov001-b-n-a.html>

## 5 Organisatoriske premisser

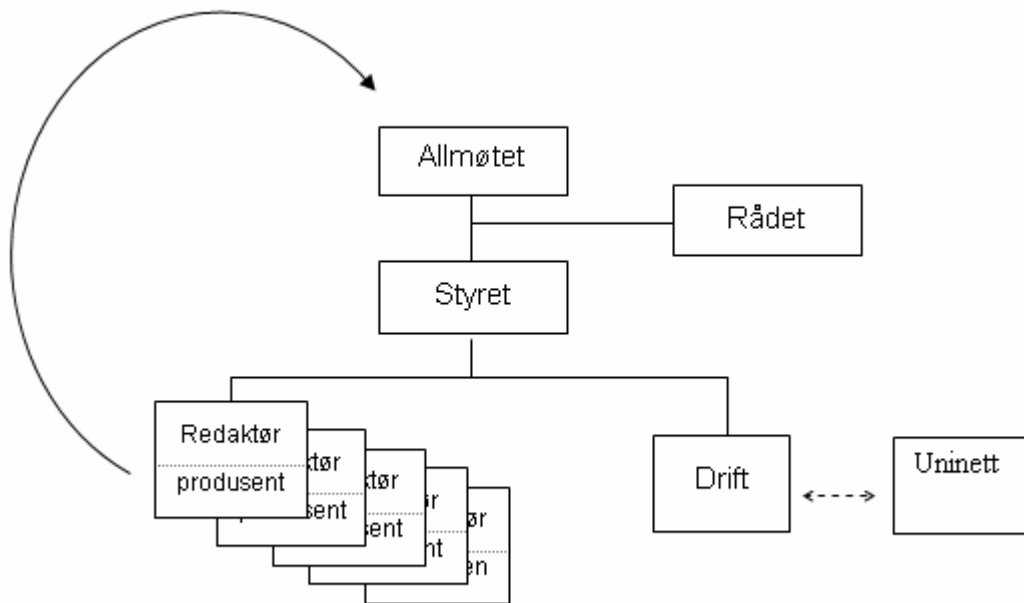
I en åpen kanal slipper alle til. Det gjør de ved å tegne medlemskap og utpeke en redaktør. Virksomheten tilknyttet kanalen må følge norsk lov og ikke diskriminere andre deltakere. Samtidig er *Åpen kanal* en forening hvor de produserende medlemmene fastsetter regler for bruk av felles sendeflate. Foreningstanken med selvstendige, likeverdige medlemmer, hvor *Åpen kanal* kun videreformidler deres produksjon, reder grunnen for gjennomføring av et viktig demokratisk prinsipp: Hvert produserende medlem beholder *selvstendige* redaktøransvar (jf. uavhengige aviser). Slik vil kanalen styrke den reelle ytringsfriheten, åpenheten og mangfoldet. Framfor å tenke seertall og mange grupper samtidig, vil det i *Åpen kanal* formidles program med høy verdi for små og varierte målgrupper, tydeliggjort med denne sammenligningen: Mens NRK *lager noe for alle*, lager i *Åpen kanal alle program for noen*.

De produserende medlemmene tar ansvar for å ivareta også framtidige deltakere. Videre må Norges åpne fjernsynskanal selv være et demokrati til etterfølgelse så vel som benytte de nyankomne demokratiske mulighetene fjernsynsteknologien nå bringer. Vi står overfor et gjennombrudd som ytterligere fører alle på like fot, uavhengig av bosted. Spredt befolkning og langstrakte land har fremtvunget representativt demokrati, enten etter en liberalistisk (flertallstyre) eller etter en republikansk modell (egentlig sterkere lokalstyre hvor lokalt valgte velger landets leder, avgjørelser sak for sak kan inngå). Til forskjell fra dette, som følge av teknologi og høye forsett, er *Åpen kanal* organisert for å bli det første riksdekkende *deltakende demokratiet* i moderne tid. Hvert produserende medlem har plass i styret.

Produsentene og deres redaktører utgjør *allmøtet*, som er kanalens øverste ledelse. Allmøtet er halvårlige, og hver medlemsorganisasjon kan stille med tre deltakere med én stemme hver. Grunnen til tre representanter framfor én, er å sikre bredest mulig deltakelse. Viktigheten av å legge til rette for bred deltakelse utdypes nedenfor. Tre representanter gjør også de produserende medlemmene mer fleksible overfor problemstillinger som oppstår i selve allmøtet. Til støtte i avgjørelser som krever en uavhengig part har de produserende medlemmene valgt inn et råd. Rådet ser også til at vedtektene respekteres.

Altså: Det er deltakerne i kanalen som utgjør allmøtet og styret. I allmøtet er de suverene (fastsetter regler/vedtekter), i styret utøver de, mens under produksjon forholder de seg til reglene

og vedtektene fastsatt av dem selv.<sup>4</sup> (Rousseau, 1752) Rådet er å sammenligne med Stortingets kontroll- og konstitusjonskomité, men er altså søkt uavhengig produsentene ved å trekke på ”samfunnsstøtter” med lengre fartstid, og som selv ikke har til formål å sende.



Figur 5-1: Organisasjonsstruktur for Åpen kanal

### 5.1 En oppskrift for framtidens folkestyre

Vi deltar i samfunnet for å bidra, men også for å få dekket eksistensielle behov. Nytt bidragene, eller hvis vi i det minste blir hørt, øker innsatsen, viljen og oppslutningen. Dette ser ut til å gjelde: Jo mer innflytelse, desto bredere deltakelse. (Dewey, 1916) Den omvendte følgen er vel så gyldig: Jo flere deltakere, desto mer folkestyre.<sup>5</sup>

En åpen fjernsynskanal åpner for begge deler, både innflytelse og deltakelse – samtidig. Dermed vil de kunne forsterke hverandre gjensidig og styrke vårt deltakende demokrati. Så vel som at deltakelse og innflytelse går hånd i hånd, vil mange deltakere også øke videotilfanget, samarbeidet grupper imellom og kunnskapen om medieproduksjon. Dette er et godt grunnlag for

<sup>4</sup> Dette er tett opp til samfunnspakten slik Jean-Jacques Rousseau beskrev den i 1762.

<sup>5</sup> John Dewey anså deltakelse og folkestyre som to sider av samme sak.

at det utvikles severdige program, noe som atter vil bevirke økt deltakelse og innflytelse. Disse ulike virkningene henger slik sammen at en liten endring i hver av dem i riktig retning, vil kunne snu en nedadgående samfunnsdeltakelse.

Det er minst tre måter å øke deltakelsen, innflytelsen og severdigheten (heretter forenklet til *deltakelsen* alene), uten at kanalen blir kostbar:

1. *Alle må kunne nå alle*, hvilket betyr landsdekkende videoinnsamling og kringkasting.
2. *Åpent og aktuelt*: Kanalen er åpen for alle og gjenspeiler både det kulturelle mangfoldet og det som opptar oss.
3. *Autonomi*: At fjernsynskanalens deltakere må råde over seg selv og over kanalens utvikling og sendeflate, og at de selv utgjør et mønster for deltakende demokrati, hvor alle deltar i alle beslutningene som vedgår dem.

### 5.1.1 Alle skal kunne nå alle

Reell ytringsfrihet dreier seg om muligheten til å kunne delta der samfunnsdebatten pågår. Nye medier har økt ytringsfriheten, men så fremt fjernsynet fører an i ”den store samtalen”, er det avgjørende å styrke ytringsfriheten i nettopp dette mediet. Menneskets på forhånd tildelte ukrenkelige rettigheter, ytringsfrihet innbefattet, gir selvrespekt, som igjen er viktig for deltakelsen og utviklingen. Den enkeltes deltakelse og utvikling er igjen viktig for samfunnet og demokratiet, som atter er viktig for den enkeltes moralske utvikling. En tilnærmet lik framstilling finner vi hos John Rawls (Rawls, 1993), John Dewey med flere.<sup>6</sup> Uten ukrenkelige rettigheter og frihet, bringes samfunnet inn på en skjev utvikling hvor allmennheten gradvis taper innflytelse og selvrespekt. Grunnsetningen for å unngå dette er at morgendagens fjernsyn gir mottakerne av første sending mulighet til å melde tilbake i samme medium til samme seergruppe. Det være seg med fordypning, rettelser og motmening. Landsdekkende videoinnsamling og kringkasting er helt grunnleggende for ytringsfriheten og Norges åpne fjernsynskanal. Sør-Korea fører an i så måte og er først ute (2002) med landsdekkende, kringkastet åpen kanal<sup>7</sup>

---

<sup>6</sup> Rawls mente at angitte rekkefølge løp før en vekselvirkning mellom dem kom i stand. Dewey satte likhetstegn mellom demokrati og deltakelse og mente at moralen utvikles i dette samspillet.

<sup>7</sup> Public access TV i Korea: [www.rtv.or.kr/eng/info/english.jsp](http://www.rtv.or.kr/eng/info/english.jsp)

Altså, siden fjernsynssendinger ofte er nasjonale, og siden den nasjonale politikken stadig blir viktigere enn den lokale politikken, fordrer det for en åpen fjernsynskanal at alle kan nå *tilbake* til alle – uansett bosted, betalingsevne og mening. Framfor skjevheter i ytringsfriheten, er det heller seerne som må velge hva som er verdt å følge opp.

Viktigheten av landsdekkende sendinger kommer også fram ved å betrakte to studier, en fra Sverige og en fra Danmark, for det er ikke opplagt at en åpen kanal gir betydelige demokratiske gevinster. I Sverige viste Jan Erik Lindell og Peder Olssons kvalitative studie at folkebevegelsen uteblir i Öppna kanaler. Kanalene klarer ikke å være aktuelle nok til å ha innflytelse på den politiske debatten.(Lindell, 1998) Sverige er fordelt på 11 *Öppna kanaler*. Ingen av dem produserer mer enn to timer per dag. Nivået varierer, men det holder gjennomgående et lavt teknisk og innholdsmessig nivå. I Danmark bevilger kulturministeriet et beløp tilsvarende 1% av kringkastingsavgiften til ikke-kommersielle kringkastingsformål. En statlig, dansk undersøkelse av Per Jauert og Ole Prehn om ikke-kommersielle tv og radio viser at engasjement, oppriktighet, grundighet og ikke minst et rolig tempo kjennetegner Public Access tv i Danmark. Forskjellen mellom Sverige og Danmark bekrefter Jauert og Prehns slutning om at statlig støtte betaler seg (Jauert, 2002). Hvis det i Norge ikke skal skytes inn titalls millioner slik som i Danmark, er det helt nødvendig for deltakelse, innflytelse og severdighet at kanalen er landsdekkende. Først da er videotilfanget tilstrekkelig, først da blir kanalen severdig.

Mens kringkasting på enkelt vis gjøres landsdekkende, har effektiv videoinnsamling vært vanskeligere. Her er det Åpen kanals viktige samarbeid med Uninett og NTNU kommer inn. Uninett (forskningsnett i Norge) strekker over hele landet til to eller flere tettsteder/byer i hvert fylke, til sammen over 70 steder<sup>8</sup>. Alle steder hvor høyskoler, universiteter og andre kunnskapsenheter har et knutepunkt, kan allmennheten nå fram til Åpen kanal over Uninett, enten ved å laste opp egne innslag selv, eller ved at det opprettes mottaksenheter som står for overføring og opplasting. Hos NTNU bistår særlig det fjernsynstekniske miljøet omkring Midgard Media Lab med verdifulle løsninger.

Landsdekkende videoinnsamling og sending vil i særlig grad styrke både utkant-Norge og landsomfattende ideelle organisasjoners kommunikasjonsevne. For de siste vil det si at alle med felles bakgrunn, ståsted eller interesse kan samordne produksjon på tvers av landsdel og by. Et

---

<sup>8</sup> Norsk Mediebarometer 2000

slikt samarbeid vil også være et viktig bidrag til økt nivå og kanalens severdighet. Dessuten vil landsdekkende sendinger gi filmskapere med flere sikre visningsmuligheter.

Landsdekkende videoinnsamling og kringkasting likestiller produserende medlemmer i *Åpen kanal* også i forhold til ledelsen av selve kanalen. Produsentenes styremedlem vil, uavhengig av bosted, enkelt kunne delta på alle styrets møter som foregår på nettet. Fremfor å opprette et hovedsete, kan landsdeler/byer likestilles, og rullering av lederverv utprøves.<sup>9</sup> Det er også gode grunner for å trekke på andre byer enn hovedstaden, slik at Norges åpne fjernsynskanal ikke blir et "Oslo-fenomen". For driftens del, det vil si å sørge for at sendeflaten er fysisk tilgjengelig for produsentenes redaktør, vil den med fordel – til forskjell fra NRK og TV2 – legges til Trondheim. Der finnes mye av Norges fremste kompetanse på fjernsynsteknologi, hos viktige samarbeidspartnere av *Åpen kanal*, slik som Uninett og NTNU.

### 5.1.2 Åpent og aktuelt

I Norge slipper 3 % av befolkningen til i fjernsyn, og langt færre etter egen plan. Begrenset deltakelse begrenser det frie ord og ensretter kulturpåtrykket. Ingen liker å bli pådyttet meninger uten å kunne svare. Istedenfor budskapsdrevet<sup>10</sup> innhold er kommersiell media dessverre ikke bare lønns- og pengestyrt, men også eierstyrt, da eiere ansetter redaktører ut fra passende politisk ståsted<sup>11</sup>. Når mediene samles hos få, innsnevres antall meninger i omløp.

Framfor ensidig kulturpåtrykk, fordeler åpne fjernsynskanaler mediemakten. Meningsdannelsen kan utvikles i et større, levende og mangfoldig fellesskap. Viktige ordskifter om fred, verdenshandel, lokal selvbestemmelse, forurensning, ytringsfrihet, med mer kan holde fram lenge etter de er "salgbare", hvis sakene opptar oss og vi kan mene noe om dem.

Skal Norges åpne fjernsynskanal være allmenn i vid forstand, må den også ivareta framtidige deltakere. Dette er en organisatorisk utfordring, blant annet fordi det er vanskelig å forutse hvem som kommer til å delta. En egnet framgangsmåte, hvilket også er den *Åpen kanal* omsider har valgt, er å oppfordre deltakere og andre til å tenke over hvilke saker som opptar dem, og ut fra

---

<sup>9</sup> Dette er også Jean-Jacques Rousseaus forslag.

<sup>10</sup> Budskapsdrevet betyr at innholdet følger det deltakerne har på hjertet framfor ønsket om å tjene penger.

<sup>11</sup> Artikkel i Aftenposten 20/1-04 [http://www.aftenposten.no/kul\\_und/article.jhtml?articleID=713592](http://www.aftenposten.no/kul_und/article.jhtml?articleID=713592)

dette mangfoldet sette av plass til det meste. Dette forener organisasjoner og grupperinger på sak. Fordelene er flere. For det første gir det mulighet for framtidige deltakere å finne en plass – også når kanalen begynner å bli full. Denne smidige bruken av sendeflaten ivaretar åpenheten, samt forenkler fordelingen. Dernest gjør samhandling på sak kanalen saksrettet og aktuell. For det tredje gjør en saksinndeling det mulig å skape møteplasser i kanalen, hvor mennesker møter mennesker. De kobles opp i flerveis forbindelser over nettet og bringes i aktørposisjon og samhandling. På en typisk møteplass vil det utveksles erfaring og kunnskap om temaer som opptar og beriker oss. Innholdet kan være nært og virkelig, med opplevde vansker, så vel som nye muligheter i folks hverdag. Emner, eller saker, kan være nærmiljø, kunst, verdispørsmål og erfaringer fra "livet selv". Denne form for aktualitet kan også styrke deltakelsen.

*Møteplassene* kan være tverrfaglige og gå på tvers av ulike samfunnsgrupper, med likesinnede eller med annerledestenkende, uformelle og formelle, vitenskapelig eller ikke. Styrets arbeid er ordnet som pedagogisk aksjonsforskning, og vil bli videreført som en møteplass i sendeflaten, hvor styrets erfaring på handlingsrettet læring og utvikling kan tjene som forbilde på andre møteplasser. Varierte møteplasser gjør det også mulig i forskningsøyemed å trekke erfaringer for videre utvikling av deltakende demokrati, og for å viderebringe deltakende pedagogikk på mange plan i samfunnet – også i skolevesenet.

### 5.1.3 Autonomi

Selv om produserende medlemmer går sammen om å dele en felles ressurs, er det helt grunnleggende at de beholder sitt redaktøransvar. Dette er grunnen til at Åpen kanals hovedvirkemåte i all enkelhet er å sikre en rettferdig fordeling og bruk av sendeflaten.<sup>12</sup> Ingen må miste eller fraskrive seg sitt redaktør- og rettslige ansvar.

Likeledes, ut fra den demokratiske nødvendigheten av innflytelse over eget befinnende, må de produserende medlemmene selv forvalte sin sendeflate. Dette er selvsagt, og slik Public Access TV drives. Kanalene er til for å styrke yringsfriheten og demokratiet, og mangel på råderetten over sendeflaten ville ha undergravd selve forutsetningen. Dette forklarer hvorfor det ikke er plass til noe ledd mellom distributør, som bringer fram sendeflaten, og deltakerne, som forvalter

---

<sup>12</sup> Foreningen skal fremme yringsfrihet i fjernsynsmediet og har også ansvar for å ivareta at framtidige brukere kan få plass. Dessuten tilrettelegger *Åpen kanal* infrastruktur for landsomfattende videoinnsamling og sending over luft og nett.

sendeflaten. Et ikke-demokratisk mellomledd vil bety sensur og tapt tilgjengelighet og ytringsfrihet straks det skulle bestemme hvem og hva som sendes.

Med Åpen kanal blir ytringsfriheten reell, og deltakerne tilkjennes én av sine ukrenkelige rettigheter.

I Åpen kanal er det indre demokratiet reelt, og deltakerne har selvrådighet over egen sendeflate.

Autonomien er ivaretatt. Når deltakelsen er bred, er kanalen folkestyrt. Grunnleggende er at *alle* deltar i alle beslutningene som angår dem – *alle* deltar i formingen av deres overenskomster. Framfor flertallets rett, konsensus, medianer og gjennomsnitt, søkes enkle grunnregler alle er enige om, og som gir alle lik rett. Alle skal, selv om alle må gi plass til alle, vinne mer ytringsfrihet enn det de på forhånd har – og ingen regler skal ramme noen enkeltpart/deltaker. Det er den grunnleggende demokratiske fellesforståelsen som skal bevirke til bred deltakelse, gode arbeidsformer og vedtekter.

## 5.2 Fordeling og omfordeling av sendeflaten

For å ivareta ytringsfriheten, åpenheten og mangfoldet legges det opp til en tematisk inndeling av kanalen i saker som opptar organisasjoner og enkeltmennesket. Styrets forslag til slike emner er følgende:

Livssyn, religion og filosofi

Samfunn, politikk og aktualitet

Kunst og kultur

Økologi og natur

Helse og omsorg

Utdanning og forskning

Arbeid og næringsliv

Produsentene velger ut emner de kan sende innenfor, enten alene, sammen med likesinnede eller med annerledestenkende. Selve fordelingen vil foregå etter på forhånd fastsatte regler, og er foreslått til å være disse:

1. Styret ivaretar mangfold og åpenhet ved å foreslå en allsidig, tematisk inndeling av sendeflaten.
2. Allmøtet vedtar endelig inndeling.



3. Dersom allmøtet ikke oppnår full enighet, vedtar rådet fordelingen av sendeflaten.
4. Rådet kan underkjenne en fordeling hvis den strider mot vedtektene.
5. Intet enkelt medlem får i utgangspunktet mer enn én halvtimes daglig sending til bestemt tid. Udekket sendeflate blir "møteplass", samproduksjon eller på annen måte delt med andre aktører.
6. Brukere av én sendetid har fortrinnsrett til denne. En omfordeling kan først skje ved at allmøtet fatter vedtak ved 3/4 flertall.
7. Dersom flere enn ventet ønsker å behandle ett og samme tema, kan styret gi temaet økt samlet sendetid, så fremt dette lar seg gjøre uten skade for andre produserende medlemmer.

## 6 Teknisk støtte

En åpen fjernsynskanal er også muliggjort gjennom teknologiske nyvinninger. Det er nå mulig på en kostnadseffektiv måte å samle inn og distribuere mengder av medieinnhold *fra* hele landet *til* hele landet. Terskelen for produksjon har i lengre tid vært lav, og mye god, ikke-kommersiell ideell produksjon blir ofte ikke vist i det hele tatt.

Parallelt med at det politiske grunnlaget ble lagt, jobbet et miljø i Trondheim med etableringen av en infrastruktur for innsamling av innhold. Det var med utgangspunkt i etableringen av en nasjonal Student-TV at behovet for å samle inn innhold over nett vokste frem. Og en infrastruktur som koblet Student-TV i Bergen, Oslo og Trondheim sammen kunne forenes med det mer omfattende tiltaket, *Åpen kanal*. Våren 2001 ble jeg som redaktør for STV Trondheim og prosjektleder for STV Norge kontaktet av Ola Tellesbø. Vi innså begge at vi hadde felles mål og komplimenterende utgangspunkt for å nå det. Trondheim hadde den tekniske støtten som skulle til for å gi allmennheten en kontribusjonskanal. Tilbake ville forskningsmiljøet få et større tilfang av innhold som ville medføre økt severdighet som igjen ville gi økt oppslutning om prosjektet. Det var også ønskelig for STV å få tilgang til en riksdekkende sendeflate. Derfor gikk STV inn for *Åpen kanal* og avtaler med våre samarbeidspartnere ble gjeldende også for dette prosjektet. Et større perspektiv og en politisk støtte bak prinsippene muliggjorde ett større mål samt gjorde partene mer villig til å forplikte seg. Relasjonene ble sterkere og mer formalisert til både Midgard Media Lab (NTNU) og Uninett. Midgard Media Lab er sentral ettersom de stiller med lagringsplass og utstyr for å gjøre innholdet tilgjengelig på nett i det rette formatet (se figur nedenfor). Institutt for telematikk ved NTNU har også støttet *Åpen kanal* med en avviklingsenhet (Playout) i forbindelse med mitt diplomarbeid. Formelt har NTNU gått ut av *Åpen kanal* som et medlem, dette fordi store offentlige aktører gjør prosessen med å få vederlagsfri distribusjon (formidlingsplikt) vanskelig. NTNU har likevel valgt å bidra ytterligere for at prosjektet blir realisert, og dermed få muligheten til å eksponere både studentrelatert og forskningsrelatert materiale på et tidligere tidspunkt. Utstyret brukes også i øvingsopplegg, prosjektoppgaver og diplomer. *Åpen kanal* fungerer her som en motivasjonsfaktor for studentene ved å gi de "hands on"-opplevelser tidlig i studiet, og med det styrke den praktiske relevansen av faget. Medierommet ved NTNU har også bidratt med videokonferanserom, studio og enheter for innspilling og opplasting av innhold. I løpet av høsten 2006 vil også institutt for telematikk kunne

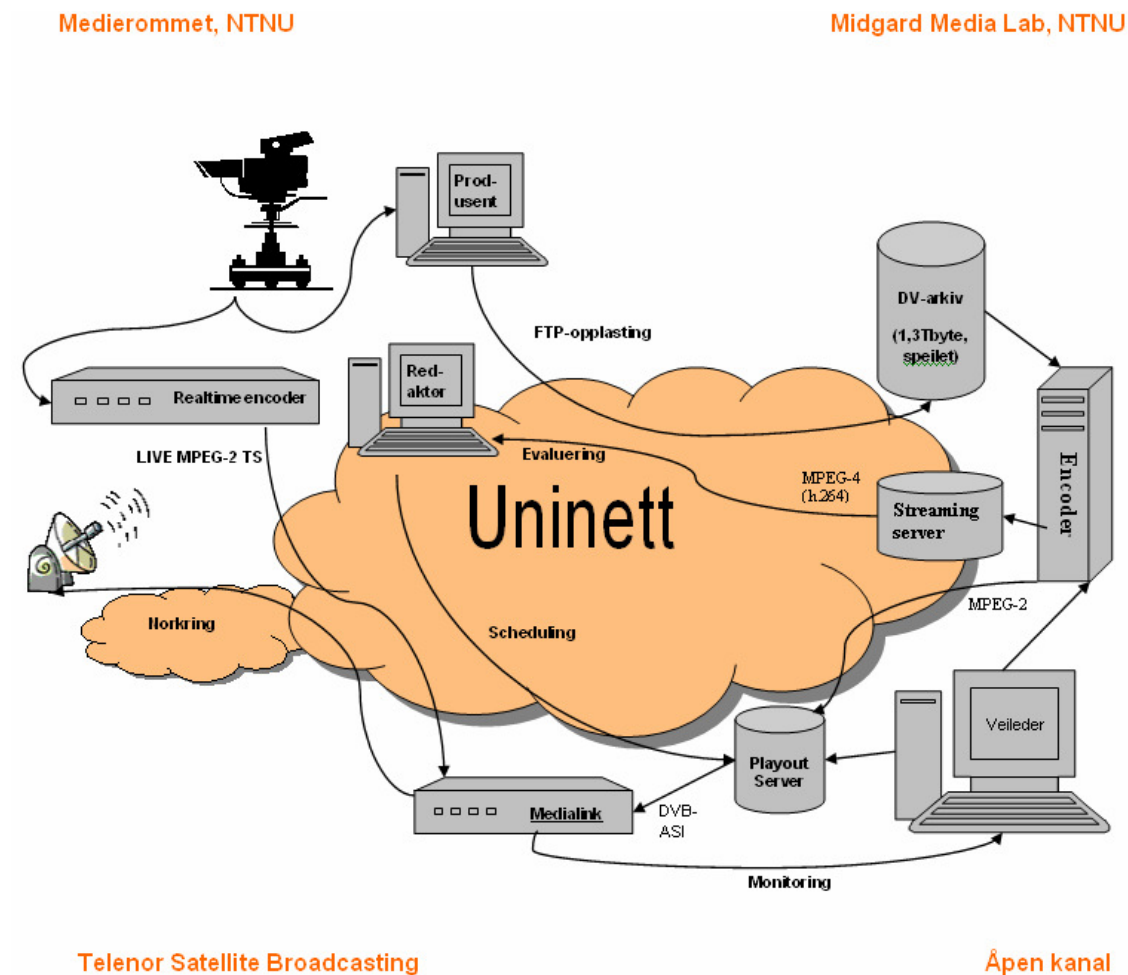
stille med ressurser i forbindelse at videreutvikling av åpen kanals pålitelighet skal inngå i et masterprosjekt. Høgskolen i Lillehammer stiller seg også bak konseptet for utprøving av distribuert samhandling med studio og utstyr for profesjonelle flerkameraproduksjoner.

Uninett, som forbinder landets høgskoler, universiteter og forskningsinstitusjoner, er en svært viktig støttespiller for *Åpen kanal*. Foreningens ulike medlemmer bruker dette nettet til innsamling av innhold der videoene lastes opp til en sentralt plassert server i Trondheim. Uninett bidrar også med linjer til distribusjon. Sendingene på nett er tilgjengelig for alle i unicast, og for brukere på Uninett også i multicast som gjør det mulig for flere å se på innholdet uten å øke belastningen på serveren. Uninett har formelt bekreftet sin støtte til prosjektet (Vedlegg 1) og argumenterer med at *Åpen kanal* vil være en verdifull arena for utprøving av ulike brukermønstre og måling av nettlast. I tillegg har Uninett investert i en rekke komponenter for live-streaming og utprøving av MPEG4 multicast. I samarbeid med Norkring eller Banetele vil Uninett også kunne sørge for distribusjonslinjer til Telenor sin jordstasjonen på Nittedal. Herfra vil videostrømmen kunne gjøres tilgjengelig på satellitt via en uplink dersom Telenor Satellite Broadcasting ser det hensiktsmessig å sette av kapasitet til *Åpen kanal*. Utstyr og programvareløsninger som utvikles og testes ut i perioden frem til 2009 vil kunne styrke kanalens fremtreden i digitalt jordbundet nett, både med tanke på teknisk pålitelighet og forutsigbarhet i kontribusjon av innhold. Ufordringer i tiden fremover blir ikke bare å utvikle brukervennlige løsninger for nettbasert kontribusjon og avvikling, men også å endre folks vaner i forhold til bruken av nett. *Åpen kanal* kan bli den første kanalen som er 100% nettbasert og når ut i et riksdekkende TV-nett. Men selv uten TV-nettet på plass vil det være mulig å prøve ut TV-sendinger. MHP innhold kan lages og spilles av på nett med utstyr vi allerede disponerer. Nettsendingene kan man se på en TV med en set-top boks som gjør signalet fra nettet visbart. Det vil også være mulig å teste ut sendinger over luft i Trondheim ved hjelp av utstyr som er tilgjengelig på Sintef og fjernsynsteknologisk laboratorium ved institutt for telematikk på NTNU.

*Åpen kanal* sender i dag MPEG4-multicast over UNINETT. Antallet potensielle seere er 200.000. Reelt antall seere bygges gradvis opp samtidig med at innholdet øker i mengde og bredde. Selv om den jevne brukeren forholdsvis enkelt kan anvende teknologien og tjenestene, trengs det å etablere nye brukervaner.

Det er på dette nettet også ønskelig å teste ut MHP over IP (MPEG-2) med hensyn på den nye funksjonaliteten som blir tilgjengelig i det digitale jordbundede nettet. Interaktivt innhold, som

MHP er tiltenkt, er ikke noe *Åpen kanal* har ressurser til å prioritere, men det kan være faglig interessant for studenter både på filmvitenskap ved Dragvoll og på ingeniørfag ved Gløshaugen å bruke *Åpen kanal* som en arena for forskning og utvikling av nye medier.



Figur 6-1: Teknisk infrastruktur for Åpen kanal

Figuren over viser hvordan hovedenhetene i kontribusjon og distribusjonssystemet fordelt mellom våre støttespillere. Mens NTNU eier utstyr som brukes, stiller Uninett linjene til disposisjon. Det eneste stedet hvor vi er avhengig av å trekke inn nye samarbeidspartnere er i distribusjonen. Norkring, som er en del av Telenor, kan bidra med linje til Nittedal der uplinken tilhørende Telenor Satellite Services er plassert. Dette innebærer ingen ekstra kostnader for Norkring. Det er kun en reservasjon av kapasitet på allerede eksisterende linjer. Fra før er Norkring koblet inn i Midgard Media Labs skap i Tyolt-tårnet der streaming serveren og DV-arkivet er plassert. Vi trenger med andre ord bare "patche" oss inn i deres nett som allerede

strekker seg helt frem til uplinken. Alternativt kan man inngå et samarbeid med Banetele. De har gitt oss et tilbud på leie av 10 Mbps linje mellom Uninett ved UiO og uplinken i Nittedal. Prisen er fremforhandlet gjennom Frode Storvik som har en rabattavtale gjennom Uninett. For ca 50.000 i etablering og 6.000 i månedlig leie kan vi føre signalet helt frem til distributør. Men det er dessverre her den store kostnaden påløper. Markedsprisen for å sende en 3,5Mbit TV-strøm på satellitt er 4500 kroner pr. time. Vi får det noe billigere gjennom en utprøvningsavtale med Telenor Satellite Broadcasting, men selv denne prisen er for høy i forhold til vårt budsjett. Åpen kanal må derfor belage seg på å vente med sending til Telenor ser det hensiktsmessig med et utvidet samarbeid, eller staten pålegger Telenor formidlingsplikt. Det sistnevnte vil etter all sannsynlighet medføre at distributør tildeles en statlig kompensasjon som følge av nye EU direktiver. I alle tilfeller er kanalen sikret plass på digitalt jordbundet nett, men før den tid bør kanalen få tilgang til distribusjon både med hensyn på behovet i dag og den verdifulle erfaringen vi trenger frem til full drift i det nye nettet (sannsynligvis i 2009).

Det har hersket mye usikkerhet om det digitale bakkenettet vil bli tilgjengelig for Åpen kanal sine deltakere i praksis. Det nye distribusjonsnettet er blitt noe helt annet enn hva Stortinget ga grønt lys for i 2004.

Etter at Telenor gikk inn i NTV styres det fullt og helt etter kommersielle vilkår. Stortingets krav om at Åpen kanal skulle få sende på rimelige vilkår, som ikke kan bety noe annet en vederlagsfritt, ser ut til å være glemt i den videre behandlingen av konsesjonsvilkårene.

KrF og Yngve Slettholm stilte ikke de rette motkravene for at NTV skulle få konsesjonen. Ap har av en eller annen grunn ikke maktet å rette opp feilene. Mangelen på motkrav blir enda mer synlig tatt i betraktning at NTV i hovedsak er finansiert ved at Stortinget økte lisensavgiften og endret momsvilkårene for Nrk. Dette ga Nrk økte inntekter i størrelsesorden 500 mill/år. På denne måten har Staten gitt Staten v/Nrk de økonomiske musklene som trengtes for å kunne kjøpe konsesjonen. Ett triks som lot seg gjennomføre etter at Telenor trakk seg ut av NTV som dermed bare bestod av NRK og TV2. Etterpå ved en ubetydelig endring av konsesjonsvilkårene kom Telenor tilbake til NTV i en omgjort konsesjonsrunde. Nå virker det som at når NTV har følt seg sikre på gjennomslag hos KKD, har selskapet kjørt opp foretningsmodellen og byttet ut lederen.

Det å ha plikt til å formidle 1,5 Mbit/s transportstrøm 20 timer/døgn (beste sendetid unntatt) for å sikre en allemannsrett i etermediet, ville knapt vært merkbart for NTV, men et gjennombrudd for "den åpne, offentlige samtalen" (motivet i den nye GrL. §100). Det dreier seg om 2 % av kapasiteten, nærmere bestemt de 2 % av kapasiteten som NTV under ingen omstendighet hadde fått solgt fordi tv-selskapene først og fremst kjøper den beste sendetiden.

Statssekretær i KKD, Yngve Slettholm, foreslo at de 2 % til Norges åpen kanal skulle koste sivilsamfunnet 9 mill/år. Om sivilsamfunnet derfor ble utelukket ble ikke ansett som et problem. Krf håpet at koalisjonen av kristne produsenter kjøpte halvparten (9-12 timer/døgn) av det som skulle være Norges åpen kanal.

Straks Telenor inntrådte i NTV falt kostnadsberegningene til det halve. Hvis Giske og hans statssekretær, Mette Gundersen, ikke har klart å rette opp noe, vil det som lskulle være folkets kanal koste kr. 4,5 millioner/år. For å bringe dette på det rene, har vi bedt om å bli veiledet om konsesjonens innhold. Dessverre nekter KKD foreningen Åpen kanal veiledning/forhåndsvarsel angivelig fordi vi ikke er part i saken. Det at foreningen Åpen kanal er part i saken slo for øvrig Justis- og politidepartementet (JD) fast i stortingsmeldingen om endringen av Grunnlovens § 100. Der framgår det at det er foreningen Åpen kanal som arbeider for at Norge skal få en åpen kanal. Dermed er vi part i forhold til det som står om åpen kanal i bakkenettkonsesjonen.

Strategien til KKD har vært slik: Sjøsett skipet før noen hindrer det. Etterpå er det for sent å få skipet inn på det tørre.

Strategien til Åpen kanal er: Få skipet til å grunnstøtte på Grunnloven.

Frem til dette skjer kan *Åpen kanal* benytte seg av utstyr vi disponerer i samarbeid med NTNU og Uninett for å realisere sendinger på nett. Med dette vil det i tillegg til MPEG4 sendinger være mulig å utvikle løsninger for MHP over IP, som vil kunne brukes på et TV-nett den dagen det blir aktuelt. Utstyr vi har tilgang på i skrivende stund er følgende:

[Envivio 4Caster:](#)

Realtime encoder for streaming av live innhold.

[Envivio 4Coder:](#)

Batch encoder for generering av ISO MPEG4 filer til VOD.

[Envivio 4Forum:](#)

Webcasting system med sanntid synkronisering av video / grafikk.

[Envivio 4Sight:](#)

Streaming server som håndterer flere multicast og unicast strømmmer.

[Envivio 4Manager:](#)

Overvåking av headend med bla. fjernstyring av 4Sight / 4Caster.

[Envivio 4Mation:](#)

(Eg. Envivio Broadcast Studio) Editor for interaktivt MPEG4 innhold.

[Cardinal Playout:](#)

Avviklingsenhet for MPEG2-innhold med scheduling konsoller.

[Cardinal Studio Professional:](#)

Editor for interaktivt innhold etter DVB-MHP standarden.

[Divine HD-COFDM modulator:](#)

Genererer signal til utprøving av digital luftsending i Trondheim.

[Avid Trilligent:](#)

Videoserver for lagring og streaming av video.

## 7 Prosjektering av *Åpen kanal*

Denne delen er skrevet i samarbeid med styret i *Åpen kanal*. Prosjekteringen gir et bilde av hva *Åpen kanal* kan realisere i en pilotperiode med statlig støtte. Prøvesendinger vil kreve midler *Åpen kanal* ikke har og det er å håpe at denne delen kan være nyttig for å få i stand en søknad om tildeling. Med et beskjedent budsjett kan *Åpen kanal* tilby drift av arkivet og tilrettelegge for en fremtidig sending. Dette er avgjørende for at kanalen skal bli severdig når det digitale bakkenettet er klart. Til dette må det opprettes stillinger, men foreløpig baserer foreningen seg på frivillig innsats:

### **Institusjoner og organisasjoner bidratt i arbeidet med at Norge får en åpen kanal:**

Amnesty Norge	Norsk filmforbund
Attac	Norsk folkemusikk- og danselag
Bergen Senter for Elektronisk Kunst	Norges fredsråd
Dyrebeskyttelsen Norge	Pro Cam Video/Inteleki
Elektronisk Forpost Norge	Proion
Fokus-TV	SKA-tv
Indymedia	Student-TV Bergen
Mediaverkstedet i Bergen	Student-TV Trondheim
Noregs Mållag	Utviklingsfondet
Normal	

### **Andre viktige støttespillere:**

Høgskolen i Lillehammer	Norsk filminstitutt
NTNU	

### **Foreningen *Åpen kanal* har et råd bestående av:**

Ada Haug Grythe	Kaare R. Nørum
Eva Joly	Arne Næss
Mosse Jørgensen	Kirsten Osen
Rosemarie Köhn	Leif Arne Rønningen

### **Styret i *Åpen kanal* består av:**

Ola Tellesbø (leder)	Anita Nyholt
Einar Kirknes	Finn Røsland
Marius Mangseth	Olav Øverland
Tor Martin Møller	



Foreningen *Åpen kanal* ble dannet for å fremme at Norge får en åpen fjernsynskanal i Norge slik som i Sverige, Danmark, Tyskland, England, Irland, Nederland, USA, Canada, Sør-Korea og andre land. Arbeidet startet med å utrede de helt nødvendige betingelsene for Norges åpen kanal. I inneværende år vil foreningen jobbe mot en prøvedrift i 2007.

### 7.1 Konseptet åpen kanal

Konseptet *åpen kanal* stammer fra Tyskland på 1980-tallet.<sup>13</sup> En åpen kanal er en kanal der alle kan delta med eget innhold og få det kringkastet til alle fjernsynsapparater, under eget redaktøransvar.

En åpen kanal er verken en fjernsynsprodusent eller kringkaster, men en infrastruktur for allmennhetens adgang til fjernsynsmediet. Denne infrastrukturen skal være til fri benyttelse for alle, og ingen eier sendeflaten. Å etablere Norges åpen kanal er å innføre en slags allemannsrett i etermediet.

Deltakelse skjer ved at deltakeren registrerer seg, blir autentisert og betaler en mindre registreringsavgift. Etter å ha blitt registrert kan deltakeren sende gratis, forutsatt at vedkommende forholder seg til norsk lov og forbudet mot å utnytte sendeflaten kommersielt. Redaktøransvaret ligger hos deltakeren. Den enkeltes redaktøransvar er i Tyskland lovfestet.<sup>14</sup>

Redaktøransvaret til den enkelte deltaker er i samsvar med ytringsfriheten og forbudet mot forhåndssensur. Straffbare ytringer er derfor gjenstand for *etterkontroll*, ikke *forhåndskontroll*.

Tilretteleggingen av allmenn deltakelse i fjernsynsmediet skal underbygge formålet om "en aaben og oplyst offentlig Samtale", jf. Grl. § 100 sjette ledd. Stortinget skrev i Innst.nr.270 (2003-2004) "Om endring av Grunnloven § 100":

---

<sup>13</sup> Den første åpne kanalen ble etablert i 1984, i Berlin 1985. Se [www.bok.de/jahrestreffen.htm](http://www.bok.de/jahrestreffen.htm)

<sup>14</sup> Se blant annet Staatsvertrag über die Zusammenarbeit zwischen Berlin und Brandenburg im Bereich des Rundfunks (MStV), § 43(5). For øvrige lover, se [www.alm.de/index.php?id=259](http://www.alm.de/index.php?id=259)

Komiteens flertall, medlemmene fra Arbeiderpartiet, Høyre, Sosialistisk Venstreparti og Kristelig Folkeparti, viser til at et flertall i familie-, kultur- og administrasjonskomiteen ved behandlingen av digitalt bakkenett for fjernsyn (Innst.S.nr.128 (2003-2004)) understreket at det som en del av det digitale bakkenettet skal settes av kapasitet til en åpen kanal, der frivillige organisasjoner, livssynsorganisasjoner og andre ikke-kommersielle virksomheter kan tildeles sendetid på rimelige vilkår. Det ble her vist til at det har vært gode erfaringer med tilsvarende kanaler i andre land, og flertallet mener det er viktig i en tid med sterk kommersiell dominans i mediebildet å sikre at ikke-kommersielle tilbud kan slippe til, og at man søker å finne finansielle løsninger som gjør at dette tilbudet også blir reelt.

Flertallet vil vise til at etableringen av en slik åpen fjernsynskanal i Norge er et eksempel på hvordan myndighetene kan legge til rette for en åpen og opplyst samtale.

En åpen og opplyst offentlig samtale forutsetter reell ytringsfrihet for alle.

## 7.2 Lokalisering

Tilretteleggingen av Norges åpen kanal vil måtte bestå i å plassere ulike servere i grensesnittet mellom internett og fjernsynsnett. Foreningen *Åpen kanal* mener det mest hensiktsmessige er å plassere disse serverne midt i det nasjonale datanettet, i Midt-Norge, nærmere bestemt Trondheim. En slik plassering vil være et positivt signal tatt i betraktning at NRK har hovedkontor i Oslo og TV2 i Bergen. Tatt hensyn til fjernsynsteknologimiljøet ved NTNU og UNINETT, vil en slik plassering også kunne gi en betydelig synergivirkning på teknologiutviklingen. Tatt hensyn til andre kulturtiltak som er under etablering i Trondheim, kan samlokalisering gi mer kultur for de samme midlene.

På dette grunnlaget har vi tillatt oss å beskrive én mulig plassering av den teknologiske infrastrukturen til Norges åpen kanal. Denne beskrivelsen utelukker ikke at andre plasseringer kan bli valgt.

Med Trondheim for øyet peker *Nova* i *Folkets Hus* seg ut. Husets eier, AØF, ønsker foreningen *Åpen kanal* velkommen til å etablere tilretteleggingen av Norges åpen kanal hos dem.

*Nova* byggetrinn 2 skal på toppen av Folkets hus bygge en sal med 1100 seter for kongress og 1400 for kino og plass til 2200 når stoler er fjernet for events. Det er foreslått at salen rigges for fjernsynsproduksjon med permanent lysrigg, kontrollrom for NRK og tilkoblings- og oppstillingsplass for NRK og andre fjernsynsstasjoners OB-buss utenfor Folkets Hus.

Plassering av Norges åpen kanal på *Nova* inngår også i *AV Arena Norway*'s<sup>15</sup> strategiske satsing for å forsterke og synliggjøre Trondheim og Norge som et internasjonalt ledende kunnskapssenter innenfor den digitale medieindustrien. *AV Arena Norway* vil blant annet realisere forretningsmuligheter som åpner seg gjennom grunnforskning og anvendt forskning.

---

<sup>15</sup> *AV Arena Norway*'s styringsgruppe: prof. Andrew Perkis (styreleder Midgard Media Lab/NTNU), Karl Klingsheim (leder Technology transfer Office/NTNU), Egil Akselsen (adm.dir. Trondheim Kino), Tore Sandvik (fylkesordfører Sør-Trøndelag Fylkeskommune), Tore Mortensen (adm.dir. SONY Norge) og Morten Westvik (Innovasjon Norge i Korea).

Strategien er å forene teknologiutvikling og anvendelse som må forstås som to integrerte deler som fungerer best sammen. Sentrale stikkord er nye verdikjeder, teknologiutfordringer, forretningsmuligheter, alliansebygging, finansiering og gjennomføring av prosjekter og tiltak. Viktige prosjekter er blant annet etableringen av fiberringen i Trondheim, samarbeidet mellom St. Olav Hospital og SONY, Kosmorama, Rock og Popsenter (ABM i Dora) og Mediehuset. Et annet viktig prosjekt er nettopp utbyggingen av NOVA trinn2 med tilliggende utviklingsplaner. Dette omfatter etablering av digitale produksjons- og visningsfasiliteter, testlab m.m..

Foreningen *Åpen kanal* anser Nova som egnet for realisering av både den internettbaserte adgangen til Norges åpne kanal, distribuert samhandling, Media Literacy og, ikke minst, for å koble sendingene opp til fjernsynsnettet. En plassering på Nova vil kunne bidra til at pilotens milepæler nås innenfor gitte rammer. Et annet forhold er at plassering på Nova i vesentlig grad reduserer usikkerhetsfaktorer omkring båndbredde og tilkobling til fjernsynsnett, som ellers kan være kostnadskreven.

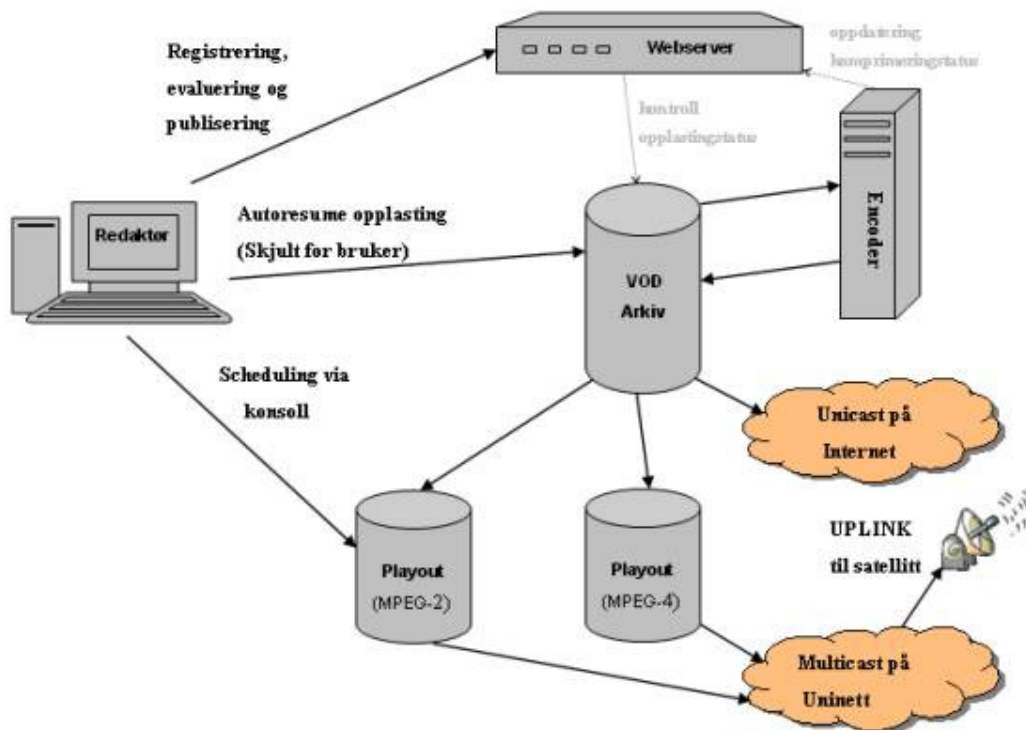
Leiepris i Folkets hus er i utgangspunktet kr 1100/m<sup>2</sup>/år. inkl. felleskostnader med strøm. Det kan være mulig å plassere pilotstaben sammen med *Midgard Media Lab* (NTNU) på de foreslåtte 60 m<sup>2</sup> for videoredigeringsstudioer. Foruten denne plassen har prosjekteringen tatt høyde for annen bruk av fellesareal, scenen innbefattet med dertil kostnader.

Budsjettet angir kr 360.000 for lokaler. Dette beløpet er et estimat som ikke binder foreningens virksomhet til et bestemt sted og vil være tilstrekkelig for Nova.

### 7.3 Oppgradering av eksisterende system

Foreningen *Åpen kanal* har med frivillig innsats og hjelp av NTNU og Uninett utviklet et video- og avspillingssystem som i ett års tid ble utprøvd. Alle med tilgang til bredbånd som hadde registrert seg som deltakere, fikk lastet opp sin video til et videoarkiv. Videoen ble tilgjengelig både for Video On Demand (VOD/unicast) og nettsending (multicast) døgnet rundt. Video- og avspillingssystemet ble sydd sammen med et egenutviklet dataprogram. Programmet hadde til formål å demonstrere systemet og hadde enkelte tekniske ”snarveier”. Piloten skal gjenopprette og utbedre det allerede eksisterende systemet. Opplastet videofil til videoarkivet skal foruten automatisk komprimering for unicast (VOD over nett) og multicast (”fjernsynssending” over nett) også komprimeres for broadcast (satellitt/bakkenett).

Deltakeren skal videre kunne fastsette tidspunktet for avspilling. Skjematisk ser systemet slik ut:



Figur 7-1: Video- og avspillingssystemet

Det å sikre alles adgang til Norges åpen kanal er pilotens andre milepæl og vil, når etablert, være det første virkemiddelet til å stimulere deltakelsen. De erfaringene foreningen *Åpen kanal* har gjort tilsier at deltakelsen i løpet av ett års tid vil nærme seg 100 aktive deltakere. Dette fordi at kun sendinger på internett vil være operativt på dette trinnet, ikke regulære fjernsynssendinger.

*Åpen kanal* anser den åpne standarden MPEG4<sup>16</sup> å være riktig format å satse på i motsetning til de vurderte systemene Real og Windows-media. Når standardformatet for fjernsynssendinger skifter fra MPEG2 til MPEG4, vil dermed all teknologi utviklet i piloten videreføres og anvendes.

<sup>16</sup> MPEG4 utviklet av MPEG (Moving Picture Experts Group) er en ISO/IEC standard mellom digital-TV, interaktive grafiske applikasjoner og multimedia. Ikke-profit organisasjon *MPEG4 Industri Forum* jobber for etableringen av standarden blant utviklere, tjenestetilbydere og brukere.

## 7.4 Etablering av sikre rutiner for registrering av redaktører

Det legges opp til følgende prosedyre for å sikre at den som registrerer seg, er den han/hun gir seg ut for å være: Den som ønsker å benytte seg av Norges åpen kanal i pilotperioden må registrere seg med fullt navn, adresse, epost og personnummer, og betale en registreringsavgift.

Autentiseringen av redaktørene skjer mot postforsendelse til oppgitte adresse for at deltakeren kan skrive under på sitt redaktøransvar, mot oppgitte email for å tildele brukernavn og passord, mot gyldigheten av personnummeret (basert på sjekksummer) og mot at deltakerne er villig til å betale en mindre sum for å sende.<sup>17</sup> Pilotsystemet vil automatisk videresende registrerte redaktører til *Medietilsynet*, jf. kringkastingslovens § 2-1 tredje ledd og kringkastingsforskriftens § 1-3 om registreringsplikt. Foreningen *Åpen kanal* anser det nødvendig for sikker autentisering at oppgitte adresse sjekkes mot folkeregisteret. Tilgang til folkeregisteret vil det derfor søkes om.<sup>18</sup>

Autentiseringen med tilhørende personopplysninger medfører at video- og avspillingssystemet må meldes til datatilsynet, jf. personopplysningsloven (pol) § 31. På grunn av systemets sentrale plass, anser foreningen *Åpen kanal* det riktig at styreleder står som behandlingsansvarlig, jf. pol § 2 nr. 4. Databehandler er daglig leder, eventuelt teknisk ansvarlig, jf. pol § 2 nr. 5.

Den som sender tar redaktøransvar. Vedkommende redaktør plikter å sette seg grundig inn i redaktøransvaret i henhold til norsk lov, derunder å gjøre det tydelig hvem som sender og er redaktør, jf. kringkastingsforskriften § 7-8. Dertil forplikter redaktøren seg til at ”sendingene skal være ikke-kommersielle. Reklame, snikreklame, produktplassering, pengeinnsamling, sponsoreksponering eller virksomhet som tar sikte på økonomisk gevinst, er utelukket.”<sup>19</sup>

Det ikke-kommersielle hensynet understreket Stortinget i Innst.S.nr.128 (2003-2004) om digitalt bakkenett for fjernsyn og Innst.S.nr.270 (2003-2004) om endring av Grl. § 100: “det

---

<sup>17</sup> Foreningen *Åpen kanal* foreslår kr 1000 i registreringsavgift.

<sup>18</sup> En alternativ autentisering vil være å benytte *Sikkerhetsportalen* (<http://www.brreg.no/sikkerhetsportal/index.html>). Foreningen *Åpen kanal* vurderer denne.

<sup>19</sup> Ordlyden er hentet fra foreningen *Åpen kanals* vedtekter, pkt. 3 ii.

er viktig i en tid med sterk kommersiell dominans i mediebildet å sikre at ikke-kommersielle tilbud kan slippe til". I Tyskland er det ikke-kommersielle hensynet lovfestet.<sup>20</sup>

Den sanksjonsmuligheten foreningen *Åpen kanal* anser mulig rettslig sett, er *utestenging* fra Norges åpen kanal ved brudd på norsk lov og det ikke-kommersielle vilkåret.<sup>21</sup> Utestenging må forutsette at vedkommende blir straffet for ytringen eller bryter det ikke-kommersielle vilkåret. Tap av adgangen til Norges åpen kanal kan oppfattes som tap av ytringsfriheten. Dette er ikke tilfellet. Det er kun adgangen til Norges åpen kanal som tapes, ikke til andre medier.

Teknisk sett vil utestenging skje ved at vedkommendes brukerkonto til kanalen stenges, videoinnslag på sendeplanen fjernes og VOD-innslag fjernes. Stengt brukerkonto betyr at verken innslag kan settes på sendeplanen eller nye innslag kan lastes til videoarkivet.

For å forenkle problemstillingene i piloten, som begrenser seg til tre år, foreslår foreningen *Åpen kanal* at utestenging varer piloten ut.

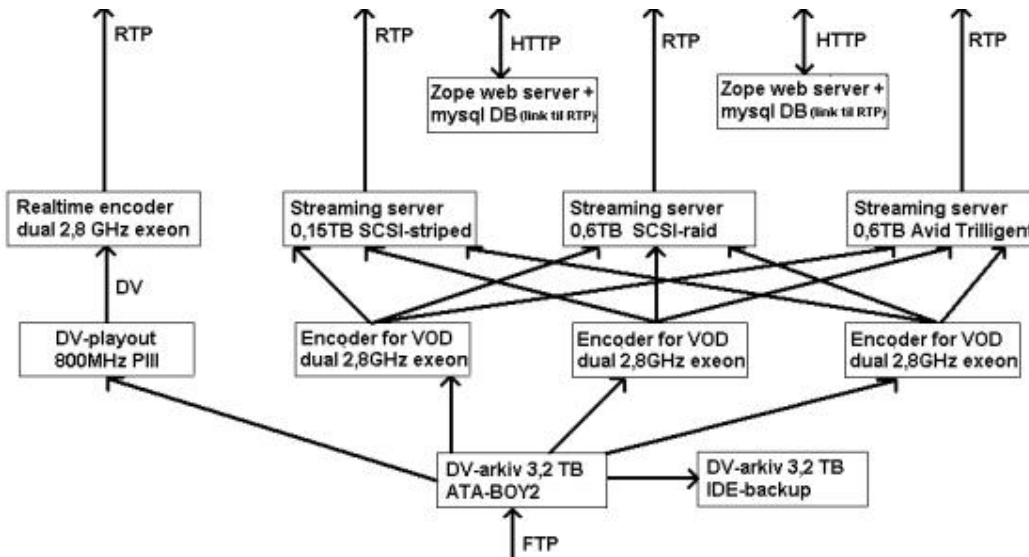
## **7.5 Kvalitetssikring av driften**

Når systemet settes i drift, blir det nødvendig med flere enheter for å sikre oppetid og tilgjengelighet. I figuren nedenfor er en driftssikker løsning foreslått, basert på dagens teknologi. Alle komponentene kan oppgraderes i takt med framtidige behov. Dette er meget nyttig med tanke på diskplass og støtte for framtidens HD-tv. Det er også nødvendig å etablere gode serviceavtaler. For øvrig vil det første pilotåret anses som en test av egenutviklet programvare.

---

<sup>20</sup> Se MStV (ovenfor) § 43(3)

<sup>21</sup> Se Foreningen *Åpen kanals* vedlagte rettslige redegjørelse « Lovfestet ytringsrett, ikke bare ytringsfrihet». Også tilgjengelig på [www.openchannel.no](http://www.openchannel.no)



Figur 7-2: Infrastruktur for kontinuerlig drift

## 7.6 Etablering av Media Literacy for barn og ungdom

Den mediale offentligheten er nærmest blitt synonymt med selve offentlighetsbegrepet. Dette har skapt et demokratisk underskudd for de som selv ikke har en kanal inn i denne offentligheten. Media Literacy, eller mediekyndighet som uttrykket kan oversettes til på norsk, viser til personers evne til å forholde seg til den mediale offentligheten. Innen akademiske disipliner som pedagogikk har man med begrepet mediekyndighet gjerne fokusert på hvordan personer fortolker og blir påvirket av ulike medietuttrykk. Man tar altså utgangspunkt i *personen som mediekonsument* (seer, leser, mottaker av andres budskap). For åpne kanaler eller public access-tv er derimot utgangspunktet *personen som medieprodusent*. Dette forholdet gjenspeiles i de Media Literacy-tiltakene som finnes innenfor slike kanaler.

## 7.7 Flytting av produksjonslinjen over til satellitt/fjernsyn: En mulighetsstudie

En utprøving på satellitt vil være helt nødvendig for å vinne erfaring om hva Norges åpen kanal blir og kan bli. En slik utprøving vil forutsette at fjernsynsdistributører pålegges formidlingsplikt av Norges åpen kanal. En generell formidlingsplikt vil sikre at sendingene ikke bare går opp på satellitt, men også fram til up-linken, som skyter signalet opp til satellitten, og ut til kabelanleggene.



For å sende på satellitt vil det i fra video- og avspillingssystemet måtte være en dedikert linje til en up-link. For Telenors del vil det si en up-link til Nittedal jordstasjon. Derfra går signalet opp til satellitt og ned til paraboler. Hvert kabelanlegg har egne paraboler slik at det som går over satellitt, også går ut i kabelanleggene forutsatt formidlingsplikt. Formidlingsplikten må innbefatte linjen til up-linken.

Markedsprisen for satellitt og dedikert linje til up-linken er henholdsvis i størrelsesorden 3 millioner og ½ million per år. De faktiske kostnadene av å bli pålagt formidlingsplikt av en ekstra kanal på allerede eksisterende infrastruktur er betydelig lavere. Evalueringen bør estimere denne så vel som inntjeningen som følger av økt salg av parabol og tilknytning til satellitt på grunn av Norges åpen kanal.

Antall deltakere vil med utprøving av fjernsynssendinger over satellitt øke dramatisk, og kan på få år nå produksjonsmengder opp mot 8-10 timer ny produksjon/dag hvilket er tilfellet for Berlin med sine 3,5 millioner innbyggere. Dette forutsetter selvsagt at de geografiske avstandene i Norge overvinnes med det internettbaserte opplastings- og avspillingssystemet foreningen *Åpen kanal* har utviklet og som vil bli oppgradert i pilotens første år.

Et eksempel fra Tyskland er prosjektet *Kids On Media* ([www.kids-on-media.de](http://www.kids-on-media.de)) hvor barn i 8-9 års alderen slipper til både bak og foran kameraet (selvsagt med samtykke fra foreldrene). Gjennom denne typen tiltak lærer deltakerne hvordan man selv kan lage tv eller radio. Dermed beveger de seg fra kun å være mediekonsumenter til også å bli produsenter. I tillegg avmystifiseres viktige sider ved media – å lede eller lage programmer er noe alle kan gjøre!

Oppslutningen rundt "reality-tv"-program som Big Brother og Idol viser stor interesse blant barn og ungdom for det å være på tv eller se på andre som en selv. Felles for deltakelse i "reality-tv"-program er derimot at rammene er lagt av andre enn deltakerne selv. Føringerne for denne utviklingen er tungt kommersielle, og resultatet ligger nært opp til Andy Warhols påstand om at alle i framtiden vil få deres "15 minutes of fame". Sett fra et demokratisk ståsted gir denne utviklingen derimot en type mediekompetanse som gjør personer til mer selvstendige medieskapere.

Norges åpen kanal vil være viktig for å øke den generelle mediekyndigheten blant innbyggerne. Det er derfor verdifullt at Norges åpen kanal kan gi et allment tilbud til personer

som måtte ønske opplæring i og tilgang på utstyr for medieproduksjon. Som allerede nevnt er interessen for medieprosjekter størst i samfunnsgruppen barn og ungdom. Derfor vil piloten orientere seg mot skoler og andre barne- og ungdomsfora for å tilby kompetanse, fasiliteter og en infrastruktur for medieproduksjon. Dette blir et enkeltstående Media Literacy-tilbud testet og utprøvd i Trondheim (se avsnittet "Etablering av prosjektet"). Tilbudet skal utvikle egne prosjekter og tilrettelegge for at Media Literacy kan kombineres med skolens øvrige medieundervisning.

"Media Literacy"-tilbudet skal styrke deltakernes evne til å fortolke hvilken mediepåvirkning vi utsettes for og gi deltakerne forutsetninger for selv å gripe inn i den mediale offentligheten gjennom egenproduserte, medierte uttrykk. Det vil bli vurdert hvorvidt tilbudet skal forsøke å gi deltakerne en innføring i kildekritikk som et redskap for å etterprøve og skille i mediemangfoldet.

En mediearbeider vil fra ansettelsen forberede etableringen av Media Literacy. Vedkommende vil også vie seg til Media Literacy også etter etableringen.

### *7.8 Distribuert samhandling/møteplasser: En mulighetsstudie*

Distribuert samhandling kan defineres som nærhet og tilstedeværelse i et fjernsynsprogram der deltakere befinner seg på geografisk atskilte steder. Samhandling skjer ved hjelp av teknologiske hjelpemidler til å styrke opplevelsen av nærvær mellom programdeltakere som befinner seg på ulike steder. En mulig framtidig distribuert samhandling vil være at deltakere fritt kan tilslutte seg programmet underveis. En utredning for et slikt system er å finne i avsnittet "Ekseprimentet med distribuert samhandling" der vi med et erfaringsgrunlag fra Høgskolen i Lillehammer og Gjøvik forsøker å sette fingeren på kritiske faktorer samt komme med forslag til fremtidige utbedringer.

Framtidens fjernsynsmedium vil i økende grad gå over fra å være et informasjonsapparat til å bli et kommunikasjonsapparat der seere i økende grad trekkes inn som deltakere, ikke som tilskuere. Fordelene er flere. For det første "vider det ut" sendeflaten på den måten at flere deltakere kan sende samtidig og med hverandre. Dette alene er et middel til å ivareta og øke kanalens åpenhet. Deltakerne kobles opp i flerveis forbindelser over nettet for samhandling i saker som opptar dem. Dernest, og vel så viktig, skaper dette virtuelle *møteplasser* for samhandling på aktuelle saker, hvor mennesker møter mennesker i *aktørposisjon*. En typisk

møteplass vil være egnet for å utveksle erfaring og kunnskap om temaer som opptar og beriker oss. *Møtepllassene* kan være tverrfaglige og gå på tvers av ulike samfunnsgrupper, med likesinnede eller med annerledestenkende, uformelle og formelle, vitenskapelig eller ikke. Innholdet kan være nært og virkelig med deling av opplevde vansker så vel som inspirerende debatter om nye muligheter i folks hverdag. Distribuert samhandling i form av møteplasser vil i seg selv gi aktualitet og slik kunne øke deltakelsen. Møtepllassen vil også kunne nyttegjøres til en gjensidig evaluering av deltakernes innhold og på den måten sørge for en raskere forbedring av sendingens kvalitet.

Piloten vil ved hjelp av distribuert samhandling vinne erfaring på handlingsrettet læring og utvikling. Varierte *møtepllasser* gjør det også mulig i forskningsøyemed å trekke erfaringer for videre utvikling av deltakende demokrati, og for å viderebringe deltakende pedagogikk på mange plan i samfunnet – også i skolevesenet.

### 7.8.1 Distribuert samhandling, teknologiske forutsetninger

MPEG4-standarden er fremtidsrettet og tilpasset distribuert samhandling (interaktivitet). Ikke minst gir objekt delen i standarden spennende muligheter. Dette er fortsatt ikke utprøvd i et større produksjons-/distribusjonsnett. I MPEG4 kan medieobjekter fra ulike fysiske steder settes sammen til én audiovisuell scene (vha BIFFS). Deltakerne i programmet vil i en eller annen grad kunne oppleve at de samhandler i et delt virtuelt rom. For å oppnå nærhet mellom aktørene har samhandlingstester utført mellom høyskolen i Gjøvik og Lillehammer vist at man må etterstrebe naturlige forhold som blikkontakt og dybdesyn (3D). I testene realiserte de blikkontakt med halvgjennomsiktig speil der kamera lå bak speilet som reflekterte bildet av personen man kommuniserer med. For å overføre bildet i 3-dimensjoner sendte de to videostrømmer fra to kameraer som representerte venstre og høyre øyne til deltakeren. Disse ble sendt ukomprimert (for å oppnå liten og lik tidsforsinkelse) over hver sin dedikerte fiber. MPEG 4 gjør det mulig å synkronisere to kameraer som kan stå på hver side av skjermen og generere et bilde i mottaker enden som både har ivaretatt dybden (3D) og blikkontakten uten kompliserte oppsett med speil og dedikerte fibere. Åpen kanal deltok eksperimentene ved HiL og HiG og ønsker å fortsette utviklingen i retning av noe som kan ha mer allmenn nytte.

Overføring av digitale fjernsynsbilder i nettverk er i seg selv svært krevende. Distribuert samhandling vil for hver deltaker som kobler seg opp fra et nytt sted kreve tilsvarende mangfoldig båndbredde hvor scenene skal settes sammen. Det er derfor viktig at Åpen kanal

har stor båndbredde i den sentrale delen av nettverket. Forsinkelse kan også virke svært forstyrrende i en samhandling der interaksjon mellom aktørene er en forutsetning. Selv med mye prosessorkraft medfører overføring av en enkel fjernsynssending på Internett betydelig forsinkelse i kodingen. Forsinkelsen kan reduseres ved å benytte hardware baserte kodere men dette er ikke å finne hos åpen kanal sine deltakere i dag. Den mest praktiske løsningen på kort sikt vil sannsynligvis være å benytte enkle kodingsalgoritmer (intraframe codec) fremfor de mer bitrateffektive (interframe codec) i kontribusjonen. Dette vil igjen kreve stor båndbredde i Åpen kanal sitt sentrale nettverk.

Etter hvert som MPEG4-standarden realiseres i helhet vil mange av disse overføringsvanskene overvinne og forenkle distribuert samhandling betydelig. Som nevnt vil piloten benytte MPEG4 for overføringen. Det er et enormt potensiale i denne standarden for å redusere bitraten og kodingsforsinkelsen. Det utvikles i dag algoritmer for å skille bevegelige objekter fra den ofte mer stillestående bakgrunnen slik at disse kan kodes forskjellig. Bakgrunnen vil i prinsippet kunne sendes over som et stillbilde (en sprite) mens man kun koder endringen i det bevegelige objektet. I den mest ekstreme realiseringen overføres også det bevegelige objektet som et stillbilde, typisk et ansikt, der man kun sender over styringsparametere til en maske som animerer ansiktet. På denne måten kan en person medieres uten synlige tap av kvalitet eller forstyrrende tidsforsinkelse på en linje med noen få kilobits kapasitet. Dette kan også få stor betydning for kontribusjon fra mobile terminaler. Hva som blir viktig for deltakerne i åpen kanal er vanskelig å forutse, men ved tilrettelegging og utprøving av ny teknologi i et reelt produksjonsmiljø vil man kunne bygge opp en unik kompetanse og forhåpentligvis bidra til at folk flest ser nytten av nye brukermønstre fjernsynsmediet.



**Figur 7-3: Etter sending. De fire til venstre er på Gjøvik, de fem til høyre på Lillehammer.**

Distribuert samhandling er også mulig å realisere ned tradisjonell fjernsynsteknikk. Dette har vært prøvd mellom Høgskolen i Lillehammer (HiL) og Høgskolen i Gjøvik (HiG). Her sendte man to ukomprimerte videostrømmer fra Gjøvik over et dedikert fiberpar til Lillehammer der det delte virtuelle rommet ble laget i en split-screen. Returen gikk via videokonferanse utstyr der signalet ble komprimert og fikk en forsinkelse én vei (på grunn av encoding og decoding) på om lag 380ms. Dette virket forstyrrende, men trente deltakere slik som skuespillere, mestret dette i sin improvisasjon.<sup>22</sup> Bildet over viser en spontan distribuert samhandling mellom flere etter at de planlagte testsendingene var over.

---

<sup>22</sup> Sigmund Andresen & Claus J. S. Knudsen *Eksperimenter med kommunikasjonskvalitet og tilstedeværelse* 2005, Høgskolen i Lillehammer/Gjøvik

## 7.9 Testsendinger på bakkenettet: En mulighetsstudie

Foreningen *Åpen kanal* har sammen med *Institutt for telematikk* og *Institutt for teleteknikk* ved NTNU og produsenten *Cardinal* testet og vunnet erfaring på fjernsynssendinger over nett. Testen anvendte det samme utstyret som ble anvendt i utprøvingen av det digitale bakkenettet i Sverige.<sup>23</sup> Den eneste forskjellen var at signalene gikk over kabel, ikke over luft. Oppsettet var likt det vi vil finne i de tusen hjem etter etableringen av bakkenettet.

Piloten bør innbefatte tilsvarende testsendinger. Det kan da være av stor verdi å gjennomføre kringkastning på luft parallelt med kringkasting over internett. I det helt enkle kan en luftsending skje ved å simulere en mast i bakkenettet med en begrenset distribusjon over NTNU som besitter en lisens for luftsending i Trondheim.<sup>24</sup> Det kan også arbeides for en mer omfattende uttesting i samarbeid med *Telenor Norkring* og *NTV*.

Parallell kringkasting over internett vil være det som heter MHP over IP (Multimedia Home Platform over Internet Protocol). Det er forventet at kommende set-top-bokser til fjernsynsapparatene vil kunne benytte IP for mottak av sending, ikke kun for returkanal som i dag. Blant andre utvikler *Cardinal* et playoutsystem for IP-sending. Foreningen *Åpen kanal* kan i piloten tilpasse denne og andre enheter for Norges åpen kanal, derunder generere interaktivt innhold der returkanalene over IP vil komme til sin rett. Det vil i tillegg til *Cardinals* playout-system være hensiktsmessig å prøve utstyr fra andre leverandører som f.eks Sony. *Cardinals* fortrinn har vært fordelaktige priser og et godt utviklingssamarbeid.

Set-top-boksene vil når bakkenettet er en realitet, etter all sannsynlighet ha støtte for MPEG4 med dertil støtte for mottak av mer effektiv komprimering. Piloten bør innbefatte testsendinger med beta versjoner av kommende set-top-bokser for å finne ut hvor mye båndbredden kan reduseres før den subjektive kvaliteten reduseres. Testsendingene vil avgjøre hvor mye båndbredde som er tilstrekkelig for Norges åpen kanal på bakkenettet. MPEG4 med tilhørende komprimeringsalgoritmer vil gi betydelig spart båndbredde. I stedet

---

<sup>23</sup> Sendingen ble avviklet fra et DVB-MHP-program til en set-top-boks, direkte foret av en COFDM modulert MPEG2TS strøm (DVB-T).

<sup>24</sup> Det kan benyttes en antenneforsterker hos *Institutt for teleteknikk* som blir tilkoblet COFDM modulatorene. En *Cardinal* playout-server eller annen server vil kunne fore COFDM modulatorene med et DVB-MHP signal.

for at NTV må avsette 5 % av kapasiteten til formålet, slik MPEG2 forutsetter, medfører MPEG4 at NTV ikke trenger å bli pålagt å sette av mer enn 2 % av kapasiteten. Norges åpen kanal blir dermed snarere en ekstrakanal enn en kanal som fortrenge mulige kunder hos NTV. Valg av MPEG4 bevirker således til at formidlingsplikten ikke utgjør en ubetydelig merkostnad for NTV.

## 7.10 Uavhengig evaluering: En mulighetsstudie

Foreningen *Åpen kanal* foreslår at det nedsettes et bredt sammensatt offentlig utvalg. Styrets medlemmer stiller seg til disposisjon for å sitte i et slikt utvalg.

Utvalget vil kunne:

- Evaluere pilotens to første år.
- Vurdere prinsipper for en lov om alles rett til deltakelse i Norges åpen kanal under eget redaktøransvar.
- Beregne inntekt og kostnad for satellittdistributør.
- Anføre både båndbredden (Mbit/s) og sendeflaten (timer) som avsettes til Norges åpen kanal på det digitale bakkenettet.
- Foreslå videre tiltak for å fremme deltakende demokrati gjennom Norges *Åpen kanal*.

Piloten har til hensikt å tilrettelegge deltakelsen forutfor igangsettelsen av det digitale bakkenettet. Hvorvidt dette lykkes, vil være målbart i form av antall deltakere, men også i form av daglig mengde ny produksjon/opplasting. Hvis satellittsendinger inngår i piloten vil 500 deltakere så vel som 3 timer daglig opplasting, være en ubetinget suksess.

Det er viktig å vinne erfaring om hvilke rettslige utfordringer som dukker opp underveis i piloten. En oversikt over misbruk av ytringsfriheten/-muligheten bør legges fram for evalueringsutvalget. De erfaringene som gjøres vil gi en pekepinn på nødvendigheten av en lovregulert ytringsrett samt kunne veilede om denne lovens innhold. En rettslig utfordring som allerede er kjent, er hvorvidt en utestenging fra kanalen kan forsvares. Vil denne utestengingen være for straff å regne? Disse og andre spørsmål må besvares. Se for øvrig vedlagte rettslige redegjørelse.

En uttesting av Norges åpen kanal på satellitt bør sammenfalle med en fortløpende beregning av faktiske utgifter, arbeidsbelastning og inntekter for distributør.

Evalueringen bør angi hvor stor sendeflate (timer) som bør avsettes ved oppstarten på bakkenettet og hvordan denne bør økes i takt med deltakermengden. Det bør også anføres hvilken båndbredde (Mbit/s) som anses tilstrekkelig for Norges åpen kanal.



Om noen år kan de nye mediene ha skapt andre kommunikasjonsformer som er vel så viktige som fjernsyn. Skal disse nye kommunikasjonsformene bakes inn i Norges åpen kanal? Hvordan vil distribuert samhandling utvikle seg? Er det et blindspor eller inngangen til en helt ny offentlighet? Hva utenom det tekniske kan legge til rette for og styrke den offentlige samtalen? Svar på disse spørsmålene vil kunne frambringe framtidige tiltak for å fremme deltakende demokrati gjennom Norges åpen kanal. Det bør derfor i evalueringen søkes å besvare disse og lignende spørsmål.

### *7.11 Avslutning av piloten*

Styret til foreningen *Åpen kanal* avslutter piloten sammen med daglig leder.

Piloten avsluttes straks samtlige milepæler er nådd og uansett etter tre år. Hvis piloten er påbegynt men det ikke foreligger tilstrekkelige midler for den videre gjennomføringen, avsluttes piloten.

Om piloten skal bringes over til vanlig drift av Norges åpen kanal, beror på vedtak utenfor styrets rettslige kompetanse. Styret må belage seg på at et forløp er at virksomheten legges helt ned, ustyret selges, leieavtale sies opp og staben avskjediges. Et eventuelt overskudd på grunn av salg av utstyr tilbakeføres til staten.

### *7.12 Ressurser for piloten basert på jevn belastning*

Piloten vil måtte ha:

- Daglig leder (1 stilling)
- Datatjeneste som drifter serverne, og sikrer tilgjengelighet og oppetid (½ stilling)
- Online informasjons- og veiledningstjeneste (½ stilling )
- Utvikler som videreutvikler programvaren og utvider tjenestefunksjonaliteten (1 stilling)
- Fjernsynstekniker/pedagog for Media Literacy (1 stilling)
- En sivilarbeider.

Datatjenesten vil i drift kreve en ½ stilling. Veiledningstjenesten vil også kreve en ½ stilling. Disse to tjenestene vil i praksis være utført av én og samme person, benevnt *teknisk ansvarlig*. Ved irregularetter i driften vil utvikler bistå datatjenesten og sivilarbeider bistå veiledningstjenesten. Dermed begrenser staben med daglig leder seg til fire faste ansatte og en sivilarbeider. Piloten vil allerede under etableringen søke om sivilarbeider som skal

vedlikeholde videoarkivet og tilse at video klart for avvikling teknisk sett er inntakt. Ellers vil foreningen *Åpen kanals* styre bistå i pilotprosjektets begynnelse og slutt. Styreleder vil også være behandlingsansvarlig for datasystemet, jf. personopplysningslovens § 2 nr. 4. Enkelte anledninger vil forutsette bruk av konsulenter. Siden datasystemet er egenutviklet vil dette behovet være minimalt. For det første driftsåret er det derfor kun budsjettert kr 50.000 til konsulenthjelp. I tillegg vil piloten måtte betale en revisor.<sup>25</sup>

Daglig leder vil, med unntak av evalueringen og termineringen av piloten, lede gjennomføringen fram mot samtlige milepæler.

### 7.12.1 Stillingsbeskrivelser

I det følgende beskrives de fire stillingene som trengs; daglig leder, utvikler, teknisk ansvarlig og mediarbeider. Ønsket bakgrunn og tenkte oppgaver for sivilarbeider er også beskrevet.

#### **Daglig leder**

Daglig leder leder gjennomføringen av piloten. Stillingen krever høyere utdanning innen telematikk. Vedkommende har kunnskap om fjernsynsproduksjon. Daglig leders ansvar er:

- rapportere til styret
- lede og følge opp staben
- planlegge kanalens tekniske utvikling
- oppgradere/kjøre inn utstyr
- føre regnskap
- være databehandler på vegne av behandlingsansvarlig, jf. pol § 2-5

#### **Utvikler**

Stillingen krever inngående kunnskap om programmering, datakommunikasjon og databasemodellering. Det er også en fordel å kjenne til tv-produksjon. Oppgavene er:

- oppgradere/videreutvikle gjeldende programvare for opplasting, komprimering og avvikling
- utbedre autentiseringen
- utvikle programvare for distribuert samhandling
- delta datateknisk under igangsetting av systemet, flyttingen av produksjonslinje til satellitt og testing på bakkenettet

---

<sup>25</sup> Hvilke ressurser eventuelt Medietilsynet må stille, er ikke anført her og vil bero på tildelt tilsynsrolle.

### **Teknisk ansvarlig**

Teknisk ansvarlig veileder deltakerne i opplasting, avvikling, rettighetshåndtering og ytringsfrihetens grenser. Stillingen krever kunnskap om server- og nettverksdrift. Kjennskap til fjernsynsteknologi, rettighetshåndtering, juss og redaktøransvar er en fordel. Hun/han må ha pedagogisk evner. Oppgavene er:

- delta i etableringen
- drifte og vedlikeholde serverpark og nettverk
- bistå utvikler i systemoppgraderinger
- veilede deltakere
- teknisk overvåke sendingene
- avlaste daglig leder som databehandler

### **Mediearbeider**

Mediearbeideren er utdannet pedagog/psykolog/sosionom. Det forventes erfaring med tv-produksjon. Vedkommende må ha god evne til kommunikasjon med barn og ungdom. Oppgavene er:

- delta i etableringen
- forberede, igangsette og gjennomføre Media Literacy rettet mot barn og ungdom
- videreutvikle Media Literacy-tilbudet

### **Sivilarbeider**

Sivilarbeider bistår teknisk ansvarlig med kvalitetssikring av videoarkivet. Vedkommende må være datakyndig. Oppgavene er:

- sjekke merking av innhold
- bistå teknisk ansvarlig i den tekniske overvåkingen av sendingene

## **7.13 Tidsplanen**

Tidsplanen gitt ved Gantt-diagrammet nedenfor tar utgangspunkt i fastsatte ressurser og milepæler gitt ovenfor. Planlagte ressurser er altså fire ansatte. Dermed er det nødvendig med romslige estimater på forventet tidsforbruk, samt at uforutsette, men mulige tidsklemmer kan forseres med innleide konsulenter.

Den enkelte aktivitet i diagrammet har samme navn som milepælen de når fram til. De fleste innbyrdes avhengighetene er at to aktiviteter ikke pågår samtidig med den samme ressursen

med mindre det gjelder daglig leder. Utvikler, teknisk ansvarlig og mediearbeider jobber med én aktivitet av gangen. Ledelse og planlegging vil inngå til enhver tid men er ikke innarbeidet i diagrammet. Likedan er ikke styret eller sivilarbeiders bistand innarbeidet.

Kvartal/driftsår	1/1	2/1	3/1	4/1	1/2	2/2	3/2	4/2	1/3	2/3	3/3	4/3
Aktivitet												
Etablere piloten	0 - 4											
Adgang til kanalen		3										
Media Literacy		Forbered, 1	2,5									
Satellitt			1	2,5								
Distrib. samhandling					2,5							
Test bakkenett						2,5						
Evaluering									0			
Overgang bakkenett									2,5	2,5		
Pilot slutt												1
<b>Ressursbruk</b>	0 - 4	4	4	3	4	4	4	4	4	4	3	4

Figur 7-4: Tidsplan for Åpen kanal pilot

Gantt-diagrammet viser de ulike aktivitetenes begynnelse, varighet og slutt. Selve aktiviteten er merket mørk grå, mens forberedelse og etterfølgende drift er merket lysegrått. Diagrammet angir ikke eventuelle avhengigheter mellom aktivitetenes forløp. Avhengighetene skyldes først og fremst den fast begrensede ressursen på fire ansatte som må gjøre oppgavene suksessivt. Det er derfor lagt inn i to slakk på til sammen tre månedsverk. Slakkene er viktige for å kunne gjenvinne planlagt framdrift hvis uforutsette vansker oppstår. Tallene i de enkelte aktivitetene angir ressursbruken. Ved å legge sammen disse tallene vertikalt ned til nederste rad ser vi at den samlede ressursbruken med unntak av slakkene ligger jevnt på 4 ansatte. Under etableringen vil stabens ressurser gå fra 0 til 4 personer.

### 7.14 Inntekter og utgifter for tilretteleggingens første år

Tilretteleggingen av Norges åpen kanal koster 3,5 millioner årlig. For det første året ser budsjettet slik ut:

## Åpen kanal

KOSTNADER	Beløp	Kommentarer
Utstyrsinvesteringer	690 000	Videoarkiv, avviklingsutstyr, produksjonsutstyr, nettportal
	200 000	Produksjonsutstyr
	200 000	DV-arkiv x 2
	40 000	Komprimeringsenhet x 2
	100 000	Streamingserver x 2
	10 000	Webserver x 2
	70 000	Programvare
	20 000	Utviklingsverktøy
	50 000	Diverse
Media literacy/opplæring	190 000	Fremme mediekompetanse/deltakelse hos barn/ungdom
	110 000	Opplæring, kursmateriell, med mer. For brukere
	30 000	Reise/på stedet opplæring
	50 000	Kurs og videreopplæring av ansatte
Lønn og honorar	1 838 000	4 stillinger, 1 sivilarbeider, konsulent, regnskap. Alle kostn. inkl.
	426 000	Daglig leder
	422 000	Utvikler
	405 000	Mediearbeider/bruker- og deltakerstøtte
	405 000	Teknisk ansvarlig
	120 000	Godtgjørelse & boutgifter sivilarbeider
	50 000	Konsulenttjenester
	10 000	Revisor
Driftskostnader	782 000	Teknisk vedlikehold, leiekostnader, telefon, med mer
	90 000	Teknisk vedlikehold og serviceavtaler
	360 000	Leie av kontor- og produksjonslokaler
	65 000	Telefon og bredbånd
	95 000	Frakt, porto, kopiering, kontorrekvisitt
	50 000	Stillingsannonser, ansettelse
	32 000	Opplysningsarbeid
	30 000	Allmøtet
	60 000	Reise
<b>SUM KOSTNADER</b>	<b>3 500 000</b>	
<b>INNTEKTER</b>		
Offentlig støtte	3 000 000	
Annen støtte	400 000	Særlig støtte til ulike 'media literacy'-prosjekter
Deltakeravgift	100 000	Engangsregistreringsavgift kr 1000 av ~ 100 nye redaktører
<b>SUM INNTEKTER</b>	<b>3 500 000</b>	
<b>BALANSE</b>	<b>0</b>	

Figur 7-5: Budsjett Åpen kanal pilot

## 8 Juridiske utfordringer: Ikke bare ytringsfrihet men også lovfestet ytringsrett.

Denne delen er skrevet i samarbeid med Ola Tellesbø, styreleder i Åpen kanal og gjesteforsker ved juridisk fakultet på Universitetet i Oslo.

Ytringsrett er et nytt begrep og en sterkere rett enn ytringsfrihet. Det er Åpen kanal sitt mål om å sikre ytringsretten som skiller oss fra andre kanaler der man kun vil ivareta ytringsfriheten til sine egne journalister.

Straks Norges åpen kanal sender på det digitale bakkenettet, har det norske folket fått reell ytringsfrihet, eller, med andre ord, det norske folket har fått en *de facto* ytringsrett; rett til å ytre seg når de måtte ønske det. Vesensforskjellen mellom ytringsfrihet og ytringsrett er at ytringsfriheten fastslår fravær av sensur, mens en ytringsrett vil fastslå at du *skal ha anledning* til å ytre deg.

Innledningsvis vil denne ytringsretten være *de facto* og begrenset til Norges Åpen kanal. Men siden kanalen, som i alle andre land, i essens og per definisjon er *åpen for alle* der enhver går på lufta med eget innhold under eget redaktøransvar, vil ytringsanledningen i realiteten og umiddelbart gi en rettslig ytringsrett: Alle kan kreve å få sende. Vel å merke ikke på grunnlag av ytringsfriheten, jf. Grl. § 100,<sup>26</sup> men på grunnlag av at det faktisk er at den åpne kanalen finnes.

Denne faktiske *de facto*-virkningen er dermed så reell at spørsmålet blir om ikke denne ytringsretten like gjerne bør lovfestes først som sist. Retten til å ha anledning til å ytre seg er allerede i ferd med å få sedvanestatus når internett og ytringsfrihetens totalforbud mot forhåndssensur, jf. Grl. § 100 fjerde ledd, tas i betraktning. Alle kan slippe til med eget

---

<sup>26</sup> Når Grunnloven § 100 ble oppdatert tilså lovgiver at det skal påhvile staten å legge til rette for den offentlige samtalen, jf. sjette ledd. Dette er et politisk pålegg uten rettslig kraft. Det vil si at den som ikke får anledning til å ytre seg, ikke kan anmelde staten for å ha forbrutt seg på § 100 sjette ledd. Derimot vil bestemmelsen gjøre det likefram for de folkevalgte å irettesette og instruere Regjeringen med henvisning til Grunnloven hvis Regjeringen ikke tilser at den offentlige samtalen tilrettelegges.

innhold på internett og internett kan ikke forhåndssensureres, ergo har alle i realiteten en *ytringsrett*.

Men nytten av denne ytringsretten og andre ytringsmuligheter er begrenset. En redaktør kan men må ikke innta et leserinnlegg, mens en vanlig internettside vinner liten oppmerksomhet. Straks ytringsretten omfatter fjernsynsmediet, innebærer dette en drastisk økning av ytringsretten. Ytringsrett i fjernsynsmediet er noe borgeren ikke har hatt før, og det ikke er gitt hvor omfattende denne skal bli.

Det kan tilføyes til dette at når borgeren ble tildelt stemmerett var dette ikke uten forbehold. Den kunne, og kan, under visse omstendigheter tapes. Likedan, når borgeren fikk adgang til å føre motorvogn med risiko for andre enn seg selv, var det ingen uforbeholden tildeling av livsvarig førerrett. Ut fra disse kjensgjerningene vil det heller ikke være urimelig å forutse at en rettslig ytringsrett i fjernsynsmediet vil gå hånd i hånd med en rettslig begrensning av den samme retten. Dette reiser spørsmålet om økningen i ytringsretten bør eller kan medføre enkelte begrensninger, og i så fall hvorvidt og i hvilken utstrekning?

En mulig begrensning av en ytringsrett må foruten å være forutsigbar for borgeren, ikke overstige den implisitte retten som følger av gjeldende ytringsfrihet. Altså, gjeldende ytringsfrihet, mediemangfold og samfunn gir implisitt en rekke ytringsmuligheter; borgere blir tatt med inn i studio, blir intervjuet, skriver kronikker, oppretter hjemmesider m.v. Denne ytringsvirksomheten er vernet av ytringsfriheten: Når du først har anledning til å ytre deg, da skal du heller ikke bli sensurert (såkalt negativ rett). Er denne ytringsanledningen misbrukt til å bryte norsk lov, det være seg ærekrenkelser, blasfemi, rasisme m.m., straffes vedkommende i etterkant jf. prinsippet om etterkontroll.

Når borgeren tildeles en ytringsrett i fjernsynsmediet, kan altså ikke denne retten i praksis bli svakere enn de ytringsmulighetene som følger av gjeldende ytringsfrihet, mediemangfold og samfunn. I tilfellet den hadde blitt svakere ville den såkalte "ytringsretten" i ytringsfrihetsperspektivet vært meningsløs. Det som skulle være ytringsrett ble isteden "kontroll av ytringer" med særdeles uheldige følger: Borgere som valgte å benytte den nye ytringsmuligheten framfor gjeldende muligheter ville bli forledet til å velge en ordning som begrenset deres ytringsfrihet.

Det er således nødvendig at en lovfestet ytringsrett i det minste samsvarer med dagens ytringsfrihet og de ytringsmulighetene som gjelder. Ytringsretten i fjernsynsmediet må utvide gjeldende implisitte adgang til ytring. Hvis ikke er en åpen kanal verken åpen eller til for det den var laget for.

Det er to hovedmåter å kontrollere ytringer på. Det er forhåndskontroll (sensur) og (rettslig) etterkontroll. Forhåndskontroll (sensur) er utelukket, jf. Grunnloven § 100 første og fjerde ledd.<sup>27</sup> Mulighetene for etterkontroll av ytringer i Norges åpen kanal vil, slik som er tilfellet i andre åpne kanaler i andre land, være minst like gode som etterkontrollen av andre medier. Det er således grunn til å regne med at den rettslige etterkontrollen av innholdet på Norges åpen kanal blir ivaretatt. For misbruk av ytringsfriheten kan retten idømme straff i form av inndragning, (betinget/ubetinget) bot og (betinget/ubetinget) fengsel. Dette samme vil gjelde for ytringer på Norges åpen kanal.

Mens forhåndssensur er utelukket og etterkontrollen vil virke som før, koker spørsmålet ned til straffespørsmålet. Tilsier reelle hensyn andre former for rettslige straff? Det å få rett til å sende på Norges åpen kanal vil som nevnt løfte den reelle ytringsretten betydelig. Når denne retten misbrukes, kan det mest nærliggende sanksjonen (straffen) være at denne retten tapes for en nærmere angitt tid, mer bestemt, tap av retten til å sende på Norges åpen kanal for en gitt tid.

En slik straff får to følger: (1) misbruket av ytringsfriheten i form av lovbrudd blir straffet og (2) straffen, som er tap av ytringsretten for en tid, forhindrer gjentakelse. Selv om tapet av ytringsretten har en virkning fram i tid, er dette rettslig sett ingen forhåndskontroll fordi straffen settes på grunnlag av et allerede begått lovbrudd. Virkningen vil kunne for en gangs skyld kunne ha en individuell preventiv virkning så fremt det er å forvente at i løpet av den tiden ytringsretten er tapt vil den straffede endre ytringsbehov – innholdsmessig.

---

<sup>27</sup> Vel å merke foreligger en slags konstitusjonell nødrett for forhåndssensur. Se bl.a. Opsahl . Ytringsfrihetskommisjonen påpekte i NOU 1999:27 at frihetens grense stoppet når en den i forbindelse med en aktuell, ikke-imaginær fare truet selve demokratiet den sprang ut i fra. Justis- og politidepartementet trakk i St.meld.nr.26 (2003-2004) fram forhåndssensur kunne forsvares for å beskytte ”forsvarshemmeligheter av særlig sensitiv art”.



Før det anføres forslag til hvordan yringsretten kan tapes for en nærmere gitt tid, er det grunn til å ta i betraktning hva som gjelder for tap av stemmeretten og tap av førerretten.

### *8.1 Tap av stemmeretten*

Statsborgerlig stemmerett tilkommer norske borgere (fra 18 år) som ikke har tapt stemmeretten etter Grunnlovens § 53. Stemmeretten kan i følge denne paragrafen tapes som følge av “Domfældelse for straffbare Handlinger”, hvor tap av stemmeretten inngår som straff, og som følge av å ha gått i “fremmed Magts Tjeneste uden Regjeringens Samtykke”. I tillegg er det kommet et nytt tredjeledd i Grl. § 50 som går ut på at begrensninger i stemmeretten til “Personer som paa Valgdagen aabenbart lide af alvorlig sjælelig Svekkelse eller nedsat Bevidsthet” kan fastsettes ved lov.

Reglene i Grl. § 53 regnes som uttømmende, slik at stemmerettsvilkår som avskjærer andre en de som faller inn under disse bestemmelsen må anses som grunnlovsstridig. Etter straffelovens § 31 kan tap av stemmerett kun idømmes for inntil 10 år. Det er dessuten bare forbrytelser mot statens selvstendighet og sikkerhet, forbrytelser mot styresmaktene og forbrytelser under utøvelsene av statsborgerlige rettigheter (for eksempel valgfusk) som kan utløse tap av stemmerett. Denne begrensningen er begrunnet med at de som er straffedømt for andre grove, men mer vanlige forbrytelser uansett ikke vil få nevneverdi innvirkning på et valgresultat.

Et annet moment ved stemmeretten er at den forutsetter en viss tilknytning til Norge, normalt folkeregistrering.

Tilsvarende for reglene om tap av stemmerett, kan man innføre en regel om tap av yringsrett i Norges åpen kanal ved domfellelse for nærmere angitte straffbare handlinger, enten disse begrenses til straffbare handlinger knyttet til yringer på Norges åpen kanal, eller også knyttes til straffbare handlinger i andre medier eller av andre typer.

Det synes åpenbart at de reelle hensynene bak et eventuelt tap av yringsrett, tilsier en mer restriktiv linje enn for tap av stemmeretten, slik at også brudd på for eksempel kringkastingslovens § 2–7 beskyttelse av mindreårige seere kan rammes.

Tilsvarende stemmeretten kan også ytringsretten begrenses til personer som er innført i norsk folkeregister, begrunnet i et krav om en viss tilknytning til Norge for å få ytringsrett. I den graden man er bekymret for misbruk av ytringsretten, vil kravet om tilknytning til Norge være en viktig begrensende faktor, ved at vedkommende dermed utsetter seg for sitt eget samfunns moralske og rettslige reaksjoner.

Personer som flytter til Norge bør imidlertid innrømmes ytringsrett før de får stemmerett, slik at enhver som har lovlig opphold og reelt bosted i Norge også har ytringsrett her. Kortere utenlandsopphold, for eksempel av mindre enn 12 måneders varighet, bør derfor ikke føre til midlertidig opphør i ytringsretten.

## 8.2 Tap av førerretten

Har vi her en likhet mellom en ny ytringsrett og den etablerte førerretten? I så fall, kan du ut fra analogi skrive én side om tap av førerrett og hvordan et tilsvarende tap av ytringsrett kunne ha vært tenkt? Alternativt skissere ut fra vårt perspektiv hvordan tap av førerretten er regulert?

Sammenligningen mellom en ytringsrett og førerretten er selvsagt ikke fortreffelig. Bilkjøring kan drepe der og da, mens ord ikke kan ta liv uten påfølgende handling. For å bli straffet for oppfordring til vold, er det et vilkår at det er oppfordring til *iverksettelse*, jf. straffelovens § 94 annet ledd nr. 1 og § 140. ”Det er derfor bare de oppfordringer som umiddelbart har som formål å føre til handling som omfattes. Bestemmelsen [§ 94] har derfor vært brukt med forsiktighet.”<sup>28</sup>

Vi har med ytringsstoff som skal ende i ytringer som skal volde irreversible skader. Psykologisk forskning viser at det er svært vanskelig å påvirke mennesker til å foreta uvante, uplanlagte, forbudte, vanskelige og truende handlinger, særlig overfor seg selv, og særlig hvis handlingen skal skje lengre fram i tid.<sup>29</sup> Påvirkning til forbrytersk handling er gjennom menneskelige relasjoner mulig i en opphetet tilstand eller over lengre tid, men ellers usannsynlig ut fra en enkeltstående offentliggjort ytring. Derfor rammer loven kun den som ”oppfordrer til iverksettelsen”.

---

<sup>28</sup> NOU 1999:27, avsnitt 6.3.3.2

<sup>29</sup> Icek Ajzen & Martin Fishbein *Understanding attitudes and predicting social behavior* 1980:\*\*

Ytringer er på et allment plan således langt mindre farlig enn fyllekjøring. Dette er viktig å ta i betraktning når omfanget av en ytringsrett skal fastsettes.

### 8.3 Forslag til håndtering av ytringsretten og tap av denne

Ytringsretten vil måtte være, slik som ytringsfriheten, individuell. Ytringsretten er begrenset til retten til å få sende på Norges åpen kanal. Retten, så lenge den ikke er tapt, er absolutt med mindre det foreligger alvorlig eller gjentagne brudd på norsk lov og forbudet mot å utnytte tildelt sendeflate kommersielt. Det rettslige utgangspunktet er at ”alle som ønsker det skal få sendeflate på Norges åpen kanal”.<sup>30</sup>

Foreningen *Åpen kanal* foreslår følgende prosedyre for å sikre at den som registrerer seg, er den han/hun gir seg ut for å være: Den som ønsker å benytte seg av ytringsretten må registrere seg med fullt navn, adresse, email og personnummer og betale en registreringsavgift. Autentiseringen av deltakeren skjer mot postforsendelse til oppgitte adresse for at deltakeren skriver under på sitt redaktøransvar, mot oppgitte email for å tildele brukernavn og passord, mot gyldigheten av personnummeret (basert på sjekksnummer) og mot at deltakerne er villig til å betale en mindre sum for å kunne få sende.<sup>31</sup>

Den som sender tar redaktøransvar. Dette er i Tyskland lovfestet.<sup>32</sup> Vedkommende plikter å sette seg grundig inn i redaktøransvaret i henhold til norsk lov, derunder gjøre det tydelig hvem som sender og er redaktør, jf. kringkastingsforskriftens § 7-8. Dertil forplikter redaktøren seg til at ”sendingene skal være ikke-kommersielle. Reklame, snikreklame, produktplassering, pengeinnsamling, sponsoreksponering eller virksomhet som tar sikte på økonomisk gevinst, er utelukket.”<sup>33</sup> I en åpen kanal er vederlagsfri kringkasting et prinsipp slik at alle kan delta på like vilkår og uavhengig betalingsevne.<sup>34</sup> Det ikke-kommersielle hensynet understreket Stortinget i Innst.S.nr.128 (2003-2004), ‘Digitalt bakkenett for

---

<sup>30</sup> Dette prinsippet har foreningen *Åpen kanal* lagt til grunn i deres vedtekter, pkt. 3. Se [openchannel.no](http://openchannel.no).

<sup>31</sup> Foreningen *Åpen kanal* foreslår kr 1000 i registreringsavgift.

<sup>32</sup> Se blant annet Staatsvertrag über die Zusammenarbeit zwischen Berlin und Brandenburg im Bereich des Rundfunks (MStV), § 43(5). For øvrige lover, se [www.alm.de/index.php?id=259](http://www.alm.de/index.php?id=259)

<sup>33</sup> Ordlyden er hentet fra foreningen *Åpen kanals* vedtekter, pkt. 3 ii.

<sup>34</sup> Foreningen *Åpen kanal* anser en engangsregistreringsavgift til ikke å være i konflikt med dette. Avgiften er kun et virkemiddel som sikrer at deltakerne har overveid deres beslutning om å sende.

fjernsyn' og Innst.S.nr.270 (2003-2004), om endring av Grl. § 100: "det er viktig i en tid med sterk kommersiell dominans i mediebildet å sikre at ikke-kommersielle tilbud kan slippe til". I Tyskland er det ikke-kommersielle hensynet lovfestet.<sup>35</sup>

En deltaker som opptrer i samsvar med det ikke-kommersielle prinsippet og norsk lov, beholder sin ytringsrett uinnskrenket. Dermed reises spørsmålet: Hva når deltakeren bryter det ikke-kommersielle prinsippet og norsk lov? Hva er sanksjonsmulighetene?

De vanlige rettslige straffemulighetene overfor lovstridige ytringer er, som nevnt, inndragning, bøter og fengselsstraff. Disse vil med ytringsretten være uendret. Lovstridige ytringer straffes som før uavhengig om ytringen kom som en følge av benyttet ytringsrett i Norges åpen kanal eller i et annet medium. Spørsmålet reist ovenfor er om ytringsrett i Norges åpen kanal tilsier andre sanksjonsmuligheter? Sanksjonen "tap av ytringsretten" ble foreslått. Dette kan gjøres på tre måter:

Én sanksjonsmulighet kan være å stenge sendeflaten umiddelbart mens kringkastingen pågår fordi innholdet klart krenker norsk lov, en annen er midlertidig tap av ytringsretten fram til dom foreligger, mens en tredje mulighet er, ved dom, tap av ytringsretten for en nærmere angitt tid. Tap av en positiv rett slik som den tredje muligheten angir, vil være straff i rettslig forstand,<sup>36</sup> jf. inndragning av førerkort (Rt 2002:1216), mens et midlertidig tap av ytringsretten fram til dom foreligger ikke vil bli ansett som straff, jf. beslag av førerkort (Rt 2002:1032).

Skal foreslåtte sanksjonsmuligheter la seg forsvare, må, som påpekt, ytringsretten oppveie sanksjonsmulighetene samlede virkning slik at ytringsfriheten ikke svekkes. Samtidig vil det

---

<sup>35</sup> Se MStV § 43(3).

<sup>36</sup> Som følge av EMK art. 6(1) og tilleggsprotokoll nr. 7 art. 4 har begrepet straff blitt svært utvidet. Tilleggsskatt (Rt 2002:497 og Rt 2002:557), inndragning av førerkort (Rt 2002:1216), tap av dagpenger etter bedrageri (Rt 2003:264), tilleggsmoms (Rt 2003:1376), tvangsplassering av barn på grunn av grov kriminalitet (Rt 2003:1827) og nektelse av prøveløslatelse (Rt 2004:927) har blitt ansett som straff, mens tilbakekall av legelicens (Rt 2004:1074), tap av retten til dyrehold (Rt 2003:1750), konkursskarantene (Rt 2003:1221), disiplinærtiltak i fengsel (Rt 2003:111), inndragning av gjenstander brukt til straffbar handling (Rt 2002:2271), beslag av førerkort (Rt 2002:1032) og gebyr jf. befordringsvedtekter (Rt 2002:1007) har ikke det. Se Ståle Eskeland *Strafferett* 2006, kap. 6, punkt 4. Høyesteretts praksis viser at tap av en positiv rett anses som straff.

være helt nødvendig at sanksjonsmulighetene er svært begrenset og minst mulig i strid med den negative ytringsfriheten om fravær av inngrep. Det rettslige utgangspunktet er at ingen skal straffes for sine ytringer (Grl. § 100 annet ledd) eller bli utsatt for andre sanksjoner (Grl. § 100 tredje ledd).<sup>37</sup> Hvis sanksjon likevel skal ilegges, er hovedregelen at sanksjonen lar seg forsvare holdt opp i mot behovet for sannhetssøken, individets frie meningsdannelse og demokratisk utfoldelse (jf. både annet og tredje ledd).

Først betraktes muligheten for å stenge sendeflaten mens sending pågår, og i så fall hvem som bør stå for stengingen. Dernest vurderer jeg midlertidig tap av ytringsretten og et ved dom tap av ytringsretten for en gitt tid.

### 8.3.1 Stenging av sendeflaten mens sending pågår

Politiet kan midlertidig forføye ytringsmateriell uten muntlig forhandling, jf. tvangsfullbyrdelseslovens § 15-7 annet ledd, eller foreta beslag etter strpl. § 203. Dette forutsetter at politiet har kjennskap til innholdet i ytringsmaterialet hvilket er tilfellet for en sending som pågår.

Derfor kunne man tenke seg at politiet ved påtalemyndighetene kan pålegge distributør eller den som drifter et senderanlegg om å stenge sendeflaten. Men, det å gripe inn i ytringsfriheten er et svært alvorlig inngrep. Det er flere vurderinger som må foretas. Var det én enkelt ytring i løpet av sendingen som klart var lovstridig, eller var det en *fortsatt lovstridig* bruk av ytringsfriheten, slik som sending av en lengre pornofilm. Det førstnevnte tilfellet taler ikke for beslag i ytringsmaterialet. Skaden har skjedd, og de rettslige følgene vil bli ivaretatt under politiets etterforskning av lovbruddet. Det er ved det andre tilfellet, beslag av ytringsmateriell rettslig sett, teknisk sett stenging av sendeflaten, som kan tenkes å ha interesse. I dag er dette så sært at så vidt kjent har påtalemyndighetene aldri pålagt distributør å stenge sendeflaten. Dette skyldes ikke minst at terskelen for å gripe inn i en sending som pågår, er svært høy.

På den andre siden, distributør vel vitende om hva som pågår med mulighet for å stoppe sendingen, kan være redd for å bli utsatt for et større erstatningssøksmål. Slike tilfeller har *Telenor* stått ovenfor, og sendinger er stoppet.

---

<sup>37</sup> NOU 1999:27, avsnitt 10.3.5

Telenor eier og drifter senderanlegg, både over satellitt (*Telenor Satellite Services*) og bakkenett (*Telenor Norkring*). Ved enkelte anledninger har Telenor stengt sendinger hos TV-selskaper som har vært kunde hos dem fordi TV-selskapet har brutt norsk lov. Stengningen skjedde mens sendingen pågikk og før rettskraftig avgjørelse om stenging forelå. Dette har reist spørsmålet om *Telenor* dermed opptrer som redaktør. Samtidig tilsier ytringsfriheten og redaksjonell uavhengighet at *Telenor* ikke kan pålegges et overordnet redaktøransvar overfor deres kunder eller på annen måte holdes rettslig ansvarlig for innholdet på deres senderanlegg.

Det kan argumenteres at hvis distributør kjenner til lovbruddet, vil alminnelig risikobetraktning tilsi at *Telenor* gjør sitt for å begrense lovbruddets omfang. I en erstatningssak er dette uomtvistet, men overfor ytringer møtes kryssende hensyn. *Telenors* stenging av sendeflaten er ikke bare selvtekt, det er brudd på ytringsfriheten som verner ytreren også mot privates (her *Telenors*) inngrep.<sup>38</sup> Uten rettens kjennelse er *Telenors* handling rettsstridig. Avveiningen mellom å begrense skadens omfang og inngrepet i ytringsfriheten tilsier at lovbruddet må være aldeles opplagt, svært grovt og av lengre varighet med fortløpende høy skaderisiko. Videre må inngrepet kunne forsvares også når inngrepet holdes opp mot den aktuelle ytringenes verdi som sannhetssøkende, individets frie meningsdannelse og demokratiprosess, jf. § 100 annet ledd. Enda mindre må ytringen være politisk, jf. tredje ledd.

Vernet om ytringsfriheten tilsier at distributør lite sannsynlig kan stenge selv rasistiske ytringer eller ”vanlig” pornografi. Bedømmelsen av de rasistiske ytringenes politiske grad vil ofte tilsi at distributør ikke *aldeles opplagt* kan stenge sendeflaten. Når det kommer til pornografi er terskelen for hva som tillates høyere enn før. En sending som stenges må være støtende, typisk i form av ”kjønnslige skildringer der det gjøres bruk av barn, lik, dyr, vold og tvang.”<sup>39</sup> Er det sistnevnte ikke tilfellet, vil distributør gjerne stå overfor tvilstilfeller. Det vil ikke være uventet om produsenter balanserer på grensen mellom det tillate og ulovlige. Det utelukker at distributør kan stenge sendeflaten.

Ytterligere, det er svært betenkelig at Telenor på egen hånd stenger sendeflaten mens sending pågår. Alminnelig rettssikkerhet tilsier at påtalemyndighetene, som er tilgjengelig døgnet

---

<sup>38</sup> Se blant annet NOU 1999:27, avsnitt 10.3.4; St.meld.nr.26 (2003-2004), avsnitt 4.2.3

<sup>39</sup> St.meld.nr.26 (2003-2004), avsnitt 4.6.7

rundt, utsteder et pålegg om stenging av sendeflaten. Blir distributør tipset før påtalemyndighetene, vil en henvendelse fra distributør til påtalemyndighetene raskt bringe på det rene om ”fortsatt lovstridighet” pågår med dertil muligheter for pålegg om stenging.

Straks Norges åpen kanal åpnes, vil antallet registrerte redaktører og produsenter øke formidabelt. Disse redaktørene vil måtte og skal kjenne ytringsfrihetens grenser, opphavsrett, ulike straffebud, kringkastingsloven m.v.. På tross av at samtlige redaktører kjenner deres ansvar vil sjansen for sendinger med ulovlig innhold også måtte øke. Dette taler for at påtalemyndighetene inntar en mer aktiv rolle i å pålegge distributør å stenge sendeflaten.

På den andre siden er det slik at enkeltssendinger i Norges åpen kanal typisk vil vare i 10 minutter og antallet seere vil typisk være 10.000 om ikke færre. Dermed vil stengingen av sendeflaten knapt kunne iverksettes før sendingen er over samtidig som virkningen av stenging kan bli mer påaktet enn om sendingen fikk gli over av seg selv. Dette taler for at påtalemyndighetene ikke endrer praksis på dette feltet.

Skulle et nødtilfelle likevel vise seg å være av en slik art at sendingen for enhver pris bør stenges, kan distributør, NTV eller annen tilrettelegger av Norges åpen kanal ty til nødrett og stenge sendeflaten. Enda bedre vil det være at påtalemyndighetene underrettes hvorpå det tas standpunkt om stenging og beslag skal pålegges. Underretting av påtalemyndighetene bør være en mulig ordning, og ingen pliktig ordning, for distributør/tilrettelegger. Om dette eller noe annet skal være den framtidige ordningen, bør vurderes i forbindelse med en lovfestet ytringsrett.

Hvem skal i så tilfellet kunne stenge sendeflaten, er det distributør? For det digitale bakkenettet vil det kunne være både *NTV* og *Telenor Norkring*, for satellitt *Canal digital*, *Viasat*, *MTG* og *Telenor Satellite Services*. Det vil også kunne være de som drifter deltakernes avspillingsenhet opp mot bakkenettet/satellitt. I det følgende forutsettes det de som drifter deltakernes avspillingsenhet står for stengningen. Dette har flere praktiske fordeler, en av fordelene er at denne tilretteleggeren også kan ta hånd om ytringsmateriell som politiet begjærer beslag i.

### 8.3.2 Stenging av sendeflaten for en bestemt aktør i påvente av dom

Adgangen til sendeflaten kan inntil rettskraftig dom foreligger teknisk sett stenges for mistenkte/tiltalte for misbruk av ytringsretten/ytringsfriheten. Dette vil være et midlertidig tap av ytringsretten. Teknisk sett kan dette skje ved å stenge vedkommendes brukerkonto til kanalen, fjerne vedkommendes videoinnslag på spilleplanen og sperre innslag for Video on Demand (VOD). Stengt brukerkonto betyr at verken nye innslag kan lastes til videoarkivet eller innslag kan settes på spilleplanen.

Ytringsfriheten er slik at ingen taper sin ytringsfrihet selv om den tidligere er misbrukt. Midlertidig tap av ytringsretten kan oppfattes som midlertidig tap av ytringsfriheten. Dette er ikke tilfellet. Det er kun adgangen til Norges åpen kanal som tapes midlertidig, ikke til andre medier. Reelle hensyn tilsier at en ny tildelt ytringsrett vil måtte ha visse skranker, ikke i form av forhåndssensur, men med ordninger som gjør at man kan reagere raskt overfor misbruk av den økte ytringsmuligheten og reelle ytringsfriheten. Fysisk beslag av elektroniske kopierer vil nødvendigvis ikke forhindre gjentatte lovbrudd. En midlertidig *utestenging* fra kanalen vil derfor være vel så viktig.

En slik utestenging vil kunne lovreguleres sammen med en lovfesting av ytringsretten.

Spørsmålet blir derfor om reelle hensyn taler for en midlertidig ordning av stenging fram til ytringsretten er lovregulert. Foreningen *Åpen kanal* foreslår at i mangel av en lovregulert ytringsrett er den mest demokratiske midlertidige ordningen at deltakerne velger et styre eller råd som ved hjelp av tekniske hjelpemidler online og umiddelbart *kan* beslutte utestenging. Vilkårene for utestenging må være at lovstridigheten er klar og grov. Samtidig må det kreves at ytringene ikke er politiske og at inngrepet er til å forsvare også når tatt hensynet til sannhetssøken, individets frie meningsdannelse og demokrati.

Den midlertidige utestengingen, så fremt deltakeren ikke godtar den, må bringes uten opphold inn for domstolen. Får styret eller rådet medhold vil spørsmålet som oppstår være hvor lenge utestengingen skal virke.

På sikt, utover en pilotfase, vil det være uholdbart at et ikke-konstitusjonelt, utenomrettslig organ, slik som et selvoppnevnt styre eller råd, forvalter borgerens ytringsfrihet. Dette gjelder ikke minst ved brudd på Norges åpen kanals forbud mot ikke-kommersiell virksomhet. Det vil



være av betydning at retten, og ikke et bestemt styre/råd, kan ta stilling til om det ikke-kommersielle vilkåret er brutt. Dette understreker behovet for en lovfestet ytringsrett slik at også det ikke-kommersielle vilkåret blir rettslig forankret.

### 8.3.3 Tap av ytringsretten for en nærmere angitt tid

Tap av ytringsretten for en nærmere angitt tid vil si tap av retten til å benytte Norges åpen kanal. Dette er et rettslig spørsmål av særdeles stor prinsipiell betydning. I dette skrevet er noen tanker omkring en lovfestet ytringsrett skissert. Det som er skissert er ikke tilstrekkelig til å foreslå mulige straffeutmålinger. Spørsmålene som må vurderes er om det skal benyttes prikkbelastninger, hvilken varighet skal utestengingen ha, skal den i øvre strafferamme ha fellestrekk med tap av stemmeretten?

På tross av den noe overflatiske skissen kan det anføres enkelte tanker. Et forslag kan være at tap av ytringsretten bør vare i fra to år og oppover til 10. Den nedre rammen fordi gjennomsnittsdeltakeren typisk ikke ytrer seg på fjernsyn oftere enn én gang per år. Tap av ytringsretten i kortere tid vil således knapt merkes. Dertil kommer at skal tapet ha en individuell preventiv virkning, må tapt ytringsrett sammenfalle med en holdningsendring. En holdningsendring vil måtte ta en viss tid, kanskje så lenge at tapet av ytringsretten blir et savn?

Mot dette er det viktig å huske at en ytringsrett ikke gjør annet enn å (1) løfte fram meninger som det norske folket allerede har, (2) slippe fram ytringer som på et allment plan er langt mindre farlig enn for eksempel fyllekjøring og (3) inngå som en del ytringsfriheten som allerede fastsetter et rettslig ansvar.

Til punkt (3) bemerker foreningen *Åpen kanal*: Kan det i det hele tatt fastsettes en tilleggsstraff som i tid strekker utover idømt fengselsstraff? Dersom en borger har sonet en dom på 6 måneder for ulovlige ytringer, bør ikke vedkommende være fri til å vende tilbake til samfunnet på alle måter, derunder retten til å ytre seg på Norges åpen kanal? Kan det tas hensyn til faren for gjentakelse? Gjentakelse straffes vanligvis hardere enn tidligere ikke-straffedømte. Er ikke det et tilstrekkelig og rettslig styringsmiddel/virkemiddel?

På tross av disse vektige innvendingene foreslår foreningen *Åpen kanal* i en pilotfase som begrenser seg til tre år, for enkelhetens skyld at utestenging varer pilotfasen ut. Det forutsettes

at deltakeren ”bevisst har omgått vedtektene”,<sup>40</sup> som foruten det ikke-kommersielle kravet viser til norsk lov.

### 8.3.4 Verst tenkte tilfelle av misbruk av ytringsretten

Fusk med ytringsretten stiller oss overfor helt andre vansker enn tilfellet fusk med stemmegivningen og stemmeopptellingen. Dette selv om bruken av stemmeretten og en ytringsrett begge vil kunne springe ut fra verdivalg vedkommende deler med mange.

Misbruk av ytringsretten blir en engangsforseelse. Dette fordi utestengingen vil skje straks etter det første lovbruddet er fullbyrdet. En midlertidig utestenging som får medhold ved dom, blir opprettholdt.

Ut fra dette, hva kan være det mest utspekulerte misbruket av en ytringsrett?

På grunn av at ytringer kontrolleres i etterkant, ikke i forkant, vil trolig det mest utspekulerte misbruket av en ytringsrett være en sammensvergelse der én og én i tur bestemmer seg for å ”bruke opp” sin ytringsrett. Slik kan lovstridige ytringer gjentatte ganger slippe ut.

Et slikt forløp forutsetter flere ting. For det første forutsetter det at enkelte vil veie sin ytringsrett til å være mindre verdt enn å få ytret seg én – 1 – gang. For det andre forutsetter forløpet at disse enkelte ikke bare er én eller to, men flere. For det tredje må de som kommer senere i rekken ta innover seg at de forutgående blir sanksjonert. *For det fjerde, så fremt det ”er rimelig grunn til å undersøke om det foreligger” gjentagne og fortløpende lovbrudd i form av et komplott, jf. strpl. § 224, vil et beslag av filmmassen som ligger klar for sending ikke være forhåndskontroll rettslig sett men et inngrep ut fra allerede ”fortsatt lovbrudd”.* Beslaget og den videre etterforskningen vil opplagt måtte omfatte mulige medvirkere. Dette vil sette en effektiv stopper for en eventuell ”organisert ytringskriminalitet”.

## 8.4 Andre rettslige sider ved Norges åpen kanal

Innføringen av Norges åpen kanal vil ikke bare legge et grunnlag for en ytringsrett. Innføringen berører også kringkastingsloven. Særlig omtalt er hvilke følger det vil få om konsesjonsvilkårene gitt i § 2-1, jf. § 2-2, tolkes strengt. Mindre omtalt men også viktig er at det i Norges åpen kanal tilrettelegges for opptak av program, jf. § 2-5.

---

<sup>40</sup> Foreningen Åpen kanals vedtekter § 8 xi., se [www.openchannel.no](http://www.openchannel.no)

#### 8.4.1 Ikke konsesjon men *disposisjon*

Kringkastingslovens § 2-1 annet punktum fastslår at enhver som sender på det digitale bakkenettet må ha konsesjon, jf. § 2-2. Videre følger det av kringkastingsforskriftens § 7-8 at den som har konsesjon må ha redaktøransvar. Her er det viktig å vite at kringkastingsloven ble laget mens Norges åpen kanal ikke var påtenkt. Dette gir Kultur- og kirke departementet en viktig skjønnsfrihet ved tilretteleggingen av Norges åpen kanal. Det avgjørende er at departementet påser at Norges åpen kanal *ikke* tildeles én bestemt aktør slik konsesjonsvilkårene forutsetter. Grunnleggende for enhver åpen kanal er derimot at distributør stiller kanalen til DISPOSISJON for alle. Dermed er sendeflaten utenfor styring fra én bestemt konsesjonær. Vel så viktig knesetter alles *disposisjon* over sendeflaten en allemannsrett i etermediet der det rettslige ansvaret er fastsatt i Grunnlovens § 100, et rettslig ansvar som i tråd uskyldspresumpsjonen forhindrer forhåndskontroll: Enhver statsborger nyter tillitt under eget ansvar så lenge tilliten beviselig ikke er misligholdt. Altså, forbudet mot forhåndssensur forutsetter at enhver deltaker i Norges åpen kanal får sende under eget redaktøransvar så lenge sendingene beviselig ikke bryter norsk lov.

#### 8.4.2 Opptak av program, sikring av bevis

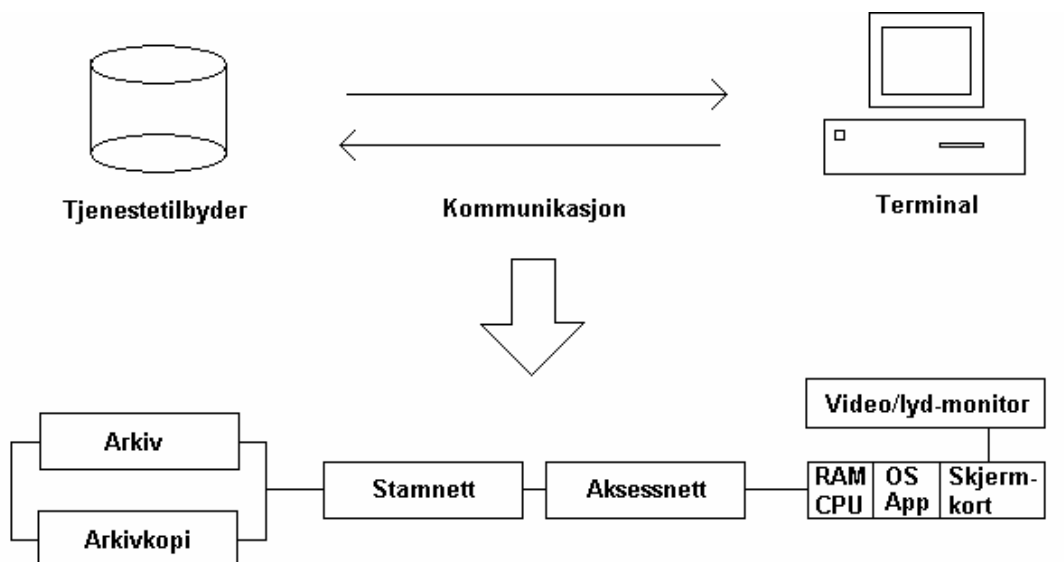
Foreningen *Åpen kanal* foreslår at tilretteleggeren bistår deltakeren i å ”oppbevare opptak av program”, jf. kringkastingslovens § 2-5. Dette vil også imøtekomme eventuelle behov om arkivering av videogram som er funnet rettsstridig, jf. filmlovens § 7.<sup>41</sup> En oppbevaring vil sikre bevis og gjør fysisk beslag jf. straffeprosesslovens § 203 overflødig.

---

<sup>41</sup> Filmloven anvendelsesområde favner ”framsyning og omsetning av film og videogram i næring”, jf. § 1, og gjelder således strengt tatt ikke innhold vist på Norges åpen kanal der næringsvirksomhet er forbudt.

## 9 Tekniske utfordringer

De tekniske utfordringene er mange både i utviklingen og driftingen av en nettbasert TV-kanal. Når det gjelder utviklingen må man ikke bare ta hensyn til brukerens krav i dag, men også ta høyde for uforutsette krav som følge av ny teknologi og nye brukermønstre. Vi skal her se på mulige valg i de ulike delene av et slikt system. Målet er ikke å komme frem til en endelig realisering, men heller å si noe om hva som er viktige aspekter.



Figur 9-1: Modell av en Åpen kanal på Internet

### 9.1 Terminaler

Terminalen er den enheten som brukeren av systemet forholder seg til. Dette kan være en PC, TV med en set-top boks eller portabelt utstyr som PDA og mobiltelefon. Disse enhetene har ulike begrensninger som ofte er gitt av størrelse eller pris. En portabel enhet har gjerne en liten skjerm og lite prosessorkraft. Det er viktig at innholdet tilpasses til terminalene de skal vises på slik at opplevd kvalitet blir høyest mulig. For å oppnå dette må man kjenne til egenskaper i de ulike streamingformatene samt begrensninger i de ulike terminalene.

### 9.1.1 Maskinvare

Den mest vanlige terminalen for mottak av en nettsending er i dag den stasjonære PC-en. Nye maskiner vil i skrivende stund kunne støtte avspilling i kringkastingskvalitet. Det man trenger er et nettverkskort, ram (128 MB) og en relativt ny prosessor (800 MHz eller mer). Ettersom dekomprimeringen vil kunne skje i software stilles det ikke spesielle krav til skjermkortet. Oppløsningen på skjermen vil heller ikke være noen begrensning ettersom det som kreves for standard TV bilde er 720 X 576 piksler i 24 bit fargedybde. I dag er det vanlig med en oppløsning på minst 800 X 600 på PC-er.

For bærbare terminaler slik som PDA eller mobil er oppløsningen på skjermen det største problemet. Prosessorer i PDA-er kan nå være opp til 400MHz, men skjermen er sjeldent større enn 240 X 320 og har en fargedybde på 16 bit. Det må derfor lages egne videostrømmer som er tilpasset PDA og mobil.

Det finnes også teknologiske løsninger som gjør at brukeren kan surfe på Internet på sin TV hjemme i stua. For å få til dette må man bruke en såkalt set-top boks som i realiteten er en liten PC med TV-utgang. Disse skal bli så billige at man kan få de gratis dersom man knytter seg til visse abonnement. Prosessoren er i disse enhetene så trege at de ikke kan spille av videostrømmene i software. Her benytter man seg av hardware akselerasjon som finnes i skjermkortet på boksen. Ettersom det er umulig å oppgradere dekoderen satser de her på ISO/IEC og ITU-T standarder i grafikkbrikkene.

### 9.1.2 Programvare

Den viktigste applikasjonen for å kunne ta nytte av videotjenesten er en mediaspiller. Avspilleren kan ha støtte for flere streamingformater i sin opprinnelige utgivelse og man kan laste ned dekodere for kommende formater dersom man har administrator rettighetene til å gjøre det. Administratorrettigheter er ikke noe man nødvendigvis har på jobb eller studiested.

Det finnes i dag flere konkurrerende formater når det gjelder multimediestreaming over Internet. De mest kjente av disse er Microsoft sin Windows Media teknologi, RealNetworks sin RealAudio/RealVideo teknologi og Apple sin QuickTime teknologi. Disse brukes alle av mange aktører på Internet i dag. I tillegg snakkes det mye om MPEG-4. MPEG-4 er en ISO/IEC standard med mange forskjellige kompleksitets- og kvalitetsnivåer som er utviklet av MPEG (Moving Pictures Expert Group).

Ikke alle disse teknologiene er like tilgjengelig på alle operativsystemer. Ifølge en artikkel fra Uninett<sup>42</sup> antar man fordelingen på de tre største operativsystemene, Windows, MacOS og Unixvarianter (Linux, FreeBSD, Solaris m.fl.) til henholdsvis 85%, 10% & 5%.

De langt fleste kjører ikke uventet Windows i en eller annen form på sin datamaskin (rundt 85 %), og her finnes alle teknologiene enkelt tilgjengelig. Windows Media Player følger med som standard på alle de nyere versjonene av Windows (Windows 98 eller nyere), og lar seg lett oppgradere gjennom den såkalte "Windows Update" tjenesten Microsoft tilbyr. Ellers er mediaspillere til RealAudio/RealVideo og QuickTime lett tilgjengelig for nedlastning fra henholdsvis RealNetworks og Apple sine hjemmesider. Anslagsvis kjører 10 % av de potensielle brukerne på en eller annen form for Apple Mac. De siste årenes Macintosh maskiner kjører som regel en av to fundamentalt forskjellige operativsystemer MacOS 9 eller MacOS X. QuickTime følger med som standard i begge, og både Windows Media Player og RealPlayer (kalt RealOne) finnes lett tilgjengelig for nedlasting. De siste 5 % av potensielle brukere plasseres i gruppen som kjører det man gjerne kaller "\*nix". De fleste av denne gruppen antas å kjøre en Linuxvariant, mens FreeBSD er et annet alternativ. Resterende varianter antas å ha marginalt med brukere. Når det gjelder mediaspillere til denne gruppen er tilstanden noe annerledes enn hos de to foregående. RealNetworks støtter utviklingen av en RealPlayer for "\*nix" i form av tekniske opplysninger om formater etc. de benytter. Real er i så måte godt støttet i denne gruppen. Spilleren kan lastes ned gratis fra RealNetworks sine hjemmesider. Når det gjelder QuickTime støtte, så finnes det ingen avspiller tilgjengelig fra Apple. Det finnes derimot kommersielle løsninger man må betale for som skal fungere (f. eks CrossOver). CrossOver koster nå US\$ 25 og støtter også eldre versjoner av Windows Media, samt RealPlayer. Det finnes også prosjekter som jobber med å utvikle gratis programvare for å avspille både QuickTime og Windows Media (f. eks mplayer, xine), men disse støtter som oftest ikke all funksjonalitet, eller siste versjoner. De har heller ingen garanti for kompatibilitet, og særlig ikke når det gjelder streamingfunksjonalitet. MPEG-4 standarden har flere grader av kompleksitet og funksjonalitet. De siste versjonene av QuickTime og Real har støtte for den grunnleggende ISO MPEG-4. Dette er imidlertid ikke nok i for å kunne tilby brukeren ekte interaktivitet. Da er det behov for en større del av funksjonaliteten i MPEG-4 standarden. Foreløpig kan dette tilbys av Envivio, som har gratis tilleggsmoduler for

---

[1] 42 Intervju av Jensen, Geir Olav, medieteknisk konsulent i Uninett, september 2003

windowsversjonene av Windows Media Player, RealOne og QuickTime for nedlastning på sin hjemmeside. De lover også å ha klar en versjon til Mac og en til Linux innen kort tid. Dersom MPEG-4 tar av i fremtiden vil det sannsynligvis ikke mangle på tilgjengelighet av avspillere for de vanligste plattformene. MPEG-4 vil i dag kreve mer av brukeren ved at det må installeres ekstra programvare, og det er det ikke sikkert at brukeren har rettigheter til dersom han/hun f. eks skal nyttiggjøre seg av tjenesten på jobb eller på en datasal ved studiestedet.

	<b>MPEG-4/ QuickTime</b>	<b>Windows Media</b>	<b>Real</b>
<b>Audio og video</b>	Standardbaserte løsninger	Proprietære	Proprietære, MPEG-4 plugin
<b>Objekt støtte</b>	Audio/video, stillbilder, 2D/3D grafikk og animasjon	Kun video og audio	Audio og video(SMIL)
<b>Interaktivitet</b>	Høy interaktivitet	Begrenset	ja(SMIL)
<b>Grafikk objekter</b>	Ja(2D og 3D)	Nei	Nei
<b>Animasjon</b>	Ja	Nei	ja(SMIL)
<b>Sanntids streaming kontroll</b>	Ja	Ja	Ja
<b>Synkronisering</b>	Alle objekter	Audio og video	Audio og video
<b>Håndtering av rettigheter</b>	Grensesnitt til tredjeparts løsninger	Microsoft DRM	Aksess kontroll til innhold
<b>Transport</b>	HTTP, UDP, RTP/RTSP, mobil, MPEG-2 TS	HTTP, UDP, RTP/RTSP, mobil	HTTP, RTP/RTSP, mobil
<b>Støtte for broadcast</b>	Ja (Hele presentasjon)	Ja (kun audio og video)	Scene må være multicast

**Tabell 9-1: Sammenligning mellom videoformater**

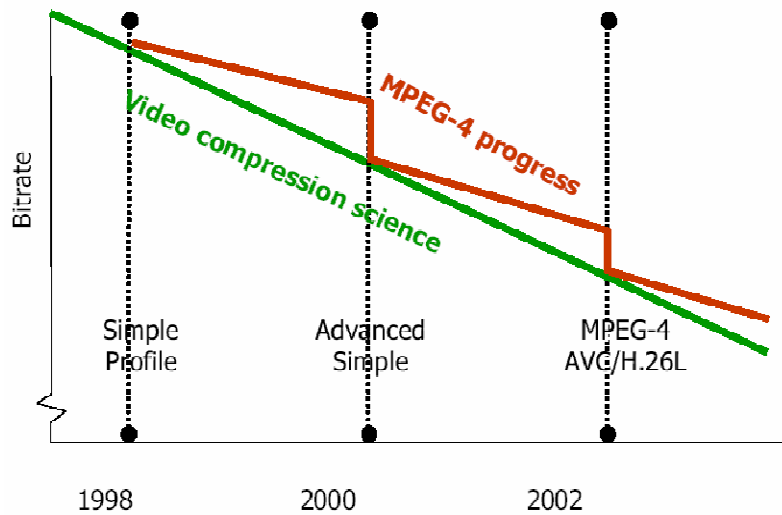
MPEG-4 standarden inneholder et rammeverk for en rekke fremtidsrettede funksjoner. Det jobbes blant annet med noe som kalles "Fine Granularity Scalability" hvor man deler inn videostrømmen i ett basis lag og ett enhancement lag<sup>43</sup>. Enhancement-laget skal fungere slik at man med utgangspunkt i strømmen i basis laget skal kunne bygge opp en videostrøm som har nøyaktig den kvaliteten terminalene og forbindelsene til den tillater. Fordelen med dette er at man slipper å lage flere komprimerte strømmer etter antagelser om terminalene. En MPEG-4 fil vil som tidligere nevnt også kunne tilby en økt grad av interaktivitet. Dette sørges for gjennom såkalte BIFFS (Binary Format For Scenes) og "Scene descriptors" der videobildet deles inn i flere objekter med hver sine egenskaper. I MPEG-4 kan man i motsetning til de andre standardene velge å kode objekter istedenfor rektangulære bilder. Dette kan være enormt besparende på bitraten ettersom ulike objekter i bildet kan ha ulike krav til oppdatering. F. eks. trenger man ikke sende et bakgrunnsbilde mer enn en gang under en nyhetsopplesning. At man også kan ha en variabel blokkstørrelse i bildet gjør at man kan skjule kodefeil bedre. Denne funksjonaliteten finner man i Advanced Video Codec (AVC) som utvikles i samarbeid mellom IEC og ITU-T (H.264). De har som mål å oppnå kringkastingskvalitet på ADSL linjer i løpet av de neste to årene. Envivio viste 10.3.2003 at deres AVC allerede har 63 % kvalitetsøkning per bit og lover en forbedring på 150 % innen årsskiftet<sup>44</sup>. Figuren nedenfor viser MPEG-4 industriforum sin forventning til utviklingen.

---

[2] 43 Hillstestad, Odd Inge, 21.03.2003, Forelesning Multimedia og signalbehandling NTNU

[3] 44 MPEG-4 Seminar NTNU 10.03.2003, Møteleder Garrec, Gall le (Envivio)



Figur 9-2: Utvikling av MPEG-4<sup>45</sup>

## 9.2 Kommunikasjon

En begrensende faktor på lyd- og bildekvalitet i en TV-kanal på Internet er hastigheten man kan sende data ut til seerne med. Problemene knyttet til dette kan deles inn i to deler. Transportnettet/stamnettet som er selve kjernenettet i Internet og aksessnettet som er den delen av nettet som går fra brukeren selv og inn til transportnettet. Tidligere var dette for privatbrukere stort sett modemforbindelser (28.8 - 56 kbit/s) og ISDN forbindelser (64 kbit/s), mens bedrifter gjerne hadde leide linjer på 64 kbit/s eller mer. I dag har det kommet mange raskere alternativer for både bedrifter og private som også er overkommelige i pris.

### 9.2.1 Stamnett

Problemer knyttet til stamnettet dreier seg i hovedsak om antall samtidige brukere. Et eksempel på en del av stamnettet i Internet er Uninett sin linje mellom Trondheim og Oslo. Dette er en kjapp linje i stamnettet med kapasitet på 2,5 Gbit/s. Hvis man for enkelthets skyld regner med at en videostrøm trenger 1 Mbit/s linjekapasitet, kan det gå 2500 slike videostrømmer over denne linja før den er mettet. Det må også påberegnes noe "overhead", slik at antall videostrømmer som kan sendes over linja er noe mindre. 2.5 Gbit/s er en meget

[4] 45 Envivo (2003), [http://www.envivio.com/images/products/etv\\_021217.pdf](http://www.envivio.com/images/products/etv_021217.pdf)

kjapp linje i stamnettet og mange steder er linjene mye tregere. Andre vanlige hastigheter i Uninett sitt nett er 33 og 155 Mbit/s. Båndbredden på disse linjene er heller ikke dedikert til en bestemt tjeneste.

En mulig løsning på kapasitetsproblemer i stamnettet kan være å innføre multikast. Sending av data på Internet i dag foregår stort sett ved såkalt unicast overføring. Dataene en bruker mottar er kun adressert til denne brukeren, og en eventuell nabo (nettopologisk sett) som vil motta de samme dataene må få de gjennom en egen datastrøm. Multikast forsøker å løse dette ved at dataene kan sendes som en strøm inntil et punkt (ruter) før det deles og sendes som en strøm til hver mottaker. Vi får dermed mindre belastning på nettet fram til dette punktet. Problemer med dette er at ruterne i Internett må støtte multikast og det må aktiveres. Multikast på tvers av forskjellige nettleverandører har også vært problematisk. Trafikkfordelingen på de forskjellige nettene kan bli ujevn og taksering av trafikk blir et problem slik at en del nettleverandører lar vente på seg med å aktivere multikast i nettet (Diot, 2000). Multikast kan være en løsning som vil fungere godt, men det er viktig å huske på at det vil bare løse en del av mulige problemer i forbindelse med stamnettet. Multikast vil fungere bra på streaming av en fastsatt sendeplan. Det kan også brukes i en Near Video On Demand (NVOD) løsning der man sender et eller flere innslag i karusell. I ekte Video On Demand (VOD) løsninger der man kan gå inn å se et innslag når som helst, vil ikke multikast kunne hjelpe.

Et generelt problem stamnettet kan møte, er at hastighetsøkningen i stamnettet ikke klarer å holde følge med hastighetsøkningen i aksessnettet. En vanlig ADSL linje i dag har en hastighet på rundt 1 Mbit/s. Dette er ca. 30 ganger så raskt som en analog modemlinje. Stamnettet har ikke i like stor grad økt i hastighet, og 1 Mbit/s er på ingen måte en maksimumsgrense for aksessnettet i dag, da det allerede leveres 20 Mbit/s av flere leverandører (ADSL2+). Når brukerne blir vant til den økte hastigheten og tar i bruk tjenester som krever større hastighet kan det være stamnettet som er den største flaskehalsen, og ikke aksessnettet som man har vært vant til, dersom stamnettet ikke klarer å holde følge hastighetsmessig.

## 9.2.2 Aksessnett

I den senere tid har det blitt tilgjengelig en del forskjellige løsninger for å få større hastighet i aksessnettet. De mest utbredte løsningene kjører på den kobberkabelen som allerede er gravd ned til de fleste hus, primært for bruk til fasttelefon. Det er ingenting som tyder på at det vil

blir storstilt graving av nye kabler (for eksempel optisk fiber) til private husholdninger i nærmeste framtid. Stadig utvikles det nye teknologier som utnytter den gamle kobberkabelen bedre og bedre. Av løsninger er det xDSL (digital subscriber line) familien med ADSL (asymmetric DSL) i spissen som er mest utbredt i dag. "Asymmetric" i denne sammenheng betyr at det er forskjellig hastighet på nedlasting og opplasting. Dette fordi det er mye mer vanlig å laste ned data enn å sende data ut, og man utnytter dette ved å ha høyere nedlastingshastighet enn opplastingshastighet.

	Telenor	NextGentel	Tiscali	Bluecom
Maks hastighet (ned/opp)	1024/256 kbit/s	2048/640 kbit/s	2048/256 kbit/s	2048/256 kbit/s
Billigste hastighet (ned/opp)	704/128 kbit/s	704/384 kbit/s	384/128 kbit/s	400/128 kbit/s

**Tabell 9-2: ADSL tilbud fra noen store teleleverandører august 2003**

	Telenor	NextGentel	Tiscali	Bluecom
Maks hastighet (ned/opp)	6000/500 kbit/s	20000/1200 kbit/s	som Telenor	20000/1000 kbit/s
Billigste hastighet (ned/opp)	700/160 kbit/s	1500/350 kbit/s	som Telenor	1500/448 kbit/s

**Tabell 9-3: ADSL tilbud fra noen store teleleverandører april 2006**

Disse tabellene viser at det i perioden for pilotutviklingen (2003 – 2006) av Åpen kanal også har vært en rask utvikling av bredbåndskapasiteten. Utviklingen er mest markant for nedlastingshastigheten. Brukerne av bredbåndsnettet er i dag først og fremst konsumenter av innhold. Med forbedrede kodingsteknikker som ADSL2+ kan bredbåndsløseleverandørene dermed tilby IPTV uten å legge opp til at man skal kunne snu strømmen med likeverdig kvalitet. Det tilbys en litt raskere opplastingshastighet i dag, men kapasitetene langt ifra tilfredsstillende for kontribusjon. På den beste ADSL-løsningene må man vente 20 ganger spilletiden når man laster opp DV-materiale (25Mbps videostrøm over 1,2Mbps linje).

Det er mer hensiktsmessig å satse på symmetriske linjer for deltakerne i Åpen kanal. Dersom man er flere med en felles bestilling (som f.eks beboere i et borettslag) kan man få

opplastingshastigheter nære ADSL2+ sin nedlastingshastighet men omtrent til samme pris<sup>46</sup>. Både Bredbåndsfabrikken og Lyse<sup>47</sup> energi tilbyr gode og rimelige symetriske (SDSL) linjer, men det kan være vanskelig å få tilgang til tilbudet dersom du bor langt utenfor byen. Bedbåndsleverandøren tjener mest på å bygge ut der flest kan koble seg til og prioriterer derfor byene

ADSL2+ kan heller ikke leveres til alle som har kobberlinje. Det er begrensninger på hvor langt kabelstreck man har fra telefonsentralen. Det vil også være forskjell i hvor store hastigheter som kan leveres ut fra både denne avstanden og hva slags stamnett det er mot telefonsentralen. Generelt vil det ikke være mulig med ADSL for folk som har mer enn 6 km kabelstreck til sentralen sin, og da med maks hastighet på 1,5 Mbit/s ned. Dette øker gradvis jo kortere avstand man har til telefonsentralen, og de som har under 2 km kabellengde til sentralen kan potensielt oppnå maks nedlastingshastighet 8 Mbit/s på vanlig ADSL (Tannenbaum, 1996).

Andre aksessmetoder som finnes i dag med hastighet utover ISDN er i hovedsak Internett fra kabel-TV leverandører. Problemer med dette er at det tradisjonelt har vært slik at alle som er tilknyttet samme kabel må dele på en gitt bitrate, slik at det enten har gått tregere jo flere som har brukt tjenesten, eller at man i utgangspunktet har blitt tildelt en forholdsvis lav bitrate, og da særlig lav opplastingsrate.

I fremtiden er det flere muligheter for å få bredbånd hjem til alle. Det har i lengre tid vært snakket om bredbånd over strømmettet uten at dette har kommet tilfredsstillende løsninger for å ta dette i bruk. Trådløse varianter ved bruk av radio og satellitt har vært utprøvd i forskjellige prosjekter, men disse systemene har ofte en svært lav returhastighet gjerne realisert med vanlig modem (Tannenbaum, 1996).

### **9.3 Tjenestetilbyder**

Tjenestetilbyder er i denne sammenheng organisasjonen bak en *Åpen kanal* som stiller med maskinvare og programvare for kontribusjon og distribusjon av video. I tillegg til aksessnett og stamnett må man ha en sentralt plassert enhet som sørger for at videoene blir lagret, altså

---

<sup>46</sup> <http://www.bredband.no/>

<sup>47</sup> <https://www.lyse.no/produkter/bredband/?3060>

en server. Det er viktig at denne serveren er pålitelig og at den kan tjene mange brukere samtidig. At disse brukerne har lav teknisk kompetanse bør også forventes. På serveren må det derfor finnes brukervennlig programvare for opplasting og streaming.

### 9.3.1 Maskinvare

Det viktige med maskinvaren er at denne kort og godt må klare å kode data raskt nok, samt å levere data raskt nok ut på nettet. I all hovedsak er dette mest et spørsmål om penger. Det viktigste å tenke på er at systemet skalerer godt. For å få til dette vil man nesten uten unntak snakke om klyngeløsninger, der man kan legge til enheter etter hvert som behovet melder seg. Fordeler med dette er at man også vil få en redundant løsning, slik at tjenesten ikke går ned selv om en enhet skulle feile. Det vil fortsette å kjøre, men med noe mindre kapasitet.

Hovedproblemet til maskinvaren vil være å ha raske nok diskløsninger når den skal levere kanskje flere tusen videostrømmer. En løsning som alltid vil hjelpe er å ha lastbalansering mellom identiske og atskilte systemer. For eksempel kan man med en enkel algoritme og to like systemer få til en enkel lastbalansering ved å tildele annenhver forespørsel til hver maskin. Det største problemet med å duplisere hele systemet flere ganger er at dette fort blir veldig dyrt.

Diskaksess er som tidligere nevnt den største potensielle flaskehalsen. Bruk av såkalt RAID (redundant array of independent disks) vil være nødvendig for å hjelpe på dette i et slikt system. Kort fortalt går RAID ut på at man leser og skriver data til mange diskene samtidig og kan dermed få mye større ytelse enn om man bare hadde en disk. Man øker også datasikkerheten ved at det lagres paritetsdata og selv om en disk skulle ryke vil man kunne gjenopprette alle data. RAID uten denne ekstra sikkerheten er RAID 0, som egentlig ikke er ekte RAID (mangler redundans). RAID finnes i flere typer (0-6), der type 5 eller en kombinasjon av 0 og 1 (10 eller 0+1) vil være mest aktuelt for streamingsystemer. Grensesnittet resten av maskinen skal snakke med diskene over vil typisk også være annen teknologi enn det som er vanlig i standard pc-er. Fiberchannel er en typisk teknologi som brukes til slike formål. Fiberchannel har en overføringshastighet på opptil 1 Gbit/s, med mulighet for å bruke flere kanaler for å oppnå enda større hastighet<sup>48</sup>.

---

[5] 48 Avid (2000), [http://www.dimedia.de/pdf/avid/triligent\\_datasheet.pdf](http://www.dimedia.de/pdf/avid/triligent_datasheet.pdf)

Avid bruker fiberchannel sammen med et proprietært diskspeilingssystem som ligner en kombinasjon av RAID 0 og 1 for å oppnå god nok overføringshastighet i flere av sine produkter spesifikt beregnet for multimediestreaming. Deres Trilligent løsning påstås å skalere opp til 5000 enkeltstrømmer med video på 1 Mbit/s. For å skalere til flere samtidige brukere kan dupliserte systemer på forskjellige geografiske steder med lastbalansering være en løsning.

### 9.3.2 Programvare

Det som er spesielt med en *Åpen kanal* er at seeren også kan være bidragsyter. For at bidraget skal bli tilgjengelig for andre må det overføres til en sentral enhet i nettverket, som er serveren. Overføringen fra bidragsyteren til serveren kalles for opplasting. Applikasjoner som muliggjør dette er ofte basert på ftp eller en sikker utgave som heter sftp (secure file transfer protocol). Felles for disse applikasjonene er at man må skrive inn en ip-adresse til serveren der innslaget skal plasseres og man må gjennom mange tastetrykk før overføringen finner sted. Når overføringene er i gang er det ingen garanti for at den fullføres. Ved brudd må overføringene fortsettes (Resume) eller startes på nytt av brukeren. Med tanke på at brukeren i utgangspunktet skal kunne være uerfaren, bør det lages applikasjoner som gjør dette automatisk. Serveren har en fast adresse, så denne sammen med diverse parametere kan settes av kanalen. Ideelt sett burde det være et program som åpner i det man kommer inn på en opplastingsside til kanalen, der det eneste bidragsyteren trenger å gjøre er å dra fila si over et vindu og slippe. Dette kan f. eks. realiseres som en javaapplet med autoresume.

Når videoinnslaget er kommet frem til serveren må det legges ut på en sendeplan. Applikasjonen for dette må være adgangsbegrenset ettersom det er ved publiseringen at man er tyngt av redaktøransvaret. Redaktørene må kunne søke i innslag under de temaene de er interessert i og evaluere innslaget før de gir det en sendetid. Dersom de ikke finner en sendetid til innslaget, men mener det er forsvarlig å sende det kan de legge det i en videodatabase. Denne databasen skal ikke være adgangsbegrenset. Her skal alle og enhver kunne se på innslagene etter eget ønske og eventuelt sette de på sendeplanen i en "folkets time" dersom man mener andre bør se det.

En annen viktig applikasjon på serversiden er komprimering. Det vil si å gjøre videofilen streambar slik at den kan spilles av på nett. Denne prosessen kan skje i siste trinn under avviklingen. Da må videoen som gjerne er en DV-fil i 30 Mbit/s komprimeres i sanntid til en strøm i et streamingformat med en bitrate som typisk er mindre enn 1 Mbit/s.

Denne prosessen er meget ressurskrevende og gir ikke et like godt resultat som det man kan få ved å gå over videofilen flere ganger (multipass). En annen løsning kan derfor være at videofilen blir komprimert av et program rett etter at den har blitt lastet opp til serveren. Når filen omsider er komprimert kan den bli gjort tilgjengelig for avvikling og Video On Demand.

## 9.4 Pålitelighet

Påliteligheten har betydning for hvordan brukerne oppfatter systemet. Vil i dette kapitlet se nærmere på tilgjengelighet og funksjonssikkerhet. Tilgjengeligheten sier noen om tilgang til tjeneste, mens funksjonssikkerheten sier noe om tjenesten kan tilbys uten avbrudd. Disse to egenskapene spiller en viktig rolle for brukerne. Dersom man gjentatte ganger ikke får tilgang til eller har problem med avbrudd i sendingene, kan det gjøre at potensielle seere ikke ønsker å benytte seg av en åpen kanal.

Først ser vi på disse egenskapene generelt, deretter prøver vi å knytte disse opp mot mulige krav fra brukerne. Vil også prøve å lage en modell for systemet og se på konsekvenser og kritiske komponenter.

### 9.4.1 Tilgjengelighet

Tilgjengelighet er definert som:

$$A(t) = P(I(t) = \text{oppe})$$

som sier hva sannsynligheten er for at systemet er oppe,  $I(t)$  er tilstanden til systemet. Tilsvarende finnes det uttrykk for utilgjengeligheten, som er den inverse av  $A(t)$ :

$$U(t) = 1 - A(t)$$

Ovenstående uttrykk kalles øyeblikks(u)tilgjengelighet, og gir (u)tilgjengeligheten ved et gitt tidspunkt  $t$ . Når tiden går mot uendelig får vi en får vi den asymptotiske (u)tilgjengeligheten som beskriver systemet ved et tilfeldig valgt tidspunkt i fremtiden:

$$A = \lim_{t \rightarrow \infty} A(t)$$

$$U = 1 - A$$

### 9.4.2 Funksjonssikkerhet

Som for tilgjengeligheten, finnes det også noen relevante mål for funksjonssikkerheten:

- 1) Funksjonssannsynligheten er sannsynligheten for at man ikke får noe feil i en gitt tidsperiode  $t$ :

$$R(t) = P(T_{FF} > t) , \text{ der } T_{FF} \text{ er tid til første feil.}$$

- 2)  $R(t)$  kan videre brukes til å avlede midlere tid til første feil (MTFF), som er et mål for funksjonssikkerheten til et system:

$$MTFF = E(T_{FF}) = \int_0^{\infty} R(t) dt$$

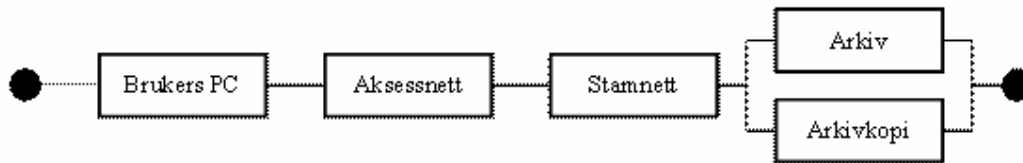
### 9.4.3 Modell for systemet

For å kunne analysere et system, og si noe om påliteligheten finnes det to modeller som er mye brukt<sup>49</sup>. Det er tilstandsmodeller og strukturmodeller. Her bruker vi strukturmodeller som gir en oversiktlig grafisk fremstilling av systemet. Elementer kan da representeres som blokker i det som kalles et pålitelighetsblokkskjema. Disse blokkskjemaene kan brukes for å bestemme tilgjengelighet og funksjonssannsynlighet. Hvert element har sin tilgjengelighet og funksjonssannsynlighet. Når man kombinerer elementene i et system, kan man da finne målene man ønsker for det totale systemet. Dersom man har en seriestruktur, må alle elementene fungere for at systemet skal virke. Et annet tilfelle er dersom man har parallellstruktur, da må minst et av elementene fungere for at systemet skal virke. Det er igjen mulig å kombinere serie- og parallellstruktur.

---

[6] 49 Emstad, Heegaard, Helvik (2001) Pålitelighet og ytelse i informasjons- og kommunikasjonssystem, Institutt for telematikk NTNU 2001





Figur 9-2: Blokkskjema for beregning av pålitelighet

Figuren over viser et mulig enkelt blokkskjema for påliteligheten til en åpen kanal sett fra brukeren. Denne modellen viser en seriestruktur som inkluderer en parallellstruktur. Modellen er en forenkling av systemet, men viser prinsippet for virkemåten.

Bruker sin PC er datamaskinen han bruker for å se på en åpen kanal, og aksessnett er aksesslinjen som knytter bruker til Internet. Disse to elementene må virke for å kunne benytte seg av en åpen kanal. Det er flere ting som kan gå galt i disse to elementene, men vi har her valg å modellere de hver med en blokk, som samlet har en tilgjengelighet og funksjonssannsynlighet som dekker alle aktuelle feil. Elementet stamnett, er nettverksforbindelsen fra en åpen kanal og frem til brukerens aksesslinje. Dersom linker eller noder går ned, vil en pågående trafikk likevel komme frem grunnet omruting i nettet. Feilrate knyttet til dette elementet vil derfor være lite i forhold til andre feilrater i skjemaet. Serverne til en åpen kanal er modellert som en parallellstruktur. Dette for å ta med redundans. Med redundans mener man i forbindelse med pålitelighet ekstra ressurser, som brukes i forbindelse med feiltolerante system. I et slikt system vil et element kunne ta over for et annet tilsvarende, dersom det skulle feile. Ekstra ressurser kan være hardware, software, informasjon eller tid.

Viser videre med et lite regneeksempel, hvordan redundans kan bidra til å øke pålitelighet:

Antar at en server feiler 5 ganger i måneden, og nedetid hver gang da er 2 timer. Tiden til feil er negativt eksponensialfordelt. Hva er da sannsynligheten for at den kan fungere et døgn uten avbrudd?

- 1) Ser først på pålitelighet for en enkelt server. Et slikt system vil da ikke fungere dersom denne serveren går ned.



Figur 9-3: Blokkskjema av enkel server

Tilgjengeligheten på et tilfeldig tidspunkt:

$$A = 0,986$$

Funksjonssannsynligheten for et døgn:

$$R(24) = 0,846$$

2) Ser videre på et system som har to like servere i parallell. Her kreves det at en av de to serverne fungerer.



Tilgjengeligheten på et tilfeldig tidspunkt, der tilgjengeligheten for hver server er lik:

Figur 9-4: Blokkskjema av redundant server  $A_{\text{parallell}} = 0,9998$

Funksjonssannsynligheten for et døgn blir:

$$R_{\text{parallell}}(24) = 0,976$$

Utrekninger viser at redundans i dette regneeksempelet øker påliteligheten til et system. Ser her at tilgjengeligheten og funksjonssannsynligheten øker vesentlig fra en til to servere. Dette skyldes at sannsynligheten for at begge servere feiler i samme tidsrom er liten. Ved å legge til enda flere servere i parallell, vil funksjonssannsynligheten bli enda større. Det er videre slik at effekten av antall elementer i en parallellstruktur er mest tydelig den første tiden.

## 10 Brukerbasert registrerings- og opplastningssystem:

I prosjektoppgaven ble det utviklet et system for registrering av brukere og organisasjoner med formål å ha en felles og helhetlig oversikt (database) over medlemmer. Det ble også utviklet som et system for opplastning av innslag med registrering av informasjon samt publisering på nett.

Systemet skulle være brukervennlig uten at det går nevneverdig ut over sikkerheten.

Dette ble løst ved at en bruker har en applikasjon å forholde seg til, nemlig den webleseren som brukeren liker best. En webleser, ett brukernavn og ett passord. All funksjonalitet skal gå gjennom webleseren.

### 10.1 Virkemåte

1) Redaktør registrerer seg og sin medlemsorganisasjon i MySQL databasen.

(Det skrives ut et skjema som signeres og sendes til statens medieforvaltning der redaktøren blir formelt registrert for sending på satellitt.)

2) Redaktør registrerer informasjon om sitt innslag i MySQL databasen.

(Her må redaktøren registrere TEMA og alt rettighetsbelagt materiale som er brukt og det må bekreftes at festerettigheter er avklart. Under registreringen bør det være mulig å laste opp innslaget til katalog med organisasjonsnavnet på FTP-serveren.)

3) Et script flytter innslaget fra FTP-serveren til en komprimeringsenhet der DV-fila komprimeres til en streambar MPEG-4 fil og en broadcastkvalitet MPEG-2 fil for hhv evaluering på nett og satellittsending. DV-fila på komprimeringsenheten slettes og de komprimerte filene flyttes til hver sin katalog på FTP serveren.

4) Når komprimeringa av MPEG-4 er ferdig endres flagget "komprimert MPEG-4" fra 0 til 1.

5) Det ble laget en intern nettside som alle redaktører har tilgang til. Denne viser ved hjelp av en SQL-spørring hva som er registrert i databasen sortert etter tema/organisasjon/tittel. Dersom flagget "komprimert MPEG-4" er satt til 1 lages det også en link som gjør at man kan streame innslaget direkte.

### 10.2 Kravspesifikasjon for publiseringssystemet:

#### 10.2.1 Prosedyre for registrert bruker

1. Bruker laster opp et innslag.

2. Bruker legger inn informasjon om innslaget.

3. Innslaget blir automatisk (re-)komprimert til nødvendige format.
4. Bruker krysser av for om innslaget skal bli gitt vederlagsfritt til Åpen kanal.
5. Innslaget blir tilgjengelig på en offentlig nettside.

### 10.2.2      Krav til sikkerhet

- All brukerinmatning må valideres for "ødeleggende" (potensiell destruktiv) kode.
- All brukerinmatning må valideres for duplikater, inntastningsfeil, verdier utenfor lovlig område etc.
- Ikke gi tilgang til scripts eller funksjoner uten at brukeren har en godkjent og registrert konto.
- Ikke gi tilgang til scripts kildekode under noen omstendighet.
- Sikre passord i script: ikke ha passord i klartekst i scripts, putt passord inn i en tekstfil utenfor allment tilgjengelige filområder.
- Ikke gi tilgang til funksjonalitet som en bruker ikke skal ha tilgang til.
- På sikt bør det legges en plan for løpende oppgradering av programvare. Dette krever en fast driftsperson.
- Løpende vurdering av eksisterende egenlaget kode med tanke på "farlige" feil som kan utnyttes eller som kan utgjøre et potensielt problem i fremtiden. (Vil kreve en fast person som utviklere eller en allsidig driftsperson.)

### Innslagenes gang i systemet

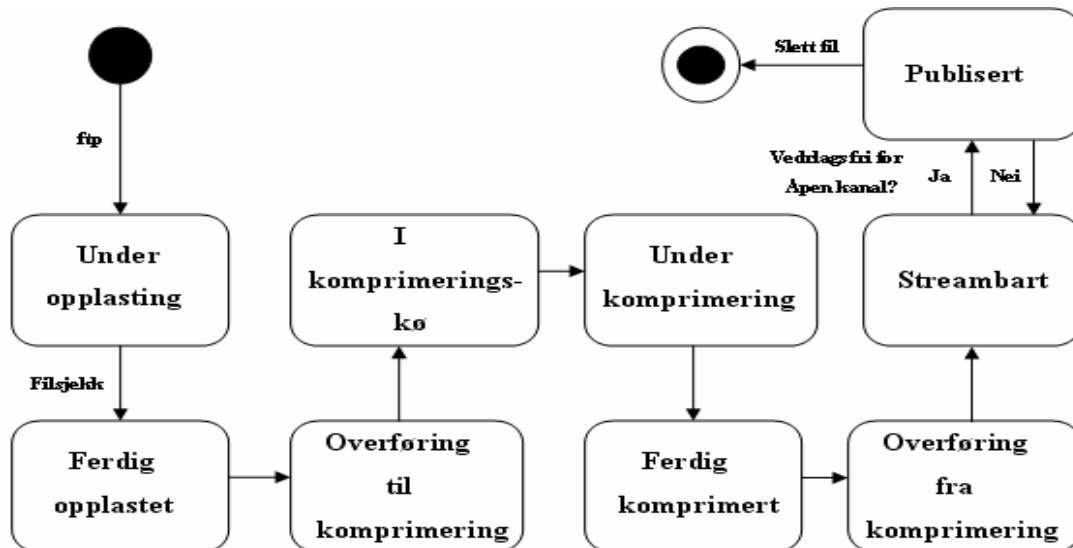
Tilstanden til innslagene i systemet må endres etter hvert som de blir prosessert. Til det er det skrevet noen PHP script som kommuniserer med en database der filenes tilstand lagres

- Skript som automatisk oppdaterer databasen med hvilke filer som er tilgjengelig.
- Skript som komprimerer ferdig opplastede filer til nødvendige formater og setter et flagg som sier at originalfila er ferdig komprimert.
- System som detekterer om en fil er ferdig lastet opp og endrer status på tilhørende flagg.

Det er mulig å lagre informasjon om innslaget mens det lastes opp. Dette for å unngå at redaktøren skal behøve å gå inn på nettsiden to ganger.

Brukeren får underveis se hvilken tilstand fila er i.

For brukeren vises kun tilstandene ”under opplasting”, ”under komprimering”, ”ferdig komprimert”, og ”publisert”. ”Under komprimering” på menyen kan for enkelhets skyld inkludere ”ferdig opplastet”, ”overføring til komprimering”, ”i komprimeringskø”, ”under komprimering.”, ”ferdig komprimert”, ”overføring fra komprimering” i tilstandsdiagrammet nedenfor. Valget ”ferdig komprimert” på menyen er kun tilstanden ”streambar”, men med tanke på at de komprimerte filene er meget små vil tidsavstanden mellom den egentlige ”ferdig komprimert” tilstanden og denne kun være noen få sekunder. Tilstanden publisert oppnås kun ved å krysse av ja på ”vedlagsfritt for åpen kanal” for en streambar fil. Da skal fila bli tilgjengelig på den offentlige nettsiden under valgt tema. Ved å krysse av nei på denne er tilstanden ”streambar” og fila blir igjen kun tilgjengelig på organisasjonens private side.



Figur 10-1: Tilstandsdiagram for et innslag

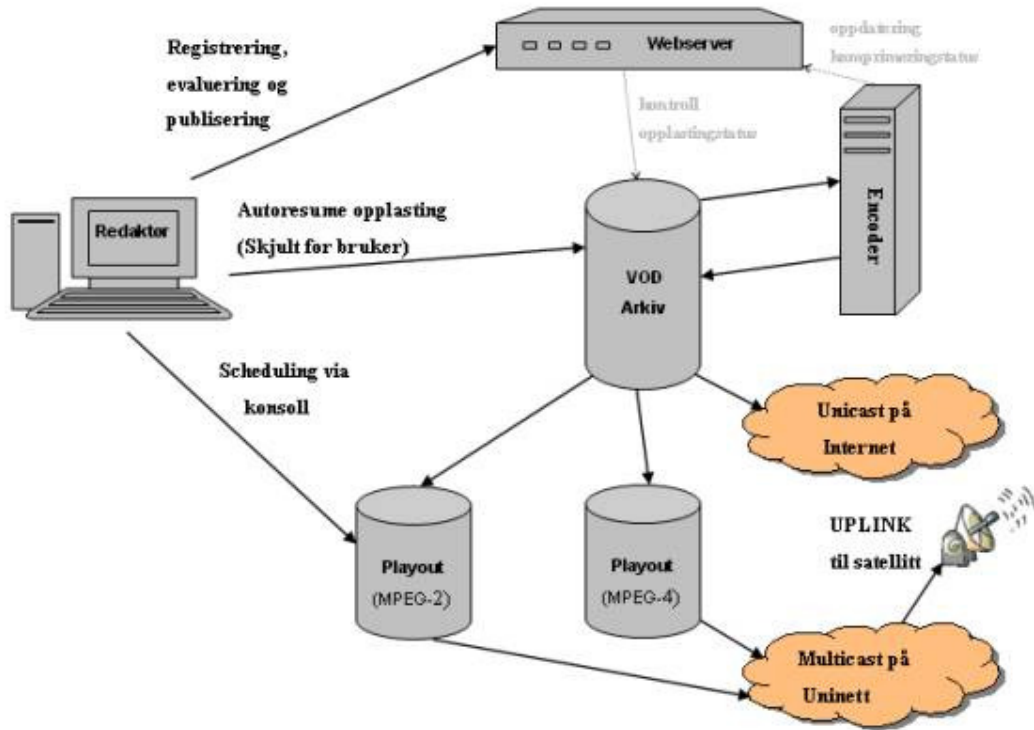
## 11 Implementering av demonstrasjonsmodell

### 11.1 Infrastruktur

Av figuren nedenfor kan man se at Video On Demand arkivet er den mest sentrale komponenten. Dette er en FTP-server med redundans og som forklart tidligere i oppgaven styrker dette systemets pålitelighet. Komprimeringsenheten bruker ISO MPEG-4 og det er det vi har funnet mest hensiktsmessig å satse på videre i motsetning til de proprietære systemene som ble vurdert (Quicktime, Real, Windowsmedia). Dette kan begrunnes med at formatet er mer fremtidsrettet både med tanke på muligheter innenfor interaktivitet og videre utvikling for å oppnå effektive komprimeringsalgoritmer. En åpen standard som H.264 eller MPEG AVC vil også kunne vinne frem i kampen om å bli det nye formatet i TV-nettet. MPEG-2 er modent for utskiftning og videoencoderen i MPEG 4 blir vurdert som den beste arvtageren av bla teknisk personell i Telenor Holding, eieren av Telenor Sattelite Services. Med andre ord kan vi ved ligge langt fremme med vår utprøving av dette formatet når et slikt skifte kommer også være godt rustet til en TV-sending. Det er likevel MPEG-2 som vurderes som den mest sannsynlige løsningen ved Åpen kanal sin sendestart. Foreløpig vil det derfor måtte kjøres to parallelle systemer for nett og TV. Men det som en dag kan sendes på satellitt i DVB MHP (MPEG-2) kan også testes ut over IP.

Systemet fungerer slik: Redaktør laster opp, registrerer, evaluerer og til slutt publiserer innslag. Innslagene blir lastet opp i Video On Demand-arkivet (VOD Arkiv) mens registrering, evaluering og publisering foregår via webserveren. Når et innslag er lastet opp, tar encodingstasjonen tak i fila og (re-)komprimerer det til andre formater, blant annet mpeg4 i forskjellige bitrater og etter hvert i MPEG-2.

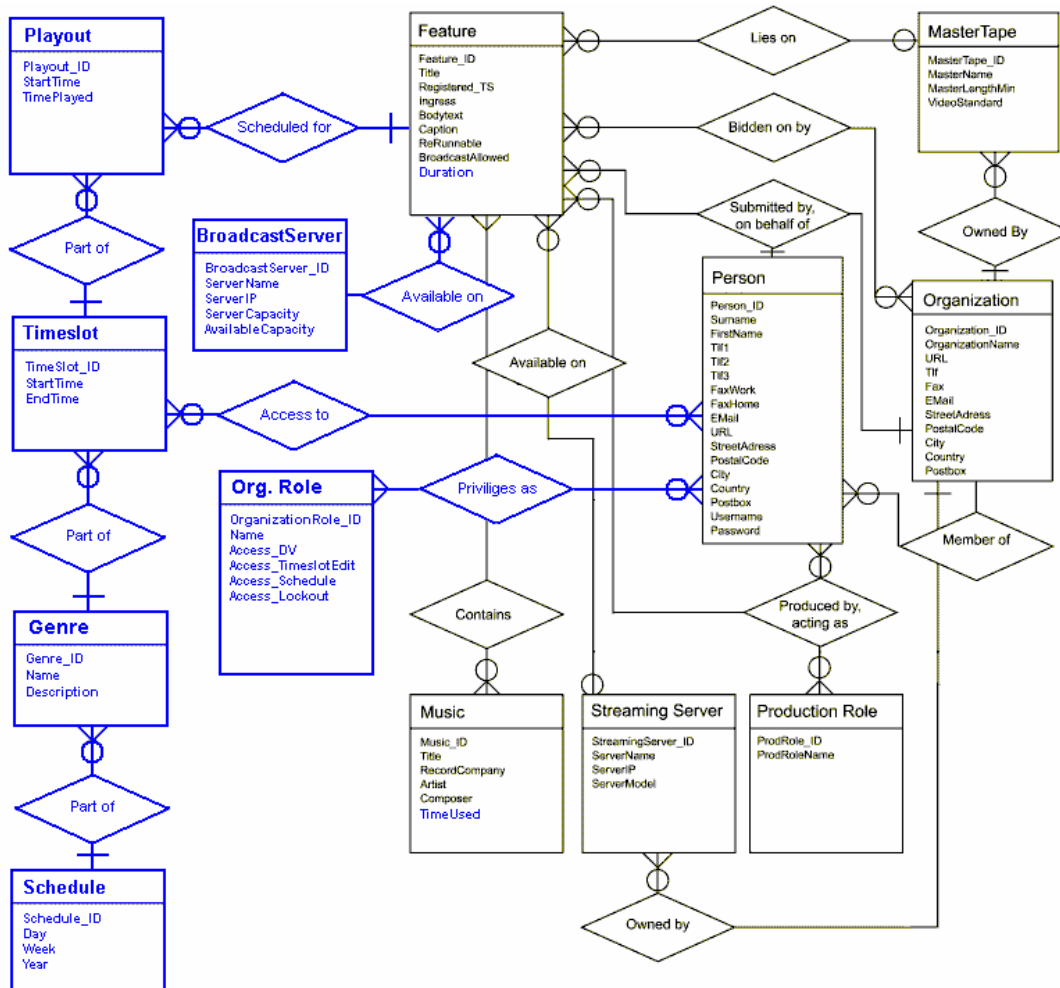
Når et innslag er ferdig komprimert kan redaktøren så evaluere innslaget, det vil si se på det gjennom video on demand. Dersom redaktøren velger å publisere det, vil det bli tilgjengelig for offentligheten. Det kan også legges ut på en TV sending via en konsoll fra Cardinal som styrer playout-enheten med MPEG-2 filer. (Denne vil etter hvert kunne styres fra nettsiden)



Figur 11-1: Infrastruktur for demonstrasjonsmodell

## 11.2 Database

Databasen skulle opprinnelig baseres på en modell utviklet i samarbeid med Student TV i Trondheim. Denne hadde ikke tatt høyde for avviklingssystemet og ble derfor utvidet med noen tabeller og attributter. Tatt i betraktning formålet for systemet, som ikke lenger var en arkivtjeneste virket denne unødige komplisert og vi valgte derfor å forkaste hele modellen.



Figur 11-2: Databasemodell i opprinnelig kravspesifikasjon (Arkivtjeneste)

Den nye modellen beskrives her med utskrifter fra MySQL databasen. Argumentene for hvorfor de ulike tabellene med deres attributter er valgt er å finne under hvert punkt som der tabellene beskrives.



```

+-----+
| Tables_in_openchannel |
+-----+
| category              |
| comments              |
| content               |
| log                   |
| organisasjoner       |
| priv                  |
| privgroup             |
| privkeynames          |
| redaktorer            |
| users                 |
+-----+

```

**Category**

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| categoryid | int(10) unsigned   |      | PRI | NULL    | auto_increment |
| name       | varchar(50)        | YES  |     | NULL    |                |
| comment    | varchar(100)       | YES  |     | NULL    |                |
| deleted    | enum('yes','no')   | YES  |     | no      |                |
+-----+-----+-----+-----+-----+-----+

```

**Comments**

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| commentid  | int(10) unsigned   |      | PRI | NULL    | auto_increment |
| comment    | varchar(255)       | YES  |     | NULL    |                |
| userid     | int(10) unsigned   | YES  |     | NULL    |                |
| dt         | timestamp(14)      | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+

```

**Content**

```

+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| contentid  | int(10) unsigned   |      | PRI | NULL    | auto_increment |
| filename   | varchar(255)       | YES  |     | NULL    |                |
| orgid      | int(10) unsigned   | YES  |     | NULL    |                |
| description | varchar(255)       | YES  |     | NULL    |                |
| title      | varchar(50)        | YES  |     | NULL    |                |
| tapename   | varchar(100)       | YES  |     | NULL    |                |
| length     | int(10) unsigned   | YES  |     | NULL    |                |
| tc_in      | varchar(8)         | YES  |     | NULL    |                |
| tc_out     | varchar(8)         | YES  |     | NULL    |                |
| categoryid | int(10) unsigned   | YES  |     | NULL    |                |
| filesize   | int(10) unsigned   | YES  |     | 0       |                |
+-----+-----+-----+-----+-----+-----+

```

## Åpen kanal

fin_upload	enum('yes','no')	YES		no	
begin_compress	enum('yes','no')	YES		no	
fin_compress	enum('yes','no')	YES		no	
gen_script	enum('yes','no')	YES		no	
randfilename	varchar(50)	YES		NULL	
number_played	int(10) unsigned	YES		0	
fexists	enum('yes','no')	YES		yes	
vfri	enum('yes','no','')	YES			

### Log

Field	Type	Null	Key	Default	Extra
logid	int(10) unsigned		PRI	NULL	auto_increment
userid	int(10) unsigned	YES		0	
dt	timestamp(14)	YES		NULL	
comment	varchar(250)	YES		NULL	
ip	varchar(15)	YES		NULL	
additional	varchar(100)	YES		NULL	

### Organisasjoner

Field	Type	Null	Key	Default	Extra
orgid	int(10) unsigned		PRI	NULL	auto_increment
orgnavn	varchar(40)	YES		NULL	
orgnummer	varchar(20)	YES		NULL	
telefon	varchar(15)	YES		NULL	
lonnpersonell	varchar(5)	YES		NULL	
fjernsynvideoprod	varchar(15)	YES		NULL	
forvsendstart	varchar(20)	YES		NULL	
formidlesmedium	varchar(40)	YES		NULL	
timerperuke	varchar(5)	YES		NULL	
adresse1	varchar(40)	YES		NULL	
adresse2	varchar(40)	YES		NULL	
webaddress	varchar(100)	YES		NULL	

### Priv

Field	Type	Null	Key	Default	Extra
privid	int(10) unsigned		PRI	NULL	auto_increment
keyname	varchar(30)	YES		NULL	
keyvalue	varchar(30)	YES		NULL	
privgroupid	int(10) unsigned	YES		NULL	

### Privgroup

Field	Type	Null	Key	Default	Extra
privgroupid	int(10) unsigned		PRI	NULL	auto_increment
groupname	varchar(30)	YES		NULL	

#### Privkeynames

Field	Type	Null	Key	Default	Extra
privkeynameid	int(10) unsigned		PRI	NULL	auto_increment
keyname	varchar(30)	YES		NULL	

#### Redaktorer

Field	Type	Null	Key	Default	Extra
redid	int(10) unsigned		PRI	NULL	auto_increment
fullnavn	varchar(40)	YES		NULL	
personnummer	varchar(20)	YES		NULL	
adresse1	varchar(40)	YES		NULL	
adresse2	varchar(40)	YES		NULL	
jobbtelefon	varchar(15)	YES		NULL	
hjemtelefon	varchar(15)	YES		NULL	
mobiltelefon	varchar(15)	YES		NULL	
epost	varchar(40)	YES		NULL	
telefaks	varchar(15)	YES		NULL	

#### Users

Field	Type	Null	Key	Default	Extra
userid	int(10) unsigned		PRI	NULL	auto_increment
username	varchar(40)	YES		NULL	
password	varchar(40)	YES		NULL	
lastip	varchar(16)	YES		NULL	
sessid	varchar(32)	YES		NULL	
orgid	int(10) unsigned	YES		NULL	
redid	int(10) unsigned	YES		NULL	
step	int(11)	YES		0	
privid	int(10) unsigned	YES		0	

## 11.2.1 Beskrivelse av tabeller i databasen

### “users”

Denne tabellen knytter sammen hele databasen. Det er det første objektet som blir “kontaktet” når en bruker logger seg på og gjennom fremmednøklerne kan man hente mer informasjon om denne brukeren, hvilken organisasjon brukeren tilhører og hvilke rettigheter han eller henne har.

De fleste feltene er selvforklarende: username og password brukes for å identifisere brukeren før og under pålogging. Lastip og sessid brukes for å identifisere brukeren etter pålogging og deretter under videre bruk. Orgid, redid og privid er alle fremmednøkler til henholdsvis sin organisasjon, ytterligere informasjon om seg selv og hvilke rettigheter brukeren har. Step er et tall som indikerer hvor i registrerings-prosessen en bruker befinner seg.

### “redaktorer”

Her legges ytterligere informasjon om en registrert bruker. Tabellen burde egentlig hatt et mer generelt navn da det er rettighetssystemet som avgjør hvilken type bruker en person er. Her er alle feltene selvforklarende.

### “organisasjoner”

Organisasjoner inneholder informasjon om organisasjonene som brukerne er tilknyttet. Feltet fjernsynvideoprod brukes for å fortelle om en organisasjon har slik produksjon per i dag. Lonnpersonell indikerer om organisasjonen har lønnet personell. Formidlesmedium indikerer om organisasjonen ønsker å formidle på fjernsyn og nett eller kun fjernsyn. Dersom organisasjonen har planer om å sende på fjernsyn må ekstra informasjon samles inn.

### “content”

Denne tabellen er basis for hele “browsing-“ systemet. Det er en tabell med oversikt over alle filene som er lastet opp, hvilken status filene har samt detaljert informasjon om innslagene som filene representerer.

Filename er selve fila. Orgid forteller hvilken organisasjon som eier innslaget. Title, description, tapename, tc\_in og tc\_out forteller hva innslaget heter samt litt

bakgrunnsinformasjon og hvor man kan finne igjen innslaget om det skulle forsvinne fra systemet. Categoryid forteller hvilken kategori innslaget ligger inne under.

Filesize brukes av et automatisert script for å finne ut om en fil er lastet opp. Når filstørrelsen er lik som forrige gang scriptet kjørte, er innslaget lastet opp. Da settes fin\_upload til "yes" hvilket indikerer at fila kan komprimeres. Dette er en løsning som har et potensiale til å feile. Dersom en bruker laster opp en fil og opplastningen blir brutt, vil dette systemet "oppdage" filen som ferdig opplastet. Dette skal senere løses med et skreddersydd opplastingssystem.

Feltet fin\_upload indikerer for et automatisk komprimeringsscript at fila er klar til komprimering. Script blir generert, gen\_script settes til "yes". Når de genererte scripta begynner å kjøre settes begin\_compress til "yes" og når komprimering er ferdig settes fin\_compress til "yes". I en senere versjon vil disse feltene bli slått sammen til et enumerert statusfelt.

Number\_played forteller hvor mange ganger innslaget er spilt. Dette er nyttig informasjon i forhold til eventuelle Tonorettigheter. Fexists forteller om en fil ikke er eller har vært tilgjengelig sist noen trykket på en link for å se et innslag. Denne kontrollen skal etter hvert flyttes til et opprydningsscript som foreløpig ikke eksisterer. Vfri er kort for vederlagsfritt og indikerer hvorvidt et innslag kan benyttes vederlagsfritt for Åpen Kanal.

### **"category"**

Tabellen category er rett og slett en liste over temaer. Hvert innslag hører inn under et tema. Feltet deleted brukes for å indikere at et tema ikke lenger skal brukes. I og med at det er mange filer som kan peke på et tema kan det ikke bare slettes uten videre. Derfor er dette flagget lagt til slik at en eventuell opprydningsrutine kan slette når ingen innslag lenger peker på et tema.

### **"priv", "privgroup", "privkeys"**

Disse tabellene utgjør rettighetssystemet. Privgroup definerer grupper som "admin", "redaktør", "vara", "gjestebruker" og så videre. Priv-tabellen inneholder nøkkelord som "edit\_own"="yes" og/eller "delete\_permanent"="yes". Disse nøkkelordene er hardkodet i php-koden slik at en bruker blir validert for hver operasjon han eller henne ønsker å utføre.

Privkeys er bare en oversikt over alle privilegier, altså nøkkelord som kan brukes slik at det skal være lett for en administrator å sette rettigheter på en enkel og grei måte.

**”log”, ”comments”**

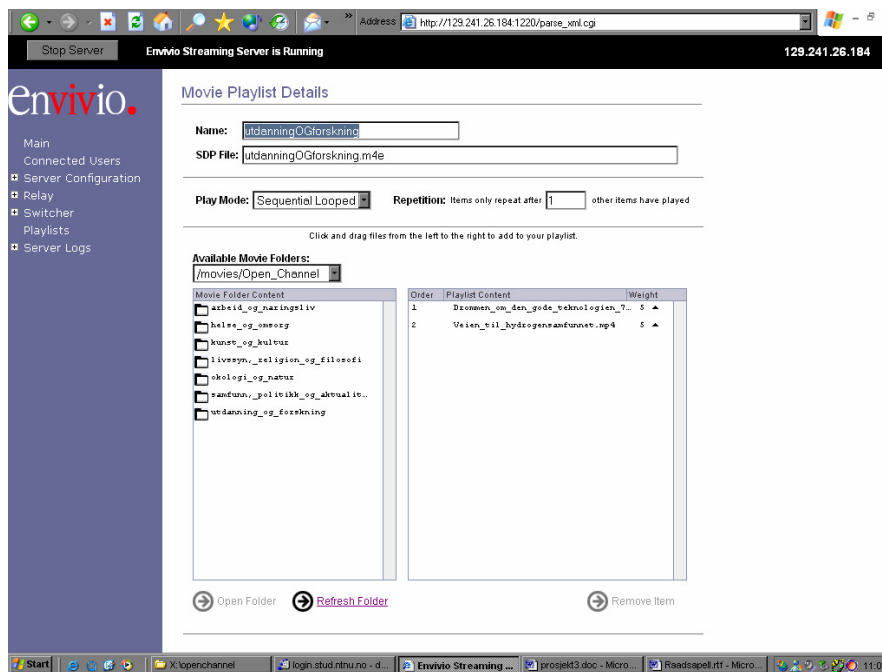
Log brukes for å logge hendelser i systemet mens comments brukes for å gi brukerne en mulighet til å komme med tilbakemeldinger.

## 12 Forslag til utbedring av systemet

### 12.1 Automatisk multicast scheduling

Åpen kanal kjører i dag MPEG-4 multicast sendinger fra Envivio 4Caster som spiller disse i en karusell. Vi har valgt å spille av innslagene sekvensielt i en loop. Alternativt kan de spilles random med ulik vekt, eller bare en gang i den rekkefølgen de står i. Når man skal lage en playlist må filene ligge lokalt og da må de lastes ned fra arkivet ved hjelp av en SSH klient som er logget på 4Caster. Arkivet har ikke en SSH server installert så man må skrive inn FTP-kommandoene manuelt (get <filnavn> osv.) fra SSH konsollen. Dette er en komplisert prosess å lære opp folk i. Ikke bare må de kunne komandolinje-nedlasting, men de må også lære seg å bruke web-grensesnittet til Envivio for å sette innslaget på karusellen. Det hadde derfor vært ønskelig at dette ble utført automatisk hver gang en redaktør publiserer et innslag på nett.

Figuren nedenfor viser web-grensesnittet til Envivio 4Caster: <http://129.241.26.184:1220>



Figur 12-1: Web-grensesnitt for programmering av sendeplan i multicast strøm

Filene som brukes i playlisten ligger her: (venstre vindu i figuren over)

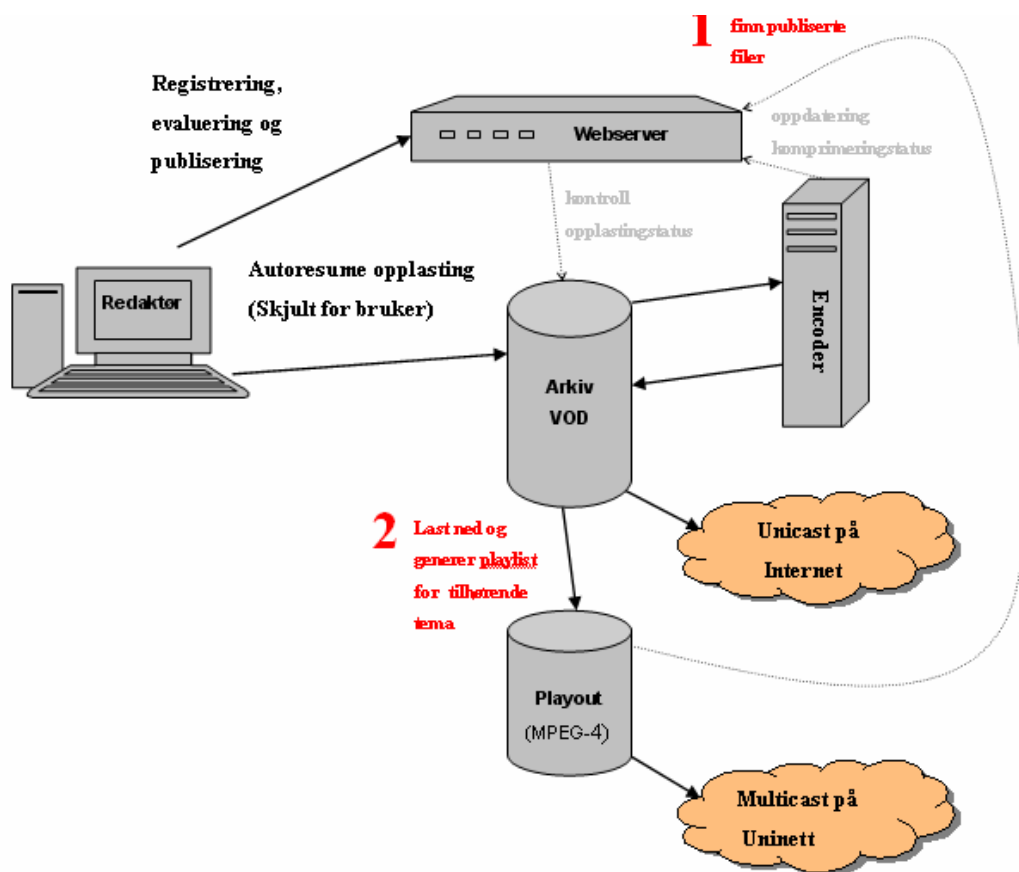
129.241.26.184/movies/Open\_Channel/<tema>

Selve playlisten ligger her: (Tilgjengelig fra web som <http://129.241.26.184/<Name>/<SDP>>)

129.241.26.184/usr/Envivio/ESS/Movies/<tema>

### 12.1.1 Kravspesifikasjon

Envivio 4Caster kan styres ved hjelp av et script. Syntaksen er godt beskrevet i Envivio sin manual for programmet. Det som skal lages er et system som først spør databasen i webserveren hvilke filer som er publisert på nettsiden. Disse skal lastes ned fra det ftp baserte arkivet. Deretter skal det genereres et script som lager playlisten med alle filer som er publisert i en sekvensiell loop for temaet de tilhører. Det skal med andre ord genereres en playlist for hvert tema i databasen der det finnes innhold som er publisert. Innslag som ikke lenger har status som publisert bør fjernes fra palylisten. Disse operasjonene bør skje jevnlig slik at man får med seg eventuelle endringer. Systemet er Linux-basert. En cron job kan derfor ta seg av dette. Playlistene skal være tilgjengelig fra videoarkivet på nettsiden [www.openchannel.no](http://www.openchannel.no).



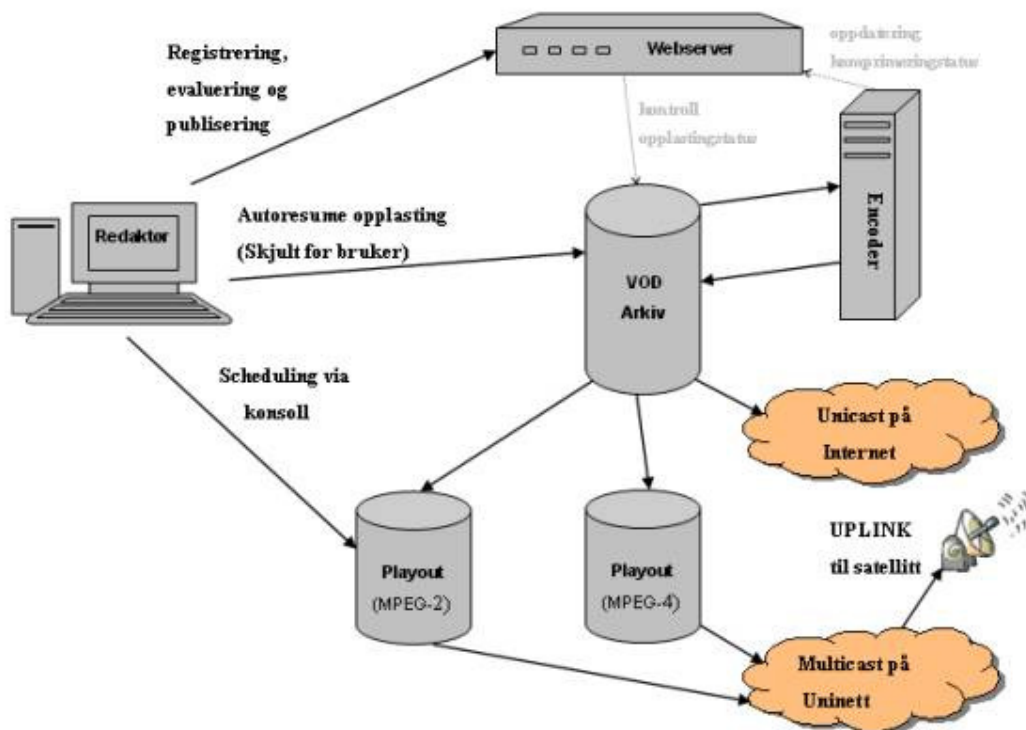
Figur 12-2: Infrastruktur for realisering av automatisk sendeplansprogrammering



## 12.2 Viderutvikling for lokale sendinger og økt redundans

Det opprinnelige systemet med kun en sentral TV-avviklingsenhet er det som vil bli tatt i bruk i de første prøvesendingene. Dette kan videreutvikles til å ha flere lokale avviklingsenheter.

### 12.2.1 En landsdekkende sending i prøvedrift

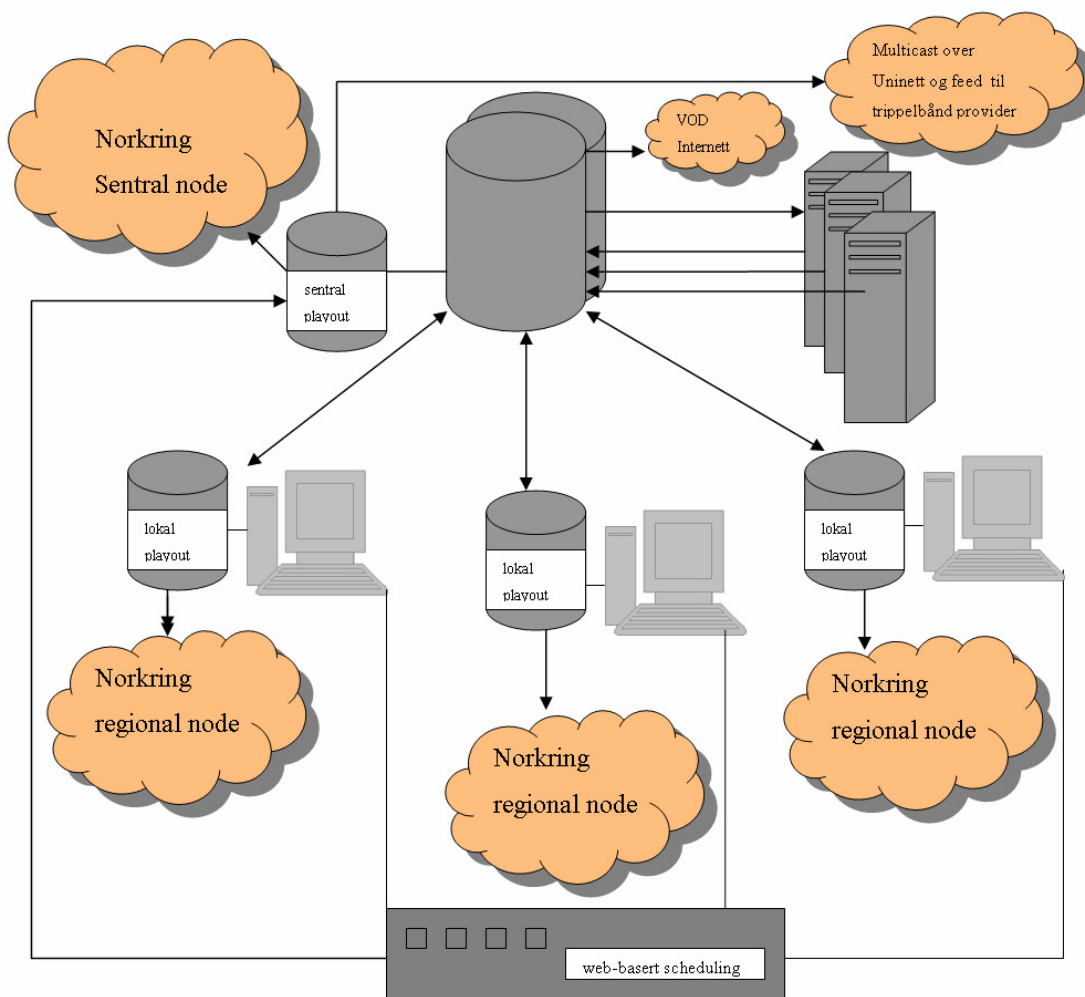


Figur 12-3: Systemets infrastruktur i dag

Prøvedrift vil begynne i april 2006. Bruken er da som følger:

- 1.Redaktør logger seg på Åpen kanal sin nettside (webserver).
- 2.Redaktør laster opp innslaget med en skreddersydd opplastingsklient (til VOD-arkiv).
- 3.Klienten håndterer alle bakenforliggende prosesser etter at den lokale filen er valgt.
- 4.Redaktør registrerer informasjon om innslaget og angir om det skal publiseres.
- 5.Encoderen komprimerer innslaget til MPEG4, H.264 og MPEG2TS (manuelt).
- 6.MPEG4 legges ut på nettsiden dersom redaktøren vil publisere det (VOD)
7. Filer av typen H264 over MPEG2TS legges i et TV-arkiv for en fremtidig sending
- 9.I prøvedrift simuleres også TV-sending ved at man avvikler MPEG2TS på hylleware
- 10.Enheten som gjør dette kan plasseres i et fjersynsnett når formidlingsplikten er på plass.

12.2.2 Lokale og landsdekkende sendinger i full drift



**Figur 12-4: Infrastruktur med mulighet for lokale sendinger og økt redundans**

Endringer fra tidligere system:

1. Automatisk synkronisering av lokalt og sentralt innhold med backup i full kvalitet
2. Uavhengige komprimeringsenheter for WM, H.264 og MPEG2TS
3. Samme grensesnitt for programmering av nett- og TV-sending
4. Uavhengige lokale playoutenheter for regionale sendinger
5. System for utveksling av innhold i full kvalitet
6. Mulighet for distribuert samhandling (live)

## 13 Utvikling i diplomoppgaven

Åpen kanal har gjennom utprøving av det nettbaserte publiseringssystemet høstet mange verdifulle erfaringer fra medlemmene. Det som utmerker seg mest som en terskel i bruken av systemet er kontribusjonen. Dette er den delen som skiller Åpen kanal mest fra andre TV-kanaler i et teknisk perspektiv. Det vil være avgjørende for kanalens åpenhet at opplastingen blir så enkel at folk uten teknisk erfaring mestrer den. I Åpen kanal er det ikke teknikere som bygger opp arkivet slik som i andre kanaler, men helt vanlige folk med noe på hjertet. Dette unike behovet gjør det vanskelig å få tilgang på hyllevare for å realisere brukervennligheten deltakerne ønsker. Hovedvekten av utviklingen i denne oppgaven har derfor bestått i å utvikle en skreddersydd opplastingsapplikasjon skrevet i C (kildekode er vedlagt).

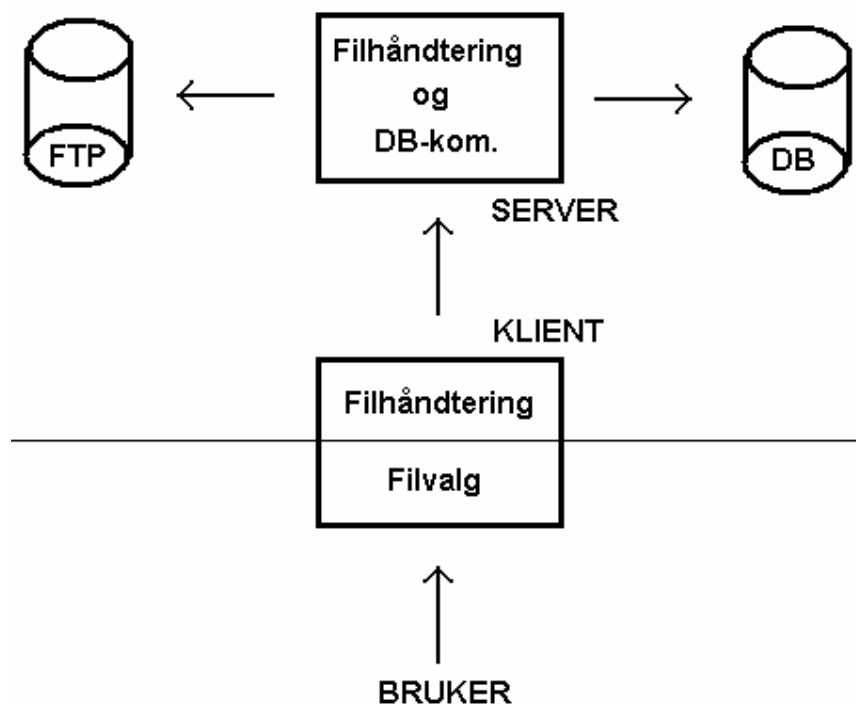
En annen problemstilling som fikk lite fokus i tidligere utvikling var avvikling for TV. Systemet er opprinnelig konstruert for å integreres med en playout-enhet, men denne delen ble aldri realisert på grunn av mangel på utstyr. NTNU kjøpte inn en playout-enhet fra Cardinal året etter det første utviklingstrinnet. Med denne kunne deltakerne styre sendeplanen over nett via en konsoll. Det viste seg at systemet til Cardinal inneholdt mange sikkerhetsfeil og serveren gikk ned på grunn av et hackerangrep kort tid etter at den var satt opp. Dette er en tjeneste som er viktig for produsentene i Åpen kanal å få erfaring med ettersom TV-sending snart kan være en realitet. NTV har allerede begynt testsendinger der de benytter tilsvarende system (MHP). Åpen kanal har i henhold til NTV sin utbyggerkonsesjon en reservert plass og trenger erfaring med reelle TV-avviklinger før innholdet skal gjøres tilgjengelig for alle på det riksdekkende bakkenettet. Avviklingssystemet har derfor blitt satt opp igjen, men med en større grad av sikkerhet. Åpen kanal har ikke konsesjon så utprøvingene finner sted i et lukket nett. Prosesseringen er likevel fullverdig med unntak av en antenneforsterker (Antennesignalet går over kabel direkte inn i set-top boksen).

Infrastrukturen i Åpen kanal har vist seg svært sårbar for hackerangrep. I tillegg til playout-serveren gikk FTP-serveren og streaming-serveren ned under utprøvingen. Det sistnevnte ble et langvarig problem ettersom maskinvaren ble sendt på service i forsøk på å gjenskape innholdet. Midgard Medialab har ikke lyktes med å hente ut dataene i skrivende stund. Dette er den mest sentrale komponenten i systemet. Det ble mot slutten av oppgaven klart at vi måtte bygge opp det tidligere systemet på nytt på ny maskinvare. Ettersom Åpen kanal ikke har mulighet til å kjøpe noe måtte det nye systemet baseres på gratis programvare.

Det har vært en tidkrevende prosess å få tak i utstyr og programmer som kan utføre tilsvarende oppgaver som det vi hadde tilgang på gjennom Midgard. Erfaringen har likevel gjort oss mer bevisst på behovet for redundans og skalerbarhet i programkoden. Da jeg skiftet ut deler av programkoden sørget jeg for at å tilpasse den slik at et skifte av hardware kun medfører små konfigurasjonsendringer. Det ble ellers påbegynt et arbeide med å sørge for redundans der jeg har satt opp alternative streamingservere med tilhørende komprimeringsalgoritmer. Systemet har fortsatt sentrale sårbarheter. Drøfting av pålitelighet vil bli fulgt opp videre av andre masterstudenter ved NTNU.

### *13.1 Brukervennlig opplasting*

Første trinnet i systemet, opplastingen, ble tidligere realisert med en freeware FTP-klient. Her må deltakerne taste inn brukernavn, passord, servens IP-adresse etc. Dette er parametere som kan scriptes, men det er vanskelig å lage scriptene intelligente nok til å håndtere linjebrydd, endring i serverens mappestruktur og enda viktigere skjule hele prosessen for deltakerne. Produserende medlemmer er ofte ikke teknologer og har ikke behov for å følge opplastingsprosessen. De vil fortelle hvor filen ligger, og deretter vite at den blir flyttet til rett sted i rett format. Ideelt sett ønsker Åpen kanal et felles grensesnitt for opplasting, registrering og publisering. Web-baserte grensesnitt var et naturlig valg for registrering av metadata ettersom dette kan realiseres med PHP og enkel kommunikasjon mot en MySQL database. Dette er åpne standarder som ikke begrenser seg til operativsystem og Web-baserte systemer forutsetter heller ingen installasjon. Det var derfor ønskelig å implementere opplastingsklienten i et Web-grensesnitt.



**Figur 13-1: Filhåndteringen og andre bakenforliggende prosesser skjules for brukeren**

Systemet som blir utviklet i denne oppgaven består av to enheter; en server- og en klientapplikasjon.

Serverapplikasjonen er tenkt å fungere som en FTP-server med utvidede muligheter. Brukerdatabasen bør være ekstern og felles med den Åpen kanal eller bruker for håndtering av deltakerne og metadata for deres innslag. Vanlig FTP bør støttes, med mulighet for å utvide for flere muligheter ved forespørsel fra en klient. Slik kan det sikres at brukere uavhengig av system kan laste opp filer til systemet, men at forhold ved opplastingen blir bedre ivare tatt dersom de utvidede mulighetene blir benyttet eller er støttet på klientsiden. Det at en klient må be om utvidede muligheter sikrer kompatibilitet med eksisterende FTP-klienter.

Av utvidede muligheter er det tenkt at FTP-serveren med jevne mellomrom forteller databasen status på filer som er under opplasting. Slik kan en bruker ved hjelp av et Web-system kunne se hvor i opplastingen systemet befinner seg samt legge inn informasjon om filen allerede mens den er på vei inn i systemet. Systemet skal håndtere feilsituasjoner automatisk og kjøre kontroll på at filen er korrekt lastet opp både underveis og når den er

ferdig opplastet. Ved brudd skal systemet automatisk finne igjen hvor opplastingen ble brutt. Alt dette skal skje mens brukerne av systemet er lykkelig uvitende om hva som skjer i bakgrunnen.

Klientprogrammet er tenkt startet når systemet starter og usynlig helt til en bruker som er pålogget websystemet klikker på en link om at en fil ønskes lastet opp. Klientprogrammet kan senere utvides til å svare på andre kommandoer som blir gitt av en bruker som klikker på linker på en nettside.

Det eneste som skal være synlig for brukeren er filvalgsmenyen. Filhåndteringen utføres av samme klientapplikasjonen, men denne delen må startes som en tjeneste. Kommunikasjonen mot tjener er dermed i gang fra når brukeren logger seg på PC-en. Brukeren må med det ikke forholde seg til oppkobling og filhåndtering i systemet.

Vi har fra før av en ferdig brukerdatabase som det er ønskelig å tilknytte filtjeneren. Tjenersystemet bli i så måte en slags mellommann som tar i mot filer fra en klientmaskin, sjekker opp mot en database om klienten er en gyldig bruker og hvor den mottatte filen skal lagres. Filen får sin egen plass i databasen med instansinformasjon som størrelse, status og hvor i opplastingssituasjonen den befinner seg. Når filen får status som ferdig opplastet tar autokomprimeringssystemet over videre behandling av filen.

### 13.1.1 Begrensninger i systemet

Prototype-utgaven av både tjenersystemet og klientsystemet skrives i første omgang for Microsoft sin serie av operativsystem. Det er ingen spesifikk grunn for dette annet enn at flere av støttesystemene rundt vil kjøre på denne typen plattform. Et videre prosjekt vil kanskje være å lage klienter for andre plattformer også.

Klientsystemet krever administrasjonsrettigheter under installasjon. Dette er på grunn av at registerendringer må til for å få en nettleser til å eksekvere gitte programmer på lokal maskin når en bruker klikker på en link. Etter installasjon skal alt fungere uten at en bruker skal merke det.

### 13.1.2 Prototypen og dens videre utvikling

Det endelige systemet ble en proprietær løsning for å demonstrere det nye ved konseptet vi har beskrevet over, nemlig opplasting av filer over web ved hjelp av støtteprogrammer.

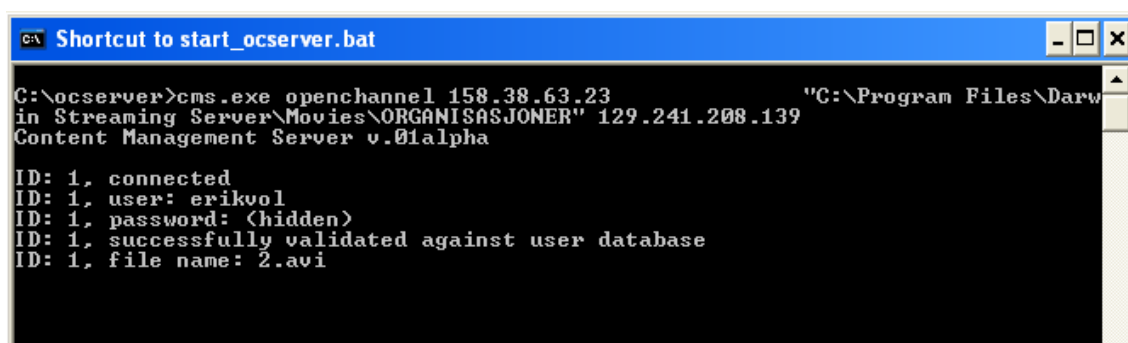
Serveren lar en bruker logge seg på, valideres mot databasen, laster opp en fil som blir lagt på rett sted, men uten feilkontroll, FTP-støtte og automatisk feilhåndtering som beskrevet over. Systemet fungerer etter oppskrift, men uten feilhåndtering er denne versjonen av systemet uegnet til noe annet enn demonstrasjon av et nytt konsept. Den videre utviklingen vil bestå i å analysere pakkene som har blitt indeksert for å sørge for korrekt retransmittering etter et eventuelt linjebrydd. Dette er muligheter som enkelt lar seg realisere med et program implementert etter vår modell.

Ettersom koden er skrevet i C vil det også være mulig å porte koden til andre operativsystem, men dette forutsetter visse endringer blant annet i registry-håndteringen som er spesifikk for windows. Noe det er vanskeligere å gjøre noe med er installasjonen. Registry-endringene forutsetter at man har administratortilgang. Det må settes visse parametere i registeret for at klienten skal kjøre som en tjeneste (starter opp i bakgrunnen når brukeren logger seg på). Det er også nødvendig med registry-endringer for å definere at OCHAN-linken på nettsiden er et kall til klientapplikasjonen. Andre gang klienten kalles åpnes et browser-vindu der deltakeren kan spesifisere hvor filen man vil laste opp ligger. Ettersom applikasjon allerede kjører i bakgrunnen vil dette skje når vedkommende trykker på OCHAN-linken på nettsiden. Denne funksjonen gjør at det ser ut som en rent Web-basert applikasjon der prosessen skjules for brukeren, som var målet med implementeringen. Men det at det forutsettes en installasjon i forkant som må utføres av en med administratortilgang kan være et problem. Det er derfor ønskelig å videreutvikle prototypen til å håndtere FTP slik at man kan få lastet opp filen med programmer som er installert. I mellomtiden håndterer systemet FTP ved hjelp av en chron jobb som jevnlig sjekker om det har blitt lastet opp noe nytt til det sentrale områdets FTP-server. Dersom dette er tilfellet oppdateres databasen på samme måte som med opplastingstjeneren.

### 13.1.3 Demonstrasjon for faglærer

Faglæreren, prof. Leif Arne Rønningen, var til stede ved en gjennomkjøring av applikasjonen. Tjenerapplikasjonen ble startet opp på serveren med remote desktop (Denne vil vanligvis kjøre konstant og trenger ikke startes opp igjen med mindre serveren har gått ned). Klienten

som har fått navnet "Silent helper" ble kjørt fra en lap-top over NTNU sitt trådløse nett. Den var ferdig installert i forkant av demonstrasjonen og kjørte som en tjeneste i bakgrunnen. Prosessen "silenthelper.exe" ble stoppet i task manager og startet opp igjen ved å trykke på exe-filen. Dette for å vise at det var en forskjell på første og annengangs-eksekvering. Andre gang den ble trykket dukket det opp et browservindu der vi valgte en videofil med 1 sekund varighet (5MB) som meget raskt ble lastet opp til tjeneren. Via remote desktop fikk vi se kvitteringen på tjenerapplikasjonen og kunne se at filen la seg under riktig organisasjonskatalog. Det sistnevnte sikres ved at organisasjonen under installasjon av klientapplikasjonen må angi brukernavn og passord som de benytter i Åpen kanal sitt Web-system. Dette identifiserer organisasjonen unikt.



Figur 13-2: Tilbakemelding fra tjenerapplikasjonen



Figur 13-3: Opplastingsklienten "Silent helper" som startes fra en link på Åpen kanal sin nettside



I andre runde ble det demonstrert hvordan systemet fungerer for en vanlig bruker med opplasting, komprimering og publisering. Alt via det Web-baserte grensesnittet. Lap-topen ble logget på som vanlig (Silent helper starter automatisk i bakgrunnen). Logget meg på som organisasjonen registrert på mitt navn (OpenChannelNo) på internsiden til openchannel.no.

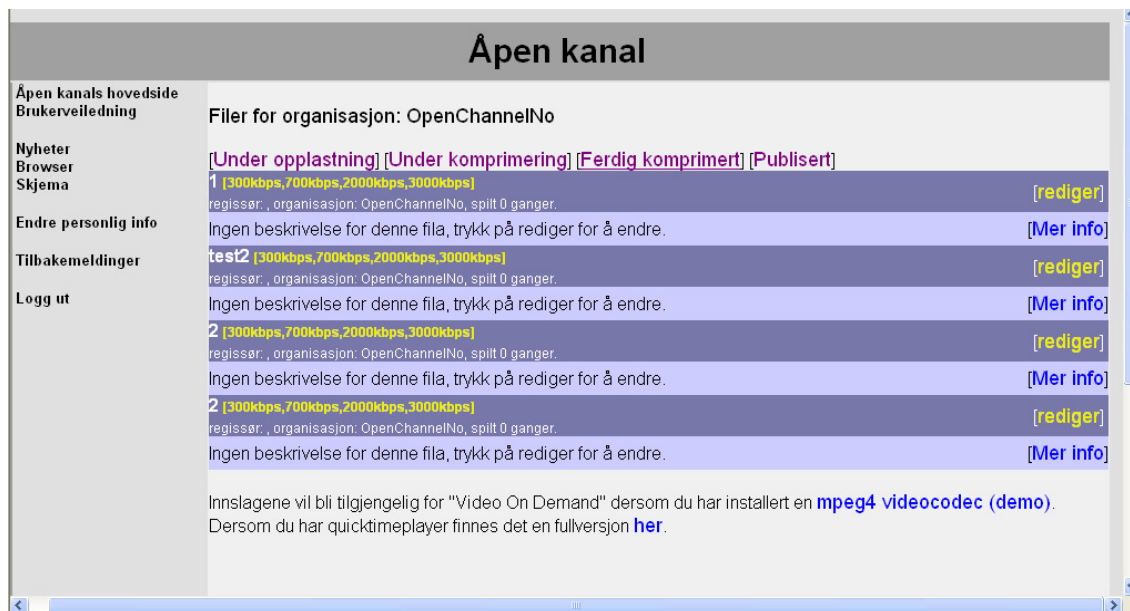
På menyen til venstre finnes en link til en browser. Den viser hva organisasjonen har av videomateriale og i hvilken status den er (Under opplasting, under komprimering, ferdig komprimert og publisert).

Her er det lagt ut en OCHAN-link (kaller på Silent helper) der man kan velge filen man vil laste opp. Dette fungerte utmerket selv med en større fil på 23 sekunder (85MB). Opplastingen gikk noe tregt over det trådløse nettet men linken var stabil. I mellomtiden var det mulig å registrere informasjon om innslaget (link til filen er registrert i DB selv om filen ikke er endelig lastet opp) som hvilken kategori den tilhører, tittel og hvor den er mastret (Tape nr. og tidskode, kjekt å vite hvis systemet går ned). Tono relatert informasjon kan også registreres for flere rettighetsbelagte musikksnutter som er benyttet i innslaget.

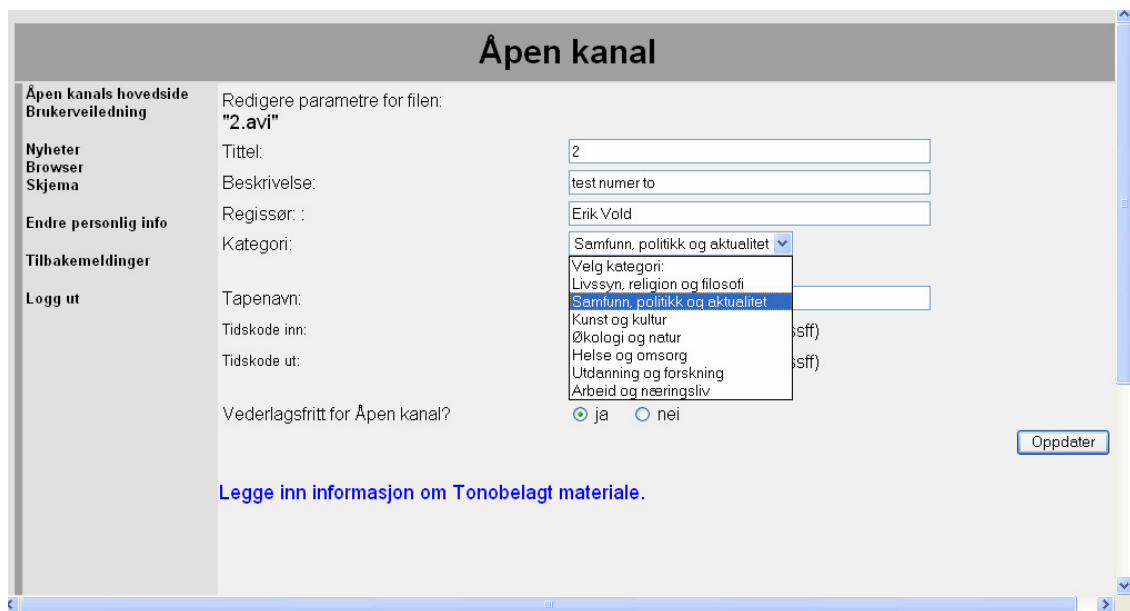
Før registreringen var ferdig hadde filen fått status som "under komprimering". Dette er en prosess som tar lang tid og jeg ville utnytte tiden til å vise faglærer hva som skjedde på komprimeringsenheten via remote desktop. Her fulgte vi prosessen og så at pascalkoden automatisk genererte script for ulike komprimeringsjobber. Vi så på videofilene som var komprimert med ulike bitrater direkte på serveren. Da jeg ved en feiltagelse aksesserte en av filene som var under komprimering gikk serveren ned. Dette skyldes en adgangskonflikt i operativsystemet.

Denne feilen kan ikke forekomme for vanlige brukere ettersom de ikke har tilgang direkte til filene via remote desktop, men kun gjennom streamingsserveren som nektes tilgang under komprimering (streaminglinken legges først ut på nettsiden når komprimeringen er fullført). Etter omstart av systemet måtte innslaget slettes i databasen for at prosessen skulle fortsette. Dette viser likevel at det er dårlig feilhåndtering i systemet. Det bør være mulig for scriptet å hente seg inn igjen der prosessen stoppet ut ifra tilstandsflaggene som er satt i databasen. Pascal koden jobber ut ifra forutsetningen om at komprimeringsprosessen fullføres og håndterer derfor ikke et brudd i nødvendige ressurser. Feilhåndtering er ikke prioritert i denne pilotutviklingen ettersom det er et konsept man vil demonstrere. Dette bør det likevel jobbes videre med mot en lansering av tjenesten for prøvedrift.

Ved videre utprøving av systemet forholdt vi oss til Web-grensesnittet. I browseren på internsiden kunne vi se at filen endret status til ”ferdig komprimert”. Vi så på innslaget i ulike bitrater via streaming-linkene som ble lagt ut i browseren og valgte deretter å publisere det. Når det var publisert ble innslaget også tilgjengelig i VOD-videoarkivet på Åpen kanal sin offentlige nettside, listet under kategorien som ble valgt.



Figur 13-4: Intern browser med streaminglinker og statusoversikt



Figur 13-5: Publisering av innslaget under en gitt kategori



Figur 13-6: Streaming på Åpen kanal sin offentlige nettside

Komprimeringen ble gjennomført av det nye sgen-scriptet som henter informasjon om de ulike enhetene fra en tekstfil. Det er enklere å endre på IP-adresser, brukernavn og passord i en tekstfil enn i flere pascalkodede programmer som må kompileres. Forflytning av hardware som medfører endringer i slike parametere må man påregne i driftsmiljøet til Åpen kanal der man baserer seg på lånt utstyr. Denne delen ble implementert når vi forsto at systemet måtte bygges opp igjen på ny hardware. Vi ønsket ikke en tilsvarende arbeidsmengde neste gang deler av systemet går ned (Sgen scriptet er vedlagt i kildekoden).

Demonstrasjonen var vellykket og fikk belyst brukervennligheten så vel som forbedringspotensialer i systemet. Opplastingsklienten og tilhørende tjenerapplikasjon fungerte som forventet.

### 13.2 Avvikling av TV-sending

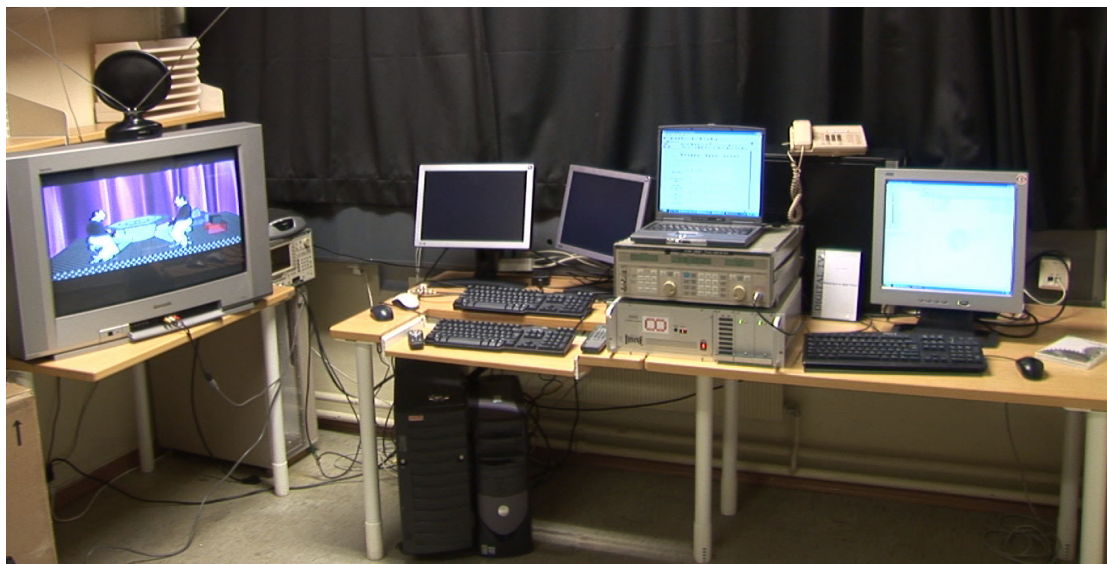
Åpen kanal har en reservert plass på det nye digitale bakkenettet. Sendestart vil sannsynligvis drøye til 2009, men NRK har allerede begynt regionale prøvesendinger. Det er ønskelig for Åpen kanal å være med på utprøvingen for å kunne høste tekniske så vel som administrative erfaringer. Dette er nødvendig for at konseptet skal fungere når kanalen forhåpentligvis når et større publikum i 2009. En public access modell som Åpen kanal er bygget etter har hittil uprøvde utfordringer med tanke på nettbasert avvikling. Spørsmål som ønskes besvart i

prøveperioden er bla hvordan man kan håndtere sendetidsfordeling blant et stadig økende antall redaktør, hvordan man kan gjøre avviklingssystemet brukervennlig og hvilke komprimeringsalgoritmer som gir best kvalitet for seeren.

### 13.2.1 Testoppsett for avvikling av TV-sending

Åpen kanal har gjennom denne masteroppgaven fått tilgang på utstyr som stilles til disposisjon av NTNU og SINTEF. NTNU investerte i en Playout-enhet i forbindelse med Matti Pedersen (tidligere siv.ing student) sin prosjektoppgave i 2004. På grunn av en defekt set-top boks fikk han ikke demonstrert systemet i sin helhet men dokumenterte installasjonsprosedyrene og verifiserte at avviklingen fungerte med en spektrumanalysator.

Ettersom systemet ble hacket og måtte reinstallerer kom denne dokumentasjonen til stor nytte.

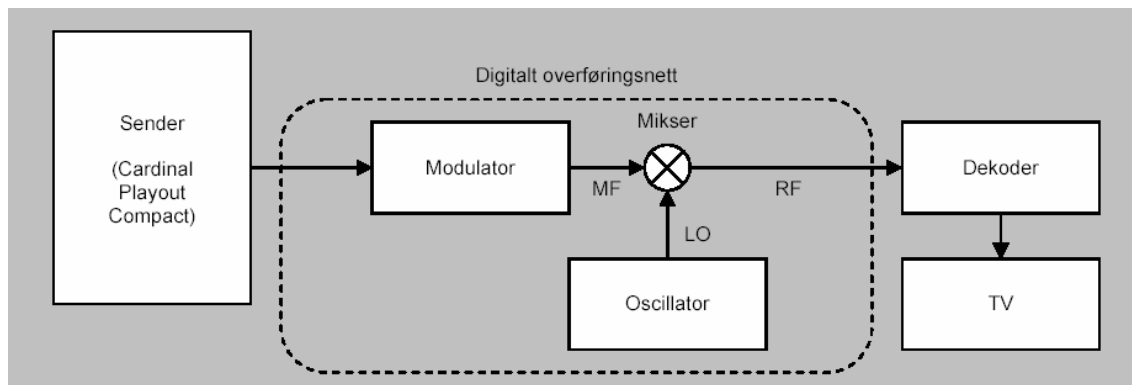


**Figur 13-7: Åpen kanal på et bord**

Testoppsettet består av en playout-enhet fra Cardinal (Cardinal Playout Compact 1.3), en prototyp COFDM modulator fra SINTEF (Divine HD EM-30), en mikser (MCL SRA-A1), en oscillator (Anritsu MG3602A), en spektrumanalysator (Anritsu MS2601A), en set-top boks (Nokia Mediamaster) og en TV (Philips 100Hz). Det ble også koblet til en PC som styrer avviklingsenheten med en konsoll-applikasjon fra Cardinal. Vanligvis vil denne styringen skje i et åpent IP-nett, men med sikkerhetshullene som fortsatt eksisterer valgte vi heller å ha en direkte kobling mellom server og PC (med en kryssset TP-kabel). Dersom Åpen kanal

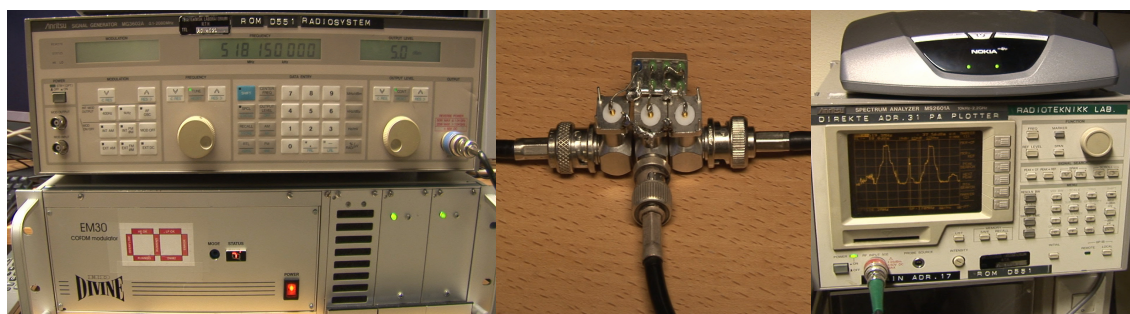
deltakere ønsker å prøve ut avviklingen over nett må de aksessere PC-en med playout-konsollen via remote deskstop (sendingen grabbes fra TV-utgangen og streames på nett).

### 13.2.2 Virkemåte



Figur 13-8: Blokkjema for TV-systemet (Pedersen, 2003)

TV-sendingen ble ikke utført over luft under utprøvingen. Dette fordi Åpen kanal ikke har konsesjon til å bruke det nødvendige frekvensbåndet. Det ble derfor ikke koblet inn noen antenneforsterker. Antennesignalet ble ført direkte inn i set-top boksen med en coax-kabel. For å generere antennesignalet måtte jeg lodde bnc-terminaler til en frekvensmikser som blander nyttesignalet og bærebølgen på 518,15MHz. Bærebølgen ble generert av oscillatoren. Nyttesignalet kommer fra COFDM-modulatoren som igjen blir foret med en DVB-ASI strøm fra playout enheten. DVB-ASI strømmen er innholdet i TV sendingen (MPEG2TS) som genereres i playout-enheten og sendes ut på et ASI-kort som fulgte med Cardinal PC-en.



Figur 13-9: Oscillator / COFDM-mod.(venstre), mikser (midten), set-top-boks / analysator (høyre)

Bildene over viser de nødvendige komponentene for å realisere en TV-sending etter standarden som vil bli tatt i bruk på digitalt jordbundet nett (DVB-T). Av

spektrumanalysatoren kan vi se at det er en markant amplitudeøkning ved frekvensen 518,15 MHz. Dette er bærebølgen til signalet som sendes. Informasjonen i TV-sendingen er i de to frekvensbåndene som ligger symmetrisk rundt denne. Senterfrekvensen på det nedre frekvensbåndet er det 482 MHz. Det er ved denne frekvensen set-top boksen finner samtlige kanaler når vi kjører automatisk kanalsøking. Informasjonen som ligger kodet i dette båndet er generert av COFDM-modulatoren ut ifra DVB-ASI strømmen. Den benytter en kodingsalgoritme som er robust mot interferens og støyproblematikk som oppstår ved overføring i luft. DVB-ASI strømmen inneholder en MPEG 2 transport strøm med flere elementære strømmer (ES) som i praksis er TV-kanalene vi avvikler. Den øvre grensen for antall kanaler man kan sende i denne strømmen er gitt av bitraten som COFDM-modulatoren er satt til (Denne må være eksakt den samme som fra Playout enheten).

Dette gjør det mulig å sende flere digitale TV-kanaler for hver analoge TV-kanal hvilket er kostnadsbesparende for distributør. NTV vil i denne sammenheng måtte vurdere hvilke bitrater som gir tilfredsstillende kvalitet på hver kanal. Høye bitrater vil gi god kvalitet men begrense antall kanaler og dermed inntjeningspotensialet i bakkenettet. I forsøket så den nedre grensen for hver kanal ut til å ligge på 3,5 Mbit pr sekund med vår MPEG2TS kodingsalgoritme (VLC). Med denne bitraten kunne vi avvikle 5 TV-kanaler samtidig i det gitte frekvensbåndet.

### 13.2.3 Utfordringer i testoppsettet

Installasjonen av Cardinal sin playout-enhet er i utgangspunktet enkel å få til, men i forsøk på å tette sikkerhetshullene støtet vi på en del problemer. Til dette fikk jeg hjelp av en driftsingeniør på NTNU, Pål Sæther. Vi hentet ned den nyeste versjon av operativsystemet "Head End Linux" fra Cardinal sin nettside. Etter installasjonen forsøkte vi å legge til komponentene som utgjør Playout applikasjonen. Dette fungerte ikke, i følge Cardinal for de operativsystemet manglet muxer-drivere. Ettersom NTNU ikke lenger har en support lisens som inkluderer oppgradering, måtte vi gå tilbake til den opprinnelige installasjonen.

Denne måtte ha en lisensfil som leses fra diskettstasjonen. Disketten var ikke å spore opp så vi kontaktet Cardinal for å få tak i innholdet av den. Filen fikk vi ikke tilsendt fra Cardinal, men Pål Sæther hadde den heldigvis liggende fra tidligere mailkorrespondanse med Matti Pedersen. Totalt sett ble installasjonen en mer tidkrevende prosess enn forventet.

Installasjons CD-en inneholdt en nyere versjon av Playout applikasjonen enn det Matti Pedersen benyttet, men et eldre operativsystem som ikke er sikkert. For å kunne bruke systemet uten fare for ytterligere hackerangrep ble ikke serveren koblet på nett, men direkte til

PC-en med kontroller-konsollen (krysset TP kabel). Dette gjør at deltakerne i Åpen kanal ikke kan styre playout fra sin lokale PC med en nedlastet Cardinal-konsoll, men må aksessere konsollen på den sentrale PC-en via remote desktop. Dette er likevel hensiktsmessig ettersom PC-en også er den sentrale filserveren med det originale videomaterialet. Dette materialet må komprimeres til MPEG2TS før det kan lastes opp til playout-serveren via Cardinal Konsollen. Komprimeringen skulle ideelt sett vært automatisert og startet av opplastingstjeneren når filene er ferdig lastet opp. Foreløpig er denne funksjonen realisert med VLC-player. VLC eksporterer filer som blir sluppet over ikonet for et bat-script. Dette ”drag-and-drop” scriptet eksporterer til det rette formatet og flytter filen inn i en MPEG2TS katalog. Her kan den aksessere med Cardinal playout konsollen. Ulempen med denne løsningen er at den forutsetter en manuell prosess med å slippe videofilen over scriptikonet. Hadde komprimeringen vært automatisert ville deltakerne ikke behøvd å være inne på systemet før etter at komprimeringen er utført. Det ble ikke prioritert å implementere i tjenerapplikasjonen, men med utvidet kommunikasjon mot databasen kan automatiseringen enkelt realiseres.

#### 13.2.4 Testoppsett for sammenligning av videoformater

Komprimeringsscriptet bruker MPEG2 som videocodec. Lyd og bilde blir multiplekset inn i en MPEG2TS fil. Det er også mulig å bruke en mer biteffektiv videocodec, H264, som også kan multiplekset inn i en MPEG2TS-fil. Det ble laget et drag-and-drop-script også for denne codecen, med tilsvarende bitrate. Playout serveren avvirket den uten feilmeldinger, men kvalitetsforbedingen kunne ikke verifiseres ettersom Nokia sin set-top boks ikke har støtte for visning av H264-video.

Komprimering til video kodet med MPEG2 over en MPEG2TS (transport strøm) ble realisert med en bat-fil med følgende commandolinje parametere til programmet VLC:

```
vlc -vvv %1
:sout=#transcode{vcodec=mp2v,vb=3500,scale=1,acodec=mpga,ab=192,channels=2}
:duplicate{dst=std{access=file,mux=ts,url=%1.ts}} vlc:quit
```

%1 er filnavnet til videoen som blir sluppet over scriptikonet. Skriptet som er brukt håndterer også overføring til FTP-server, kopiering til redundante streamingservere og opprydding av temporære filer. Skriptene i sin helhet er vedlagt med sensurerte FTP-passord.

Ved å erstatte ”vcodec” parameteren (video codec) med ”h264” og ”acodec” parameteren (audio codec) med ”aac” er det mulig å oppnå tilfredsstillende kvalitet ved 1 Mbit pr sekund (vb=1000) som vi fikk se i demonstrasjonen. Dette er en mer biteffektiv komprimeringsalgoritme enn MPEG2 og forventes å bli tatt i bruk i bakkenettet spesielt i

forbindelse med HD-sendinger. HD-sendinger kan med h264 realiseres uten å legge beslag på vesentlig mer kapasitet enn det SD-sendingene gjør i dag (Poynton, 2003).

Kvaliteten på h264-video kunne ikke verifiseres på set-top boksen der denne codecen ikke er støttet. Videoen ble derfor lastet opp til en quicktime streaming server, men for at den kunne spilles av derfra over nett må den ”hintes”. Dette utføres av et program som heter mp4creator med følgende commandolinje parametere i scriptet:

```
mp4creator -list %1.mp4
mp4creator -hint=1 %1.mp4
mp4creator -hint=2 %1.mp4
```

Her finner programmet tracks i MP4-strømmen (”mux”-parameteren i VLC scriptet måtte settes til ”mp4”) og hinter de som har tracknummer 1 og 2 (hhv lyd og bilde).

H264 codecen ble i demonstrasjonen sammenlignet med Windows Media 9 som er en konkurrerende komprimeringsteknologi også relevant for bakkenettet. Også denne videoen ble realisert med et komprimeringscript der Windows Media Encoder selv sørger for hinting:

```
CScript.exe wmcmd.vbs -input %1 -output %1.wmv -a_mode 1 -a_setting
128_48_2 -a_codec WMA9STD -v_preproc 1 -v_mode 1 -v_framerate 25 -v_bitrate
1000000 -v_codec WMV9
```

Videoen som skal streames på nett kan ikke benytte samme visningsmetode som i TV-distribusjonen. På TV-nettet i dag veksles det mellom odde og partallsnummererte horisontale linjer i videobildet 50 ganger pr sekund. Dette gir jevnere bevegelser (en høyere tidsoppløsning) i videostrømmen enn om man skulle vist 25 helbilder pr sekund slik man gjør på en PC-skjerm (progressivt). Teknikken kalles interlacing og fører kun til en subjektiv romlig oppløsningsreduksjon på 30% til tross for at datamengden er redusert med 50% (Hjernen interpolerer resulterende helbilder). Dette fungerer godt på TV-skjermer som viser linjene vekselvis, men dersom alle linjene vises samtidig 25 ganger pr sekund, som på en PC-skjerm, vil det bli synlige striper i bildet der man har raske bevegelser. For å gjøre interlaced video anvendelig i streaming må derfor videoen ”de-interlaces”. Dette er en prosess som tilsvarende hjernen utfører når man ser en interlaced videostrøm. Algoritmen interpolerer en frame fra de to delbildene (fields) ved å finne gjennomsnittlig plassering i henhold til nærliggende piksler. I Windows Media Encoder-scriptet utføres de-interlacing når man setter parameteren v\_preproc lik 1 (video pre processing). I VLC må det skrives inn følgende globale parameter i slutten av scriptet: --sout-transcode-deinterlace.



Disse komprimeringscriptene kunne vært skrevet inn i allerede eksisterende komprimeringscript som er automatisert på en egen maskin (Envivo MPEG4). Kommandoene over ville i så fall måtte genereres av pascal koden "sgen" der filnavnet (her %1) hentes fra databasen. Med det ville man kunne fulgt statusen på komprimeringen via browser som forklart i tidligere avsnitt. Denne endringer innebærer kun noen få kodelinjer, men det er likevel ikke ønskelig å implementere det av hensyn til redundans. Envivo komprimeringsenheten vi har automatisert er en lånt maskin. Den kan bli brukt til andre vitenskapelige formål av folk ved Q2S senteret på NTNU. Vi ønsker derfor på sikt å lage en komprimeringsprosess som kan kjøre parallelt på Åpen kanal sin egen server i tilfellet Envivo-enheten ikke lenger er tilgjengelig. Prosessen kan startes av opplastingstjeneren slik Envivo enheten blir i dag.

### 13.2.5 Demonstrasjon for faglærer

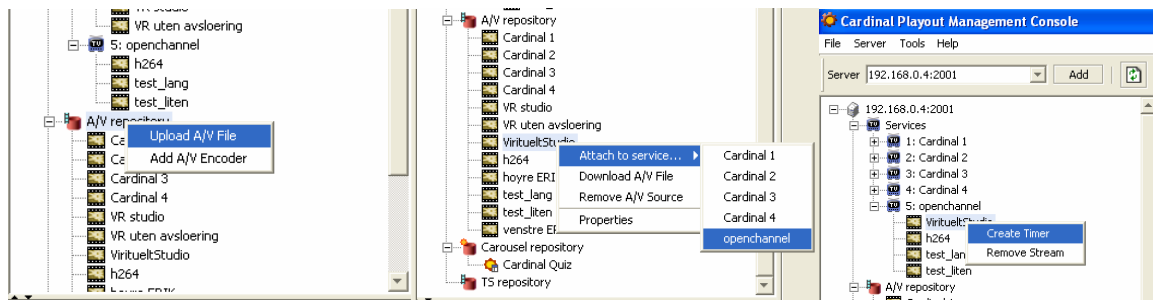
Prof. Leif Arne Rønningen, min faglærer, var til stede under den første prøvesendingen. Her ble innholdet som var lastet opp med den skreddersydde klienten i forrige demonstrasjon komprimert til MPEG2TS med "drag-and-drop" scriptet. Det ble også demonstrert komprimering til andre formater som H264 og Windows Media 9. For å vurdere innholdet av disse ble filene lastet opp til henholdsvis en quicktime streaming server og en Windows Media streamingsserver. De ble aksessert via en link som på forhånd var lagt ut på Åpen kanal sin nettside. Denne prosessen er kun halvautomatisert på tilsvarende måte som de andre "drag-and-drop"-scriptene. Hensikten var å vurdere kvalitetsforskjellen på ulike codec'er ved 1 Mbit per sekund. Deretter ble innholdet sammenlignet med en MPEG2 strøm i tilsvarende bitrate, og kvaliteten var langt dårligere på denne enn de to andre. Forskjellen mellom H264 og Windows Media 9 var mindre. WM9 var lett å spille av jevnt men hadde dårlig kontrast i forhold til H264 og et litt for soft bilde ("softing" er en teknikk som brukes for å skjule "blocking"-feil i komprimeringen). H264 var vanskelig å spille av jevnt (mer komplisert dekodingsalgoritme) men detaljene, spesielt i mørke partier kom bedre frem enn i WM9. Begge de sistnevnte formatene er relevante i det bakkebunnede nettet ettersom det med disse er mulig å utvikle en TV kanal med tilfredsstillende kvalitet mellom 1 og 2 Mbit pr sekund. Dette er svært plassbesparende mhp bakkenettets kapaistet og kan gi rom for flere kanaler og større inntjening for NTV. Microsoft sin Windows media 9 advanced codec har i denne sammenheng fått navnet VC1 etter en SMPTE-standardisering. Microsoft jobber hardt for å komme inn i broadcast med sine DRM løsninger og Microsoft media center som gir god rettighetshåndtering og fleksible visningsmuligheter hos mottakeren. Dette er en trussel for

MPEG4 og MHP standardiserte systemer. MHP er videre truet av lisensproblemer som påfører distributør kostnader som i dag ikke kan dekkes inn i følge Leif Arne Rønningen (Fra samtale med teknisk sjef i NRK).

Etter en diskusjon rundt de ulike formatene gikk demonstrasjonen videre til å prøve programmering av sendeplan med det formatet som er støttet av set-top-boksen, MPEG2.

Service	Active	Running	User	Stream name	Type	Start	Stop
1			demo	Cardinal Quiz	Carousel	04.04.2006 11:24:58	05.04.2006 11:24:58
1		⊕	erikvol	hoyre ERIK	A/V	06.04.2006 15:59:50	07.04.2006 15:59:50
2		⊕	erikvol	venstre ERIK	A/V	06.04.2006 15:59:56	07.04.2006 15:59:56
3		⊕	erikvol	VirtueltStudio	A/V	06.04.2006 16:00:01	07.04.2006 16:00:01
4		⊕	erikvol	VR studio	A/V	06.04.2006 16:00:08	07.04.2006 16:00:08
5		⊕	erikvol	test_liten	A/V	05.04.2006 13:55:31	05.04.2006 13:56:30
5		⊕	erikvol	h264	A/V	05.04.2006 18:42:48	06.04.2006 18:42:48
5		⊕	erikvol	test_lang	A/V	06.04.2006 14:53:18	07.04.2006 14:53:18
5	✓	⊗	erikvol	test_lang	A/V	06.04.2006 16:00:18	07.04.2006 16:00:18
5	✓	⊗	erikvol	VirtueltStudio	A/V	07.04.2006 21:12:27	08.04.2006 21:12:27

Figur 13-10: Schedulert sending med timere



Figur 13-11: Opplasting av video (venstre), tilknytning til kanal (midten), aktivering av timer (høyre)

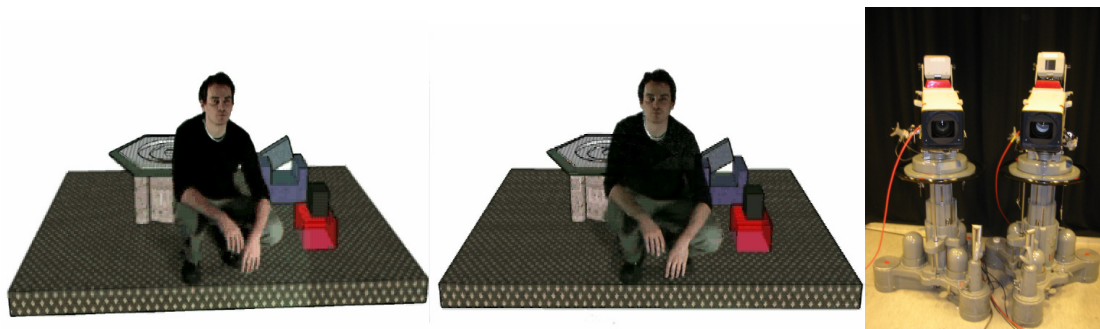
For at innholdet skal bli tilgjengelig på avviklingsserveren må det lastes opp med Cardinal konsollen på den tilkoblede PC-en. Dette gjør man ved å høyreklikke på AV-repository og velge upload A/V-file. Deretter velges filen i et browservindu og man gir den et navn som ikke behøver være det samme som filnavnet. Disse dataene lagres i avviklingsserverens database sammen med lokasjonen til filen som blir lastet opp (Dersom filen er av typen Program Strøm (PS) vil den bli konvertert til Transport Strøm (TS). I vårt tilfelle var filen en MPEG2TS som ikke måtte konverteres). Når filen dukker opp i AV-repository må man velge hvilken kanal den skal sendes på. Kanaler i Cardinal-systemet kalles for "service". Valget utføres ved å høyreklikke på filen og velge "attach to service". I demonstrasjonen opprettet vi først en ny kanal ("openchannel") ved å høyreklikke på serviceikonet og velge "add service".

Etter at denne kanalen var valgt for videofilen dukket innholdet opp under vår kanal med navnet "openchannel".

For å bestemme når denne skal sendes må man høyreklikke på videofilen under openchannel og velge "add timer". Her setter man dato og klokkeslett når videoen skal begynne og slutte.

I dette tidsintervallet går videofilen i loop. Dersom tidsintervallet er overlappende med det som allerede ligger i sendeplanen vil det siste innslaget bli kansellert og få en rød hake på timer symbolet (klokka). Resultatet av sendetidsprogrammeringen kunne vi først se etter at vi hadde søkt inn kanalene på ny. Da dukket kanalen openchannel opp på set-top boksen med vårt innhold. Det ble i etterkant lastet opp innhold til flere kanaler på serveren inntil den øvre grensen for maksimal samlet bitrate var nådd. Denne bitraten ble i konsollen satt til det samme som COFDM-modulatoren (19905882 bits pr sekund). Når man passerte denne ble enkelte rammer kastet og avspillingen av videoen ujevn.

### 13.2.6 Stereoskopisk payout



**Figur 13-12: Venstre bilde (venstre), høyre bilde (midten) og stereoskopisk studiokamera (høyre)**

Andre strømmen som ble testet ut var fra de distribuerte samhandlings prøvene i Lillehammer. Det var en virtuell debatt med meg selv, en debatt mellom NRK og HiL i et virtuelt rom og en stereoskopisk sending der den virtuelle bakgrunnen er rotert ulikt i henhold til de to kameraposisjonene. Den sistnevnte er interessant fordi den er en demonstrasjon på at en 3D-sending med dette systemet kan realiseres. Her har jeg lagt inn et bilde fra et kamera som representerer venstre øye i den ene kanalen og høyre øye i den andre kanalen. Avviklingen på disse ble satt med en identisk timer slik at 3D-visningen kunne bli synkron. Dessverre hadde vi ikke to set-top-bokser så jeg fikk ikke testet stereoskopisk visning. Men ved å koble set-top-boksene til hver sin motsatt polariserte projektor ville man med pola-briller kunne sett et 3D-bilde avviklet over et TV-nett med bedre kvalitet en det man får til på en kanal. Over en kanal benyttes anaglyph (dårlig fargegjengivelse og 3D-separasjon) eller field sekvensiell 3D-

teknikk (dårlig romlig oppløsning). Disse teknikkene er nærmere forklart i etterfølgende kapitler. Den 3-dimensjonale sendingen kunne vært realisert i sanntid dersom vi hadde hatt to grabberkort. Livekilder inkluderes ved å høyreklikke på AV-repository og velge "add AV-encoder". Det stereoskopiske materialet som er lagt inn ble realisert i sanntid. Opptaket består i en virtuell scene der jeg er keyet inn over en virtuell bakgrunn som er rotert ulikt i henhold til de ulike kameravinklene på venstre og høyre bilde (En ulik bakgrunn er valgt for de to keyerene som ble kjørt ut på hhv preview og program utgangene på mikseren). Dette sørger for at både jeg og den virtuelle bakgrunnen blir 3-Dimensjonal. Den virtuelle bakgrunnen er laget i Scetch up der man kan velge kameravinkel fra 3D-modellen før de to 2-dimensjoanale bildene rendres ut. Den stereoskopiske avviklingen ble vist faglærer men ikke med 3D-visning. Dette er en mindre viktig del av oppgaven og det ble dermed ikke prioritert å kjøpe inn utstyr. Prinsippene viser likevel at det er mulig å få til en høykvalitets 3D-sending for åpen kanal dersom de kan reservere ytterligere en kanal i sendenet. Dette kan være interessant i en samkjøring med andre kanaler som uansett kan ha glede av den ene halvdel som fremstår som en 2D-sending for seere med kun en set-top-boks. Løsningen er likevel ikke å forvente å bli tatt i bruk. Leif Arne påpekte at det må finnes mer brukervennlige visningssystemer som ikke forutsetter rigging og vanskelig tilgjengelig utstyr. Med autostereoskopiske skjermer er det nå en lovende utvikling mhp kvalitet. Med disse trenger man ikke briller ettersom de to bildene blir projisert direkte til hvert sitt øye. Dette sammen med en stereoskopisk player som kan ta in parallelle strømmen og vise det riktige formatet (kolonne-interlacing for autostereoskopiske skjermer) kan ha en fremtid. Leif Arne påpekte også at Playattion 3 har 2 skjermutganger og prosesseringskraft til å vise flere høyoppløselige strømmen. Dette vil bli en rimelig enhet som kan fungere som en stereoskopisk set-top-boks. Enda har vi likevel ikke funnet noe utviklingsinitiativ for PS3 rettet mot stereoskopi.

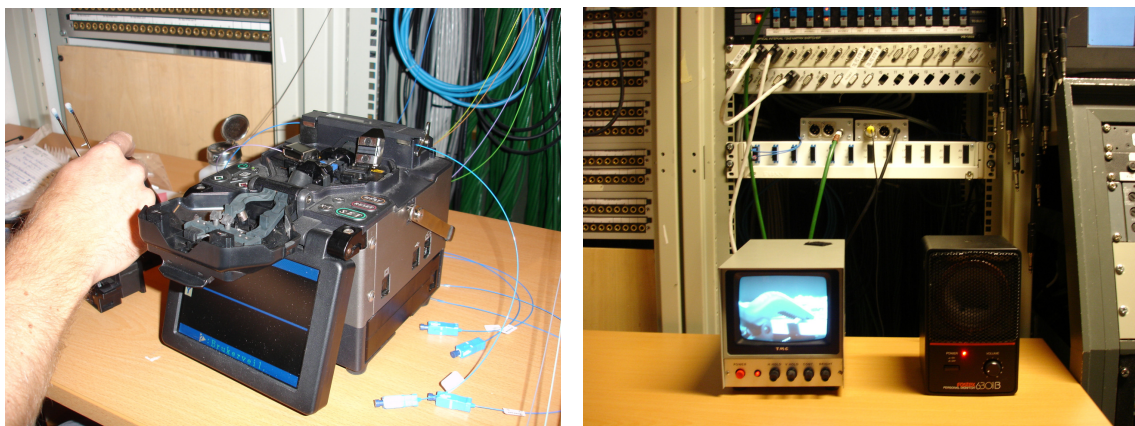
### 13.2.7 Utbedringer av avviklingsystemet

Avviklingssystemet er etter min vurdering ikke brukervennlig nok for Åpen kanal sine deltakere. Komprimering til rett format må startes manuelt og grensesnittet for programmering av sendeplan er unødvendig komplisert. Her er det mulig å gjøre fatale feil, blant annet ved å sette overlappende avviklingstidspunkt som medfører at det siste innslaget ikke blir avviklet. Det er ellers tilgang til å endre parametere som kan ødelegge for avviklingen av andres innslag. For Åpen kanal bør det utvikles et system med samme grensesnitt som det øvrige systemet. Med et Web-basert grensesnitt og kommunikasjon mot playout-serverens database kan nødvendig funksjonalitet realiseres. Dette muliggjør også en

mer intelligent håndtering av videoinnholdet ved programmering av sendeplan. Innslagene kan settes på sendeplanen som blokker med kjent lengde der overlapping ikke kan forekomme. Et mer grafisk grensesnitt vil være mer brukervennlig og gi en bedre oversikt over tilgjengelig tid på sendeplanen. Systemet kan også ekskludere brukeren fra muligheten til å ødelegge for andre deltakere. Man kan sette eierskap og med det adgangsnivå til de ulike video objektene slik det blir gjort i det Web-baserte publiseringsystemet som benyttes i Åpen kanal i dag.

## 14 Eksperimenter med distribuert samhandling v/HiL

Som student ved fjernsynsteknikk-studiet på Lillehammer har jeg fått tilgang på videoteknisk utstyr til eksperimentering. Skolen er på mange måter en NRK-skole der elevene blir opplært i utstyr som benyttes i profesjonelle miljøer både for enkamera og flerkameraproduksjoner. Studioet har skolen fått etter OL 94 og utstyr har i stor grad blitt gitt fra NRK. Enda heldigere er det at skolen har tilgang på ressurspersoner som besitter unik kunnskap samt har en evne til å sette dette ut i praksis. Allerede de første ukene etter opptaket kom jeg godt inn i fagmiljøet og fikk interessante kontakter også på Høgskolen i Gjøvik (HiG). Claus Knudsen ved HiG viste stor interesse for konseptet Åpen kanal og fagmiljøet ved NTNU. Vi diskuterte hvordan vi best kunne utnytte hverandres ressurser og kom frem til at jeg skulle delta i samarbeidsprosjektet TiDE-SVR (beskrevet nedenfor). Min oppgave var å rådføre transmisjonsgruppen i TiDE-SVR, observere den distribuerte samhandlingsproduksjonen og komme med forslag til eventuelle forbedringer. Med meg fikk jeg Sigmund Andresen ved Høgskolen i Lillehammer (HiL). Han har også bakgrunn fra miljøet ved NTNU, men jobber nå som lærer i fjernsynsteknikk ved HiG. Spesielt relevant var kunnskapene om komposisjon av objektbaserte videoscener fra fagene videoteknologi og multimedia signalbehandling samt erfaringene med 3D-videoproduksjon. I slutten av denne rapporten vil jeg gå nærmere inn på hvordan disse teknologiene kan nyttegjøres for å nærme seg prosjekts mål ytterligere. Teknologien har her blitt prøvd ut i miljøet det er tiltenkt, men med relativt få involverte. I November har vi som mål å implementere dette i et større prosjekt for vitenskaplig dokumentasjon av virkningene. Enn så lenge står mitt bidrag som en pilot og utredning for praktisk gjennomføring. Denne utredningen samt erfaringer gjort i TiDE-SVR regnes som svært relevante for Åpen kanal. Målet til kanalen har alltid vært å aktivisere seerne og inkludere de i samfunnsdebatten. Med distribuert samhandling kan man skape programmer der også seeren utgjør en aktiv del av sendingen. Vi søker i dette eksperimentet å finne løsninger som gir deltakerene en følelse av nærhet til tross for geografiske avstander. Dette bør være relevant for Åpen kanal hvor den nære, naturlige og enkle samtalen står i fokus. Åpen kanal kan bli den første deltakerstyrte debattarena med god tilsdervedværelse og dybde i tema. Teknikken er i det følgende eksperimentet basert på eksisterende teknologi, men med visse ideelle forutsetninger. Folk flest vil ikke ha dedikerte fiber-linjer som her, men forsøket regnes likevel som relevant på et konseptuelt plan. Hvordan tilvarende produksjoner kan utføres i realistiske miljøer blir drøftet i slutten av rapporten.



Figur 14-1: Første fiber dedikerte fibersammenslutning mellom HiL og HiG. Termineringsverktøy nødvendig for sammenslutning (til venstre) og gjennomslag av første bilde (til høyre)

### 14.1 Samproduksjon mellom HiL og HiG

Forskningsprosjektet Television in Digital Enviroment (TiDE) er et paraplyprosjekt som knytter sammen ulike delprosjekter som hver for seg tar utgangspunkt i hovedmålet for prosjektet: å analysere konsekvenser av digitalisering, konvergens og nye medier for fjernsyn. Prosjektet består av både rene forskningsprosjekter og utviklingsprosjekter som utprøver ulike løsninger og dokumenter er resultatene på forskjellige måter. Gjennom prosjektet ønskes det å skape en møteplass for forskere og fjernsynsfolk omkring et ytterst viktig tema for utviklingen av fjernsynet (TiDE, 2005). Denne delen er skrevet i samarbeid med bachelor og masterstudenter ved Høgskolen i Gjøvik samt min veileder ved HiL, Sigmund Andresen.



Figur 14-2: Sentrale bidragsyttere til TiDE prosjektet. Mats Erixon, KTH (til venstre), Sigmund Andresen, HiL (i midten) og Claus Knudsen, HiG (til høyre).



Shared Virtual Reality (SVR) ble gjennomført som en del av prosjektet Television in Digital Environment (TIDE). Shared Virtual Reality er betegnelser som brukes for å beskrive nærhet og tilstedeværelse i digitale medier. Formålet med TIDE/SVR-prosjektet er å undersøke i hvilken grad nærhet og tilstedeværelse kan produseres i et fjernsynsprogram der deltakere befinner seg på geografisk forskjellige steder. Deltagerne i prosjektet har vært studenter og ansatte fra Høgskolen i Lillehammer (HiL) og Høgskolen i Gjøvik (HiG) samt en masterstudent fra NTNU (meg).



Figur 14-3: Samling av deltakerne i SVR-prosjektet ved Høgskolen i Lillehammer

### *14.2 utfordringer i TV-produksjoner med distribuert samhandling*

I fjernsynsproduksjon i dag er nærhet og tilstedeværelse, i forbindelse med kommunikasjon over geografiske avstander, et problem. I situasjoner der man har et intervjuobjekt eller en debattant som sitter i et annet studio enn der programmet blir produsert, vil vedkommende bare høre og ikke se de man snakker med i hovedstudio. Samtidig har de i hovedstudio også begrensede muligheter å se personen som befinner seg i det andre studioet. Dette legger ikke til rette for god kommunikasjon, der blant annet kroppsspråk og øyekontakt er nøkkelkomponenter.

Dette problemet er det SVR-prosjektet forsøkte å finne en løsning på. Hvordan øke følelsen av nærhet og tilstedeværelse hos deltagerne i studioene. Mediatoren stod i sentrum for løsningen som SVR-prosjektet testet. Den har egenskap til å vise video samtidig som den kan ha et kamera som filmer der videoen vises. Dette betyr at man kan oppnå videokonferanse der man ser rett inn i et kamera, mens man likevel kan se på en mottager i andre enden. Det er på dette området som mediatoren kan ha muligheten til å forbedre en TV-produksjon. SVR-



prosjektet undersøkte om bruken av mediatoren kunne tilføre nærhet og tilstedeværelse i en TV-produksjon der deltagere befinner seg på geografisk atskilte steder.

### 14.3 Terminologi

Det benyttes mange fagbegrep innen feltet presens forskning, og mange av ordene er låneord fra andre språk, samt ord hentet fra andre fagfelt. For å unngå misforståelser er de viktigste begrepene som benyttes i denne rapporten beskrevet her.

- Transparent teknologi er et begrep som benyttes for å beskrive teknologiske løsninger som er ergonomisk og intuitivt utformet slik at brukeren legger minst mulig merke til at det benyttes teknologi for å oppleve mediert nærvær. Transparent teknologi muliggjør en innlevelse som kan føre til at brukeren overser selve teknologien, og dermed kan konsentrere seg kun om opplevelsen.
- Virtuelt rom brukes om det «rommet» som oppstår når to eller flere personer kommuniserer gjennom en kanal. Personene møtes ikke i de rommene de fysisk befinner seg i, men i et skapt rom som er manipulerbart.
- Mediatoren er en applikasjon som skaper øyekontakt mellom deltakerne på de ulike geografisk atskilte lokasjonene. Den fysiske innretningen består av en monitor som viser bildet speilvendt og et speil som reflekterer bildet slik at det vises rett. I tillegg er det et kamera bak speilet, og en kobling til en overføringsenhet som kan ringe opp andre videokonferanse enheter.



Figur 14-4: Mediatoren. Det halvgjennomskinnelige speilet sørger for blikk-kontakt ved at bildet fra personen man kommuniserer med blir reflektert foran linsen til kamera (Samme prinsipp som i en telepromter).

### 14.4 Laboratorieoppsett

SVR-prosjektet disponerte høsten 2005 to studioer ved Høgskolen i Gjøvik (HiG) og Høgskolen i Lillehammer (HiL). Studioene ble koblet sammen og brukt i en TV-produksjon. I begge studioene ble det bygget en mediator. Mediatoren er en installasjon som tillater toveis videokommunikasjon der man har øyekontakt med personen i andre enden. Dette er en av nøkkelkomponentene i forsøket som var avgjørende for å kunne påvirke deltakerens opplevelse av tilstedeværelse. Det siste hovedtema som ble undersøkt under dette prosjektet var hvordan en TV-produksjon i dette miljøet tok seg ut for en testgruppe i forhold til en TV produksjon laget med tradisjonell teknologi.

### 14.5 Gjennomføring



**Figur 14-5: Kontrollrom for sammensetting av det virtuelle rommet (til venstre). Claus og meg i studio, jobber for å opprette IP-forbindelse til Gjøvik via videokonferanse-enheter (til høyre).**

Forsøket ble gjennomført med to delproduksjonen fordelt over to dager. I del 1 var to deltagere med i en fjernsynsproduksjon der de skulle ha en filosofisk samtale omkring ulike stikkord gitt av programleder. Programleder og en deltager var tilstede i studio på Lillehammer, mens den andre deltageren satt i studio på Gjøvik. Deltagerne kjente hverandre godt gjennom jobb, og begge hadde kjennskap til og erfaring med bruk av vanlig videokonferanseutstyr. Del 2 var også en fjernsynsproduksjon. Her deltok to profesjonelle skuespillere som hadde i oppgave å utøve teatersport basert på følgende temaer:

- En romantisk middag
- En opphetet debatt
- Forsøk med bevegelse i virtuelt rom.

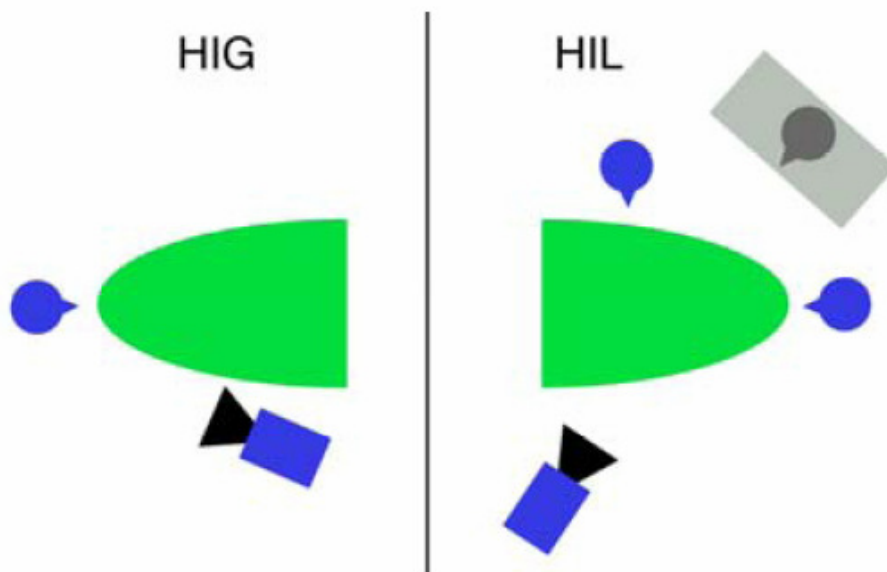
Del 2 ble også gjennomført ved at en skuespiller befant seg på Lillehammer mens den andre var på Gjøvik. Hensikten i både del 1 og 2 var å se om Gjøvik og Lillehammer kunne mikses til et virtuelt studio der deltagerne kunne føle at de møttes.

Vi ønsket også å undersøke hvilken verdi den distribuerte samhandlingsproduksjonen kunne ha for en TV-seer. Dette ble gjort ved å vise et utdrag på rundt 12 minutter delt opp i 3 sekvenser (A, B og C). Sekvens A var på rundt 3 minutter og viste rektorene i diskusjon med tradisjonell teknologi. Sekvens B var om lag like lang og viste rektorene i samme setting, men med mediatorene i bruk. Siste sekvens (C) var på de resterende 6 minuttene og viste de to skuespillerne i et felles «rom» som har et romantisk måltid sammen.

### 14.6 Rektoriintervju

Rektoriintervjuene ble gjennomført med to forskjellige settinger. Den første settingen er et klassisk oppsett der det er en programleder og et intervjuobjekt i studio mens et annet intervjuobjekt befant seg på et annet sted med kun et kamera og lyden fra programleder og meddebattant. I den andre settingen har intervjuobjektet på distanse en mediator og kan se meddebattanten. Overgangen mellom de to forskjellige settingene ble gjort glidende. Det ble ikke produsert som to separate opptak.

#### 14.6.1 Intervju av person på distanse uten mediator

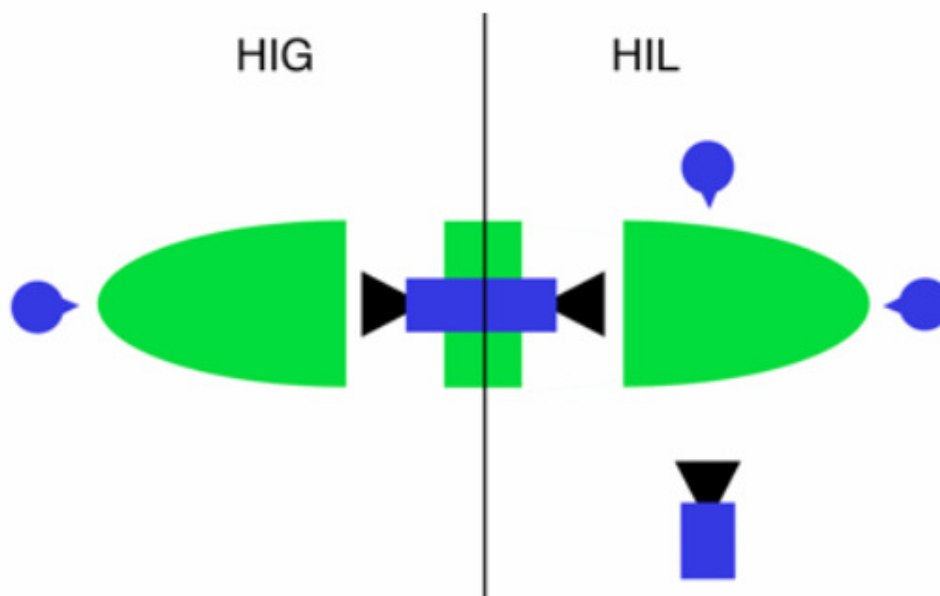


Figur 14-6: Intervju uten mediator, prinsippkisse

Dette er settingen som brukes i normal tv-produksjon i dag. Som vi kan se av illustrasjonen har ikke deltageren på HiG noen skjerm hvor han kan se de to andre. Programlederen og deltageren på HiL Har en skjerm men denne skjermen er plassert bak de to slik at de må snu seg for å få kontakt med deltageren på Gjøvik. Kameraet har mulighet for å få med de to deltagerne på lillehammer pluss skjermen som viser Gjøvik og på den måten konstruere et treskudd hvor alle de tre deltagerne vises. Jørn Wroldsen som er intervjuobjektet på HiG sitter alene i et studio og ser inn i et kamera. Han har en propp i øret hvor han får lyden fra de to andre deltagerne men han har ikke noen skjerm som gir han bilde av de to som sitter på Lillehammer.

#### 14.6.2 Intervju av person på distanse med mediator.

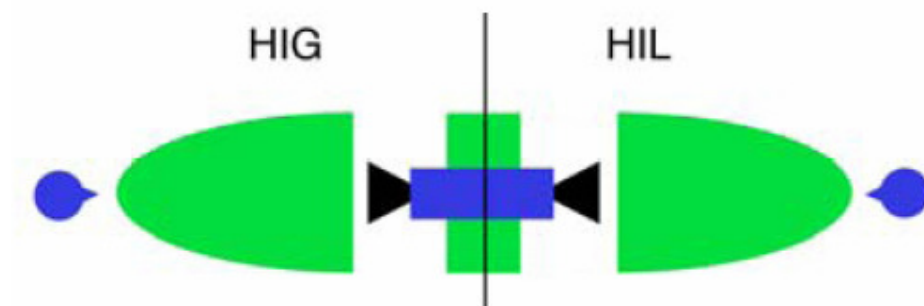
I neste del av intervjuet får deltagerne hver sin mediator. Oppsettet blir som vist i illustrasjonen. Nå er kameraet på hvert av stedene på linje med deltagerens øyne. Deltageren på HiG har nå fått en monitor i mediatoren som gjør at han kan se sin motdebattant rett i øynene. Programlederen og deltageren på HiL har en tilsvarende visning av HiG-debatanten. Et kamera ved siden av mediatoren filmer programlederen og gjør det mulig å også legge inn bildet av programlederen på skjermen på Gjøvik dersom det er ønskelig.



Figur 14-7: Intervju ved bruk av mediator, prinsippkisse

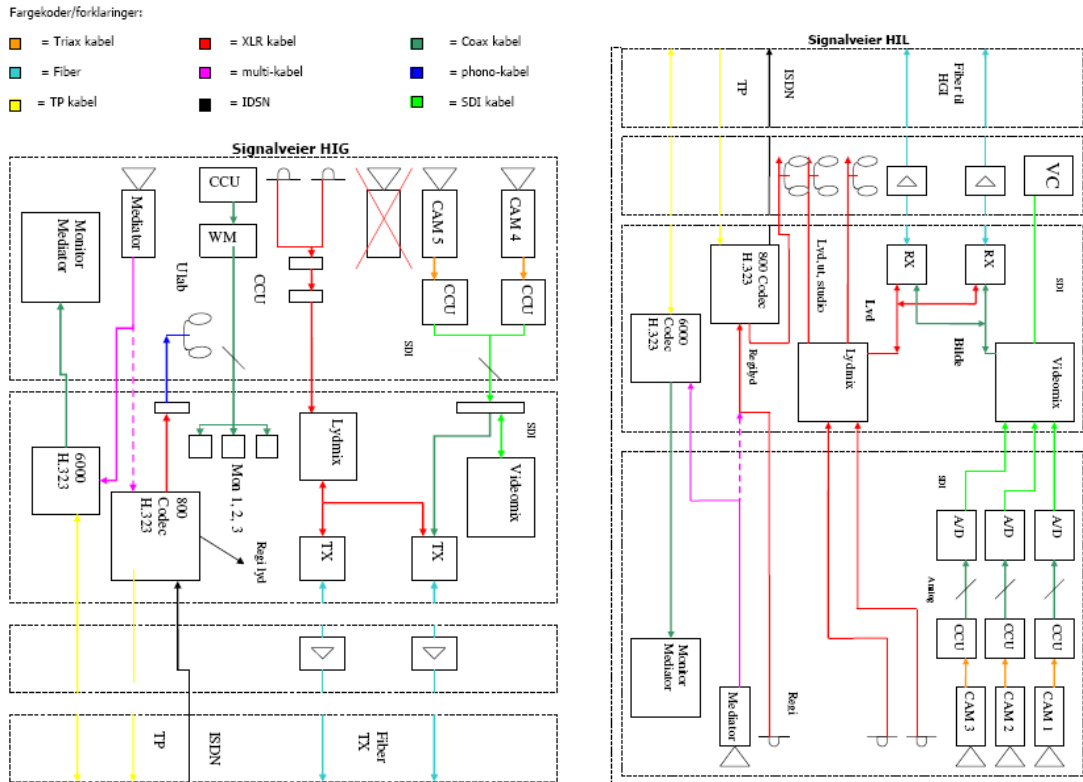
## 14.7 Skuespillere

Under alle forsøkene med skuespillere ble mediatoren brukt. Ved å bruke skuespillere i stedet for «vanlige mennesker» som i intervjuene med rektorene, kunne man fremprovosere helt andre situasjoner og reaksjoner på grunn av at skuespillerne er profesjonelle. De bruker kroppen sin og følelsene sine som verktøy i det daglige. De er profesjonelle på å fremprovosere og tolke reaksjoner hos motspilleren. Det ble ikke gjort noen endringer på oppsettet mellom de forskjellige situasjonene, «Romantisk middag», «Debatt» og «Lek». I alle tre situasjonene var oppsettet som illustrasjon 3 viser. De fire kameraene som befinner seg utenfor mediatoren, ett på Gjøvik og tre på Lillehammer, er bevegelige kameraer med kameraoperatører som kunne beveges etter ønske og regis kommando for å gi forskjellige utsnitt og vinklinger.



Figur 14-8: Skuespiller med mediator, prinsippskisse

## 14.8 Infrastruktur



Figur 14-9: Signalveier ved hhv. Høgskolen i Gjøvik og Høgskolen i Lillehammer.

Mediatoren ble koblet til nettverket mellom høgskolene som består av en fiberlinje fra Uninett med en kapasitet på 155Mbps (Andresen s. og Knudsen C., 2005). Forsinkelsen på linjen mellom de to skolene ble målt til 6,3 millisekund (brukte kommandoen ping i kommandolinje verktøyet for windows gjentatte ganger og regnet ut gjennomsnittlig forsinkelse). En større andel av forsinkelsen tilskrives konverteringen fra analoge til digitale bilder i bildemikserens konvertere. Det formatet som ble brukt i bildebehandlingsskjeden heter SDI. Ved å bruke et slikt ukomprimert format kan forsinkelsen reduseres til 40 ms fra signalet treffer CCD-brikken til det går inn i bildemikseren eller blir distribuert over en linje (DV kamera med firewire-tilkobling som ofte forekommer i Åpen kanal sitt produksjonsmiljø vil ha langt større tidsforsinkelse på grunn av komprimering i kamera). Bildemikseren bufferer også en frame og forårsaker en ytterligere forsinkelse på 40ms. SDI (med kapasitet til embedded audio der lyd multiplekseres inn i SDI-signalet) krever en overføringshastighet på 270Mbps. HiG/HiL sitt eget nettverk har ikke tilstrekkelig med kapasitet, siden det også er annen trafikk enn bilder/lyd som går frem og tilbake. Det ble etablert trådpar med fibere, som gjorde det mulig å

sende to fjernsynskanaler. Videokonferanseutstyret var det utstyret som sto for den største forsinkelsen og tap av frames under produksjonen. Videokonferanseenheterne som ble brukt til TV-produksjonen var av typen Tandberg 6000. Dette er en av de nyere videokonferanse enhetene som Tandberg har som også har en den nye kodeken H.264. Den hadde mulighet til å jobbe på en hastighet på 4Mbps over IP. På grunn av ustabile switcher forandret hastigheten i nettet seg som førte til at det uheldigvis ble brukt den eldre kodeken H.263. Den har bare mulighet til å jobbe på 1536Kbps, ca 1,5Mbps, og med begrenset båndbredde ble det tap av "frames" under produksjonen (Andresen S. og Knudsen C., 2005). Under opptak ble det lagret 25 bilder/frames per sekund lokalt. Etter at bildene ble sendt til Lillehammer og tilbake igjen var det 16 frames offset mellom de to bildene. Det tilsvarte en forsinkelse på 320ms en vei, noe som klart ble lagt merke til under produksjonen. Det ble i etterkant av prosjektet gjennomført vellykkede tester med H.264, og forsinkelsen ble målt til 4 frames mellom Lillehammer og Gjøvik.

Siden dette skulle være en fjernsynsproduksjon i Norge, måtte signalet være av standarden PAL (Andresen s. og Knudsen C., 2005). Bildene som ble vist på skjermen i mediatoren ble sendt via videokonferanse-enheterne og komprimert før den digitale overføringen på IP,

Her ble det brukt en kodeks som reduserer kvaliteten og innfører betydelig tidsforsinkelse. For å konvertere videosignalene som skulle inn i bildemiksen samt de to lydkanalene ble det benyttet en mer høykvalitets digitalisering. Med minimal tidsforsinkelse samples bildelinjene en for en uten kompresjon før samplene sendes over en fiberoptisk dedikert linje (sort fiber). For dette prosjektet ble det brukt en Marenus-boks. Lyden ble samlet i 40KSps/18bit mens video ble samlet i 20MSps/8bit. Denne versjonen av Marenus sampler kompositt video som ikke regnes for å ha tilfredsstillende kvalitet i en profesjonell fjernsynsproduksjon (digital SDI eller analog komponent video benyttes vanligvis der de ulike fargekomponentene Y, Y-B og Y-R har hver sin kanal). Det viste seg likevel ikke å være en avgjørende kvalitetsbegrensning i forhold til andre faktorer. Marenus sin samplingsteknikk (tilsvarende for deres SDI-utgave) var svært heldig med tanke på den kritiske faktoren forsinkelse.

Siden hver linje kodes og dekodes hver for seg i bildene, blir forsinkelsen minimal. Man behøver mao ikke vente til hele bildet er mottatt (40ms) før det kan vises.

### 14.9 Målsetting

Prosjektets overordnede mål har vært å bruke kommunikasjonen mellom deltakerne til å forsøke å optimalisere følgende kritiske faktorer i produksjon av nærhet og tilstedeværelse når deltagerne ikke er fysisk tilstede i samme rom:

- blikkontakt som en faktor for å skape nærhet
- reduksjon av tidsforsinkelse av det overførte signalet for å gi en bedre dynamikk.

Blikkontakt ble oppnådd med hjelp av en "mediator", en optisk innretning som gjør at en bruker ser rett inn i et kamera samtidig som han/hun ser på en TV monitor. En mediator ble fysisk plassert på hvert av stedene der personene som skulle kommunisere befant seg. Selve settingen for forsøket var at det skulle produseres et fjernsynsprogram der de deltagende personene ikke var tilstede fysisk i samme rom. Vi ønsket å undersøke hvordan personene opplevde dette, og hvilke faktorer som påvirket denne opplevelsen.

Undersøkelserområdene for SVR-prosjektet ble delt inn i fem:

- 1) fortellerteknikk på handlings- og fortellerplan
- 2) interaktivitet med fokus på både menneske og maskin
- 3) betydningen av transparent teknologi
- 4) distansearbeid i staben
- 5) opplevelse av rom og kropp hos deltagerne.

Denne rapporten fokuserer interaktivitet mellom mennesker, og på programdeltagernes opplevelse av interaksjon som deltagere i en fjernsynsproduksjon. Den søker også å beskrive responsen tv-seere har gitt på grunnlag av materiell produsert i prosjektet.

Eksperimentet regnes som relevant i forhold til Åpen kanal der distribuert samhandling med tid og stund vil være en realitet for deltakere samt seere. Som evalueringskriterium for seerens opplevelse av produksjonen ble det også satt sammen en film av råmateriale for visning til en testgruppe. Til denne ble det laget et spørreskjema. Dette skjemaet ble presentert for testgruppen etter visning av filmen. Skjemaets formål er å hente ut seerens erfaringer ved visningen av resultatfilmen. Spørreskjemaet går hovedsakelig inn på seerens opplevelses av kontakt aktørene imellom og hvordan dette formidles ut til seeren. Skjemaet består av 21 spørsmål. Dette er spørsmål med svaralternativer og oppfølgingsspørsmål med mulighet for supplementærinformasjon i tekstfelt.

### 14.10 Seerundersøkelsens relevans

Seeren skulle oppfatte filmen som en vanlig tv-sending. Når det gjaldt visningsomgivelser



skulle også disse reflektere hvordan man vanligvis ser på tv. Det ble satt opp et tv-apparat på en benk i et hjørne i et stille rom, med fem-seks enkle lenestoler foran.

Det var ikke ønskelig å vise programmet til flere en fem personer samtidig, derfor ble det satt opp en visningsplan for de originalt 50 inviterte. Denne ble senere modifisert etter behov da antallet oppmøtte til hver visning varierte. I påfølgende undersøkelse spurte vi om seerne syntes omgivelsene de ble plassert i var behagelige. Her svarte 33 % at de syntes de var avslappende, 57 % syntes det var helt greit og 10 % syntes det var vanskelig å konsentrere seg. Det er dermed rimelig å konkludere med at testmiljøet var tilfredsstillende for å gjøre målinger på hvordan en seer av åpen kanal ville hatt glede av SVR-teknologien i dag.

Det bør likevel bemerkes at produksjoner av denne typen ligger noe frem i tid og at miljøet kan gi et feil bilde av hvordan seeren vil konsumere innholdet i fremtiden. Åpen kanal søker å aktivisere seerne. Et mer interaktivt innhold vil kunne forutsette at seeren sitter på noe som ligner mer på en arbeidsstasjon der man blant annet kan ha tilgang på et webkamera eller andre medierende redskaper. Med digitale sendinger (spesielt IPTV) er det også naturlig at PC-en fungerer som en TV (set-top boksene som gjør om det digitale TV signalet slik at det kan vises på en analog TV er i prinsippet skreddersydde PC-er). Tradisjonelt har likevel TV-en vist seg å ha en plassering i hjemmet som ikke ligner på PC'ens og som ikke legger opp til aktiv seerdeltakelse. En undersøkelse av 15 ulike familier i Oslo sitt forhold til PC i hjemmet viser at det er sterk motstand mot å ha en PC i stuen (Ling, Rich og Thrane, Kristin, 2000).

Den bryter med det man ønsker å uttrykke med rommet. Stuen er scenen med rekvisitter som uttrykker familiens kulturelle opphav og status. Stuen er også et område der familien kan trekke seg vekk fra arbeidslivets mas. PC-en representerer i denne sammenheng et uønsket funksjonelt og arbeidsrelatert element (Ling, R and Thrane, K, 2002).

Det er i stuen vi i hovedsak konsumerer innhold. Det gjør vi passivt og skjeldent gjennom en PC. Men det er grunn til å tro at "stue-tradisjonen" vil bli utfordret slik den ble ved innføringen av TV-apparatet på 50- og 60-tallet. TV i stuen brøt den gang med at stuen var stedet for konversasjon og lesning (Guttu, J. Jørgensen og Nørve S.,1985). Undersøkelser utført av Telenor R&D (Ling, R and Thrane, K, 2002) viser at det nå er et generasjonsskifte i bruk av PC som TV. Foreldrene i undersøkelsen lar seg underholde gjennom den mer rigide kringkastningsstrukturen (TV og radio) mens ungdom i stadig større grad søker underholdning gjennom interaktive mediestrukturer (Video messenger, VOD og Peer to Peer fildeling). Undersøkelsen viser også at TV i liten grad har evne til å engasjere, mens PC åpner for både sosialt og pedagogisk engasjerende underholdning. For Åpen kanal blir det

interessant å se hvordan deltakerne kan benytte videokonferanseutstyr som nå blir stadig mer tilgjengelig med gratis videoklienter (f.eks Skype video) samt billige og gode web-kamera. Spillkonsoller har allerede en plass i stua og kan også bidra til å gjøre interaktivitet "stuerent". Kvaliteten på disse kan snart konkurrere med dyre videokonferansesystemer. Sony Playstation3 vil komme med en ny utgave av webkameraet som heter Eyetoy. Denne løsningen støtter HD-oppløsning og har mer enn nok prosesseringskraft til å formidle et bilde i kringkastningskvalitet (6 Cell prosessorer som jobber i parallell yter 2Tflops). Tar man dette i betraktning kan det se ut som om en TV-produksjon med distribuert samhandling om kort tid kan realiseres. Men selv om teknologien hjelper oss er det mange flere problemer som må løses. En god samhandling er avhengig av minimal tidsforsinkelse, øyekontakt med deltakerne og integrerende scenekonstruksjoner. Dette er problemer som er svært relevant for deltakernes følelse av interaktivitet. Det er også problemer teknologimiljøene i stor grad har oversett. Vi ønsker her å gjøre praktiske forsøk for å avdekke mer spesifikt hvor problemene oppstår i en TV produksjon med distribuert samhandling.

### *14.11 Interaktivitetsbegrepet*

Et begrep som er spesielt interessant i forhold til deltakerens opplevelse av tilstedeværelse er interaktivitet. Det er mange ulike måter å definere interaktivitet på. Begrepet interaksjon har en lengre tradisjon, og oppfattes gjerne som utgangspunktet for interaktivitetsbegrepet. Dette begrepet er knyttet til mellommenneskelig kommunikasjon, gjensidighet, samspill og vekselvirkning (Haugsbakk 2000). Interaksjonsbegrepet oppstod i programmeringsmiljøet i forbindelse med kompilering av programkode. Det var en stor framgang at programmereren kunne få løpende tilbakemeldinger på feil i koden i stedet for å gjøre ferdig all kode og sende denne til en kø for bearbeiding av en egnet datamaskin. Det ble opplevd som å være i interaksjon med maskinen, og ble derfor kalt interaktiv programmering. Interaktivitetsbegrepet blir i dag brukt på mange områder, blant annet innen medievitenskapene. Et forsøk på å definere interaktivitet innen medievitenskapene er beskrevet av Jens F. Jensen: "Interaktivitet kan bredt defineres som et mål for mediets potensielle muligheter for å flade brukeren øve innflydelse på den medieformidlede kommunikations indhold og/eller form" (Jensen, 1998).

Jensen bruker tre kriterier til å bestemme hvor interaktivt et medium er; valg – som gir tilgang til informasjon, konversasjon – der brukeren deler sin informasjon med andre, og registrering – der systemet samler informasjon for å tilpasse seg brukeren. I medievitenskapene står interaktivitetsbegrepet som en del av begrepet presence. Jonathan Steuer beskriver

interaktivitet som en av forutsetningene for at personer som deltar i mediert kommunikasjon skal føle tilstedeværelse. Han peker på tre faktorer som påvirker graden av interaktivitet: responstiden til teknologien som brukes (hastighet), brukerens mulighet til å endre systemets oppsett (range), og sammenhengen mellom brukernes input og responsen (mapping) som gis (Steuer 1992).

### *14.12 Interaktivitet i TIDE prosjektet*

Interaktivitet en av flere presens faktorer (Knudsen, 2004). Interaktivitet kan igjen deles inn i hastighet, range og mapping. (Steuer,1993) Siden dette var en tv-produksjon, satte det begrensinger for hva som var mulig. Hadde det vært et opptak som skulle redigeres i ettertid, kunne det legges på effekter som for eksempel bevegelige gjenstander i det virtuelle rommet. Hvis noe gikk galt, brudd på forbindelsen, kunne det bare spilles inn på nytt. For å skape det virtuelle miljøet ble det for prosjektet brukte en mediator som hadde som hensikt å skape øyekontakt mellom brukerne. Den besto blant annet av en video kodek (koder - dekode) som vanligvis brukes til videokonferanse.

#### 14.12.1 Hastighet

Hastighet er en av de tre faktorene som interaktivitet kan deles inn i (Steuer, 1993 s. 15). Det å ha et VR-miljø som operer i mest mulig realtime er viktig, men enklere i et begrenset miljø enn i vårt tilfelle. Det som gjør det enda mer vanskelig her er at dette prosjektet benytter kameraer som må gjennom bildebehandlingsenheter med forsinkelse (spesielt stor for videokonferanse kodekene). Video og lyd skal dessuten overføres over et stort nettverk som medfører lang propagasjonstid (tiden det tar å forflytte lyssignalene gjennom fiberen) samt betydelige forsinkelser i rutere og switcher for det som sendes over IP. Utfordringene mhp hastighet er nærmere beskrevet under punktet "Infrastruktur".

#### 14.12.2 Range

Range blir definert ut ifra antall forskjellige muligheter for påvirkning og i hvor stor grad en bruker de påvirkningene kan endre omgivelsene (Steuer, 1993). Eksempler på dette kan være at deltakeren tar i bruk videomiksere for å forandre hvor bildet kommer fra og på den måten

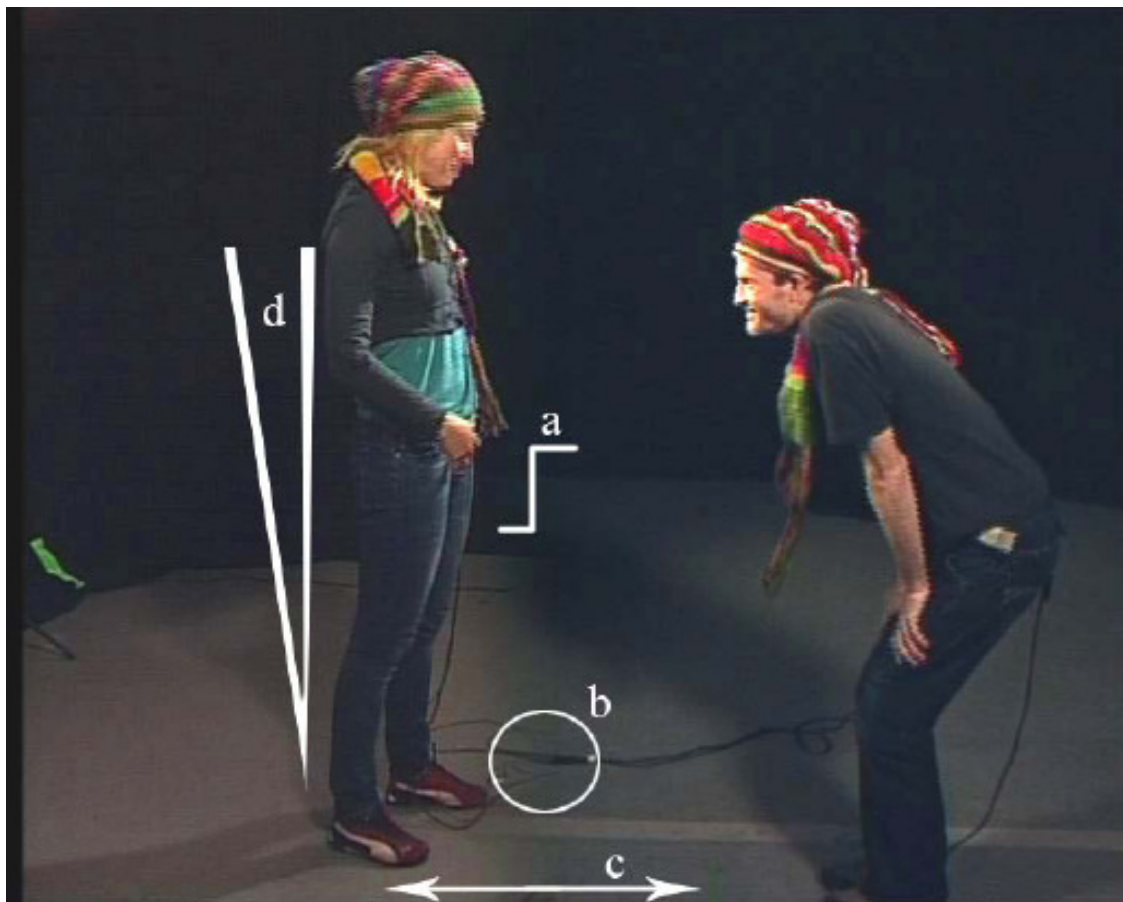
endre VR-rommet, hva slags utsnitt kamera skal ha på et gitt tidspunkt, fjerne farger fra bildet og erstatte det med for eksempel en annen videokilde (Knudsen, 2004).



**Figur 14-10:** Viser skillelinjen mellom de to bildene fra Gjøvik (til venstre) og Lillehammer (til høyre). I området i midten er det en "soft edge" på "wipe"-effekten der det foranliggende bildet til venstre blir gradvis mer transparent mot det bakenforliggende bildet til høyre. Dette området kan ikke brukes i produksjonen uten å bryte illusjonen (Prøveopptak TiDE, 2005).

Den siste dagen av produksjon, var målet å skape en "romantisk middag" og en diskusjon mellom to skuespillerne. For den romantiske middagen ble det kjøpt inn to like bord, stoler og andre effekter som det var behov for. (Andresen s. og Knudsen C., 2005. s. 10-11) Det ble brukt mye tid på å lage lik bakgrunn og få effektene til å se riktig plassert ut. Bildene fra Lillehammer og Gjøvik ble klippet sammen slik at når skuespillerne satt ved bordet så det ut som om de satt rett overfor hverandre. Det ble en skillelinje mellom de to bildene men den var nesten ikke synelig. Så lenge de to skuespillerne holdt seg innenfor sin side gikk det fint, men beveget de en hånd utenfor området sitt, forsvant de ut av bildet. Til slutt skulle skuespillerne prøve å bevege seg sammen i et felles virtuelt rom. De skulle stå overfor hverandre og leke sammen men de så ikke hvor skillet i det virtuelle rommet var. Det eneste de fikk se var et

monitorbilde av seg selv. Det var her en stor utfordring med å få riktig størrelsesforhold mellom deltakerne.



Figur 14-11: Bildet viser et dårlig sammensatt virtuelt rom på grunn av a) eleveasjonsforskjell, b) ikke sammenfallende objekt, c) ulik chroma/farge-metning og d) ulik vatring på kamera. (Prøveopptak TiDE, 2005)

### 14.12.3 Mapping

Dette kan sammenliknes med en kartlegging av i hvilke (hvis noen) menneskelige bevegeleser og eller kommandoer ved stemmebruk, som hadde en effekt på det virtuelle miljøet. Et eksempel på det er å bruke en virtuell hanske. Hvis brukeren simulerer et kast av en ball med hansken, vil ballen faktisk bevege på seg i miljøet (Steuer, 1993) . Et annet eksempel kunne være at en bestemt bevegelse styrte lydstyrken eller hvor mye et kamera blir zoomet inn.

Det er ikke blitt brukt noen form for mapping i dette prosjektet. Verken når rektorene snakket sammen, eller når skuespillerne jobbet sammen. I mediatorene så de hverandre i øyene, eller i profil. Det ble ikke laget noe kommandoer som de kunne benytte for å forandre miljøet. De

som styrte kamera kunnet etter kommando ta nærbilder eller flytte på kamera osv. men de som skulle være del av det virtuelle miljøet hadde ingen kontroll over hva som foregikk rundt dem. De kunne bevege på seg innenfor området sitt og hva de gjorde hadde ingen effekt på miljøet. De kunne se på hverandre og tilpasse seg hverandre men det var helt tilfeldig hva som ble gjort.

### *14.13 Drøfting*

#### 14.13.1 Interaktivitet og fortellerteknikk

Selve systemet med bruk av mediator i mediert kommunikasjon syntes å være relativt godt egnet til å skape interaksjon mellom de deltagende parter. En vurdering av systemet opp mot Steuers tre faktorer viser at brukerne opplever systemets responstid som meget bra. Ingen av programdeltagerne har latt seg forstyrre av forsinkelser eller dårlig synkronisering. Når det gjelder brukerens adgang til å påvirke systemet, er denne svært begrenset. Det er opp til regi å endre bildet eller utsnittet som brukeren får se på mediatoren. Det gjør at brukeren ikke opplever systemet som spesielt interaktivt. Dette ble mest merkbart under det tredje tema der skuespillerne skulle improvisere og bevege seg i rommet. Regi skiftet bilder på en måte som gjorde at skuespillerne ikke følte at de hadde kontroll. De hadde heller ikke noen mulighet til å formidle sine ønsker om eventuelle bildeskift underveis. Også deltagerne ønsket å kunne hatt en større mulighet til å påvirke systemet. Dette gikk på opplevelse av at avstanden var for stor, og at nærheten med et fiksert bilde ble for stor. Mediatoren har fungert godt i forhold til at det er stor forutsigbarhet i hvordan den reagerer på input fra brukeren. Signalet blir overført med en kvalitet som ingen av brukerne har reagert negativt på. Det har heller ikke oppstått større artefakter som har forstyrret kommunikasjonen. Det ligger derfor tilrette for god interaksjon på dette området.

#### 14.13.2 En-til-mange

Første del av forsøket der to deltagere var med i et samtaleprogram, ble preget av at deltagerne kjente hverandre godt fra før. De hadde en høflig tone, og ventet til det naturlig var deres tur til å snakke. Programmet ble enkelt styrt av programleder, og det oppstod ingen opphetet stemning. På tross av dette formidlet de to deltagerne to relativt forskjellige inntrykk etter å ha deltatt i forsøket (Beskrevet som ”å lytte etter noe udefinerbart” av en av skuespillerne).

Dette ser ut til å kunne ha sammenheng med følgende faktorer:

- programleder var tilstede på Lillehammer
- de to studioenes hadde ulike utforming og innredning
- personene hadde ulik bakgrunn, forventning og kunnskap

Det kan være flere faktorer som innvirket, men de nevnte ser ut til å ha størst betydning. Oppsettet var slik at programleder kunne se deltager i Gjøvik, men deltager i Gjøvik kunne ikke se programleder. Dette gjorde at deltager i Gjøvik følte det var god interaksjon mellom programleder og deltager i Lillehammer. Det ble opplevd at den fysiske nærheten mellom de to i studio på Lillehammer ikke kunne kompenseres ved at deltager i Gjøvik og deltager i Lillehammer hadde blikkontakt. Det viser seg altså å være mer komplisert å produsere nærhet i en en-til-mange situasjon. Hvis deltager i Gjøvik hadde oppnådd visuell kontakt med programleder, kunne dette ha økt opplevelsen av interaksjonen mellom Gjøvik og Lillehammer.

### 14.13.3 Ulike deltakererfaringer

Det var i utgangspunktet et mål å utforme de to laboratoriene mest mulig like. På det tekniske området så dette ut til å fungere bra. Derimot ble det blant annet på grunn av forskjellen i rommenes størrelse og utforming, vanskelig å skape to visuelt like rom. Laboratoriet på Gjøvik ble bekledd med plast og molton for å hindre lysinnslag og for å dekke til vegger/flater som kunne gi refleksjon. Dette ga et rotete og litt kaotisk inntrykk. De to mediatorene var bygget etter samme mal, men endelig utforming og funksjon ble noe forskjellig. En av forskjellene var at speilene som ble brukt hadde ulik størrelse og tykkelse. Tykkere speil førte til at mediator på Lillehammer hadde en større refleks enn den som stod på Gjøvik. En annen forskjell var hvordan mediator var innkledd med molton. På Lillehammer så dette ordentlig ut med en stram og skikkelig kledning, mens det på Gjøvik så ut som om kledningen bare var "surret" rundt rammen. Deltager i Gjøvik opplevde at disposisjonen av laboratoriet og mengden av synlig teknologi i sterk grad hindret ham i å leve seg inn i den delte virtuelle virkelighet som ble skapt. I utgangspunktet var han heller ikke forberedt på at dette skulle være en fjernsynsproduksjon. I tillegg til disposisjonen av laboratoriet, virket det også forstyrrende på interaksjonen at det var en stor avstand fram til mediatorbildet. Dette ble beskrevet som en følelse av å sitte i hver sin ende av et større rom. Disse ulike faktorene bidrar til å redusere følelsen av interaktivitet.

Deltager i Gjøvik reagerte spesielt på disse forholdene, mens skuespilleren lot seg påvirke i mindre grad. Årsaken til dette er trolig de ulike rollene de to hadde i forsøket. Skuespilleren skulle gjøre en jobb og var fokusert på denne. Deltager skulle kun ha en samtale med en kollega og forventet noe annet av omgivelsene.

#### 14.13.4 Erfaringer og kunnskap

De to deltagerne kom fra miljøer med ulik tradisjon. Skuespilleren i Gjøvik hadde god kjennskap til filmmiljøet. Han følte seg derfor komfortabel med studioomgivelsene og det å delta i en fjernsynsproduksjon. Rektoren i Gjøvik hadde erfaring med videokonferansesystemer, og så på oppsettet som er omkring mediator som svært komplisert og omfattende. Skuespillerne er først og fremst innstilt på å gjøre en jobb. De er vant til å bruke sin kunnskap og sine erfaringer til å skape seg et virtuelt rom der skuespillet foregår. Det er derfor ikke så stor forskjell på å spille en rolle på en scene og i et fjernsynsstudio. Utfordringen for disse to er i større grad å tilpasse seg et nytt miljø og finne andre nøkler for å løse oppgaven. Dette viser tre ulike måter å forholde seg til teknologien, noe som krever at resultatene vurderes sett i lys av disse opplysningene. Skuespillerne utrykte at de var overasket over at det var mulig å formidle følelser med bruk av denne teknologien. Det var tydelig at øyenkontakt var med å øke interaksjonen, og at kontakten ble tettere. Dette ble tydelig når skuespillerne spilte det siste temaet der de ikke lengre oppnådde å ha øyenkontakt. Tilbakemeldingen da var at det var vanskeligere å gjøre jobben fordi de mistet kontakten. Skuespilleren i Gjøvik bekreftet også at øyenkontakt gjorde noe med interaksjonen i mediert kommunikasjon. Han sa imidlertid også at den nærheten han opplevde kunne bli for intim med utgangspunkt i at bildet var for statisk. Nærheten kan på denne måten føles ubehagelig.

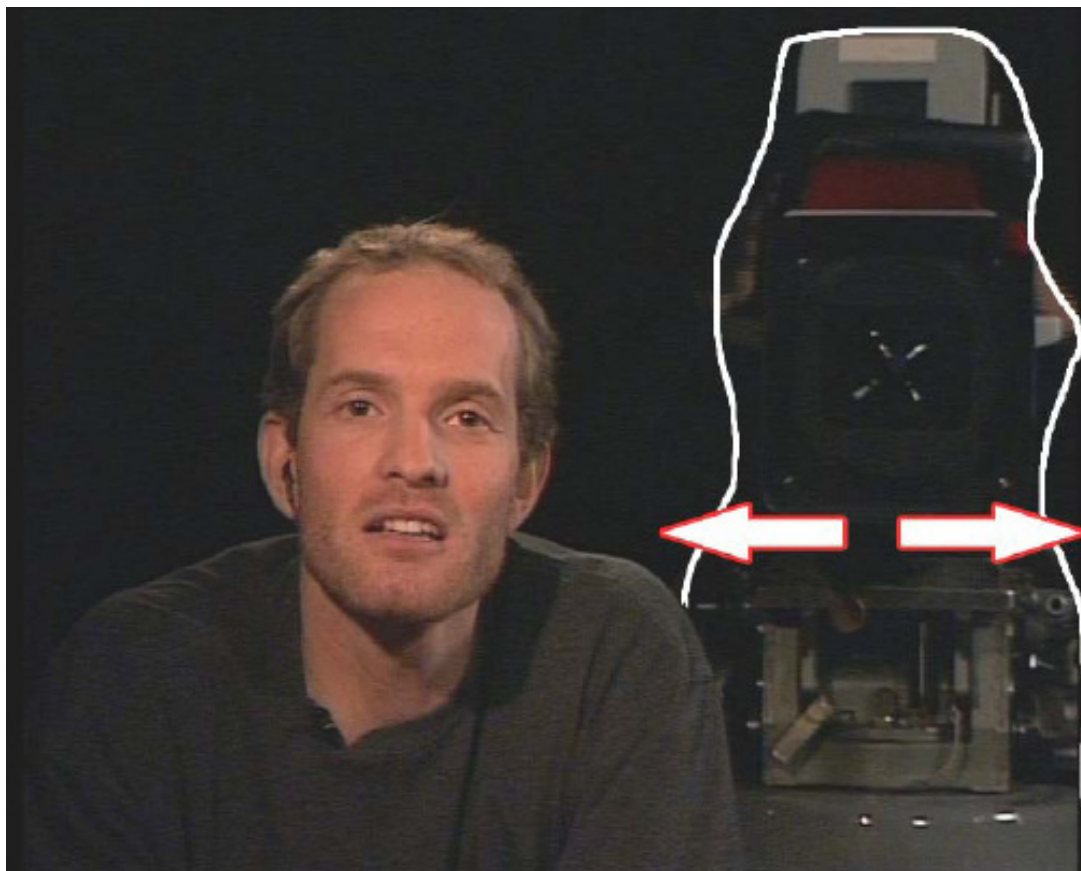
#### *14.14 Konklusjon for deltakere*

Bruk av mediator i en slik type fjernsynsproduksjon ser tilsynelatende ut til å øke interaksjonen mellom deltagerne ved følelse av mediert nærvær og tilstedeværelse. Noen forutsetninger ser allikevel ut til å måtte være tilstede for å oppnå denne følelsen.

- Varierende med erfaring og kunnskap, ser det ut til at omgivelsene kan påvirke opplevelsen av tilstedeværelse og interaktivitet. Ved gjennomføring en enkel samtale kan det være en alminnelig oppfatning at utstyret og teknologien som brukes også er enkelt uten et altfor omfattende oppsett. Også omgivelsene og teknologiens synelighet ser ut til å påvirke



opplevelsen av interaktivitet ved at dette er forstyrrende elementer. Graden av påvirkning ser ut til å ha sammenheng med erfaring og kunnskap.



**Figur 14-12: Objekter som beveger seg i bakgrunnen (her: kamera) kan virke svært forstyrrende på deltakernes blikkontakt og synlig teknologi kan bryte illusjonen av et delt rom. (TiDE film, 2005)**

- Systemets responstid er viktig for at interaksjonen skal oppleves som god. Forsinkelse der lyd og bilde ikke er fullt ut synkronisert kan medføre en viss forvirring hos deltagerne. Dette fører igjen til at deltagerne må bruke mer energi på å dekode informasjonen, og at systemet ikke oppleves som så interaktivt.
- Forsøket ga ikke deltagerne noen mulighet for å påvirke systemet gjennom selv å kunne endre kameravinkel eller bildeutsnitt. Dette reduserer i følge Steuer sin definisjon, graden av interaktivitet. Dette bekreftes også av personene i forsøket. Muligheten til å endre avstand til mediatorbilde, variere vinkel eller utsnitt ville tydelig gitt brukeren en annen opplevelse av interaktivitet.
- Det ligger også utfordringer knyttet til en-til-mange kommunikasjon. I del 1 der rektoren i Gjøvik ikke så programleder, ble kommunikasjonen mellom dem vanskelig. Følelsen av

nærhet var ikke tilstede, og dermed heller ikke følelsen av interaksjon. En-til-mange eller mange-til-mange kommunikasjon krever derfor en mer kompleks produksjon.

Når det kommer til problemstillingen om vi har greid å gjøre teknologien transparent for deltagerne i tv-programmet, er det vanskelig å kunne trekke noen klare slutninger ut fra forsøket. Til det er en referansegruppe på fire mennesker for lite. Det vi derimot har greid å oppnå med prosjektet er en teknisk gjennomførbarhetsprøve. I forkant av SVR virket det høyst usikkert om det teknisk sett lot seg gjøre å gjennomføre prosjektet med de ressurser som var tilgjengelig. I ettertid ser vi at det ble gjennomført på en tilfredsstillende måte og vi sitter igjen med en god indikasjon på hvordan mediatoren kan brukes som et element i tv-produksjon.

### *14.15 Produksjonsresultat for visning*

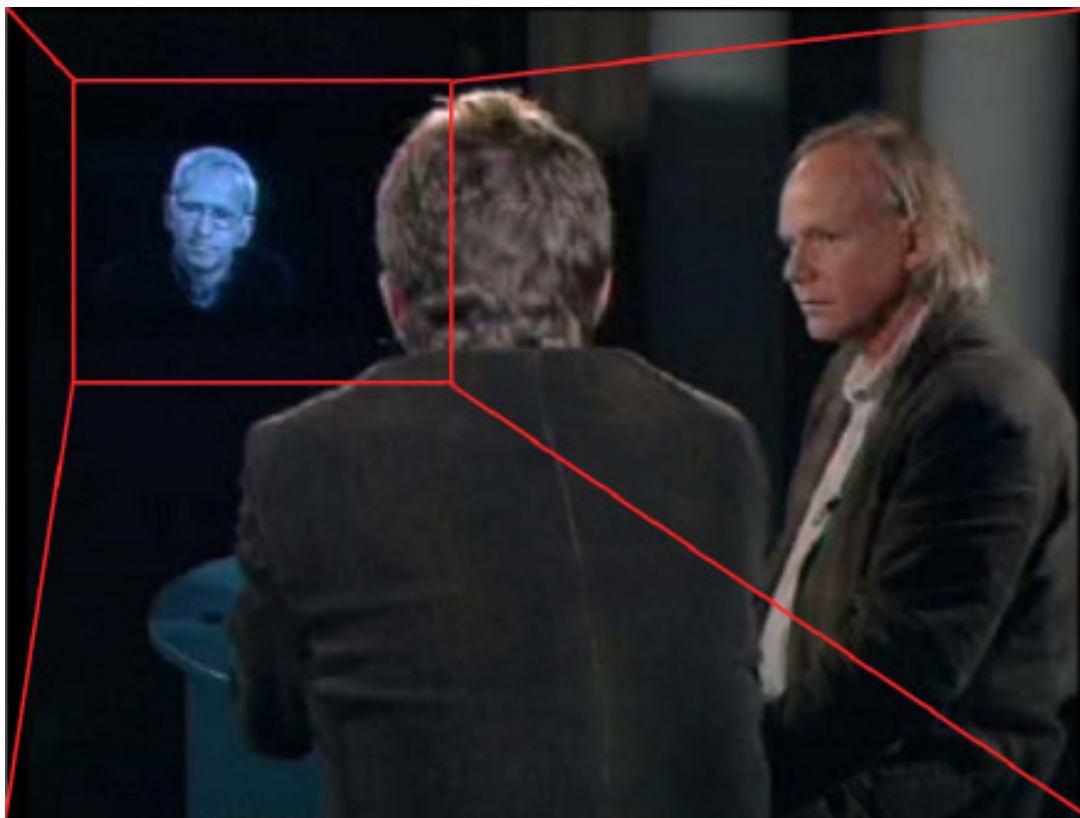
Som tidligere nevnt ble det etter prosjektuken satt sammen en film for påsyn av seere. Denne besto av tre sekvenser. Dette er sekvenser som representerer et sammendrag av innspilt råmateriale. Regimessig var sekvensene allerede klippet av produsent på direkten under innspilling, så hver sekvens var utdrag det ferdigproduserte tv-programmet. Presentasjonsfilmen ble 12 minutter og 24 sekunder lang. De 3 sekvensene bruker forskjellige bildutsnitt og har forskjellig innhold. I sekvens A og B ser vi prosjektleder Claus Knudsen lede en samtale mellom rektor Jørn Wroldsen ved Høgskolen i Gjøvik og rektor Gudmund Moren ved Høgskolen i Lillehammer. Knudsen fører samtalen med ledende stikkord som de to rektorene reflekterer over. Innholdsmessig er disse sekvensene forholdsvis like, men regissør Cecilie Giørtz har prøvd å ta i bruk presence-teknologien forskjellig. Denne varierende bruken av presence-teknologien er gjort for å gi seeren innsyn i forskjellige anvendelsesområder for denne teknologien.

I sekvens A vises alle tre aktører i samme utsnitt ved at Wroldsen som er lokalisert på Gjøvik vises på en tv-skjerm mellom Knudsen og Moren som er i studio på Lillehammer (figur nedenfor). Dette er en metode som også blir brukt i ordinær tv-produksjoner i dag.



**Figur 14-13: Screenshot- sekvens A, alle tre aktører i bilde. Knudsen (til venstre) og Moren (til høyre) i studio Lillehammer. Wroldsens (midten) fra studio Gjøvik (TIDE film, 2005).**

I sekvens B blir dette utsnittet ikke tatt i bruk. I stedet velger regissør å bruke et 3-skudd over Morens skulder (figur nedenfor). Man kan da se alle tre aktørene og at de har øyekontakt gjennom mediatoren. Noe som er ment å gi seeren en økt følelse av Wroldsens tilstedeværelse og interaksjonen med andre aktører.



**Figur 14-14: Screenshot- sekvens B, Wroldsen avskåret Moren. 3-skudd over skulder. Størrelsesforholdet mellom rektor og mediator gir ikke likeverd (TiDE film, 2005).**

I sekvens C ser vi to skuespillere, en kvinne og en mann, improvisere en scene fra en romantisk middag. Her bruker regi utsnittet over skulder fra sekvens B (figur nedenfor) sammen med et bilde med wipe-effekt for å representere begge aktører i samme bilde. For bildet med wipe-effekt filmes begge skuespillerne fra siden, med et kamera på Gjøvik og et på Lillehammer. Disse Sammenføres i midten ved en glidende overgang til motsatt bilde (figur nedenfor). Dette krever at man har like kulisser på begge lokalisasjoner. I vårt tilfelle like bord, stoler, duk og bordsetting. Det ferdige bildet med wipe-effekten er det som vises til seer. Innholdsmessig fremstår filmen som 3 innslag fra en vanlig tv-sending og er ment å ha stor gjenkjennelsesfaktor for seere ved å bruke både tv-debatt- og dramatiserings settinger.



Figur 14-15: Screenshot – sekvens C, De to bildene fra studio i Gjøvik og studio i Lillehammer, sammenføyd med wipe-effekt. (TIDE film, 2005).

#### 14.16 Kontakt og likeverdighet

Gjennom programmet prøver man å fremstille aktører med et likeverd selv om de er lokalisert forskjellig. Med likeverd menes her at man er like mye visuelt tilstede eller at man er like godt representert i det felles rom man befinner seg i. Ved produksjoner der alle aktører befinner seg i samme studio vil et slikt likeverd være mye lettere oppnåelig gjennom en balansert bruk av kameravinkler, balanser tid i bildeutsnitt også videre. I denne produksjonen ble bruken av presence-teknologi en tilleggsfaktor i utfordringen. Her kan kommunikasjonskvalitet mellom deltagerne sterkt påvirke hvilke syn seer får på aktørene. I sekvens A og B ser vi likheter med debattprogrammer. Hvis en aktør skulle fremkomme svak eller fraværende på grunn av de teknologiske løsningene som blir brukt vil dette kunne skade vedkommendes sak. Når seerne ble spurt om hvor god kontakt de følte aktørene hadde så vi

en vesentlig forskjell mellom sekvens A og B. 25 % mente at kontakten var god i sekvens A, 64 % mente at kontakten kunne vært bedre mens 3 % mente at kontakten var dårlig. Dette var innslaget der Wroldsen var presentert på en tv-skjerm i studio mellom de to andre aktørene. Seernes oppfatning av aktørenes kontakt tok seg betraktelig opp i sekvens B. Her svarte 50 % at aktørene hadde god kontakt og 50 % at kontakten kunne vært bedre. For de tre aktørene var det ikke gjort noen endring i form for kommunikasjon, men måten det ble presentert for seer var endret. Dette vil si at bruken av 3-skudd for å representere alle aktører kan regnes som en sterk forbedring. Dette gjenspeiler seg også litt i seernes syn på aktørenes likeverd, men riktignok ikke i like stor grad. 32 % mente de to rektorene ikke var likeverdige i sekvens A, mens 17 % mente dette for sekvens B. Altså ser vi en sammenheng mellom hvordan aktørene ble presentert og seerens oppfatning av deres kommunikasjon og likeverd i samtalen. Gjennom undersøkelsen kom det kommentarer på at programleder skulle fokusert mer på Wroldsen som ikke var fysisk tilstede, for å skape et større likeverd. Sitat fra en av respondentene sier "...det var nok ikke tilfeldig at den som var fysisk tilstede kom til orde først og mest". Men det generelle inntrykket var at Knudsens lederform gav debattantene mulighet til å snakke i fulle setninger og at det ikke ble kamp om ordet. Noe som passer godt når en deltager ikke er i studio, men kanskje ikke provoserer nok i et forsøk som dette. I sekvens C brukte man bare bilde over skulderen og tidligere nevnte nøklet bilde fra siden for å vise begge skuespillerne i samme bildeutsnitt. Her gir seerne god respons på både kontakt og likeverdighet for aktørene. 89 % mener at kontakten var god eller medium god. Og 79 % mener at skuespillerne var likeverdige eller ganske likeverdige. Likevel kom det en del kommentarer på at når man ser utsnittet med kvinnen i en TV-skjerm over mannens skulder blir man med en gang veldig bevist på at de ikke er i samme rom. Noe som går ut over inntrykk av både kontakt og likeverd. Under produksjon av sekvens C fikk man ikke tid til å justere størrelsesforholdet skikkelig mellom de to skuespillerne. Regi var ikke fornøyd med justeringene, men tidsplanen tillot ikke å bruke mer tid på dette. Dette førte til at man trodde at seere ville reagere på at kvinnen kunne virke større en mannen i bildet fra siden. Men i henhold til spørreundersøkelsen og kommentarer derfra var det ingen reaksjoner på dette.



**Figur 14-16:** Screenshot - sekvens C, Over-skulder mot kvinne. Likeverdig størrelsesforhold mellom aktørene (TIDE film, 2005).

### *14.17 Konklusjon for seere*

71 % av respondentene følte at det var noe fremmed i måten programmet var laget på. Når de oppdaget at ikke alle aktørene befant seg i samme studio var reaksjonene splittet. Kommentarer i undersøkelsen gir et inntrykk av at det er greit å produsere over distanser i debatt situasjoner som sekvens A og B. Men når man gjør en dramatisert scenen som i sekvens C er reaksjonene mer blandet. Noen syntes det var moro og spennende, andre så ikke helt nytteverdien og ble lettere irriterte over det. Det kom mange ytringer om elementer som ikke fungerte her. Tidsforsinkelse, dårlig bildekvalitet i mediert bilde og ekko var noen av tingene som ble trukket frem. 60 % av respondentene sier at de kan trekke likheter mellom filmen og tv-programmer de har sett før. Så å si samtlige av disse sammenligner sekvens A og B med nyhetssendinger og debattprogrammer. Ingen viser til at de har sett noe de kan sammenligne sekvens C med. Dette er selvfølgelig naturlig i og med at nyhet- og debattprogrammer i årevis har brukt lignende teknologier for å hente inn lyd og bilde av

personer utenfra. Mange registrerte altså at produksjonen var utført på en uortodoks måte og så verdien i at aktørene hadde øyekontakt. Men det var mange som mente at det ikke fungerte å vise bare tv-skjermen fra mediatoren til seer. Flere respondenter oppfattet dette som en middelmådig løsning som ikke holdt mål. ”Jørn på monitoren ble tidvis bare en utydelig masse der det eneste en la merke til var at han blunket energisk med øynene”, skrev en av seerne. Andre respondenter kom med forslag til forbedringer av dette, for eksempel å vise aktøren som ikke var i studio på en storskjerm eller ved projeksjon. Generelt fikk seerne inntrykk av at produksjonsformen fungerte. Deres respons på likeverdighet og kommunikasjon aktører imellom var tilfredsstillende og sekvens A og B ble knyttet sterkt opp til hvordan teknologien kan brukes i debatt og nyhetssammenheng. Hvor spennende det vil være å produsere dramatiseringer med denne typen teknologi var respondentene usikre på. Men det var enighet om at det fungerte i denne sammenheng. Det var enighet blant respondentene om at utsnitt som i sekvens A bør unngås til fordel for 3-skudd i sekvens B. På tross av kommentarer om usynkronisert lyd, dårlig bildekvalitet i mediator og ekko har man lyktes med å holde seeren engasjert. Noe som har gitt en givende og vellykket respons. Kombinasjonen fjernsyn og SVR har vist seg å fungere i de situasjoner vi prøvde og er nå klar for å bli provosert i mer utfordrende settinger.

### *14.18 Drøfting av forbedringspotensialer*

#### 14.18.1 Fysisk

Som rektor Vroldsen påpeker i intervjuet rett etter sesjonene med rektor Moren er selve mediatoren lite ergonomisk utformet. Den er stor og klumpete, og tar veldig mye av oppmerksomheten til brukeren vekk fra den forholdsvis lille skjermen. En faktor som spiller en stor rolle når det gjelder å oppnå transparent teknologi er at bildet blir så klart som mulig, og dette igjen krever en helt ren skjerm, og ikke minst et rent speil. For å oppnå dette trengs andre metoder for rengjøring av speilene. Trykkluft og linsepapir fungerer etter min mening ikke godt nok. Lyssettingen gjorde at problemet ikke var merkbart under gjennomføringen, men i en mer naturlig lyssetting vil nok smuss bli mer fremtredende for brukerne og dermed gi en redusert opplevelse av nærvær.



### 14.18.2 Teknisk

Bildekvaliteten ved kjøring over IP var fortsatt for dårlig selv ved bruk av de nye Tandberg 6000 enhetene. Blokkdannelser og forsinkelser gjorde at deltakerne hele tiden ble minnet på at de benyttet komplisert teknikk for å kommunisere, og illusjonen om et felles virtuelt rom ble satt på prøve som følge av dette. For å oppnå tilfredsstillende resultater er nok en dedikert linje eneste garanti. En slik linje vil kunne gi en jevn datastrøm mellom enhetene, og dermed ikke skape problemer som forsinkelse og hard komprimering. Problemet med en slik løsning er kostnader. En dedikert linje vil bli for dyr for de fleste formål, så det er nok lite realistisk å se for seg dette som en god løsning i nærmeste fremtid. Diffserv. og Intserv. er teknikker som ved prioritering av pakker med medieinnhold kan sørge for en sikker bitrate (QoS). Dette er systemer som vanskelig lar seg realisere, så det er mer nærliggende å tro at codecutviklingen samt stadig raskere linjer vil gi større margin for varierende bitrater. Enn så lenge kan man sørge for å benytte nettverkstier med færrest mulig hopp samt sørge for at alle enheter involvert har støtte for de beste kodekene (i skrivende stund h264/MPEG4 AVC).

Det gikk tydelig frem i undersøkelsen at mediatorbildet brøt illusjonen. Det er vanlig i nyhetssammenheng å vise skjermbilder som en del av scenen men da blir fargetemperaturen og størrelsen på skjermen tilpasset forholdene i studio. Det ble ikke gjort på en tilfredsstillende måte her. Et studiolyt gir typisk en fargetemperatur på 3600 Kelvin på aktørene. Dvs. at energispekteret fra en hvit flate tilsvarer energispekteret til et absolutt sort legeme med temperaturen 3600 Kelvin. Det som kjennetegner dette lyset er at det er varmt gult i forhold til de blå-aktige fargene som blir gjengitt i skjermen (høyere fargetemperatur). For å kompensere for dette burde servicemanualen til skjermen vært fremskaffet i forkant av produksjonen. Med denne er det mulig å finne frem til hvordan fargetemperaturen på skjermen kan stilles til 3600 Kelvin.

### 14.18.3 Fortellerteknisk

Siden dette var første gang teknikken er blitt brukt i TV- sammenheng er det mange faktorer som kunne gjort resultatet bedre. En del av faktorene er helt sikkert ikke avdekket enda, men observasjonene har likevel avdekket noen vesentlige muligheter for forbedring. En stor utfordring i en slik produksjon er å finne gode kameravinkler som ikke forstyrrer illusjonen om nærhet mellom personene på skjermen. Det er viktig å ivareta denne illusjonen både for deltakere for å skape kontakt mellom dem, og for seere slik at de ikke fokuserer på noe annet enn selve interaksjonen mellom personene på skjermen. Teknikken bør skjules slik at dette

ikke kommer inn som et irritasjons – eller forstyrrelsesmoment. Scener som viser en person inne i en TV gjør at teknikken blir unødig eksponert for publikum. Svarene fra spørreskjemaet som seerne fylte ut viser som nevnt tidligere at en stor andel syntes det var noe merkelig med filmen. Skuddet over skulderen til deltakeren på Lillehammer i del c forvirret mange av seerne. Helt fram til det skuddet var illusjonen om at deltakerne befant seg i samme rom opprettholdt, men i det øyeblikk seerne ble presentert en scene der den ene deltakeren befinner seg inne i en TV- skjerm ble denne illusjonen brutt for resten av programmet. Det finnes eksempler på situasjoner hvor det er greit å vise at en person ikke er fysisk tilstede, for eksempel dersom en person rapporterer fra et annet sted, og det skal gå frem at personen faktisk er på det stedet, men i disse tilfellene vil ikke mediatoren være et hensiktsmessig hjelpemiddel.

Skjermen i mediatoren var ellers noe liten og vil ikke kunne gi et inntrykk av likeverdige samtalepartnere uten en DVE-manipulasjon i bildemikseren.. En plasmaskjerm på 50 tommer ville gitt et mer riktig størrelsesforhold men sannsynligvis blitt for kostbart i innkjøp. En god løsning ville vært å bruke projektorer (tilgjengelig på IT-avdelingen) for å lage et bakprojisert mediator-bilde med en tilfredsstillende størrelse. Med to like projektorer, polariseringsfilter og et spesielt lerret vil man også kunne laget et 3-dimensjonalt bilde. Mer om dette i 3D-avsnittet.

Et annet problem med denne produksjonsteknikken er scenekonstruksjonen. Det ble ikke påpekt i seerundersøkelsen fordi det ikke var begrensende i dette tilfellet. Med en split-screen scenekonstruksjon kan skuespillerne ikke krysse hverandre. Midtlinjen er et absolutt skille med et grenseområde der deltakerne vil være halvgjennomskinnelig. En slik begrensning kan føre til et unaturlig samspill der en virtuell berøring ikke kan forekomme. Dybdeeffekten i rommet blir også noe begrenset da aktørene ikke kan overlape hverandre.

En alternativ teknikk for scenekonstruksjon ble utviklet og testet i samarbeid med NRK to måneder senere. Denne piloten, beskrevet i neste kapittel, vil kunne brukes som grunnlag for en etterfølgende SVR-produksjon planlagt i november 2006.

### 14.19 Samproduksjon mellom Høgskolen i Lillehammer og NRK

10.Mars 2006 stilte NRK Hedmark og Oppland sitt nyhetsstudio til disposisjon samt en TOM (Technical Operational Manager) for å prøve ut en alternativ scenekonstruksjon for distribuert samhandling. Forsøket ble gjennomført med kun én masterstudent fra Høgskolen i Lillehammer/NTNU som var undertegnende. Produksjonens kompleksitet begrenset seg deretter. Det var ikke mulig å gjennomføre en produksjon som krever manøvrering av kamera eller instruksjon til deltakerne. Piloten var kun ment å tjene som en teknologi-studie for å forberede neste SVR-produksjon som er planlagt i november 2006. Erfaringene vil også kunne være nyttig for Åpen kanal der man med tiden vil realisere distribuerte og skalerbare møteplasser .Høgskolen i Lillehammer stilte med det samme studioet som ble brukt under den første SVR-produksjonen. Dette ble knyttet sammen med NRK Hedmark og Oppland sitt studio via linjer som allerede var installert mellom de to bygdene etter OL 94.



Figur 14-17: Bildet til venstre viser kontrollrommet og bildet til høyre studio ved HiL.



Figur 14-18: Bildet til venstre viser kontrollrommet og bildet til høyre studio ved HiG.

### 14.19.1 Formål

Vi søker i dette eksperimentet å finne tekniske løsninger på enkelte problemer som oppstod under det første SVR-prosjektet, samt drøfte mulige utbedringer med fremtidig teknologi.

Ønskede utbedringer blir også sett i forhold til hva som er nyttig i en Åpen kanal-produksjon, der deltakerne er fysisk adskilt og i mindre grad kan koordinere produksjonen i forkant.

### 14.19.2 Svakheter ved tidligere benyttet scenekomposisjon

Scenekomposisjonen som bli benyttet var etter min mening den største begrensningen for fremtidige produksjoner av distribuert samhandling. Dette skyldes splitscreen teknikken. At hver aktør har sin begrensede del av skjermen og ikke kan krysse denne uten at illusjonen brytes virker begrensende på samspillet og kunstig for den som ser på. Denne teknikken er også avhengig av at begge studioene har tilsvarende rekvisitter og bakgrunn ettersom disse skal sømnløst limes sammen til ett bilde. Det blir mye logistikk i relaiseringen av et slikt nettverk dersom alle skal behøve å ha tilgang på de samme rekvisittene. Det vil forutsette koordinering av samproduksjonen lang tid i forveien og oppfordrer ikke til spontane debatter som det bør legges opp til i en Åpen kanal. Møter med mange til mange relasjoner har det også vist seg vanskelig å realisere med split-screen teknikken (TiDE-SVR 2005). Dette er en setting som er svært relevant for Åpen kanal ettersom antall deltakere vil variere dynamisk.

### 14.19.3 Forsøk med objektbasert scenekomposisjon

Vi forsøker i dette eksperimentet å vise hvordan deltakerne som objekter i en scene vil frigjøre ressurser til fri ytring og begrense koordineringen av produksjonen. Med objektbasert scenekomposisjon vil man kunne ha en felles bakgrunn som kun må være tilgjengelig for den som setter sammen scenen. Da behøver man heller ikke instruere debattanter til å måtte begrense gestikuleringen til sitt område på skjermen. Når deltakerne ikke trenger å forholde seg så mye til teknologien vil fokuset forhåpentligvis skifte til innholdet de vil formidle.

Seerundersøkelsen viser at mediatorbildet ikke kan integreres i scenen uten at illusjonen brytes. Her vil jeg drøfte hvordan objektbaserte metoder kan gjøre komposisjoner med ”over skulder” bilder enklere samt åpne for multiparts konferanser. 3D-bilder vil med en objektbasert teknikk også enklere kunne realiseres. I eksperimentet benyttes 3D-video kun til kontakt mellom deltakerne for å styrke nærhetsfølelsen. Det kan også være hensiktsmessig å bruke objektbaserte metoder for å realisere flere 3-dimensjonale synsvinkler i en scene (multi

view) ettersom metoden begrenser datamengden som må overføres mellom deltakerne. Teknikkene bak 3D er mer inngående beskrevet under ”3D i samhandling og produksjon”.

#### 14.19.4 Gjennomføring

I studioet på NRK ble det satt opp en grønn duk (green screen) som bakgrunn for deltakeren. En tilsvarende scene ble satt opp på HiL, men der med en blå duk (blue screen) bak deltakeren. Bakgrunnen ble jevnt belyst og duken strekt slik at bakgrunnsfargen skulle inneholde færrest mulig nyanser. Dette er nødvendig for at bildemikseren skal klare å skille bakgrunnen fra forgrunnen (Chroma-keying). Det bildemikseren gjorde var å ertstatte fargen med et bilde som passer inn som en felles bakgrunn for begge deltakerne. Det ser da ut som om de sitter i ett og samme rom til tross for avstanden mellom de er stor, altså en enkel realisering av ”shared virtual reality” (SVR). For at også deltakerne skal få et inntrykk av at de deler rommet (og med det realiserer distribuert samhandling) ble det satt opp en toveis lyd- og bilde-forbindelse mellom deltakerne. I studioet ved HiL kommuniserte deltakeren via mediatoren der han kunne se vedkommende han kommuniserte med i øynene samtidig som han ble filmet fra riktig vinkel (Eye gaze). Dette ble realisert som i forrige eksperiment men her var kameraet bak det halvgjennomskinnelige speilet stereoskopisk. Deltakeren på NRK kunne dermed se deltakeren på HiL i et 3-dimensjonalt bilde. For å realisere det 3-dimensjonale bildet på NRK ble baksiden av en tavle benyttet som lerret. Denne hadde den egenskapen at den ikke endret polarisasjonen til bildet på de to projektorene som lyste på den (metallisk overflate). Projektoren som viste bildet fra det venstre kamera (tilsvarende venstre øye) i det stereoskopiske oppsettet var horisontalt polarisert. Den andre projektoren som viste bildet fra det høyre kameraet (tilsvarende høyre øye) var vertikalt polarisert. Når disse belyser samme overflate er det umulig å skille de fra hverandre dersom det som skiller de ved filtrering ikke bevares. Filtreringen i dette tilfellet var basert på polarisering og refleksjonen fra lerretet vil bevare polariseringen på grunn av den metalliske overflaten (tilsvarende silver-screen). Når deltakeren på NRK tok på seg 3D-brillen med et horisontalt og et vertikalt polariseringsfilter foran hhv høyre og venstre øye ble kun det bildet som var tiltenkt det rette øyet synlig (et bilde som har gått gjennom vertikal og horisontal polarisering vil ikke synes). Det ble med andre ord mulig å se deltakeren på HiL i 3-dimensjoner. Deltakeren på HiL kunne dessverre ikke se deltakeren på NRK i noe annet enn 2D. Dette skyldes mangel på ressurser. Det var vanskelig å få tilgang på ytterligere to like projektorer og et lerret som ikke endret polariseringen. Ideelt sett skulle vi hatt en autostereoskopisk 3D-skjerm i mediatoren der man ikke trenger briller, men dette er for kostbart foreløpig.

### 14.19.5 Infrastruktur

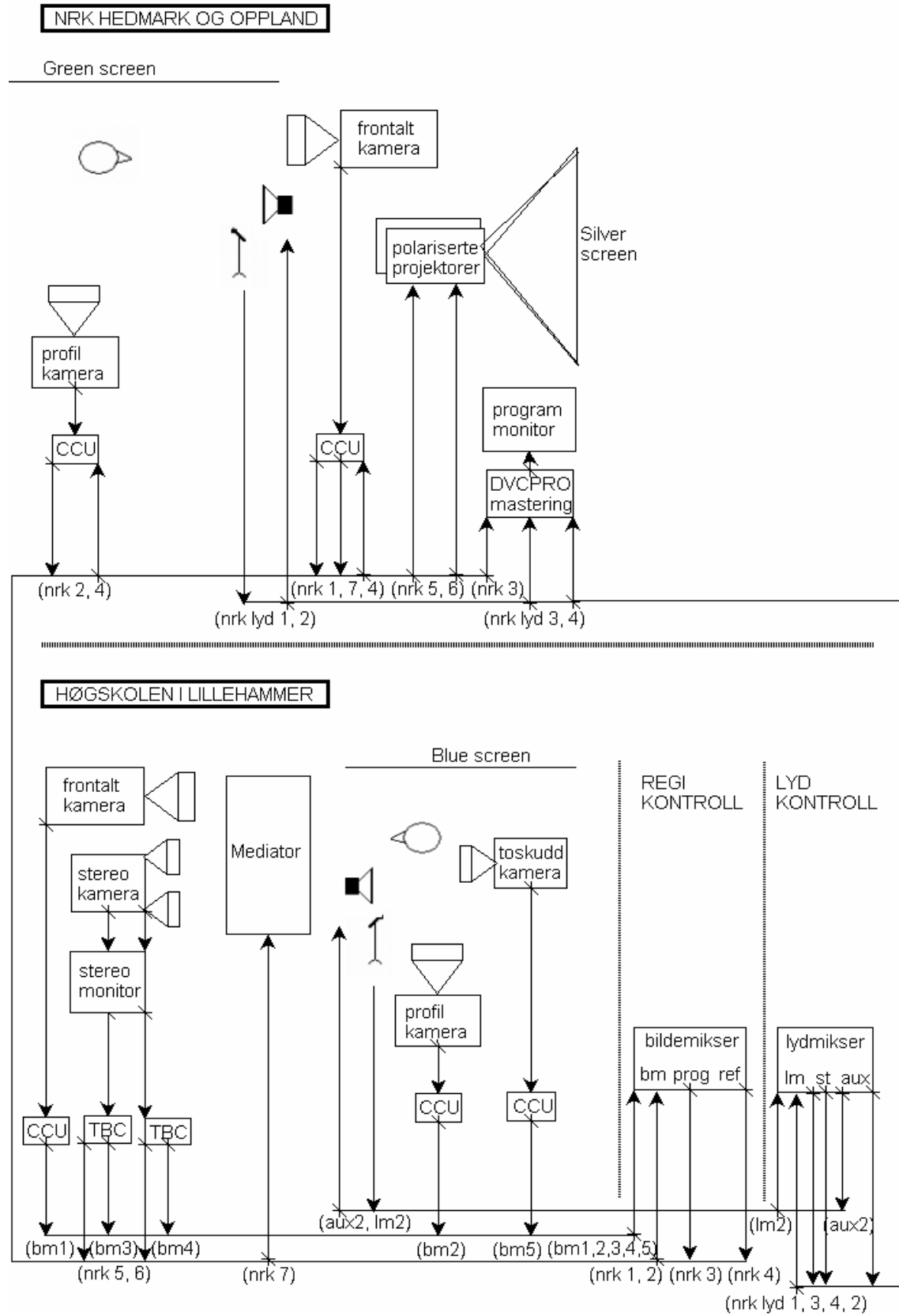
Linjene mellom NRK og HiL besto av 7 coaxial kabler (RG-59) og 4 balanserte lydlinjer (XLR). Vi benyttet samtlige linjer. Fra NRK gikk det kun ett mikrofonsignal som ble tappet ut til studio på HiL. Lyden fra HiL ble tappet tilbake til studio på NRK og den endelige miksen gikk i retur til NRK der det ble gjort opptak av produksjonen på en DVCPRO-recorder. På grunn av de balanserte lydlinjene var støyen på lyden ikke merkbar (Støy som tilføres kablet tilføres begge spenningspotensialer som ligger i motfase og blir dermed kansellert). Bilder som gikk inn i scenekomposisjonen ble overført med SDI (komponent digital video) som gir best kvalitet. Bilder som ble benyttet til kommunikasjon mellom deltakerne ble overført med analog compositt video der luminans og fargeverdier moduleres inn i en og samme kabel. Dette gir bildet noe mer støy, men det var ikke merkbart ettersom skjermteknologien som ble benyttet i begge ender hadde større svakheter. Bak mediatoren på HiL sto som tidligere nevnt et stereoskopisk kamera, men det ble også plassert inn et studiokamera her som kunne gi et bilde med blikk-kontakt som hadde god nok kvalitet til å inngå i produksjonen.

For å korrigere fargebalansen (samt konvertere fra analog triax til SDI) måtte samtlige studiokamera kobles via en CCU (Camera Control Unit) på vei inn i bildemikseren. Dette er viktig for at bildene skal se mest mulig like ut slik at man får en kontinuitet når man klipper mellom dem. Korrigeringen utføres på et eget rom som kalles KK (Kamera Kontroll) hvor man har mulighet til å måle luminans (oscilloscope) og fargebalanse (vectorscope) samt at



Figur 14-19: Sigmund Andresen i KK

man har en god referansemonitor. I hvert studio var det et frontalt kamera og et profilkamera. Det var også et toskuddkamera på HiL som kunne klippes inn for å vise et "over skulder" bilde med NRK-deltakeren i mediatoren (som kommuniserer med deltakeren på HiL). Når man klippet til profilkamera var det mulig å skape illusjonen av at de befant seg i samme rom for i denne vinkelen hadde vi dekning med blå-skjerm/grønn-skjerm i bakgrunnen. Dersom vi hadde hatt nok duk til å dekke sidene ville man også kunnet skiftet til frontalt kamera (samtidig med et skift av bakgrunn som korresponderer med den valgte vinkelen) og opprettholdt illusjonen.

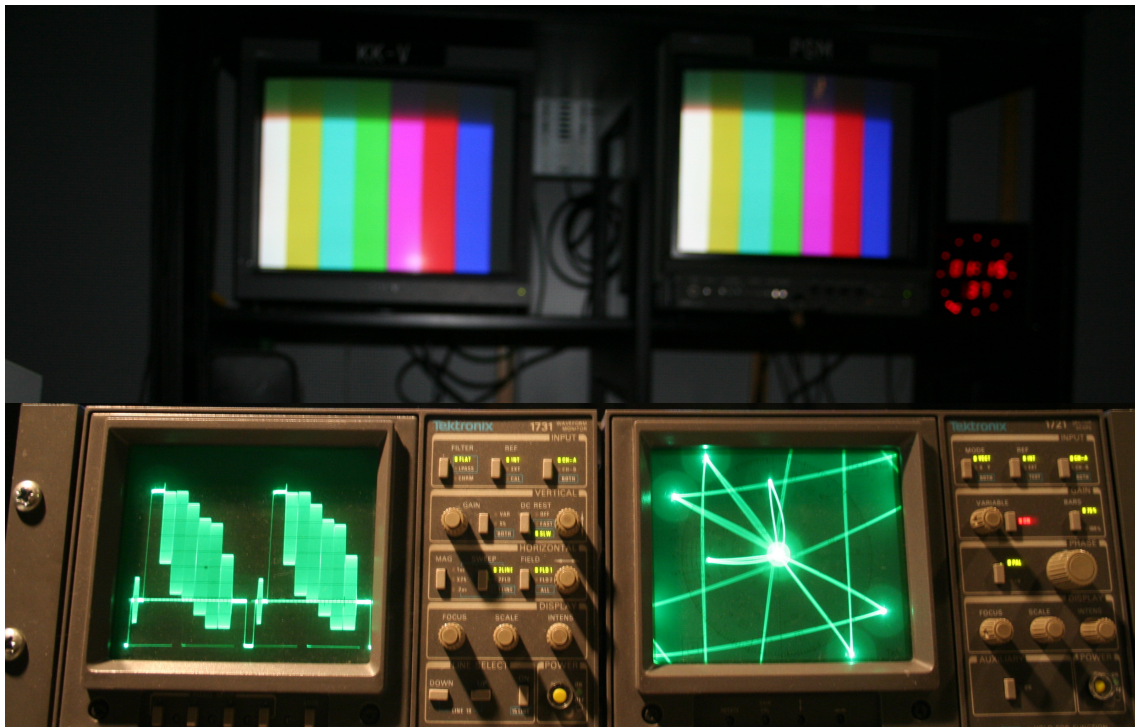


Figur 14-20: Infrastruktur for distribuert samhandling mellom studio på NRK og studio på HiL.



### 14.19.6 Tekniske erfaringer

For å skille bakgrunnen fra forgrunnen (deltakeren på stol) ble det benyttet ulike farger på dukene. Det var tydelig at den grønne veggen var lettere å skille fra forgrunnen enn den blå. Dette har årsak i at den grønne komponenten gir større bidrag til luminansverdien som vi kan se av bildet tatt av en colourbar i KK. På oscilloscopet kan man se spenningsverdien på en horisontal videolinje slik den endrer seg når den forplantes fra venstre til høyre i skjermen. Her ser vi at den grønne kolonnen gir større spenningsverdi enn den blå kolonnen (større andel av det maksimale spenningsutslaget på 1V). Det vil si at den også inneholder mer informasjon (og er mindre følsom for støy) og er et bedre grunnlag for nøkling. Det samme kan vi se av luminansligningen:  $Y=0,3R+0,59G+0,11B$  (Klevsand, 1989) der grønn har den største andelen av bidraget til luminansen Y.



**Figur 14-21: Oscilloscopet (til venstre) viser fallende luminansverdi for fargene fra venstre til høyre som er hvit, gul, cyan, grønn, magenta, rød, blå og sort. Vektorskopet (til høyre) viser fargeverdien (hue) som vinkel (fase) i forhold til de korte vektorene som peker til venstre og rett opp (blackburst har to komponenter som er 90 grader ute av fase på grunn av "Phase Alternating Line" teknikken i PAL). Fargemetningen kan leses ut av lengden på vektoren.**

Oscilloscopet på figuren viser også linjesignalet som lager en liten dip fra 3V til 0V. Denne ligger i forkant av hver horisontale videolinje og forteller når linjen skal tegnes opp. Denne sammen med den høyfrekvente "blackbursten" som følger like etter danner et



videoreferansesignal. Den sistnevnte komponenten er en referanse for fargene som ligger modulert inn over luminansverdien der avvik i fase fra blackburst-frekvensen angir fargeverdi (fargeverdi/hue kommer frem av vinkelen på vektorscopet til høyre i figuren) og amplituden angir fargemetning (fargemetning/saturation kommer frem av lengden på vektoren). Det viste seg i forsøket at vi var avhengig av linjesignalet for å synkronisere kameraene til NRK. Utstyret fra HiL var synkronisert med en felles videoreferanse og utstyret på NRK med en annen (house sync).

Når vi koblet inn profilkameraet fra NRK inn i bildemikseren begynte dette bildet å rulle. Det var mulig på bildemikseren å sette framesync der bildemikseren tilpasset seg dette signalet i stedet for "house-syncen", men da begynte i stedet bildene fra HiL å rulle. Løsningen var derfor å sende HiL sin videoreferanse på en egen linje til profilkameraets CCU på NRK.

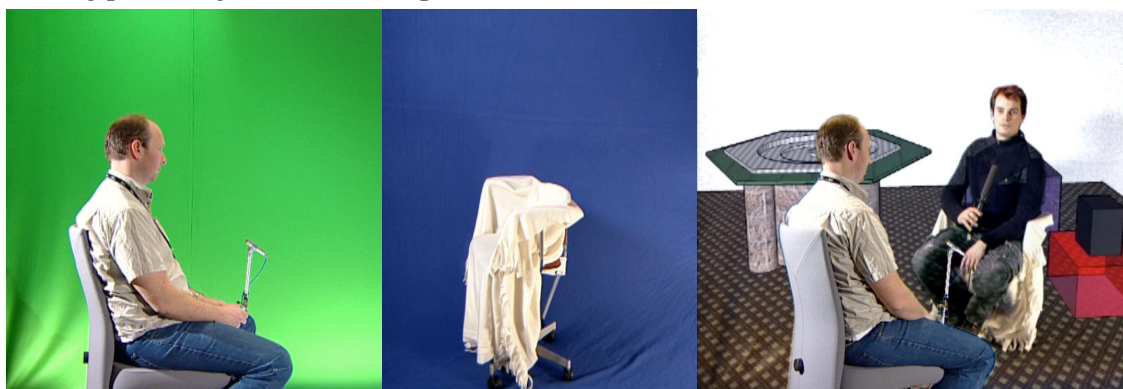
Da var alle kameraene som skulle brukes i miksen i sync. Videolinjene mellom mediatorene måtte ikke synkroniseres ettersom de gikk direkte til hver sin enhet som fant riktig synk for denne ene kilden. I distribuerte samhandlingsproduksjoner hvor det er begrenset med kapasitet, ville man ikke kastet bort en linje til videosynkronisering. Dette kan utføres av eksterne enheter som leser synkroniseringen fra inngående videosignaler og tilpasser denne til house-syncen før signalet går inn i bildemikseren. På HiL kunne dette utføres av en såkalt TBC-enhet, men denne var allerede i bruk for å synkronisere og konvertere composite signaler (fra det stereoskopiske kameraet) i tilfelle dette skulle benyttes i bildemiksen.

For å erstatte bakgrunnsfargen med en felles bakgrunn måtte fargen som skulle gjøres transparent spesifiseres med en cousing på bildemikseren. På Kayak-mikseren har man en joystick som beveges til det punktet der man velger fargen i bildet. Valget måtte utføres to ganger på hver sin key ettersom det var ulike bakgrunnsfarger på det sentrale studioet (HiL) og det distribuerte studioet (NRK). Vi fant at dersom man velger fargen som jevnt over representerer bakgrunnen og helst nærmest forgrunnen får man fine kanter rundt objektet som skal keyes ut. Signalene som keyes (der fargen velges) er key bus og disse tilordnes hver sin key på delegate raden. Det var kun to chroma-keyer (der transparens velges på farge) på denne mikseren og det var key 1 og 2 på ME-raden (Mix Effect-raden). Dette begrenser antall mulige samtidige deltakere fra ulike distribuerte studioer, men løsningen kan skaleres ved å kjøpe eksterne keyere eller investere i en bildemikser med flere ME-rader.

Hvilket objekt som skal ligge over det andre må også velges på bildemikseren. Den keyen som tilordnes høyest prioritet er den som ligger lengst frem i bildet. I vårt forsøk valgte vi å

legge deltakeren fra NRK fremst ettersom vedkomende ikke hadde grønn-skjerm på gulvet og derfor kun var med i bildet fra stolsetet og opp. I scenekomposisjonen er han derfor større og fremstår som lengst frem i bildet, dvs at tilhørende key måtte tilordnes høyeste prioritet..

For å få med hele aktøren på HiL måtte kameraet gå så langt ut at det på sidene av bildet ikke lenger var dekning med blåskjerm. Dette bildet måtte derfor kuttes i sidene (DVE-crop) før det gikk til key-bussen ellers ville det vært områder i bildet fra HiL der bakgrunnen ikke var transparent. Bildet måtte også krympes noe (DVE-size) slik at deltakeren ved HiL fikk en naturlig plassering bak deltakeren på NRK.



**Figur 14-22:** Bilde fra NRK (venstre), fra HiL (midten) og komposisjon med felles bakgrunn (høyre).

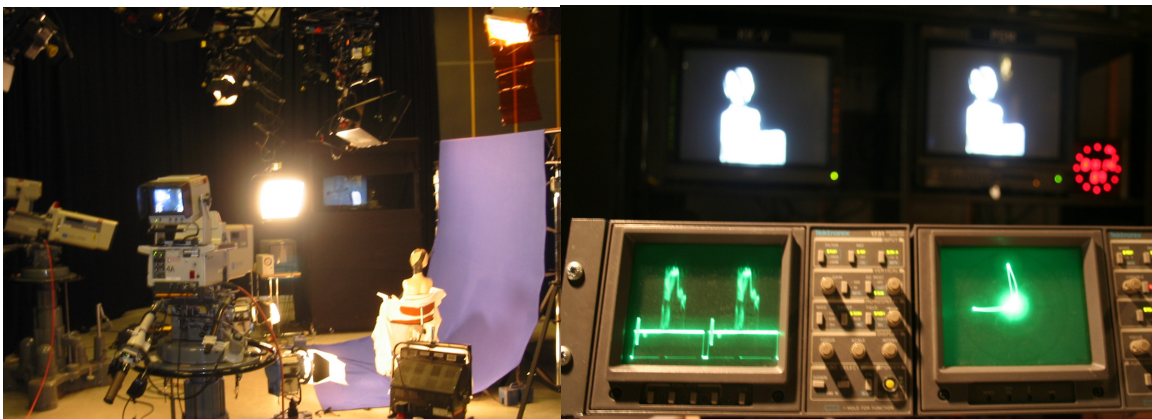
Studio på NRK hadde en grønn-vegg som var svakt belyst, kun med lysstoffrør som omsluttet duken. Dette hadde tre klare fordeler. Det er en billig løsning som lar seg realisere med utstyr de fleste i Åpen kanal vil ha tilgang på. Ettersom lyset er svakt kommer det heller ikke noe grønt gjenskinn på deltakeren og kantene på objektet blir dermed bevart (nøkkelfargen ville gjort kantene transparent). Lysstoffrøret har ellers den egenskapen at det gir god spredning og derfor et jevnt lys på duken. Dette resulterer i at hele den grønne bakgrunnen blir transparent.

For å sjekke om duken var jevnt belyst benyttet vi en funksjon på kameraet som heter zebra. Denne funksjonen vil også eksistere på semiproffesjonelle kameraer som er mer sannsynlig finne i Åpen kanal. Zebra fungerer slik at når lyset overgår en bestemt luminansverdi markeres dette med hvite striper i søkeren. I dette tilfellet satte vi zebra nivået til 75% og flyttet på lyskildene til hele duken hadde denne luminansverdien.

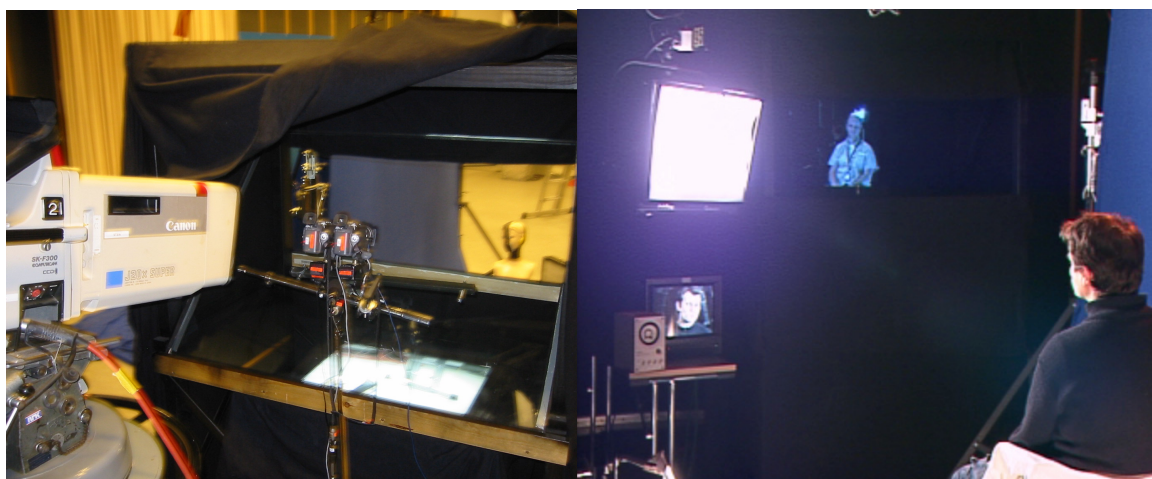
På HiL hadde vi problemer med gjenskinn fra den blå duken ettersom man her benyttet relativt kraftige studioskastere til å belyse den. Dette ble kompensert for med et orange-filtrert spisslys på deltakeren. Det orange lyset (CTO) er komplementært til blått (CTB) og sørger dermed for at den resulterende fargen på lyset bakfra blir hvitt. Kantene på deltakeren ved

HiL ble med dette lyset klarere definert. På bildet nedenfor kan man se den resulterende alfakanalen som danner en maske rundt objektet der det som er sort er transparent.

Delene av objektet som er hvitt er det som blir med i scenen mens grått gir en myk overgang.



**Figur 14-23: Orange-filtrert spisslys øverst til høyre (venstre bilde). Alfakanal i KK (høyre bilde).**



**Figur 14-24: Orange-filtrert spisslys øverst til høyre (venstre bilde). Alfakanal i KK (høyre bilde).**

#### 14.19.7 Fortellertekniske erfaringer

Kommunikasjonen mellom deltakerne var i dette eksperimentet satt opp under ideelle forhold. Det var kun en frame forsinkelse mellom de. Dette skyldes at vi hadde dedikerte coax-linjer og ikke behøvde å komprimere signalet. Følelsen av nærhet burde derfor vært bedre enn i det

tidligere eksperimentet mellom HiG og HiL. Dette var likevel ikke tilfellet på grunn av en lite gjennomarbeidet gulvplan. Ved å flytte på mediatoren nærmere profilkamera og dermed endre kameravinkelen til frontalkameraet ville det vært mulig å oppnå blikkontakt mellom aktørene

samtidig med at man lager et sammensatt bilde av de to. I dette eksperimentet hadde vi ikke fokus på deltakernes følelse av tilstedeværelse, men seerens opplevelse av et delt virtuelt rom ved utprøving av alternative teknologiske løsninger. Flytting av kamera og følgende omrigg av lys ble derfor ikke prioritert.



**Figur 14-25: Steroskopisk projektor (venstre). Frontalt kamera og silverscreen, NRK (høyre)**

En opplagt svakhet med det stereoskopiske bildet på NRK var manglende blikkontakt. For at vedkommende skulle se deltakeren på HiL i 3D måtte han ta på seg 3D-briller. Disse separerer venstre og høyre bilde ved hjelp av lineær polarisering. Brillene sperrer for en del av lyset og gjorde dermed øynene mørkere og vanskeligere å se for deltakeren på HiL. Ved NRK hadde vi heller ingen mediator. Når NRK-deltakeren så på HiL-deltakeren måtte han se i retning av et lerret som ikke var plassert sammen med kamera. Blikkretningen ble derfor heller ikke riktig for HiL-deltakeren. Lerretet med silverscreen-egenskapen hadde dessuten et kraftig gjenskinn som virket forstyrrende. Til tross for disse svakhetene var det tydelig at en 3-dimensjonal avbildning der deltakeren på HiL ble gjengitt i korrekt størrelse og dybde ga en følelse av tilstedeværelse for NRK-deltakeren. Denne følelsen av tilstedeværelse kunne vært gjensidig dersom man hadde hatt en mediator også på NRK og modifisert skjermen i begge ender til å vise et 3D bilde med korrekt størrelse. Hvordan den alternative mediatoren kan bygges opp med enkle midler er skissert i etterfølgende avsnitt.

En interessant erfaring med den objektbaserte scenekomposisjonen var at deltakeren på NRK følte mest for å se på den sammensatte scenen under dialogen. Denne skjermen ble passert ved frontalkamera og sørget for en tilfredsstillende blikkontakt på HiL. Sigmund Andresen inviterte fjernsynsteknikk-klassen til NRK og de fikk prøve seg som deltakere en for en. Det viste seg veldig enkelt og naturlig for deltakerne å finne sin posisjon i scenen ettersom denne ikke var gitt av tekniske begrensninger som med split-screen teknikken. Studentene som så på mente også det delte rommet var troverdig, selv med en kunstig bakgrunn (tegnet i Scetch up). Vi fikk også kommentarer på seerverdien av produksjonen fra fjernsynsteknikk-studentene. De var til å begynne med imponert over mulighetene ved objektbasert komposisjon, men uttrykte at vi ikke har klart å lage en engasjerende setting i selve produksjonen. Studioet var i følge studentene statisk og nærheten til hver enkelt deltaker ikke god nok.

#### *14.20 Videre utvikling av det virtuelle rom*

Med utgangspunkt i studentenes respons er det ønskelig med flere nærbilder i produksjonen. Dette kunne vært løst ved å sette en blå/grønnskjerm på siden av den skjermen som danner bakgrunnen for profilkameraet. Dette ville gitt hver deltaker en virtuell bakgrunn når man klipper til kameraet bak mediatoren som viser et frontalt nærbilde. Den virtuelle bakgrunnen måtte i dette tilfellet vært rotert i henhold til vinkelforskjellen mellom profilkamera og frontalkamera. Med tanke på at bakgrunnen er laget som en 3D-modell ville dette kun medført en ny rendring av tegningen i 3D-programmet som ble benyttet (Scetch-up). Den nøkkelfargede duken som danner bakgrunn for profilkamera behøver ikke være stor og innebærer ingen vesentlig investering. Bildet som blir tatt av frontalkameraet er et nærbilde så den trenger kun dekke et område rundt overkroppen til deltakeren.

Når det gjelder studentenes kommentar om at studioet var statisk er dette en større utfordring å gjøre noe med. Bevegelige kameraer kan løse problemet men forutsetter store investeringer. Den virtuelle bakgrunnen kan i dette tilfellet ikke ligge som faste bilder i bildemikserens minnebank, men må synkroniseres til kameraene som en 3-dimensjonal animasjon.

##### 14.20.1 Bevegelig kamera i virtuelle rom

I dette eksperimentet hadde kameraene fast posisjon og tegningen som representerer studiokulissene kunne derfor stå i ro. Resultatet er en statisk produksjon der verken dybden eller dynamikken i rommet kommer til sin rett. Der man bruker virtuelle rom i dagens TV-produksjoner har man av den grunn bevegelig kamera. Dette kompliserer den tekniske løsningen. Når kameraet beveger deg må bakgrunnen rotere slik kameraet roterer rundt



deltakeren. Står bakgrunnen i ro eller beveger seg feil i forhold til avstanden til deltakeren vil man umiddelbart se at perspektivet er kunstig. Rendringen av den 3-dimensjonale bakgrunnen er ikke spesielt krevende for dagens datamaskiner, men styringssystemene er vanskelig å realisere uten svært kostbart utstyr.



Figur 14-26: Valgdagssending NRK 03, Sensorhode (435VRM), Rendringsenhet (Virtual scenario)

Sensorene som registrerer kameraets bevegelse må kunne formidle verdiene av panorering og tilt til rendringsenheten med god presisjon. Bildet over viser et slikt sensorhode. Det finnes også sensorer for kran (elevasjonsendring) og dolly (kjøring) samt sensorer for avlesning av zoom/fokus verdiene på kamera (RadAmecc broadcast systems, 2006). Dersom kameraet er håndholdt monteres det på sensorer som kan registrere kameraets relative posisjon (IR/mikrobølge med ekko timing) og dreining (evt. med akselerometer) i rommet.

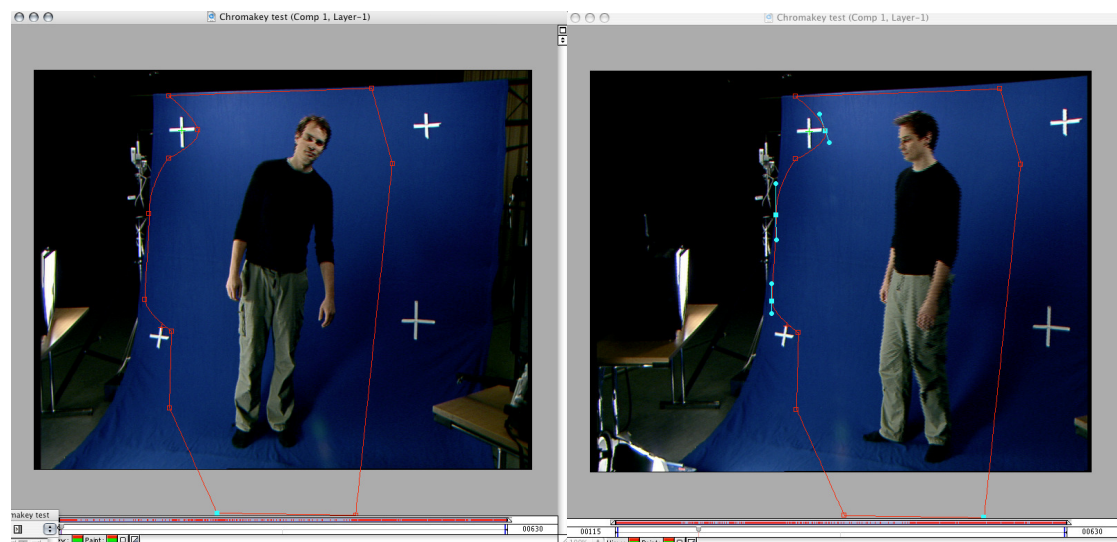
Virtuelle studio har blitt prøvd ut av NRK i valgdagssendingen, den første i 1995. Her har NRK funnet kreative løsninger som selv sjefen for grafisk design og studioutvikling i CNN, David Price, har latt seg imponere av (Propaganda, 2003). I 2003 var Lene Einang Flach prosjektleder for valgdagssendingen. Hun mottok en henvendelse fra CNN der de ønsket å lære av NRK sine metoder. Det de hadde gjort var i utgangspunktet enkelt. Med perforerte glassplater og en grønn belysning på disse, skapte de flere lag i studioet som man kunne gjøre transparente mot resten av studioet. Når man reduserte lysstyrken gikk glassplaten over fra å være en nøkkelfarget grønn-skjerm (med valgtall på) til å bli stadig mer usynlig og dermed avdekke bakgrunnen. Med flere slike moduler var det mulig å gi inntrykk av at objektene oppstod i rommet, og det uten kostbart teknisk utstyr.

Dette er teknikker også deltakere i Åpen kanal kan ha glede av i kommende distribuerte samhandlingsproduksjoner. Men når kameraet også skal bevege seg må man finne smartere

løsninger enn de kostbare sensorteknikkene. Et forslag til en overkommelig løsning er skissert i følgende avsnitt.

## 14.20.2 Bevegelig kamera og utnyttelse av visuelle parametere

Med markører på blå/grønn-skjermen er det mulig for en datamaskin å analysere bildet for deretter å kunne fortelle om kameraets bevegelse. Dette er en teknikk som benyttes i etterarbeid, der materialet allerede er spilt inn. Å utføre slike analyser og samtidig korrigere bakgrunnen for kameraforflytning krever for mye regnekraft til at man med rimelig utstyr kan gjøre dette i sanntid (Commotion som har den raskeste algoritmen renderer med 10 bilder per sekund på en dual G5). Det er likevel en teknikk som bør vurderes ettersom produsentene i Åpen kanal ikke vil ha råd til sensorbasert utstyr. Regnekraft er det som faller raskest i pris, så på det tidspunktet Åpen kanal begynner med distribuerte samhandlings produksjoner er det sannsynlig at det vil finnes programbaserte løsninger for realisering av virtuelle studio.



Figur 14-27: "Garbage mask" med trackere før (til venstre) og etter kamerabevegelse (til høyre).

Vi har utført en test som viser at man oppnår troverdige bakgrunnsendringer med eksisterende programvare. Combustion som ble benyttet er et program som ikke er laget for sanntidshåndtering men som er godt egnet til å prøve ut prinsippene.

Kryssene på blåskjermen er nyttige for å kunne analysere kameraets bevegelse, men kan ikke være en del av keybussen, dvs bildet vi plukker ut bakgrunnsfargen fra. Dersom disse markørene blir med vil det oppstå et kryss i bakgrunnen ettersom forgrunnen ikke er transparent over hele området (krysset har ikke nøklingsfarge). I eksperimentet med bevegelig kamera klistret vi på kryss i hvert hjørne av blåskjermen. Deretter sørget en

avrundet "garbage mask" (markert rødt på bildet over) for å få kastet den delen av bildet som var utenfor masken (her: kryss i kantene av blå-skjermen). Det var likevel mulig å sette en tracker på kryssene. Disse trackerne ble så satt til å styre bakgrunnsbildet. Garbage masken følger også bevegelsen av kryssene som man kan se av blåskjerms-bildet over.



**Figur 14-28: Komposisjon før (til venstre) og etter kameraforflytning (til høyre).**

Bakgrunnsbildet må ha høyere oppløsning enn videobildet ettersom det skal bevege seg inn og ut av bildrammen og derfor bør ha tilsvarende oppløsningen på det utsnittet som blir vist. 3D-tegningen av den virtuelle studioscenen, som vi har benyttet tidligere, hadde samme oppløsning som videobildet og ble derfor erstattet av et høyoppløselig bilde av en skog. På bildet der bakgrunnen er satt inn viser den røde linjen kameraets bevegelse. Alle panorering og tilt bevegelser var troverdige, men zoom fungerte ikke like godt. I komposisjonen så det ut til at vi fikk en perspektivendring. Dette kan skyldes at de fokale egenskapene i linsa ikke ble satt inn som parametere i algoritmen.

Skyggen som kommer frem i bakgrunnen var ikke mulig å fjerne i etterarbeidet uten å tegne masker bilde for bilde, noe som ville vært alt for tidkrevende. Den viser at det er viktig med jevn belysning på blå-duken (Nøklingsfargen bør begrenses i sitt fargespekter).

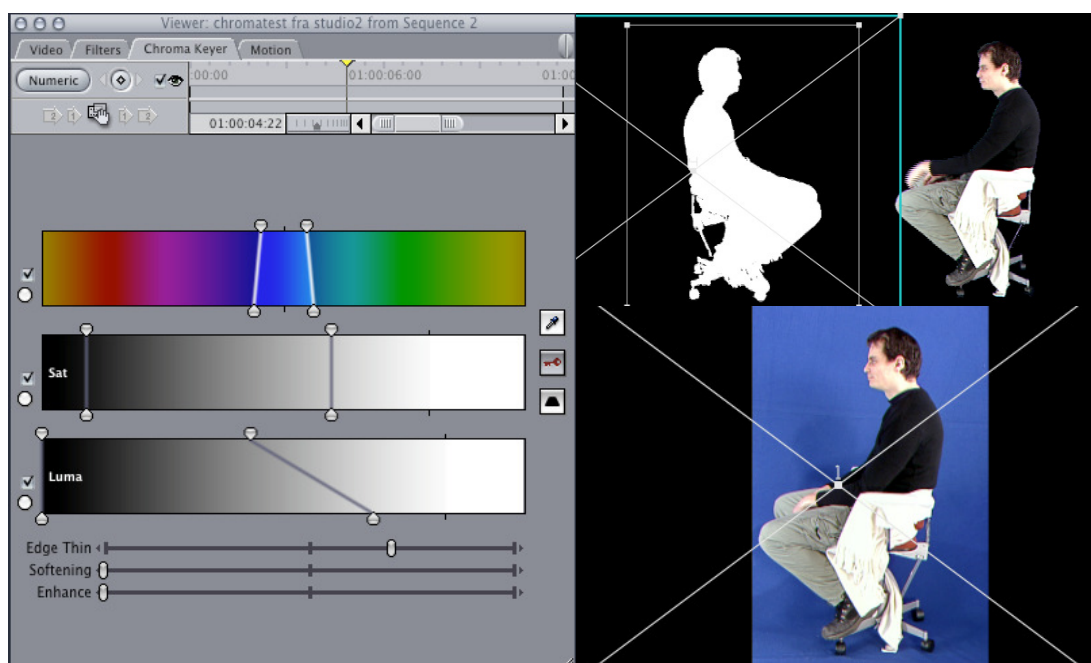
Den feilen som til å begynne med var mest merkbar var fargeforskjellen mellom det relativt kjølige bakgrunnsbildet og det varme studiolyset. I Combustion er det mulig å kopiere fargepaletten fra bakgrunnen til forgrunnen. Dette sammen med at lysretningen på studiolyset ble tilpasset lysretningen i bakgrunnen ga en troverdig komposisjon.

### 14.20.3 Virtuelle studio med rimelig software

Dersom man har mulighet til sette sammen den objektbaserte scenen i ettetid (ikke live), finnes det rimeligere løsninger for distribuert samhandling enn de hardwarebaserte. Dette er



sannsynligvis mer relevant for Åpen kanal sine deltakere i dag. Redigeringsprogrammer som allerede er i bruk blant produsentene har støtte for blue/green-screen nøkling og kan med en plugin også håndtere bevegelig kamera (motion tracking). Det er mulig å spille inn materiale på de ulikt plasserte studioene og kun ha mediator-lignende kommunikasjon via videokonferanseenheter. Deltakerne er ikke avhengig av å se den endelige komposisjonen. Det viktigste er at de er synkronisert og føler en nærhet til hverandre. Samspillet kan gjenskapes som om det foregikk i samme rom i ettertid. Dette stiller også mindre krav til linjene mellom de. Materialet i broadcast-kvalitet som skal inn i komposisjonen trenger ikke å overføres i sanntid. Det er kun videokonferanseenheter som må kommunisere i sanntid. Disse kan oppnå en tilfredsstillende kvalitet (TiDE, 2005) med 768kbit i H.264 som deltakerne kan få til på ADSL-linjer. Disse antagelsene har motivert til eksperimenter med scenekomposisjon på det mest brukte redigeringsprogrammet, Final Cut Pro.



Figur 14-29: Parametere for chroma-keying (til venstre), crop av kilde (nederst), alfamaske (øverst)

Som i eksperimentet med objektbasert scenekomposisjon i sanntid fikk vi her problemer med at blåskjermen ikke var tilstrekkelig stor. Når kameraet gikk ut i vidvinkel for å avdekke hele deltakeren ble sidene av bildet uten blå duk. For at forgrunnen skulle være transparent også i dette området måtte nøklingsbilde (forgrunnen med deltakeren foran blåskjerm) kuttes i

sidene. Til dette benyttet vi en funksjon som heter crop (sortert under videofilter av typen matte).

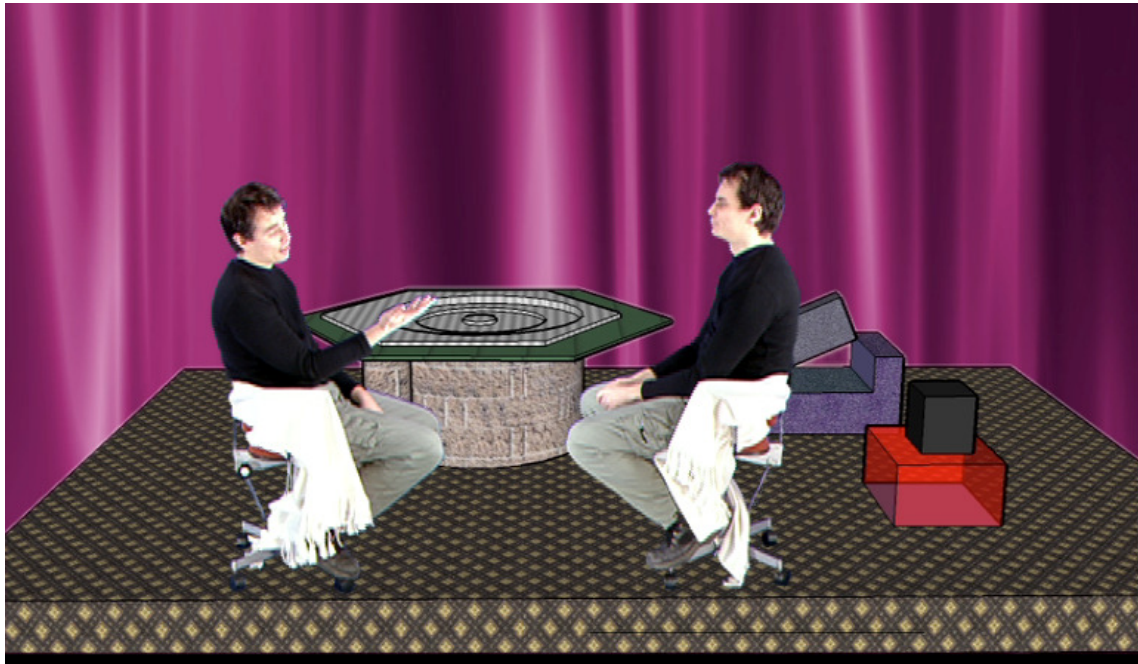
For å gjøre den blå bakgrunnen transparent gikk vi inn på et filter som heter "Chroma key" og valgte fargen vi ønsker å gjøre gjennomsiktig (her blå bakgrunn) med en pipette på bildet. Vi kunne se av alfamasken (kommer frem ved å trykke på nøkkelen) at det var enkelte hvite flekker rundt deltakeren. Disse områdene er ble ikke transparente på grunn av variasjoner i den blå nøklingsfargen (ujevn lyssetting). For at keyeren skal akseptere disse variasjonene (og dermed gjøre hele bakgrunnen transparent) måtte vi trekke nedre og øvre grense fra hverandre både i fargemetningsspekteret ("Sat" på bildet) og lysintensitetsspekteret ("Luma" på bildet). På grunn av et blått gjenskin fra bakgrunnen var det tydelig, spesielt i håret, at hele deltakeren ikke var fri for transparens. Ved å øke verdien for "Edge Thin" ble overgangsområdet mellom sort og hvit alfa-verdi redusert. Dette ga et klarere definert omriss av deltakeren.

Omrisset ble likevel ikke like klart definert som i forsøket der vi benyttet en bildemikser til å utføre separasjonen. Bildemiksen (Kayak DD) har svært kostbare kretser for å utføre dette og er dermed en bedre forutsetning for et godt resultat. Hovedkilden til problemet viste seg likevel å være kildeformatet som ble benyttet i det siste eksperimentet.

#### 14.20.4 Utfordringer med DV og HDV-formatet i chromakeying

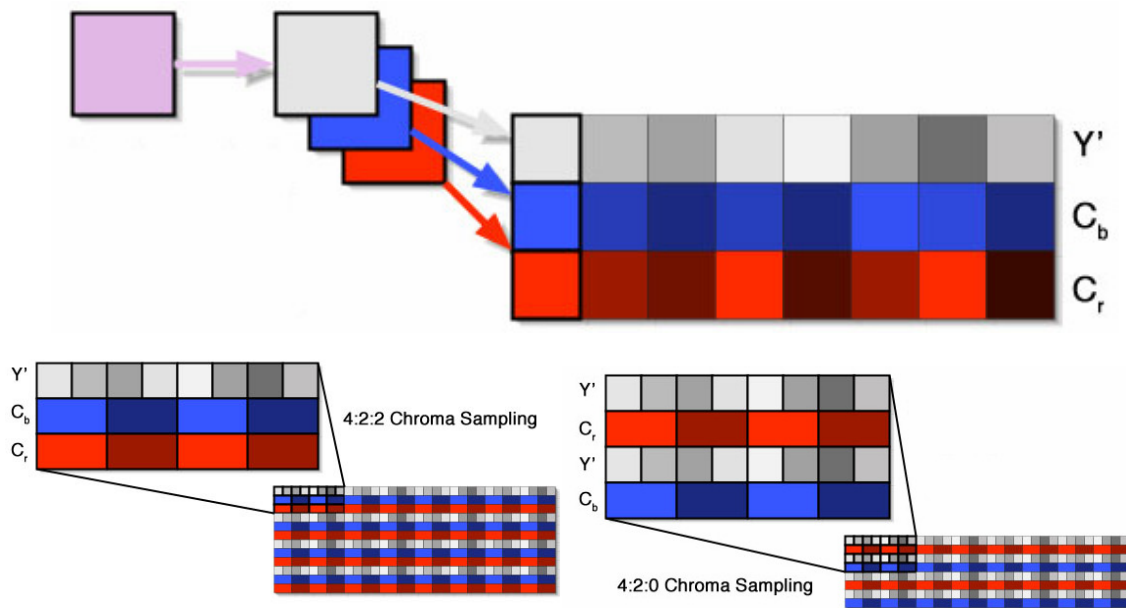
Opptaket i eksperimentet ble gjort på DV (4:2:0) ettersom dette er formatet som er mest brukt av Åpen kanal sine deltakere. Når vi i første omgang overførte kilden direkte til redigeringsenheten med firewire beholdt den sin opprinnelige kvalitet. Den viste seg mindre heldig til chromakeying. Det fremkom tydelige blokker i omrisset av deltakeren, noe vi antok hadde sammenheng med fargeoppløsningen på DV-formatet. Ved senere å konvertere kilden til et format med høyere fargeoppløsning (4:2:2) fant vi at man kunne oppnå tilsvarende kvalitet som i bildemikseren.

Konverteringen ble utført hardware i en DVCPRO-spiller som konverterte fra det komprimerte formatet DV til det ukomprimerte formatet på SDI utgangen. For å ta inn et SDI-signal må man ha et spesielt redigerings/grabber-kort (her: Decklink Blackmagic SD kort til ca 5000kr). Denne prosessen ("upresing") kan også utføres i software med en plugin til FinalCut Pro som heter G Nicer. Kvaliteten man oppnår med denne er på det beste lik det man får ut av en hardwarekonverterer (Nattress, 2005).



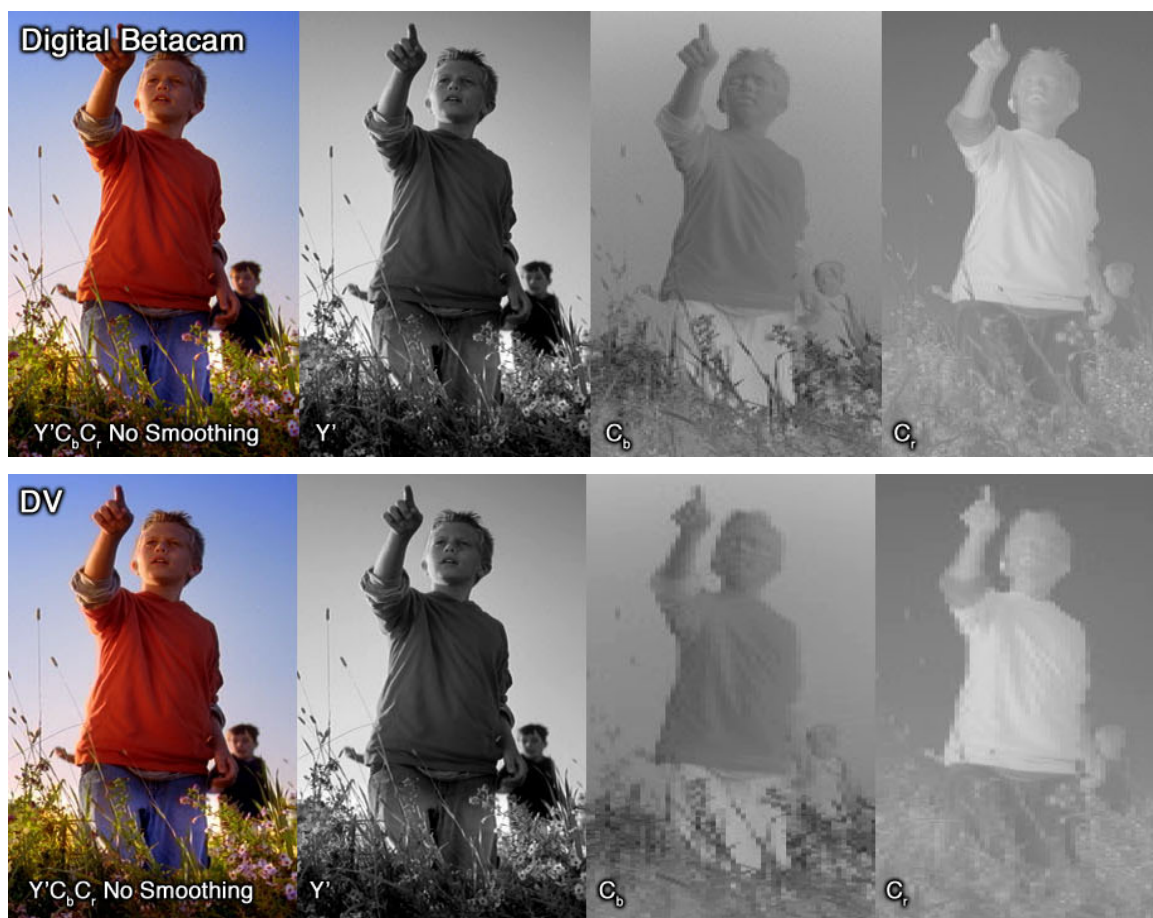
Figur 14-30: I debatt med meg selv, endelig komposisjon ("med upresing") utført etter opptak.

Fra et ingeniørteknisk synspunkt kan det virke ulogisk at man kan oppnå økt kvalitet ved å skape informasjon fra noe kildematerialet ikke har ivaretatt. Tapet av informasjon kommer som en følge av fargesamplingsteknikken i DV (og HDV formatet) der fargeinformasjonen ikke blir lagret fra samtlige piksler i CCD'en. Noe har dermed gått tapt ved lagring på tape.



Figur 14-31: Fargesampling uten reduksjon 4:4:4 (øverst), fargereduksjon 4:2:2 og 4:2:0 (nederst)

4:4:4 (f.eks HDCAM SR) heter det fargesamlingsformatet som innebærer samme oppløsning på lumakomponenten ( $Y'$ ) som på fargekomponentene ( $C_r$  og  $C_b$ ). Fargefølsomheten (tappene) er mindre enn lysfølsomheten (stavene) i øyet så det er mulig å redusere denne datamengden uten at det vil synes. I digibeta og DVCPRO50-formatene lagres kun fargeverdien i annenhver kolonne av piksler (4:2:2). I DV-formatet lagrer man også farge i annenhver piksel-kolonne, men man lagrer kun rød ( $C_r$ ) eller blå ( $C_b$ ) komponent vekselvis for hver rad som blir samlet (4:2:0). Man reduserer dermed informasjonsmengden ytterligere, men i det sistnevnte tilfellet blir det synlig at bildet mangler noe. Spesielt synlig er dette hvis omrisset er definert av en farge slik som ved nøklingen av deltakerne i eksperimentet.



Figur 14-32: Fargekomponenter i 4:2:2 fargesamling (øverst) og 4:2:0 fargesamling (nederst)

I en ”upresing”-prosess prøver man å gjenskape det som sannsynligvis var der med en interpoleringsalgoritme. Denne tar et gjennomsnitt av omkringliggende fargeverdier og analyserer forholdet til luminansverdien i området der man mangler fargeinformasjon.

Luminansverdien er ivaretatt for samtlige piksler og antas koherent (i sammenheng) med fargemetningen der informasjonen om dette mangler. På denne måten kan man gjenoppbygge fargeverdiene slik de sannsynligvis har vært før de ble kastet. Det gir en mer korrekt fargegjengivelse enn om samme fargeverdien skal gjelde for hele området der man ikke har fargeinformasjon. Uten denne algoritmen vil bakgrunnen spre seg inn som store blokker i omrisset av deltakeren vi forsøker å nøkle ut. Chromakeying utfører separasjonen av bakgrunn fra forgrunn ved hjelp av chromaverdiene (farge) og er avhengig av at disse er mest mulig korrekt gjengitt. En ”upresing” fra 4:2:0 til 4:2:2 medfører en reell høyere fargeoppløsning. Det beste resultatet oppnår vi likevel dersom opptaket ble gjort opprinnelig i 4:2:2 (dobbel fargeoppløsning), men dette forutsetter dyrere utstyr enn hva Åpen kanal sine deltakere vanligvis har tilgang på.

#### 14.20.5 Semiprofesjonelle HD-kameraer egnet for chromakeying.

Støtte for høyoppløselig video av typen 1080i eller 720p har vært vurdert av NTV, kommende konsesjonær for Åpen kanal sitt sendenett. 1080i er det formatet med høyest oppløsning (1920x1080). Her sendes 50 delbilder (fields) pr sekund. Delbildene tegner odde (upper field) og partallsnummererte (lower field) horisontale linjer i et bilde vekselvis. To delbilder (fields) danner en fullverdig bilderamme (frame) som vises 25 ganger pr sekund. Denne teknikken kalles interlacing og er ikke det som er anbefalt av EBU (European Broadcast Union). Årsaken til dette er at når man skal vise bildet på en skjerm som ikke er basert på interlacing slik som Plasma eller LCD må deinterlacing utføres i set-top boksen hos alle TV-seere. Dette er en prosess der delbildene settes sammen til helbilder. Kvaliteten på deinterlacing-kretser er svært varierende og innfører en unødvendig kostnad i set-top boksene som NTV skal subsidiere. 720p er derfor et mer sannsynlig valg for distributøren av Åpen kanal. Her er hvert bilde et helbilde (progressivt), men oppløsningen er noe lavere (1280x720). Dette muliggjør også tilfredsstillende kvalitet ved en lavere bitrate enn ved transmisjon av 1080i. For HD-sending vil man ikke benytte MPEG2-codecen som i dag, men H.264 (MPEG4 AVC) som er en mer bit-effektiv komprimeringsalgoritme. HD sendinger på bakkenettet er planlagt i 2009. NRK har allerede kjøpt inn HD-utstyr og utført produksjoner med OB-bussen HD1 som i følge Sony Norge skal ha kostet 40 millioner. Her benyttes kamera av typen HDCam SR som har et prisnivå som er uoppnåelig for deltakerne i Åpen kanal. De høyoppløselige videoformatene vil likevel måtte vurderes med hensyn til fordelene det gir i distribuert samhandling. En høyere oppløsning vil innebære klarere definerte kanter på deltakerne som

skal filmes og dermed en mer presis chromakeying. Med HDCam SR er denne presisjonsøkningen gitt direkte av forholdet mellom den gamle oppløsningen (720 x 576) og den nye (1920 x 1080) ettersom fagerkomponentene og lyskomponentene (luma) har samme oppløsning (4:4:4).

Dette vil ikke være tilfellet for de rimelige HD-formatene der oppløsningen på fargene er redusert (4:2:0 eller 4:2:2). Fargeoppløsningen er likevel vesentlig høyere enn det mye brukte DV-formatet og 4:2:2 regnes for å være tilfredsstillende for chromakeying. På sikt vil rimelige HD-formater som HDV (høyere oppløsning, men native kun 4:2:0) og P2 (4:2:2) sannsynligvis være det som tar over for DV. I denne serien finnes også kamera med støtte for progressivt opptak (JVC og Panasonic), men 720p formatet som benyttes gir en generelt lavere oppløsning enn 1080i (Canon og Sony) og det gjør de mindre egnet til chromakeying. Formater med 4:2:0 fargesampling er heller ikke egnet til chromakeying men det finnes unntak. Canon har et HDV-kamera (1080i), Canon XL2, til ca 60.000 kr med avtagbar linse og en HD-SDI utgang. Ved å benytte HD-SDI utgangen kan dette kameraet fungere som et fullverdig 4:2:2 kamera selv om det som lagres på tape er 4:2:0. Det som kommer ut av denne utgangen er ikke "upreset" men i sitt originale format 4:2:2. For å kunne nyttegjøre dette signalet må man ha et grabberkort med støtte for HD-oppløsning og terminaler av typen HD-SDI. Med den vil PC-en (eller Mac-en) fungere som en fullverdig opptak og redigeringsenhet for HD 4:2:2. Det rimeligste profesjonelle kortet i skrivende stund er Decklink Blackmagic HD til ca 10.000kr.

Dersom man skal ha mulighet til å lagre høyere fargeoppløsning på et portabelt medium finnes det en rimelig HD-løsning fra Panasonic. De har et kamera til ca 50.000 kr som lagrer 100Mbps DVCPROHD (4:2:2) på PCMCIA-baserte minnebrikker (P2). Med en så høy bitrate vil man fort trenge mange slike brikker og disse er dyre (ca 500 kr pr GB). Kameraet har riktignok en mulighet til å lagre på tape (ca 1 kr pr GB), men kun i lavoppløselig DV (4:2:0). Skal man gjøre opptak over lengre tid må minnebrikkene tømmes på en harddisk (ca 4 kr pr GB) eller man må benytte en ekstern portabel disk som lagrer mediedata direkte fra firewireutgangen. Ettersom kameraet ikke har HD-SDI utgang er Canon løsningen å foretrekke i studio sammenheng i dag. Den vil kommunisere bedre med konvensjonelt studioutstyr og muliggjør live distribuert samhandling basert på chromakeying. For reportasjebruk vil HDV opptak på tape gi tilfredsstillende kvalitet til en lav lagringskostnad. Denne muligheten har man ikke på Panasonic. Panasonic løsningen kan likevel vinne frem ettersom lagring direkte på disk faller ned mot prisen av lagring på tape. Diskbaserte løsninger



fordrer også en mer effektiv arbeidsflyt ettersom man ikke trenger å spille over materialet fra tape før redigering. Ved å koble harddisken på redigeringsenheten har man direkte tilgang på materialet rett etter opptak. Det eneste problemet for live distribuert samhandling består i at man ikke har HD-SDI utgang på Panasonic-kameraet. Dette er et mindre problem ettersom det allerede i dag finnes billige DV til SDI konvertere. Sannsynligvis vil det komme tilsvarende bokser for de høyoppløselige formatene som er i bruk i semiprofessionelle miljøer der man stadig søker mer kostandseffektive løsninger. Kameraene som er foreslått her er blant de mest kostbare av de semiprofessionelle HD formatene. Det selges i dag HDV kameraer helt ned i DV-pris. Det fysiske minste HDV kameraet til Sony ligger på ca 16000 kr<sup>50</sup>. Den synkende prisen gjør HD tilgjengelig for stadig fler produsenter i Åpen kanal, men de billige modellene har ikke egnede egenskaper med tanke på chromakeying (Mangler 4:2:2 lagrings- og utgangsmuligheter).



Figur 14-33: HDV kamera, Canon XL2 (til venstre). P2 kamera, Panasonic HVX200 (til høyre)

#### 14.20.6 Alternativer til chromakeying

I skrivende stund finnes det ikke noe godt alternativ til chromakeying når man skal ta objekter (her: deltakerne) ut fra bakgrunnen i sanntid. Chromakeying er mulig å få til med kostnadsrammen deltakerne i Åpen kanal er bundet av. Men chromakeying er ikke en enkel løsning som eksperimentene har vist. Spesielt vanskelig har det vært å få til en jevn lyssetting

---

<sup>50</sup> Samtlige priser er hentet fra Video 4, leverandør av proffesjonelt videoutsyr (<http://www.video4.no>).

og omrisset av objektene blir dårlig definert med DV-kameraene som deltakerne i Åpen kanal benytter i dag (på grunn av 4:2:0 fargesamplingen).

### **Lumakeying**

Det enkleste alternativet i dag er Lumakeying. Her separeres objektet fra bakgrunnen ved at bakgrunnen defineres som en lumaverdi (lysintensitets-verdi) fremfor en chromaverdi (fargeverdi) som i eksperimentet. At man benytter lumaverdien til separasjonen er heldig for DV formatet ettersom oppløsningen på denne komponenten ikke er redusert. Omrisset av deltakerne bør derfor i teorien bli klarere definert med utstyret Åpen kanal deltakerne benytter i dag. Denne keyingteknikken finnes det støtte for i rimeligere bildemiksere og dersom man klarer å gjøre bakgrunnen helt svart eller helt hvit kan det fungere. I forkant av samarbeidsprosjektet med NRK testet jeg ut teknikken, men uten hell. For at bakgrunnen ikke skal synes gjennom forgrunnen kan forgrunnen ikke ha helt hvite eller svarte partier. Dette begrenser forgrunnes dynamiske område og er svært vanskelig å få til i praksis. En helt hvit bakgrunn gir reflekser og vanskelige lysforhold for kamera, mens en helt sort bakgrunn må kompenseres for ved unaturlig lyse forgrunnsobjekter. Teknikken benyttes av disse grunner ikke til objektbasert scenekomposisjon.

### **Differansekeying**

En keying teknikk som kan hente ut objekter fra en naturlig bakgrunn ville vært en ideell løsning for Åpen kanal sine deltakere. Da ville man ikke hatt behov for stor kunstige elementer i studio som en blå-/grønnskjerm. Skjermen er forstyrrende og tidkrevende å lyssette. Med avansert bildeanalyse er det mulig over tid å plukke ut bakgrunnen ettersom denne står i ro. ”Difference matte”, eller differansekeying på norsk, er en slik teknikk.

Differansekeying er etter min mening et lovende alternativ til keying teknikkene som her vært vanlig å benytte frem til i dag. Det finnes rimelige software baserte løsninger som kun er tilgjengelig i nyere versjoner av etterarbeidsverktøy. Teknikken ble prøvd ut i versjon 7 av Adobe After Effects og jeg fant at resultatet var tilfredsstillende enn så lenge kameraet ikke var i bevegelse. Årsaken til dette er at teknikken separerer objekter fra bakgrunnen ved å analysere endringen fra foregående bilde til det neste. Der det ikke har vært noen endringer antar algoritmen at det er en bakgrunn. Denne antagelsen holder kun dersom kameraene ikke beveger seg. Da vil også bakgrunnen endre seg og denne blir feilaktig definert som en del av forgrunnsobjektet. Algoritmen vi testet vurderer globale bevegelser som en del av bakgrunnen



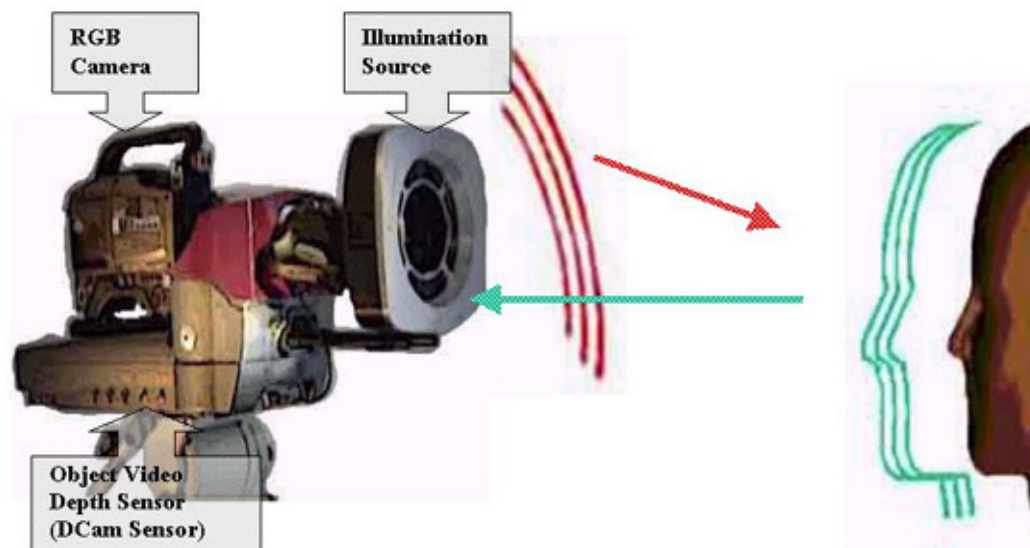
og kan på den måten håndtere kameraforflytninger, men i omrisset av objektet fikk den problemer. Dette skyldes at objektet dekker til bakgrunnen og en del av vurderingsgrunnlaget for global bevegelsestimering blir borte. Med stereoskopisk avbildning kunne denne algoritmen fått indikasjon på dybden av objektet og avdekket større områder av bakgrunnen (gitt et klarere definert omriss av objektet). Slike algoritmer er under utvikling hos Hewlett Packard og er mer inngående beskrevet under ”3D i distribuert samhandling”. HP sin løsning er foreløpig en prototyp som klarer 15 bilder pr sekund og er tiltenkt video konferanse. Utover denne har jeg ikke funnet produkter som håndterer en separasjon basert på differansekeying i sanntid. Teknikken var dermed ikke mulig å prøve ut i live distribuert samhandling.

### **Dybdekeying**

Differansekeying forutsetter avanserte algoritmer som det vil ta tid å utvikle. Spesielt vil det være vanskelig å for denne teknikken å håndtere kamerabevegelse noe vi har sett må til for å gi et virtuelt studio en nødvendig dynamikk. En teknikk som kan utvikles med enkle midler og som kan håndtere kamerabevegelse er derfor mer interessant for Åpen kanal i dag.

Teknikken bør dessuten kunne skille bakgrunnen utifra naturlige egenskaper og ikke være avhengig av kunstige elementer for å skape en separerbar bakgrunn (f.eks en blå-/grønnskjerm).

Deltakeren/objektet skiller seg fra bakgrunnen med en parameter som kan hentes ut mekanisk uten bruk av avansert bildeanalyse. Dette er dybden. I de aller fleste tilfeller sitter deltakeren nærmere kameraet enn studioveggene. En enhet som måler avstanden til kameraet vil dermed kunne fortelle hva som er forgrunn/deltaker og hva som er bakgrunn.



Figur 14-34: Prinsippskisse for dybdeavlesning

3DV Systems har utviklet et system (Zcam) der man henter ut dybdeverdien ved hjelp av en sensor som monteres på kameralinsa (3DV, 2006). Sensoren fungerer som et ekkolodd der den måler tiden det tar for et signal å reflekteres fra overflaten til objektet/deltakeren og sammenligner dette med tiden det tar før signalet reflekteres fra bakgrunnen i studioet. Objektet som er nærmere kameraet vil ha kortere gangtid og defineres dermed som nærmere. Et område av de nære dybdeverdiene settes til verdien 1 (helt hvit) og de øvrige dybdeverdiene settes til verdien 0 (helt sort) i alfakanel til video signalet. Ettersom objektet/deltakeren befinner seg i det nære området vil denne få verdien 1 i alfakanalen og dermed synes i videobildet. Bakgrunn (og forgrunn) som ikke ligger i det definerte nærområdet vil få verdien 0 i alfakanalen og dermed bli transparent (ikke en del av videobildet). Dette er en svært enkel løsning som ikke forutsetter komplisert lysrigging i studio eller kostbart bildeanalyseverktøy. Det eneste man trenger er en bildemikser som kan ta inn alfakanaler og gjøre RGB-signalet transparent. Dette er noe de fleste bildemiksere har mulighet til (Alternativt kan man benytte keyer-terminalene i tekstgeneratorer).



**Figur 14-35:** Kun den andelen av scenen som har korrekt dybde blir nøklet ut (alfakanal til høyre)

Alfakanelen kan også ta andre verdier enn 0 og 1. I dette tilfellet er ikke det nyttig ettersom man kun ønsker å definere omrisset av objektet og ikke vil ha semitransparente områder på deltakeren. Men dersom man tillater verdier mellom 0 og 1 og lar alfakanalen fortelle om det enheten faktisk tar en måling av (dybden) vil denne kanalen kunne brukes til å generere et 3D bilde (i tillegg til å separere objektet fra bakgrunnen). Streamer man alfakanalen sammen med RGB-dataene (Støttet av formatet Quicktime 16M+) har man mulighet til å overføre separate deltakere i 3 dimensjoner i kun en strøm. Alternative teknikker for 3D-opptak blir diskutert nærmere under 3D i distribuert samhandling.

## 15 3D i distribuert samhandling

### 15.1 Bakgrunn

Erfaringene fra TiDE-SVR prosjektet og den distribuerte samhandlingsproduksjonen med NRK viser at det er store forbedringspotensialer i kommunikasjon med 3D-video. I dette avsnittet ønsker jeg å sette fokus på tilgjengelig teknologi og systemer som er under utvikling. Min påstand er at man i oppbyggingen av virtuelle delte rom vil benytte den objektbaserte teknikken ettersom denne egner seg mer i den virkelige verden der man ikke har dedikerte linjer og må begrense beslaget på linjene. Spesielt relevant er dette for systemer der deltakeren i den distribuerte samhandlingen skal kunne veksle mellom ulike betraktningsvinkler. I et system som ikke er objektbasert må man i prinsippet sende uendelig antall duale strømmer ettersom kameravinkelen er den fysiske vinkelen hos avsenderen. I multiviewsystemer som er tiltenkt det kommersielle markedet separerer man deltakeren fra bakgrunnen. Det blir kun tatt bilder av dette objektet fra et antall vinkler nødvendig for å kunne regenerere en vilkårlig virtuell kameravinkel på mottakerenden. Ettersom man kun sender objektet og et fåtall variasjoner av denne er beslaget på linjen (den nødvendige båndbredden) begrenset. Problemet med disse systemene i dag er at regenereringen er prosessorkrevende som medfører en mindre tilfredsstillende bilderate. Men det er bare et tidsspørsmål før algoritmene blir gode nok og/eller PC-ene blir raske nok. De objektbaserte systemene vil også bety en enklere løsning på blikkontakt (eye-gaze) ettersom kameraene ikke trenger være fysisk plassert bak skjermen. Et bilde med en virtuell kameravinkel bak skjermen (tilsvarende den vi fikk bak speilet i mediatoren) genereres fra et sett av stereoskopiske kamera som er plassert på siden av (eller over og under) skjermen. Med dette er det ikke nødvendig med dyre havgjennomskinnelige speil for å oppnå blikkontakt og man får langt enklere og mer portable løsninger basert kun på vanlige skjermer (2D eller 3D).

### 15.2 Eksperiment med tilgjengelig teknologi

Etter at TiDE-SVR prosjektet var gjennomført ble infrastrukturen benyttet til å sette opp den første 3-dimensjonale videokonferansen mellom HiL og HiG.

#### 15.2.1 Oppsetting av visning og opptaksutstyr

På HiG fikk vi hjelp av en master student (Mats Reinsby) som satte opp stereoskopisk opptak- og visningsutstyr. Han hadde tilgang på to identiske JVC kamera av semiprofesjonell type

som han satte på en bordflate. På bordflaten justerte han vinkelen mellom de slik at det på HiL så ut som om venstre og høyre kamera var overlappende i den delen av bildet som ikke skulle stikke ut eller inn av lerretet (0-parallakse). Han fikk også hjelp av Claus Knudsen til å sette opp to tilnærmet identiske projektorer. Claus var testperson ved HiG når systemet var ferdig satt opp. Han kommenterte at det var vanskelig å perseptuere dybden i 3D-bilde på hans side. Det er rimelig å anta dette skyldes at projektorene ikke var identiske og at man har prøvd å tilpasse de overlappende skjermene med zoom og keystone korreksjoner. Har projektor-paret ulike linser er det i praksis svært vanskelig å oppnå en lik spredning av pikslene på lerretet. Resultatet blir et udefinert 3D-bilde. På HiL fikk vi låne to identiske projektorer av IT-avdelingen som ble filtrert med polariseringsfilter der lyset som slippes gjennom er hhv vertikalt og horisontalt polarisert. Filtrene som ble benyttet var ganske enkelt fra en 3D-brille som ble lagt over de to projektorene. Samme teknikken ble også benyttet på HiG. Ettersom projektorene viste hvert sitt bilde fra venstre og høyre kamera og polariseringen korresponderte med retningen i brillene man hadde på seg ble det reflekterte bildet fra lerretet oppfattet 3-dimensjonalt (Vertikalt polarisert lys slapp ikke gjennom det horisontalt polariserte brilleglasset og motsatt). Dette forutsatte selvsagt at lerretet ikke endret på polariseringen. Metalliske overflater har denne egenskapen og det fant vi på baksiden av tavlene ved HiG og HiL. Ulempen med disse "silver-screen" lerretene var at gjenskinnet fra projektorlampen var ubehagelig sterkt. På et mattere lerret ville dette ikke vært et problem. Ellers fremsto dybdevirkningen av 3D bildet på HiL noe overdrevet. Dette skyldes at kameraene på Gjøvik var for brede til å kunne gi øyeavstand (6,5 cm) mellom sensorene. Når kameraene ble plassert på avstand og bildene zoomet inn ble dybdegjengivelsen svært troverdig. Dette var en kompensasjonsteknikk vi hadde positiv erfaring med fra produksjonen av den stereoskopiske disseksjonsfilmen beskrevet under "Veiledning i stereoskopisk produksjon".

### 15.2.2 3D-kommunikasjon via videokonferanseenheter

Videokonferanseenheten Tandberg 6000 ble erstattet med en Tandberg 800 som skolen hadde tilgang til over lengre tid. Opprinnelig var planen å gjennomføre eksperimentet på konvensjonelt videokonferanseutstyr der det stereoskopiske signalet med sine to kilder skulle gå tur/retur via den samme enheten. Dette for å vise at det var mulig å gjennomføre en 3-dimensjonal distribuert samhandling med utstyr de fleste har tilgang på. Dessverre viste det seg at videokonferanseenheten ikke hadde støtte for duo-video begge veier. Dersom den ene enheten hadde satt opp en duo-video og den andre prøvde å velge inn en duo-videokilde ble

den førstnevnte erstattet med den siste. Det samme er i følge teknikere på HiG tilfellet på Tandberg 6000, men ikke tilfellet for tidligere implementasjoner av Tandberg 800. Etter oppgraderingen av 800-enheten hadde de mistet en duo-video kanal til fordel for muligheten til å kjøre høyere bitrate. Dette er et valg Tandberg har tatt på bakgrunn av at man skjeldent har behov for duo-video begge veier. Kvalitetsøkningen som man oppnår med å tilordne større minneandel til de tre kanalene er synlig og vil være det som betyr mest for de fleste.

Sony har på sine videokonferanseenheter full 4CIF-kvalitet (704x576) i to kanaler begge veier, så dette bør også være mulig å få til for Tandberg. Det er vel å anta at det er kostnadsspørsmål som begrenser tilgjengelig minne på Tandbergenhetene.

For å løse problemet (med den kostnadsrammen vi hadde) ble vi nødt til å benytte de dedikerte fiberlinjene på den stereoskopiske returen til Lillehammer. Uten komprimering kunne signalet sendes til HiL med svært kort tidsforsinkelse og med broadcast-kvalitet. Dette var en fordel med tanke på opplevd følelse av tilstedeværelse i følge testpersonen Sigmund Andresen ved HiL.



Figur 15-1: Oppsett av stereoskopisk kamera (til venstre) og stereoskopisk skjerm (til høyre)



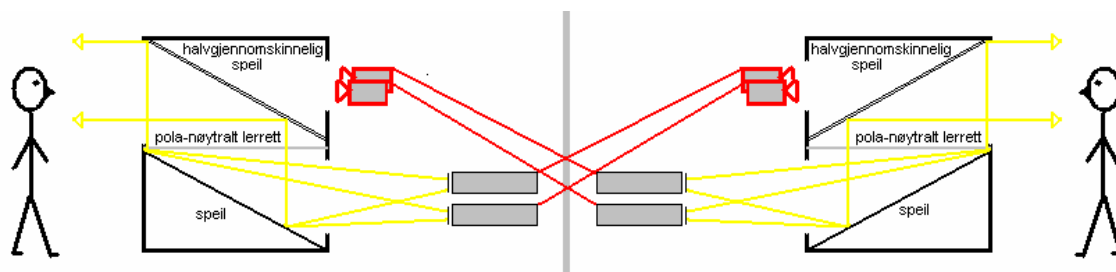
Figur 15-2: Stereoskopisk prøvebilde (til venstre). Første stereoskopiske bildet fra Gjøvik (til høyre)

Kodeken som ble benyttet til transmisjon av den stereoskopiske strømmen tilbake til Gjøvik var av typen H.264/MPEG4 AVC. Denne hadde noen blokkeffekter ved 768 kbit/sek, men med en oppløsning på 704x576 (4CIF) var kvaliteten tilstrekkelig for å oppnå tilstedeværelse ved HiG. De projiserte flatene dannet en bildeflate der størrelsen på det men medierte deltakeren korresponderer med hans virkelige størrelse. På dette punktet var visningen bedre enn i mediatoren. Oppløsningen er i utgangspunktet ikke tilstrekkelig for størrelsen på den projiserte flaten, men detaljrikdommen ble forsterket av at det var to bilder som ble projisert på samme flate (Det gikk to 4CIF strømmer fra HiL til HiG på 786kbit/sek hver).

### 15.2.3 Forslag til oppbygging av en 3-dimensjonal mediator

Der den utprøvde løsningen var dårligere enn mediatoren var på blikkontakt. I følge testpersonene var dette en svakhet ved systemet som ikke kunne oppveies med dybdesyn når det gjaldt følelsen av nærhet. Kameraene kunne ikke plasseres foran lerretet da dette ville forstyrret bildet av personen man kommuniserer med samt at kameralinsen ville fått lysinnslag direkte fra projektoren. Det sistnevnte kan man løse med et bakprojisert lerret som ikke endrer polarisasjonen, men da vil fortsatt kameraets plassering overskygge deltakerbildet.

Vi prøvde ut om det var mulig å plassere kameraene på hver side av lerretet for dermed å få et perseptuelt blikkretningsbilde i 3D (Hjernen beregner et snitt av venstre og høyre kamera ved 3D-persepsjon. Dette vil tilsvare et kamera plassert midt på lerretet). Avstanden mellom kameraene var for stor og vi så et dobbelt bilde (Teknikken vil fungere på mindre skjermer som i Samsung sin prototype-mobil med autostereoskopisk skjerm og et kamera på hver side). En enkel løsning med det utstyret vi har tilgang på ville vært å erstatte monitoren i mediatoren med en bakprojisert 3D-skjerm.



Figur 15-3: 3D-mediator, prinsippkisse

De to kameraene på bildet fanger en stereoskopisk avbildning av deltakeren bak det halvgjennomskinnelige speilet. Bildene sendes til hver sin projektor. Projektorene har ulike polarisasjonsfiltere for venstre og høyre bilde (hhv horisontalt og vertikalt polarisert). Denne polariseringen bevares dersom det bakprojiserte lerretet er pola-nøytralt. Speilene endrer ikke polariseringen og det vil være mulig for deltakeren å se to ulike bilder i det halvgjennomskinnelige speilet. De ulike bildene fremkommer når deltakeren tar på seg en brille med hhv horisontale og vertikale polafiltere foran venstre og høyre øye. Ettersom motsatte polariseringer utslukker hverandre vil hvert øye kun ser det bildet som korresponderer med polafilterets retning på projektoren. 3D-bildet av deltakeren man kommuniserer med projiseres på det halvgjennomskinnelige speilet og kameraet bak sørger for at de har øyekontakt. Projektorteknikken gjør det også mulig å vise deltakerne i en naturlig størrelse. Fra før var det kun montert inn en 32" skjerm i mediatoren, men denne kan bli 50" dersom man fyller hele den horisontale flaten (her: lerretet) med bildet av deltakeren.

Korrekt størrelse, dybde og blikkontakt vil bidra til en følelse av tilstedeværelse. Det som kan virke mot sin hensikt i dette oppsettet er at man må benytte briller. Brillene reduserer lyset og øynene vil dermed være vanskelig for deltakerne å se. Problemet kan løses ved å bytte ut lerretet med en lentikulær skjerm. Disse skjermene er autostereoskopiske som betyr at man ikke trenger briller. Venstre og høyre bilde projiseres til hvert sitt øye ved hjelp av ulike linsekonstruksjoner som ligger foran annenhver kolonne av piksler i skjermen. For at annenhver kolonne skal vise hvert sitt bilde må videosignalene gå gjennom en stereoskopisk multiplekser. Dette realiseres med et gratis program på en PC med to grabberkort. Multiplekseren er ikke spesielt kostbar, men skjermene ligger foreløpig på ca 30 000 kr<sup>51</sup> og disse er kun 19". Det finnes også skjermer som er større, men det er å anta at disse er svært kostbare<sup>52</sup>.

En rimelig mellomløsning ville vært å erstatte de lineære filterne foran projektorene med sirkulære polarisasjonsfiltere. Da er det mulig å rotere brilleglassene uten at de mister den separerende egenskapen. Dette kan nyttegjøres i pola-baserte kontaktlinser der man kan bevare blikkontakten. Vi har ikke lyktes å finne kontaktlinser med sirkulære pola-filtere. Dette har ikke noe bruksområde foruten 3D mediering, et relativt lite forskningsfelt. For

---

<sup>51</sup> Prisen er fra et tilbud fra Seereal tilsendt Jostein Halgunset ved Institutt for laboratoriemedisin ved NTNU.

<sup>52</sup> Leif Arne Rønningen ved NTNU mener det er en rask utvikling av autostereoskopiske skjermer mhp kvalitet og størrelse, men i skrivende stund er de store skjermene som prototyper å regne og svært kostbare i innkjøp.



videre eksperimenter med mediatoren må vi derfor akseptere bruken av briller i 3D-mediering.

Mediatoren er ellers en stor fysisk løsning som vil kunne erstattes av mer teknisk intelligente løsninger på sikt. I utvikling av videokonferansesystemer har man forstått viktigheten av blikkontakt og dybde i det virtuelle rom. I neste avsnitt vil jeg vise et eksempel på dette hos utviklingsavdelingen i HP.

#### 15.2.4 Relatert forskning på distribuert samhandling i virtuelle rom

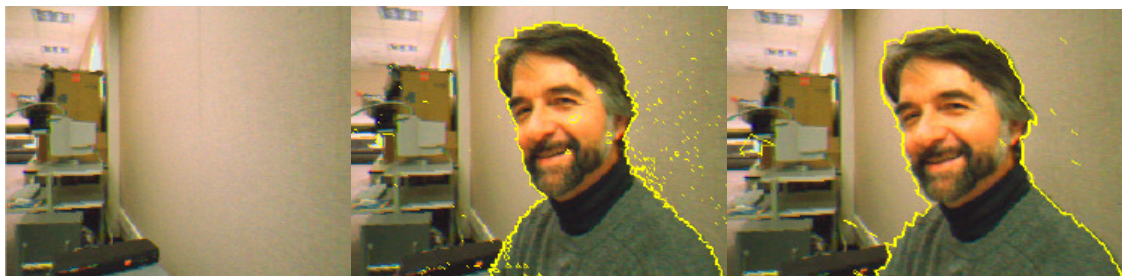


Figur 15-4: Konferanse i et virtuelt rom. Kameraoppsett (venstre), rendret deltakerbilde (høyre).

### HP

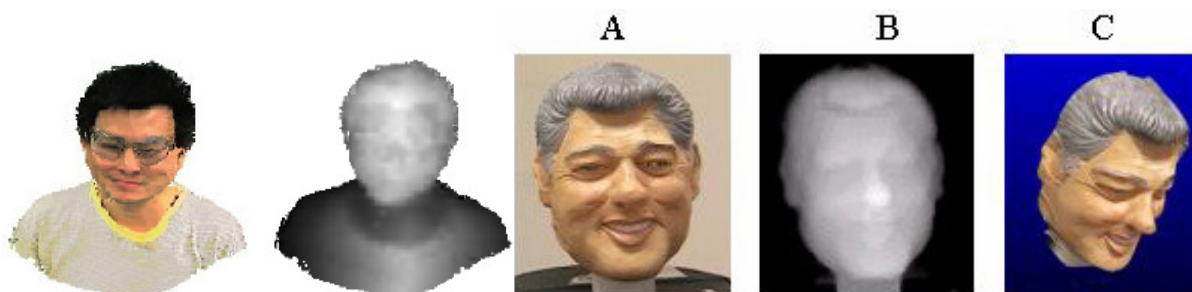
HP har siden 2003 jobbet med et prosjekt som heter Coliseum der man søker å utvikle et videokonferansesystem med mulighet for flerpartskonferanser i virtuelle rom. Systemet er ment å fungere på konvensjonelle PC-er men man har så langt ikke lyktes i å oppnå tilfredsstillende bilderater (kun 15 bilder pr sekund) på dagens maskiner (Baker, 2003). Dette skyldes kompliserte analyse og visualiseringsalgoritmer. Bildet over til venstre viser en deltaker som blir filmet fra fem forskjellige vinkler samtidig. Disse strømmene danner grunnlag for å generere en 3-dimensjonal avatar som kan plasseres i et virtuelt rom. Avataren er representasjon av deltakeren som kan roteres og tilpasses rommet slik at det ser ut som om man sitter rundt et konferansebord (bilde til høyre). Kameraene er ikke plassert bak skjermen som i mediatoren, men av den 3-dimensjonelle avataren kan man enkelt lage en virtuell kameravinkel som tilsvarer denne og dermed gir blikkontakt. For å plukke ut deltakeren fra bakgrunnen benyttes en differansekeying (forklart i tidligere avsnitt). Det lagres først en referanseframe der deltakeren ikke er i bilde og man kun ser den faste bakgrunnen. Når deltakeren så kommer inn i bildet vet algoritmen hva som er bakgrunn og hva som er deltaker utifra en sammenligning med det lagrede bildet. Området som har forandret luma- og/eller cromaverdi betydelig i pikselstrukturen (differanse ikke lik null) kan avgrenses som en maske

rundt deltakeren. Med lineær interpolasjon forenkles masken. Dette for å gjøre den videre prosesseringen mindre krevende. Masken som dannes ved de ulike kameraene danner også grunnlag for å kunne fortelle om deltakerens blikkretning. Når deltakeren snur på hodet utfører systemet ved hjelp av maskene en motion tracking og forflytter bildet av det virtuelle rommet i henhold til hodebevegelsen.



Figur 15-5: Bakgrunn/ref.bilde(venstre), differansekalkulert omriss(midten), interpolasjon(høyre)

Det virtuelle bildet genereres hos mottakeren med VRML (Virtual Reality Markup Language) der man kan bevege seg fritt i rommet uten at de andre deltakerne merker noe til det. Dette gir også en mulighet som HP ikke har drøftet. Ved å rendere ut to bilder med en differanse i kameravinkel som tilsvarer øyeavstanden vil man kunne vise et ekte 3D-bilde med på en autostereoskopisk skjerm i midten. Med blikkontakt og dybdesyn uten briller vil dette kunne føre til en uovertruffen følelse av nærhet til en overkommelig pris (Ettersom skjermen er nær deltakeren trenger den ikke være stor og da er den heller ikke uopnåelig dyr). Men for å lage den 3-dimensjonale avataren benyttes en teknikk som pr i dag ikke gir tilfredsstillende kvalitet (IBPH). Man kan se av bildet til venstre nedenfor at geometrien ikke kan gjengis presist nok til at ansiktet blir troverdig. Sorthvitt bildet til venstre viser dybden i ansiktet som blir kalkulert utifra de fem kameraene med ulik kameravinkel.



Figur 15-6: 3-dimensjonale avatarer generert med IBPH (til venstre) og Zcam (til høyre)

### 3DV

Bildene merket A, B og C er hentet fra reklamemateriell (3DV, 2006) der 3DV demonstrerer deres Zcam teknikk for måling av dybde (Teknikken er tidligere beskrevet i forbindelse med dybdekeying). Selv om denne kilden kan være manipulert av markedsmessige hensyn er det rimelig å anta at avbildningen stemmer med virkeligheten. Dette er systemer som er i bruk i profesjonell fjernsynsproduksjon der kravene til kvalitet er strengere enn for videokonferansesystemer. A viser en RGB-avbildning av deltakeren, B viser dybdeverdiene der hvit er nærmest kamera. Av bilde C, som er den 3-dimensjonale rendringen av deltakeren, kan vi se at geometrien er mer presist gjengitt enn med IBPH-teknikken. Med Zcam-teknikken er det også enklere å kalkulere en 3-dimensjonal avatar ettersom dybdeverdiene i bildet er gitt direkte av verdiene fra sensoren (ekkolodd-prinsipp). Dette vil kunne gi en tilfredsstillende bilderate. Med Zcam teknikken trenger heller ikke det fysiske kameraet ligge bak skjermen for å oppnå blikkontakt. Avataren kan roteres slik at blikkretningen blir inn mot skjermen der det virtuelle kamera i IBPH-teknikken var plassert. Med Zcam teknikken vil man også kunne gjenskape et stereoskopisk bilde ved å rendere ut to kameravinkler tilsvarende øyeavstand. Så langt ser det ikke ut som om IBPH teknikken ikke kan foretrekkes fremfor Zcam, men systemene må sees i sammenheng med brukergruppen de er tiltenkt. Zcam er laget for profesjonell TV-produksjon og dybde-sensoren med tilhørende hardware er sannsynligvis relativt dyrt. IBPH teknikken er ment å fungere på en vanlig PC der man kobler til de 5 web-kameraene (VGA) til en firewire port. Dette er den eneste hardwareinvesteringen man må gjøre med mindre man også ønsker en autostereoskopisk skjerm. All prosessering skjer i software. Investeringene er derfor relativt lave og datakraft samt presisjon i algoritmer vil med tiden utvikles til å kunne gi en tilfredsstillende dybde og bilderate. Teknikken har derfor en lovende fremtid og kan bli viktig for Åpen kanal sine deltakere.

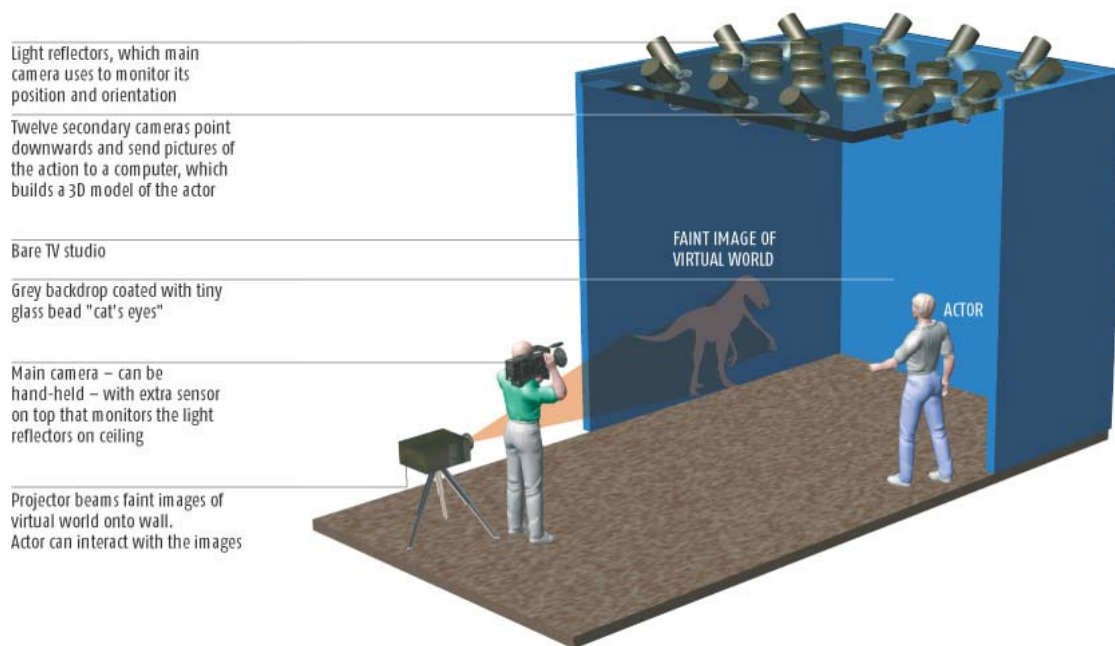
### BBC

BBC har forsket på lignende teknikker som HP, men for TV og filmproduksjon (Grau, 2004). Her begrenser man seg ikke til frontale kameraer, men foretar en bildeanalyse fra 12 kameraer plassert rundt objektet. Dette gir en større bevegelsesfrihet i rendringen. Kameradekning på baksiden av deltakeren gjør at denne siden også kan vises. Med flere kamera får objektet også større dybdepresisjon. Her benyttes ikke differensekeying, men en chromakeyng-teknikk som ligner på den bluescreen-løsningen vi har benyttet i forsøket. Chromakeying kompliserer vanligvis studiooppsettet, mer har man klart å unngå problemet med ujevn lyssetting. Studioet

dekkes med en retrorefleksiv duk som ved hjelp av små innebygde glassperler kun reflekterer i den retningen lyset kommer fra. Rundt kameralinsa plasseres en diodering med blått lys. Diodene lyser så svakt at det ikke er synlig blå-lys på forgrunnen. Når lyskilden har eksakt samme posisjon som linsa vil kun kameraet se lyset fra dioderingen. Dette er en smart teknikk som kan gjøre chormakeying langt enklere for Åpen kanal sine deltakere.

#### VIRTUAL TELEVISION STUDIO

An actor can interact with computer-generated characters in real time to produce Hollywood-style special effects



**Figur 15-7: Prinsippskisse av BBC sitt studio for generering av virtuelle rom.**

BBC har i den foreslåtte teknikken også tatt høyde for at deltakeren skal kunne forholde seg til objekter i omgivelsene. En datamaskin renderer objekter inn som en del av scenen. Disse objektene vises som en svak projisert avbildning. Som man kan se av konseptskissen var det planlagt å projisere referansebildet på den retrorefleksive duken i en posisjon som korresponderer med deltakernes blikkpunkt. Projiseringen må være for svak til ikke å bli betraktet som bakgrunn av keyeren, men sterk nok til at deltakeren kan få en referanse å spille mot. Referansebildet ble projisert på et vanlig lerret i forsøket. Årsaken kan være at lysforholdet mellom projektoren og blålyset var vanskelig å tilpasse, men dette er ikke

bekreftet av BBC. Prinsippskissen viser også at sensorer i taket sporer kamerabevegelse og renderer scenen deretter. BBC benyttet virtuelle kamera i forsøket så teknikken ble overflødig.



Figur 15-8: Studio med reelt oppsett. Interaksjonsskjerm(venstre), diodering(midten), duk(høyre).

Interaksjonsskjermen er interessant ettersom deltakerne her kan se deler av det resulterende virtuelle rommet. I eksperimentet med NRK hadde deltakerne også mulighet til dette, men oppsette vår var langt enklere å prosessere. I systemer som forutsetter 3-dimensjonale renderinger av både interaktive objekter og bakgrunn har det tidligere ikke vært mulig å kunne gi deltakerne denne referansen. BBC sin løsning er på denne måten unik.



Figur 15-9: Resultatbilder. Mesh kalkulert fra 12 kamera(venstre), multikam(midten), scene(høyre)

### Interaktivitet

For at deltakeren skal føle at han er en del av et miljø er det viktig at forutsetningene for interaktivitet er tilstede. I henhold til teorien om interaktivitet (Steuer,1993) bærer både løsningen til BBC og HP et lovende potensiale for realisering av Range og Mapping.

Range blir definert ut ifra antall forskjellige muligheter for påvirkning og i hvor stor grad en bruker kan endre omgivelsene (Steuer, 1993). Mapping angir hvilke menneskelige handlinger som styrer gitte endringer i det virtuelle miljøet. I HP sin løsning var det hodebevegelsen som styrte forflytningen i det virtuelle rommet. En slik løsning har ikke blitt utviklet i BBC sitt system, men man har her en bedre forutsetning for et større sett av mapping. Mens HP løsningen utfører sine bevegelsesanalyser utifra 2-dimensjonale masker av overkroppen har BBC hentet ut en 3-dimensjonal maske av hele kroppen fra de 12 kameraene. Med utgangspunkt i denne masken er det mulig å bruke kroppsspråk som en trigger til endringer i

omgivelsene. En ekstra dimensjon er heldig for analyse av kroppbevegelser som skjeldent utarter seg i ett plan. Men beregningsgrunnlaget er mer komplisert og forutsetter derfor mer regnekraft enn i HP sin interaktive løsning. En utfordring mhp interaktivitet i BBC sitt system kan dermed bli hastighet (Steuer, 1993). Dersom algoritmene blir for kompliserte vil tiden fra aktøren utfører en handling til at endringen fremkommer i miljøet bli så lang at deltakeren ikke vil føle at relasjonen til miljøet er naturlig. For et mindre mapping-sett kan derfor HP løsningen gi deltakerne en sterkere følelse av interaktivitet.

### 15.2.5 Enkel sammensetning av virtuelle rom med SMIL

Med en player som har støtte for objektbasert video som for eksempel Real Media One og Quicktime kan man med enkelt scriptingspråk sette sammen scener hos mottakeren. Det geniale med dette er at man kan lage individuelle komposisjoner der man kun legger beslag på den båndbredden som kreves av objektene man velger å ta inn i scenen. For å skille objektene fra bakgrunnen kan man benytte chromakeying hos mottaker eller alfamaske hos avsender. Det sistnevnte er mest plassbesparende ettersom man ikke sender med bakgrunnen.

I følgende kodeeksempel har jeg realisert en scene med to objekter (f.eks deltakere) som ligger over en felles bakgrunn (f.eks virtuelt studio). Jeg har ikke lykkes å finne streamingtjenester som håndterer alfamaske. Bakgrunnen skilles fra objektene hos mottaker ("ChromaKey Tolerance" må settes til et nivå som aksepterer den variasjonen "ChromaKey"-fargen har for at bakgrunnen skal bli transparent) og må sendes i sin helhet. Eksempelet er derfor ikke det mest biteffektive. (PS: Chromakeying fungerer kun med Open GL støtte):

```
<smil xmlns:rn="http://features.real.com/2001/SMIL20/Extensions">
  <head>
  <layout>
    <root-layout width="320" height="240"/>
    <region id="VR_scene"/>
    <topLayout width="180" height="120">
    <region id="deltakere">
    </topLayout>
    </layout>
  </head>
  <body>
  <par>
    <video src="rtsp://quicktimeserver.item.ntnu.no/bakgrunn.mp4"
    region="VR_scene" rn:chromaKey="#FFFFFF" rn:chromaKeyTolerance="#010000"
    rn:chromaKeyOpacity="#FFFFFF"/>
    <video src="rtsp://quicktimeserver.item.ntnu.no/deltaker_HiL.m4e"
    region="deltakere" rn:chromaKey="#0000FF" rn:chromaKeyTolerance="#010000"
    rn:chromaKeyOpacity="#FFFFFF"/>
    <video src="rtsp://quicktimeserver.item.ntnu.no/deltaker_NRK.m4e"
    region="deltakere" rn:chromaKey="#00FF00" rn:chromaKeyTolerance="#010000"
    rn:chromaKeyOpacity="#FFFFFF"/>
  </par>
  </body>
</smil>
```



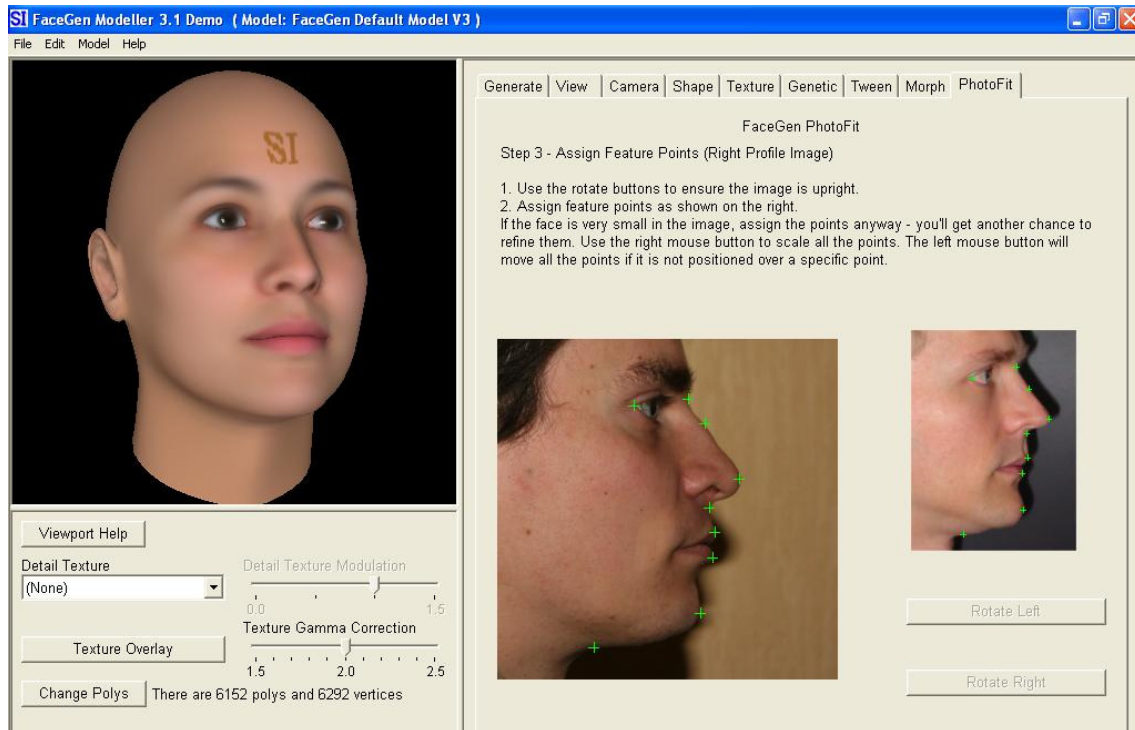
### 15.2.6 Eksperimenter med syntetiserte avatarer.

Det er mulig å ta en 3-dimensjonal avbildning av deltakerne i forkant av konferansen og styre avatarene kun med parametere som beskriver bevegelser i kropp og ansikt. Med denne teknikken overføres 3D-bildet av deltakerne med tekstur og 3D-modell (mesh) én gang til stedet der den virtuelle scenen lages. Under produksjonen av den distribuerte samhandlingen sendes kun styringsparametere, som begrenser nødvendig bitrate til noen få kbps. Dette gjør distribuert samhandling tilgjengelig for deltakere med dårlig linjekapasitet, typisk for avsidesliggende strøk og mobile enheter.

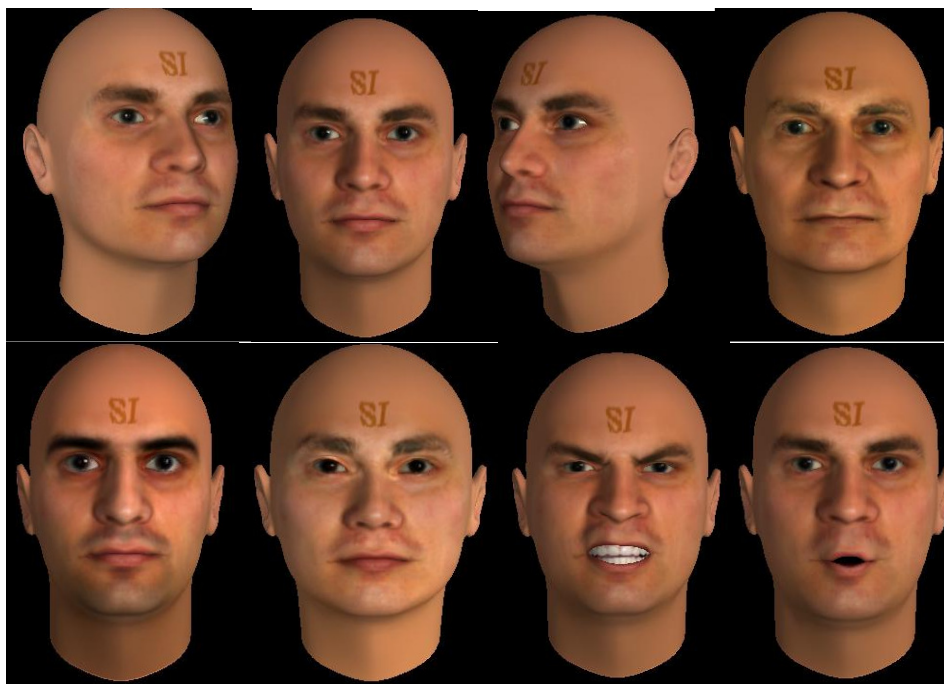


Figur 15-10: Analyse av ansiktsbevegelser og mapping til en predefinert 3D-maske (Metaxas, 1999)

Analysering av ansiktsuttrykk og kroppsbevegelser er det som er den største utfordringen i avatar-baserte systemer. Jeg har valgt å ikke fokusere på denne delen ettersom det er lite tilgang på ferdigutviklet programvare og utprøvingen ville dermed blitt for tidkrevende. Avbildningen som danner avataren og styringen av denne har jeg derimot prøvd ut i praksis. Teknikken er brukervennlig og gir tilfredsstillende kvalitet. For 3D-avbildningen benyttet jeg programmet Facegen. Her legger man inn et profilbilde og et frontaltbilde av seg selv og angir punkter i bildet som nesetipp, øyekrok etc. Ut ifra disse to bildene generer programmet en 3-dimensjonal avbildning som kan styres. I programmet som generer avataren er det mulig å legge inn verdier for styringsparametere som emosjonelle uttrykk, vokaler, konsonanter, rase, alder etc.



Figur 15-11: Konstruksjon av en 3-dimensjonal avatar med programmet FaceGen.



Figur 15-12: Meg i ulike vinkler i dag, om 30 år, inder, asiater, sint og under uttalelse av vokalen o.

MPEG4 definerer en standard for styring av avatarer. Dette gjør det lettere å utvikle systemer som kan kommunisere med hverandre. Syntetiserte avatarer har så langt vært lite brukt i



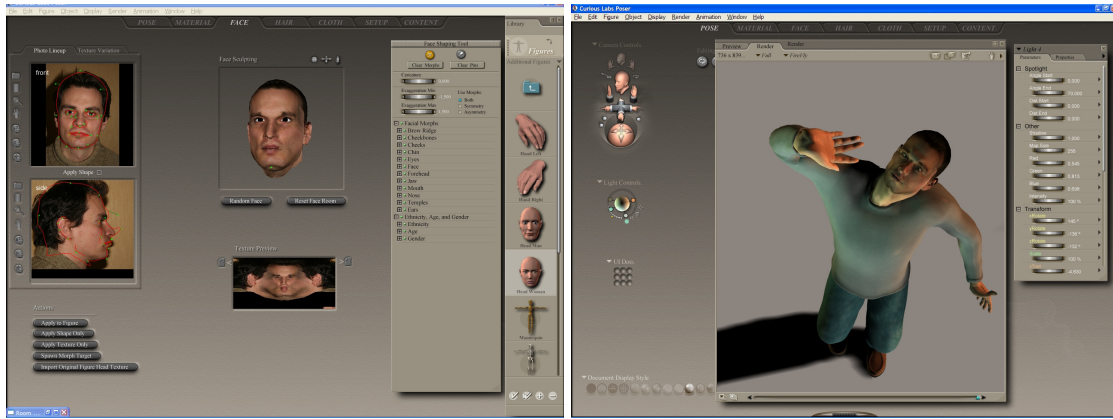
distribuert samhandling, men dersom man får til å analysere ansiktene vil teknikken kunne bli nyttig. Med et skjermkort med støtte for Open GL som benyttes her vil rendringen kreve minimalt med regnekraft. Visningen vi dermed kunne realiseres med utstyr som er tilgjengelig for Åpen kanal sine deltakere.

Eksempelet over viser kun styring av ansikter. Kroppsspråket er også viktig i en mediering. Her finner man utfordringer med å analysere kroppsbevegelsene. Forskning (Metaxas, 1999) har vist at det er en fordel med stereoskopisk avbildning i bevegelsesanalyse ettersom bevegelsen skjeldent forekommer i kun ett plan. Stereoskopi gjør også separasjonen av kroppen fra bakgrunnen enklere i naturlige miljøer (Vaghant, 2004). Det første trinnet i en bevegelsesestimering er sammenligning med kjente 3D-modeller for å gjenkjenne hva som er en hånd, en underarm etc. 3D-data gjør det deretter mulig å hente ut globale parametre slik som om hendene beveger seg i en rett linje, en sirkel eller i et plan (Metaxas, 1999).



**Figur 15-13: Gjenkjenning av kroppsbevegelser ved hjelp av multikamera analyse (Metaxas, 1999).**

Visningen av kroppen er heller ikke her noen utfordring dersom man har et Open GL-støttet skjermkort. En rendring av meg i full figur er avbildet under. Her er teksturen fra ansiktet lagt inn etter samme prinsipp som i FaceGen, men med programmet Poser. I dette programmet kan man også sette virtuelle lyskastere noe som hjelper på virkelighetsfølelsen. Avatarene ser ikke ut som ekte mennesker i dag. Det er likevel rimelig å anta at de vil det om kort tid. Spillindustrien er en viktig pådriver for å skape troverdige virtuelle miljø og naturlige spillkarakterer/avatarer.



Figur 15-14: Avatar med kropp. Generering av ansiktstekstur (venstre). Animerbar kropp (høyre)

## 16 Konklusjon

### 16.1 Konklusjon for samfunnsdelen

Åpen kanal har langt på vei funnet sin egenart og form i overensstemmelse med – og for å ivareta – målet om å styrke ytringsfriheten i fjernsynsmediet. Likeledes foreligger de tekniske beskrivelsene og nødvendig infrastruktur for å implementere denne formen. Bare to ting gjenstår: For det første må Kultur- og kirkedepartementet og *Åpen kanal* sammen komme fram til betingelsene for Norges åpne fjernsynskanal. Gratis kringkasting er opplagt. Utover dette er hva produsentene selv vektlegger: (1) alle må kunne nå alle, (2) kanalen må være åpen og mangfoldig og (3) produsentene må hver for seg råde over egen produksjon og i fellesskap, uten innblanding utenfra, råde over sendeflaten.

For det andre trengs en statlig støtte som sikrer fintilpassing av den allerede tilgjengelige infrastrukturen, og drift av denne. Alt koster til sammen tre millioner kroner per år, og er opp til sittende regjering å foreslå bevilget. Det vil gi det klarsignalet som Stortinget allerede venter på å få.

Norge kan bli et foregangsland i realiseringen av reell ytringsfrihet. "Reell ytringsfrihet" er det samme som "allmenn ytringsrett". For 150 år siden stod det om allmenn rett til å lære skriving og regning. For hundre år siden stod det om allmenn *stemmerett*. I dag står det om allmenn *ytringsrett*.

### 16.2 Konklusjon for systemdelen

For at *Åpen kanal* i realiteten skal bli åpen må også de tekniske løsningene videreutvikles. I utviklingen har vi vektlagt brukervennlighet og tatt hensyn til at *Åpen kanal* sine deltakere ikke er teknologer. Videreutviklingen i denne diplomen besto i å sørge for at deltakerne ikke må forholde seg til bakenforliggende prosesser i infrastrukturen (som for eksempel oppkobling til FTP-serveren). For publisering på nett er nå systemet helautomatisert med en opplastingsklient som dukker opp ved å trykke på en link på nettsiden. Et Web-basert grensesnitt er noe folk flest er kjent med. Løsningen vil derfor ikke ekskludere deltakere som har vanskelig for å forstå teknikken bak. Teknikken er usynliggjort, men man kan likevel følge prosessen gjennom systemet i et enkelt Web-grensesnitt. Dette gjør at systemet egner seg for feilsøking og for mer avanserte brukere, men er ikke en forutsetning for å nyttiggjøre tjenesten. Foreløpig er *Åpen kanal* kun på nett så løsningen er i dag tilfredsstillende med

tanke på brukervennlighet. TV-avvikling er ikke realisert i et Web-grensesnitt. Dette blir håndtert av en egen konsoll som etter min mening er lite brukervennlig tatt i betraktning det behovet som en åpen kanal har. Det bør realiseres et mer intelligent system for sendetidsprogrammering der brukeren ikke forholder seg til noe annet enn mulig plassering på sendeplanen. Komprimering til MPEG2 må helautomatiseres og varigheten av innslagene samt kategorien de er registrert under bør gi en begrenset tilgang på sendeplanen. Det bør ikke være mulig med overlappende sendetidspunkt slik som i eksisterende system og i henhold til vedtektene i Åpen kanal skal innslagene som sorterer under et gitt tema begrenses av tidsvinduer. Et slikt system kan også realiseres med et Web-grensesnitt som i bakgrunnen kommuniserer med databasen til playout-enheten. Dette er en essensiell funksjon for at Åpen kanal skal kunne videreføre brukervennligheten når TV-sendingene kommer i gang. Erfaringene med Playout-enheten viser at dette siste leddet ikke er teknisk krevende å realisere. Overgangen fra en nettsending til en TV-sending er i praksis bare en formatforskjell. På TV streames MPEG2TS og på nett har vi valgt å streame MPEG4. En implementering med en mer intelligent samkjøring av de to systemene er mulig. Åpen kanal har foreløpig begrensede utviklingsressurser og i den sammenheng er det mer tilrådelig å utvikle Web-baserte styringssystemer for ferdig utviklet programvare.

Når det gjelder mulighetene for livesending og distribuert samhandling er ikke dette avhengig av en TV-distribusjon. Produksjon av debatter i delte virtuelle rom kan Åpen kanal realisere i samarbeid med TV-studioer tilkoblet Uninett. Høgskolen i Lillehammer samt Universitetet i Bergen, Oslo og Trondheim er tilknyttet Uninett og har gode ressurser for flerkameraproduksjoner. Det vil i korte prøveperioder også være mulig å koble sammen de distribuerte deltakerne med dedikert fiber. Dette vil, som forsøket mellom Gjøvik og Lillehammer har vist, redusere tidsforsinkelsen og bygge opp under følelsen av nærhet.

For å inkludere deltakere som ikke har tilgang på like gode linjer bør man også prøve ut distribuert samhandling med videokonferanseenheter. Mine studier har vist at produsenter i videre utvikling av slikt utstyr tenker objektorientert for en enkel realisering av virtuelle rom. HP tar sikte på å utvikle multipart distribuert samhandling for standard PC. Det er bare et tidsspørsmål før avansert mediering med utstyr folk har hjemme blir mulig. Linjekapasiteten er også under utvikling. Slik jeg ser det kan Åpen kanal tjene på å vente til rimelige produkter dukker opp fremfor egenutvikling av systemer for distribuert samhandling, men man kan utnytte denne tiden til å gjøre erfaringer med samhandling på tilgjengelig utstyr. Forsøkene på Høgskolen i Lillehammer har vist at man med enkle midler som halvgjennomkinnelige speil

og polariseringsfilter kan oppnå en nærhet som hittil er sjeldent å finne i eksisterende kommunikasjonssystemer. Blikkontakt og 3-dimensjonale bilder er ikke utprøvd i stor grad. Med det nettbaserte systemet kanalen har i dag og den unike tilgangen på gode linjer kan Åpen kanal bli en pilot for alternativ produksjon der mennesker møtes i nærhet til tross for geografisk avstand.

## 17 Referanser

- Andresen S. og Knudsen C. (2005): Eksperimenter med kommunikasjonskvalitet og tilstedeværelse, HiL 2005
- Baker (2003): H. Harlyn Baker, Nina Bhatti, Donald Tanguay, Irwin Sobel, Dan Gelb, Michael E. Goss, John MacCormick, Kei Yuasa, W. Bruce Culbertson and Thomas Malzbender: "Computation and Performance Issues In Coliseum, An Immersive Videoconferencing System", Hewlett-Packard Laboratories 2003
- Dewey, John (1916): Democracy and education, 1916, og The public and its problems, 1927
- Diot, Levine, Lyles, Kassem, Balensiefen (2000): "Deployment issues for the IP multicast service and architecture," IEEE Network, 14(1), January/February 2000
- Grau (2004): O.Grau, T.Pullen, G.A.Thomas: "A combined studio production system for 3D capturing of live action and immersive actor feedback", BBC R&D, 2004
- Gripsrud, J (2003): Foredrag med tema "medieblikk" av Jostein Gripsrud, Blinderen oktober 2003
- Guttu, J. Jøergensen and Nørve, S. (1985), Bovaner: En undersøkelse av 30 blokkleiligheter i Oslo, Byggforsk 1985
- Haugsbakk, G. (2000): Interaktivitet, teknologi og læring, ITU skriftserie 2000
- Jauert, Per & Prehn, Ole (2002): Mangfoldighet og kvalitet. Evaluering af tilskudsordningen for lokalradio og -tv, Kulturministeriet 2002.
- Jensen, J. F. (1998): Interaktivitet & interaktive medier (Multimedier, hypermedier, interaktive medier), Ålborg universitetsforlag 1998.

Klevsand, T (1989): Fjernsynsteknikk, Vett oig viten 1989

Knudsen C. (2004): Presence Production, KTH 2004

Knudsen C. (2005): Prosjektinfo TIDE/SVR Television in Digital Environment - Shared Virtual Reality, et samarbeidsprosjekt mellom HiG og HiL. TIDE-prosjektet, HiL/HiG 2005

Lindell, Jan Erik og Olsson, Peder (1998): Demokrati-tv, En kvalitativ studie av sändarna i Öppna Kanalen i Göteborg, Göteborgs Universitet, 1998.

Ling, R and Thrane, K (2001): The management of electronic media in Norwegian households, Telenor R&D 2001

Ling, R and Thrane, K (2002): "I don't watch TV to like learn anything": The leisure use of TV and the internet, First monday 7, <http://www.firstmonday.dk> 2002

Metaxas, Dimitris (1999): "Deformable Models for Segmentation, 3D Shape and Motion Estimation and Recognition", University of Pennsylvania 1999

Nattress (2005): Graerne Nattress, "Chroma Sampling: An investigation", <http://www.kenstone.net>, 25.juli 2005

Pedersen, Matti (2003): Realiseringen av en digital TV-kanal, ITEM NTNU 2003

Poynton, C (2003): Digital video and HDTV, Morgan Kaufmann publishers 2003

Propaganda (2003): "CNN ville ha NRKs valtriiks", <http://www.propaganda.no> 2003

RadAmec broadcast systems (2006): <http://www.radamec.com> – Virtual studio division.

Rawls (1993): Political liberalism 1993, blant annet s. 278

Rousseau, Jean-Jacques (1762): Om samfunnspakten, De norske bokklubbene, Oslo 2001.

Rønningen, L (2004), Distributed Multimedia Plays: Virtual dinner, ITEM NTNU 2004

Rønningen, L (2006), Diskusjon om 3D-visning og anvendt teknologi, Caruso NTNU 2006

Steuer, J. (1992), Defining Virtual Reality: Dimensions Determining Telepresence , Journal of communication 1992.

Tannenbaum, A (1996), Computer Networks 3.edition, Prentice Hall 1996

Tellesbø, Ola (2002): ”Ytringsfrihet og Åpen kanal” i Transformator, sommer 2002.  
Tilgjengelig på [www.openchannel.no](http://www.openchannel.no)

Vaghant (2004): Vaghant, Kushai: “Realtime human body tracking in public spaces”,  
University of Canterbury 2004.

3DV (2006): 3DV Systems, <http://www.3dvsystems.com> 2006



## A – Appediks: Veiledning i stereoskopisk produksjon

### 17.1 Bakgrunn

Interessen for tredimensjonal avbildning har vært til stede siden Sir Charles Wheatstone introduserte stereoskopet i 1833, men det var først på 50-tallet da Hollywood produserte noen titalls stereoskopiske filmer at folk fikk øynene opp for konseptet. Siden har interessen vært noe varierende på grunn av kostbare og kompliserte oppsett samt begrenset tilgjengelighet (3D ble aldri noe stor suksess på TV). Med fremveksten av rimelig digital teknologi og mer praktiske visningsalternativer er 3D igjen blitt aktualisert, spesielt for medisinsk og ingeniørteknisk anvendelse.

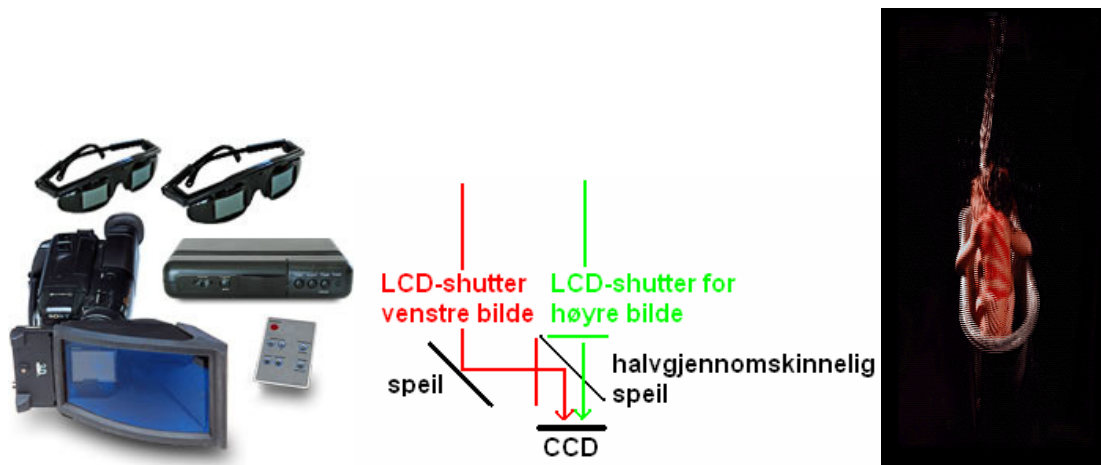
### 17.2 Avgrensninger

Dette avsnittet er tenkt som en veiledning til de i Åpen kanal som ønsker å prøve seg på 3D-produksjon eller ønsker å forstå prinsippene i praksis. Det blir i denne veiledningen tatt utgangspunkt i en produksjon som ble utført på institutt for laboratoriemedisin ved NTNU. Den 3-dimensjonale videoen skulle fungere som en opplæringsfilm i disseksjon for legestudenter og var kun ment for intern visning på ferdig oppsatt datautstyr. I skrivende stund er det ikke hensiktsmessig å sende 3-dimensjonalt innhold i det norske TV-nettet, som Åpen kanal er tiltenkt, ettersom TV-seerne ikke har tilrettelagt utstyr for visning (set-top boks med field-sekvensiell stereo splitting). På internett finnes det en mulighet for å prøve ut 3D-innhold. Streamingsystemene vi benytter i Åpen kanal har ingen begrensninger på dette. Og med en PC er det enklere å sette opp en 3D-visning ved hjelp av en stereoskopisk player (freeware) tilsvarende det vi benyttet på NTNU. Streamingsystemene gir også støtte for høyere oppløsning. I denne veiledningen tar vi utgangspunkt i det formatet som ga tilfredsstillende oppløsning til disseksjonsfilmen nemlig HDV med 1440x1080 piksler. To slike kilder som er nødvendig for å danne et 3D bilde krever en oppløsning på 2880x1080 som er større en TV-nettets fremtidige HD format (1920x1080). Skal 3D bildet sendes på TV-nettet må man halvere den horisontale oppløsningen ved å legge venstre og høyre bilde inn i hhv øvre og nedre field (en frame består av to fields), men visning krever at set-top boksen har mulighet til å dele framen igjen (og evt styre shutterglass).

Teknikken kan realiseres i kamera og er et enklere alternativ til det som blir veiledet i etterfølgende avsnitt. Nuview har utviklet en forsats til DV og HDV kameraer som veksler

mellom et bilde som er optisk forskjøvet til venstre og et bilde som går rett frem. Når denne vekslingen synkroniseres med lagringen av fields i kameraet (kompositt utgangen kobles til forsatsen) får vi en stereoskopisk (field sekvensiell) stereostrøm direkte på tape.

Denne strømmen kan redigeres med vanlige redigeringsprogrammer og avvikles og distribueres i konvensjonelle sendenett. Det må kun en spesialbygd set-top boks til for å sørge for visning. Hele kjeden er dermed det samme som for tradisjonell 2D-video i dag, med unntak av noen svært billige tillegg i endepunktene (set top boks til ca 700kr og forsats til ca 1500kr<sup>53</sup>)



Figur A-1: Nuview 3D-utstyr (venstre), prinsippskisse forsats (midten), field sekv. resultat (høyre)

Teknikken kunne revolusjonert 3DTV, men dessverre vil fremtidige sendesystemer i Europa kunne gi en uoverkommelig teknologisk begrensning.

Distribusjonsteknikken forutsetter at TV-sendenettet er basert på interlacing. Dette er mindre sannsynlig for Åpen kanal sitt sendenett ettersom EBU anbefaler det progressive formatet 720P og NTV sannsynligvis vil følge denne anbefalingen. Vi har derfor valgt å se bort ifra denne teknikken i veiledningen.

Stereoskopisk video er den formen for 3D som enklest lar seg produsere, distribuere og vise. Ideelt sett skulle vi gitt en veiledning i en form for 3D der man avdekker større deler av bildet ved å endre betraktningvinkel. Dette kalles naturlig 3D og er slik vi oppfatter visuelle objekter i virkeligheten. For å kunne se rundt bevegelige objekter i en avbildning av virkeligheten, må man gjøre opptak av bildet venstre og høyre øye oppfatter ved flere

<sup>53</sup> Basert på produkter som selges i USA. Prisene er hentet fra <http://www.nuview.com>

kameravinkler samtidig. Dette er ikke i seg selv noe stort problem. Studier<sup>54</sup> har vist at man kan oppnå en tilfredsstillende naturlig 3D effekt i bredderetningen med 17 ulike vinkler, totalt 34 kameraer.

Problemet oppstår når man skal overføre og vise alle disse kameravinklene samtidig. Med 30Mbps per kamera (DV-strøm) vil en overføring på nettet legge beslag på en hel gigabit forbindelse, eller 30 gigabit med naturlig 3D også i høyderetningen. Med 34 ulike projiseringer horisontalt og vertikalt kreves det en oppløsning på displayet (for å vise standard PAL-bilde) på 24480 x 19584, totalt 480 megapiksler! (I praksis løser man dette med et lerret som reflekterer kun i den retningen projektoren lyser fra og setter opp 17 stereoskopiske projektorsett posisjonert på hver betraktningvinkel. Dersom man kun har en bruker av systemet er det mulig å vise en vinkel av gangen på en standard PAL skjerm styrt av bevegelses sensorer som måler mottakerens betraktningvinkel. CAVE benytter et slikt system på 4 skjermer). På sikt vil det være mulig å redusere datamengden ved overføring av multi-view 3D ved at synsvinkelen hos mottakeren genereres som virtuelle kameraer utifra et mindre sett av 3D-parametre. En slik objektbasert modell krever mye regnekapasitet og har i skrivende stund ikke latt seg realisere på dagens PC-er med tilfredsstillende bilderate.

Naturlig 3D er av disse grunner på forskningsnivå og Åpen kanal velger å veilede i stereoskopisk video for å kunne gjennomføre 3D-produksjon og distribusjon i praksis.

### *17.3 Stereoskopiske prinsipper*

For å oppfatte dybde i et bilde er vi avhengig av to typer "hint". Det vi kan oppfatte med ett øye (monocular cues) og det vi kan oppfatte med to øyne (binocular cues). (10% av befolkningen har ikke evne til å oppfatte dybde på grunnlag av mangler i dette settet med "hint"). Følgende dybdeparametre er vi i stand til å oppfatte med kun ett øye:

*Bevegelse pallakse:* Objekter som beveger seg i nærheten dekker en større andel av vårt synsfelt enn objekter med lengre avstand.

*Relativ størrelse:* Objekter vi husker som store oppfattes som nærmere dersom de dekker like stor del av synsfeltet som et objekt vi husker som lite.

*Lineært perspektiv:* Vinkelen mellom objekter avtar med økende avstand. (Parallele linjer møtes i horisonten, for eksempel toglinjer, veier, kabelstrekk etc).

*Aeriel perspektiv:* Objekter med sterke kontraster og farger oppfattes som nært. (Farger blir mindre intense og kanter mindre skarpe på avstand pga atmosfæriske forstyrrelser).

---

<sup>54</sup> Muntlig overlevert fra professor Leif Arne Rønningen på utveksling i Berkley University 2005

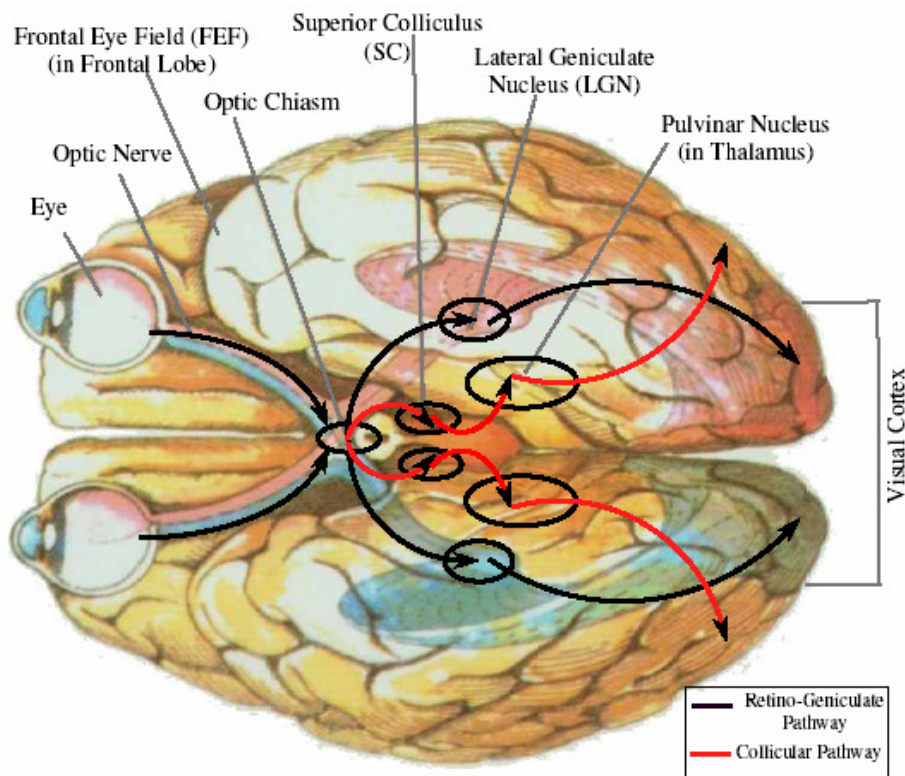
*Farge og lystone:* Et objekt med stor lys- og fargeintensitet oppfattes som nært.

*Skygge:* Et objekt som ligger foran et annet kan kaste skygge på det bakenforliggende.

Skyggen på objektet kan i seg selv også avdekke størrelse og dermed avstand.

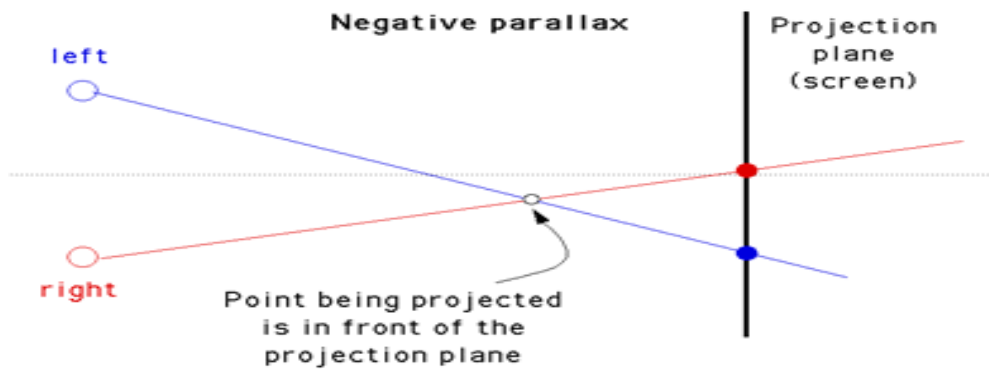
*Okklusjon:* Et objekt som dekker over et annet vil oppfattes nært i forhold til objektet som dekkes over.

Det er dybdeparametre vi kan oppfatte kun med to øyne som nyttegjøres spesielt i stereoskopisk video. Venstre og høyre øyne har noe ulikt perspektiv på grunn av avstanden mellom de i kraniet. Denne forskjellen er med på å skape et dybdebilde i hjernens visuelle system som er avbildet nedenfor.



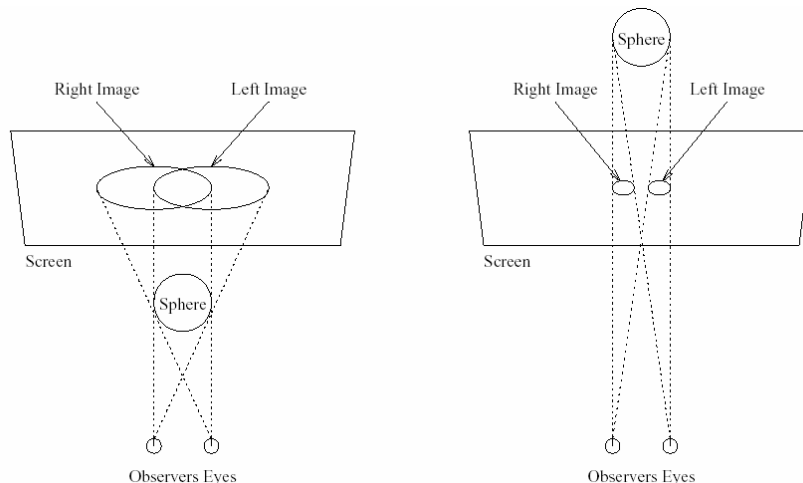
**Figur A-2: Menneskets visuelle system**

Dybdeeffekten kan gjenskapes ved at man gjør opptak av video fra kameraposisjoner som tilsvarer øynenes plassering i kraniet for deretter å projisere de ulike bildene separat til høyre og venstre øye (slik bildene oppfattes med personens øyevstand).



Figur A-3: Prosjeksjonsplan og parallakse

Øynene vil rettes inn mot fikseringspunktet. Dersom et objekt i venstre øye står til venstre for objektet i høyre øye oppfattes dette som om objektet befinner seg bak projeksjonsplanet. Dersom objektet i venstre og høyre øye er krysset, dvs at objektet i venstre øye står til høyre for objektet i høyre øye, vil det oppfattes som om objektet befinner seg foran projeksjonsplanet (negativ parallakse). Parallaksen kan justeres i flere stereoskopiske visningsprogrammer (f.eks. Amira og Stereoscopic player).



Figur A-4: Stereoskopisk dybdeillusjon

### 17.4 Tilgjengelig teknologi

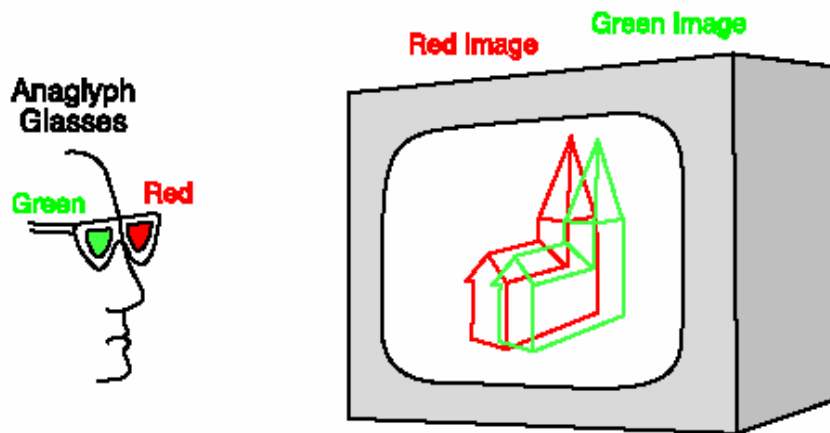
Utstyr for å gjøre opptak av stereoskopisk video er allment tilgjengelig. Man kan benytte seg av to helt enkle DV-kameraer som plasseres på et kamerastativ med en naturlig øyeavstand.

Redigering av stereoskopisk video kan også gjøres med konvensjonelt utstyr. Ved å klippe høyre og venstre kamera likt i hvert sitt videospor på en og samme tidslinje (f. eks i Adobe Premiere Pro) og deretter eksportere hvert spor hver for seg til fil eller tape, bevares synkroniseringen mellom de to kameraene.

I tillegg til redigering av video består etterarbeidet i å legge de to bildene i en og samme stereoskopiske videostrøm, såkalt stereoskopisk multipleksing. Dette er en noe krevende prosess med et begrenset utvalg av verktøy. Det mest vanlige er å bruke *Stereo Movie Maker* eller *Adobe After Effects* der man legger to videoer side om side eller over hverandre i bildet. (Stereo Movie Maker tar inn MPEG2TS direkte. Dette er det samme formatet som går ut til HDV-tapen. Videoen som eksporteres til tape mellomlagres som en .m2t-fil og må kun omdøpes til .m2v før man kan ta det inn i programmet. Ingen omkonvertering er nødvendig bortsett fra lagringen av stereovideoen i windowsmedia. Prosessen er langt mindre tidkrevende enn ved bruk av Adobe After Effects der man må komponere bildene manuelt.)

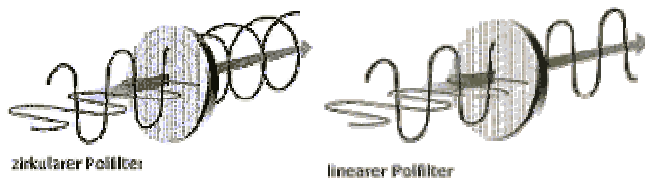
Når det gjelder visning av stereoskopisk video er det flere muligheter. Felles for disse er at man ønsker å projisere to ulike bilder på venstre og høyre øye som korresponderer med bildet fra hhv venstre og høyre kamera. Nedenfor er de ulike teknikkene beskrevet.

**Fargeseparasjon:** Dette er teknikken som brukes i stereoskopiske TV-sendinger i dag. Venstre bilde blir filtrert med rødt og høyre bilde blir filtrert med grønt (alternativt cyan). Bildene legges over hverandre og vises på en vanlig TV-skjerm. Med briller blir bildene så separert, der kun det ene bildet er synlig for hvert øye. Det røde filteret i venstre brilleglass slipper gjennom det røde bildet og sperrer for det komplementære grønne bildet. Grønnfilteret i høyre brilleglass slipper gjennom det grønne bildet og sperrer for det komplementære røde bildet. Denne teknikken er enkel å realisere i vanlige TV-nett og har akseptabel kvalitet dersom ikke fargegjengivelsen er av betydning.



Figur A-5: Prinsippskisse fargeseparasjon

**Separasjon med polarisering:** Her benyttes projektorer med ulike filtre for å skille venstre og høyre bilde fra hverandre. Begge bildene projiseres over hverandre på et reflekterende lerret som ikke endrer polarisasjonen ("Silerscreen"). Polarisasjonen angir retningen de elektromagnetiske bølgene forplanter seg i rommet og den er enten sirkulær eller lineær. På NTNU har vi filtre for å separere bildene på lineær polarisasjon og det er denne teknikken som er beskrevet under punktet "Stereoskopisk visning".



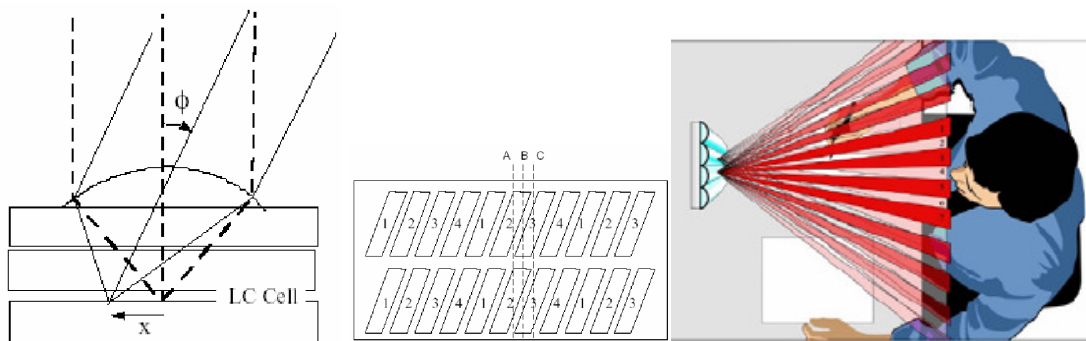
Figur A-6: Sirkulær og lineær polarisasjonsfiltrering

Venstre bilde filtreres med et polarisasjonsfilter som kun slipper gjennom lys forplantet diagonalt mellom øvre venstre og nedre høyre hjørne. Venstre brilleglass har et tilsvarende filter som slipper gjennom bildet med denne polarisasjonen men sperrer for bilder med motsatt polarisasjon. Høyre bilde polariseres motsatt med et filter som kun slipper gjennom lys forplantet diagonalt mellom nedre venstre og øvre høyre hjørne. Grunnen til at man ikke filtrerer på kun horisontal og vertikal polarisasjon er at LCD-projektorene som benyttes polariserer fargekomponentene ulikt i disse retningene. Grønt lys er polarisert vertikalt mens rødt og blått lys er polarisert horisontalt i projektoren. For ikke å filtrere bort noen av

fargekomponentene plasseres derfor polarisasjonsfiltrene 45 grader på alle fargekomponentene.

(”Stereo Bright” er en teknologi som nyttegjør dette på en mer intelligent måte. I stedet for at man reduserer 50% av lysstyrken ved filtrering, roterer man polarisasjonen +45 grader på venstre projektor og projiserer her den grønne komponenten fra høyre bilde. På høyre projektor roterer man polarisasjonen -45 grader og projiserer den grønne komponenten fra venstre bilde. Seeren har på seg de samme brillene. Ved å bytte grønn komponent på projektorene og benytte tapsfri rotasjon kan man dermed få større lysstyrke, men samme resultat. Denne teknologien er ikke aktuell dersom man har DLP-baserte projektorer.)

**Lentikulær separasjon:** Denne teknikken har hittil vært svært kostbar, men stadig reduserte produksjonskostnader på LCD-panel vil medføre at dette blir en mer utbredt visningsteknologi for stereoskopisk video. Den er også enklere å sette opp og man trenger ikke benytte briller. Her separeres høyre og venstre bilde i skjermen ved at bildene tildeles hver sine lyspunkter. LCD-panelet er dekket av små linser der de som sitter på lyspunktene for venstre og høyre bilde har ulik vinkel. Det venstre bildet rettes mot det venstre øyet og det høyre bildet rettes mot det høyre øyet på en person som sitter med riktig avstand i en sentrert posisjon foran skjermen. Et autostereoskopisk alternativ til denne teknikken er en ”grid” som legges foran pixel-matrisen. Den sperrer for pikslene i venstre kolonne når betraktningsvinkelen inn mot panelet tilsvarer vinkelen til det høyre øyet (og motsatt). Den lentikulære teknikken benyttes av bla Phillips som også har lyktes i å lage glidende overganger mellom ulike 3D-views ved hjelp av skråstilte piksel-slisser (Rønningen, 2006).



Figur A-7: Lentikulær multiview, prinsippsskisse (Rønningen, 2004)



**Shutter separasjon:** Dette er den billigste teknikken for å realisere 3D-visning i høy kvalitet (Nvidia Ti4200 koster ca 2500 kr). Venstre og høyre videostrøm tidsmultiplekseres og sendes til en helt vanlig skjerm med god oppfriskningsrate. Dette kan være en CRT projektor (som i NTNU sin Cave) eller en vanlig CRT PC-skjerm (LCD-skjermer kan ha for treg responstid). In Focus (depth Q) har utviklet en DLP-projektor som håndterer bilderater opp til 120Hz. Med denne vil det være mulig å vise et tilnærmet flimmerfritt 3D-bilde til store forsamlinger. Det som gjør at man oppnår 3D-effekten er et par aktive briller som lukker hhv venstre og høyre øye ved hjelp at et LCD display i brilleglasset. Når denne lukkerhastigheten korresponderer med hastigheten av vekslingen mellom høyre og venstre videostrøm ser man kun de bildene som er tiltenkt hvert øye. Flimringen kan være noe belastende for den som skal se på, men dersom man øker oppfriskningsraten til 120Hz vil det nesten ikke være merkbart. Med flere briller kan 3D-bildet deles av flere. (Problemet med høy oppfriskningsrate er dels prisen på skjermene som håndterer dette og dels tregheten i brillene som gir redusert lysinntak med økende shutterhastighet.)

## 17.5 Gjennomføring

For å realisere en stereoskopisk video må man gjennom tre prosesser: opptak, etterarbeid og visning. Ved å belyse viktige aspekter under hver av disse prosessene i et praktisk perspektiv kan teorien enkelt prøves ut (3D-videoproduksjon har frem til nå ikke vært tilgjengelig for folk flest, men det er god grunn til å tro at det vil bli det. Etter hvert som 3D-LCD skjermer blir billigere og stereoskopisk multipleksing blir integrert i redigeringsprogrammer, vil ikke 3D video kreve stort mer ressurser enn 2D video).

### 17.5.1 Stereoskopisk opptak

Opptak av stereoskopisk video kan gjøres med tradisjonelt videoutstyr. Det eneste som er spesielt er at man må benytte to kameraer samtidig. Kameraene monteres på et stativ med en kameraplate som har en kameraskruer til hvert av de to kameraene. Ideelt sett bør avstanden mellom kameraskruene (eg CCD brikkene) være lik avstanden mellom øynene.

Vanligvis benytter man små DV-kameraer til 3D-opptak ettersom disse er smale nok til å settes opp i øyeavstand. På NTNU har man valgt å erstatte de to DV-kameraene med HDV-kameraer. Dette gir større detaljrikdom (1440x1080 kontra 720x576) og mer nyansert fargegjengivelse (10 bit kontra 8 bit) samt jevnere film (høyere bilderate). De er mekanisk sett svært lik dagens DV-kameraer, og derfor ikke mye dyrere. (Tapen er den samme, men

MPEG-2 komprimeringen er basert på interframe-koding i stedet for DV sin intraframe-koding som gjør det mulig å få plass til mer bildeinformasjon).

### 17.5.2 Kameraoppsett

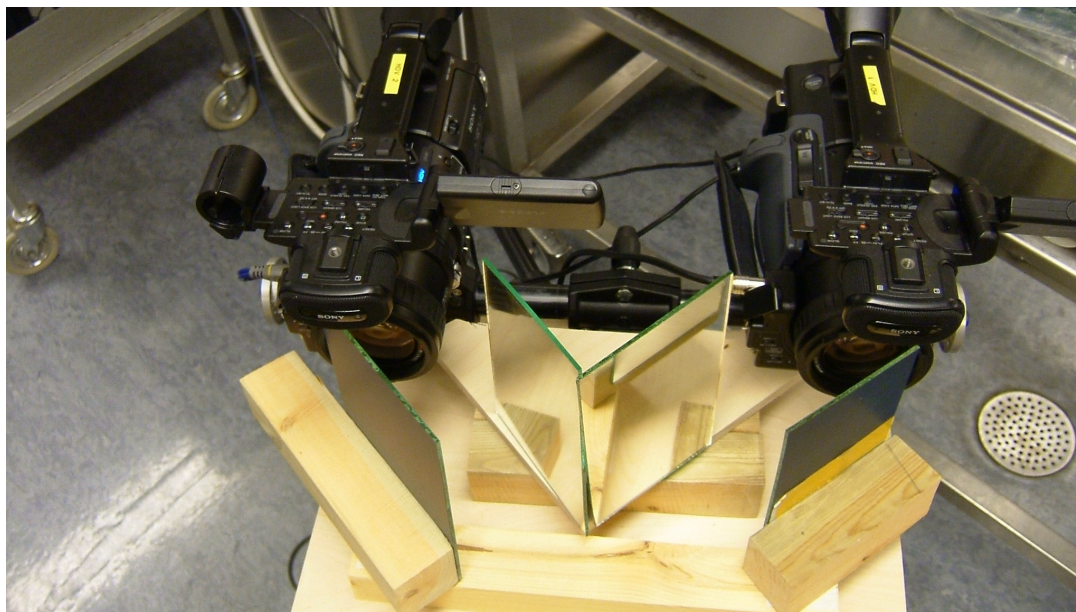
HDV-kameraene (Sony HVR-Z1E) som er tilgjengelig på NTNU er for brede til at man kan plassere de med øyeavstand på kameraplatten. Den økte avstanden mellom "øynene" må derfor kompenseres med en økt avstand til objektet man skal filme. Dersom avstanden til objektet er for liten vil dybden i 3D-bildet virke kunstig stor. Med 4,5 til 6 meter mellom kameraene og objektet viser det seg at dybden blir riktig. Dette krever at man benytter seg av zoom/tele på kameraet for å få anvendelige utsnitt (overvinklede bilder blir med det vanskelig å realisere i rom der det er lavt under taket).



**Figur A-8: Stereoskopisk kamera og monitorrigg (shutterglass-basert)**

Det har blitt bygget et periskop for å kompensere for øyeavstanden og dermed unngå den store avstanden til objektet, men kvaliteten på speilene var ikke tilstrekkelig (stor fargedispersjon). Man oppnår også en "heldig" effekt ved å benytte zoom/tele, bakgrunnen

kommer ut av fokus på grunn av liten dybdeskarphet og objektet i forgrunnen blir således tydeligere separert fra forstyrrende detaljer.



**Figur A-9: Utprøving av periskop for å redusere ”øyeavstanden”**

### **Monitoring i 2D med CRT**

Når man setter opp kameraene er det hensiktsmessig å bruke en monitor som kan ta inn to videokilder og har mulighet til å vise disse samtidig eller vekselvis.

Kameraene må vise så likt bilde som mulig. Helst bør kameraene være identiske spesielt mhp linse og CCD-brikke. Innstillinger som fokus, zoom, hvitbalanse og lukkertider bør også justeres tilsvarende. Det er enkelt å sammenligne disse dersom man har muligheten til å bytte mellom de to videokildene på monitoren slik man kan på de fleste profesjonelle videomonitorer. (Det finnes en slik på Midgard Media Lab ved NTNU på Gløshaugen)

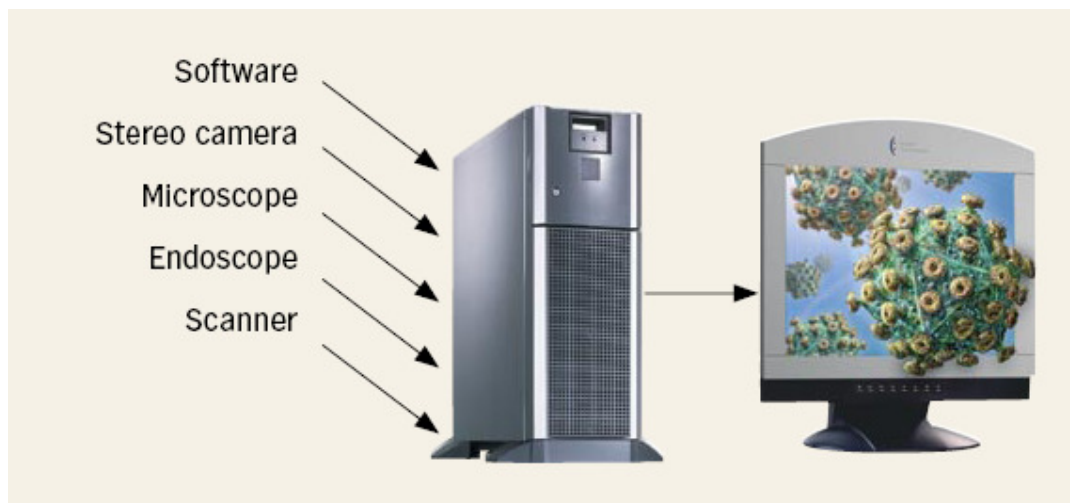
Denne funksjonen er også viktig for å justere vinkelen på kameraene. Fikseringspunktet, der parallaksen er null (i plan med lerretet), skal ha samme posisjon i høyre og venstre bilde. For hver nye innstilling/bilde må derfor kameravinklene tilpasses avstanden til objektet man skal filme. Jo nærmere objektet jo er kraftigere blir vinklene (som øyne som sjeler mot et nært objekt). Først kan man på øyemål vinkle de inn mot hverandre slik at krysningen treffer objektet. Deretter plasserer man et merke på monitoren for så å velge den andre kilden og justere vinkelen på kameraet man ser på til objektet treffer merket. Dersom objektet ikke har samme posisjon vil det i visning enten fremstå foran eller bak lerretet, og uten en god vinkel

mellom kameraene vil generelt det stereoskopiske bildet være anstrengende for seeren. Denne justeringen kan bli noe tidkrevende dersom avstanden til objektet endres ofte. Ideelt sett bør derfor panoreringen være motorisert og styrt av en avstandsmåler.

### **Monitoring i 3D med lentikulær LCD-skjerm**

Man kan også benytte seg av en monitor som kan vise begge bilder samtidig og aller helst i 3D. Det enkleste er i så fall å benytte lentikulære LCD skjermer der venstre og høyre bilde projiseres til hvert sitt øye. Firmaet Seereal fører slike med en akseptabel kvalitet tatt i betraktning prisen på 3200 euro (gjelder Cn-versjonen som ikke har tracking teknikk slik som C-i og C-s der linsene dreies for å tilpasse synsvinkelen). Disse skjermene (med unntak av konkurrerende visningssystemer til ca 300.000 kr) har ikke to kamerainnganger og er derfor avhengig av en PC for å kunne ta opp og generere 3D-bildet. Bildet som skjermen trenger består av to videostrømmer (venstre og høyre øye) som må ligge i annenhver kolonne (column interlacing). Programmet for å lage slike strømmer er gratis. Man installerer ganske enkelt "stereoscopic player" og "stereoscopic multiplexer" på en PC med to firewire innganger som kameraene er koblet til. Det er den stereoskopiske playeren som lager det nødvendige 3D-bildet ved at man velger (view -> viewing method -> column interlaced, left line first). "Stereoscopic multiplexer" er programmet som tar inn videostrømmene fra firewireinngangene. Dette programmet installerer seg i "stereoscopic player" og er tilgjengelig via menyen: file->live video->stereoscopic multiplexer. Instillingene her må stemme overens med instillingene på kameraet. HDV kameraet må konvertere sitt høyoppløselige format til DV (velges i i menyen på kameraet: i-link conversion -> on) for at multiplexeren skal kunne ta det inn (oppløsningen blir da 720 x 576). Selvom kameraet foretar konvertering på firewire/i-link utgangen vil opptak til tape være i HDV (1440 x 1080).

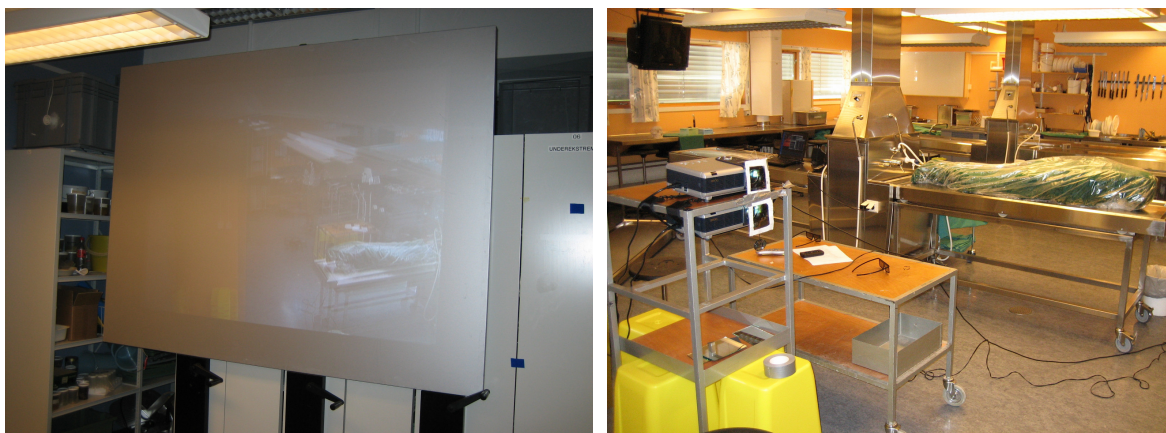




Figur A-10: Autosereoskopisk visningssystem fra Seereal Technologies.

### Monitoring i 3D med projektor

Dersom flere skal kunne se bildet i 3D samtidig er det vanskelig å få til dette med en lentikulær 3D skjerm ettersom betraktningvinkelen er svært begrenset. Man må sitte midt foran skjermen for at venstre og høyre bilde skal projiseres til riktig øye, de som sitter ved siden av vil ikke oppfatte et 3D-bilde. Med to polariserte projektorer kan man lage et stort 3D-bilde på et silverscreen lerret der alle med polariserte briller innenfor en stor betraktningvinkel vil kunne få korrekt dybdevirkning. Med denne visningsteknikken er det også enklere å realisere monitoring fra kameraene. Her kan man plugge kameraenes s-video signal direkte inn i hver projektor uten å gå via PC.



Figur A-11: Stereoskopisk visning. Projektorer med polafilter (venstre), silverscreen-lerret (høyre)

Det som er ulempen med dette systemet er at det tar stort plass og er vanskelig å sette opp. Projektorene må treffe likt på hele det projiserte feltet noe som i praksis er vanskelig å få til når de har ulik posisjon foran lerretet. (Kun mulig med optiske tilpasninger, men digital keystone kompenserer for det meste og finnes i de fleste billige projektorer)

Det er anbefalt å ha et slikt visningssystem for den som utfører handlingen i opptaket ettersom man da kan bevege seg fritt omkring. Kameraene har flere utganger så dette kan realiseres som et supplement til kameramannens eget monitoringsystem.

### **Monitoring i 3D med PC-skjerm**

Det er også mulig å få tak i skjermkort til PC som gjør at man kan benytte en vanlig PC-monitor til høykvalitets 3D-visning. Det er en billig løsning (2500kr) der man må ta på seg et sett med aktive LCD-briller som lukker for venstre og høyre øye vekselvis. I samme hastighet sender man hhv høyre og venstre bilde til skjermen slik at man kun ser det bildet man er ment å se på hvert øye. CRT skjermer har vanligvis en god nok oppfriskningsrate (mer enn 100Hz) til å vise dette relativt flimmerfritt. LCD-skjermene er foreløpig for trege, men også disse er i ferd å bli anvendelige med en responstid på mindre enn 8 ms. En ting man bør være klar over med slike visningssystemer er at brillene stjeler en del av lyset. Man bør derfor ta av brillene når man stiller blenderåpningen på kameraet. Det er også anbefalt å gjøre det når man stiller vinkelen mellom kameraene. Uten briller vil det se ut som om man har to halvgjennomskinnelige bilder der man enkelt kan sjekke at de to ligger over hverandre i punkter som skal ligge i "lerretplanet", altså hverken stikke ut eller inn av lerretet ved visning.



**Figur A-12: Høy kameraplassering for realisering av overvinklet bilde med nødvendig tele**

### **Monitoring i 3D med laptop**

Dersom man ønsker en bærbar 3D-monitoring kan det være kjekt å realisere dette i en laptop. Alt man trenger er et innstikkskort til PCMCIA med to firewire-innganger. Tekram (TR-1394C) leverer slike til ca 1000kr. Man benytter seg av ”stereoscopic player” og ”stereoscopic multiplexer” programmene som forklart under punktet ”monitoring med lentikulær LCD-skjerm”, men viewing method settes til ”Color anaglyph red – cyan”. Med dette vil det legges et rødt filter over bildet fra venstre kamera og et cyan filter over bildet fra høyre kamera. Man kan dermed separere de to bildene og oppnå en 3D effekt ved å benytte briller med tilsvarende komplementære farger i glassene (Slike briller kan man få tak i ved å kjøpe 3D-filmer som Shrek eller småspioner der det følger med 4 glass i hver utgivelse). Fargegjengivelsen blir ikke god men det er mulig å benytte dette som en dybdereferanse. Alternativt fører Sharp en 3D-laptop med en litt dårlig lentikulær LCD skjerm (Viewing method settes til Sharp 3D display)





**Figur A-13: : Portabelt stereoskopisk visningssystem**

### *17.6 Noen tips under opptak*

#### **Lys**

Det er ofte ønskelig med minst mulig skygger på objektet man skal filme. Man oppnår dette ved å skape et diffust lysfelt emitert fra et størst mulig område. Enkleste og billigste måten å sørge for dette er å sende lyset på en stor flate og la det reflekteres med god spredning. Man kan f.eks bruke et hvitt takpanel, men med det får man også lite kontroll over lyset. Den mest anbefalte teknikken er å benytte soft box (fra f.eks Chimera) som monteres på vanlige lyskastere (som bør stå i ”flod”) eller store felt med fargenøytrale lysstoffrør (Kino Flo).

For å belyse noen områder litt ekstra trenger man også noen små spotter. På disse bør man montere såkalte ”låvedører” for å begrense lysets spredning. Det kan også være ønskelig å myke opp lyset slik at heller ikke spottene kaster skygger. Dette kan enkelt gjøres ved klype fast (bruk treklyper, ikke plast som kan smelte!) en diffunderende duk (frost eller spun) på låvedørene. Med tre spotter kan man belyse objektet bakfra og gi det en spiss som forsterker

dybdevirkningen samt sørge for at man både har det myke fyll-lyset og hovedlyset i front. Dersom man ikke klarer å holde sollyset ute får man et blandingslys det kan være vanskelig å kompensere for i etterarbeidet. Man er da nødt til å legge et blått filter foran lampene slik at de får samme fargetemperatur som sollyset. Dette ”blåstikket” kompenseres for ved å stille hvitbalansen til ”utelys” på kameraet (alternativt ved å legge på komplementærfargen oransje i etterarbeidet.).



Figur A-14: : Lysrigg med Chimera og spotlys

## Lyd

For å gjøre opptak av detaljelydene og begrense romlyden er det en fordel med en retningsstyrt mikrofon som retter seg inn på det området der lyden genereres. Avstanden til objektet bør være kortest mulig ettersom lydintensiteten avtar med kvadratet av avstanden og det vil i praksis øke støynivået relativt. Ideelt sett burde man ha en mikrofon med hyperkardioide-karakteristikk (som f.eks. Sennheiser MKH 416). Kommentar til handlingen eller dialoger kan man gjøre på trådløse mygger (f.eks. Sony UWP C1), men det er anbefalt at det meste legges på i etterarbeidet for ikke å forstyrre detaljelydene (Kan spilles inn via kameraet som har balanserte lydinn ganger i motsetning til lyd kortene i de fleste PC-er). Nivået inn på kameraet bør være høyest mulig men ikke overstyre. Dette justeres automatisk (kan medføre ”pumping” i støynivå) eller manuelt på baksiden av HDV-kameraet.



Figur A-15: : Trådløs mikrofon med sender og mottaker (venstre), retningsstyrte mikrofoner (høyre).

### Bildekomposisjon

Når man gjør opptak av 3D-video er det noen ekstra hensyn man må ta i komposisjonen. For enkelt å kunne oppfatte dybden bør hele objektet være i bilderammen til både venstre og høyre kamera. Dette er vanskelig i praksis, men man bør i hvert fall sørge for at det som skal være i fokus ikke er i nærheten av høyre eller venstre bildekant slik at det kun oppfattes av hhv venstre eller høyre øye. Det er også en fordel med en mest mulig diffus bakgrunn (ikke reflekterende eller gjennomskinnelig, gjerne en matt duk) ettersom denne i alle tilfeller må ha deler av seg utenfor bilderammen og dermed kan forstyrre dybdeoppfattelsen av objektet i forgrunnen. Spesielt gjelder det at de delene av objektet som skal stikke ut av skjermen er vanskelig å oppfatte dybden på dersom begge bildene på denne delen ikke er komplette. Generelt bør man derfor legge hele objektet inn i 3D-bildet ved stille på vinkelen mellom kameraene. På de punktene der venstre og høyre bilde ligger eksakt over hverandre er objektet i visningsplanet. Når man utfører justeringen kan man legge et spisst objekt på et slikt punkt og vha en monitor som viser begge bilder samtidig (evt. svitsjer mellom de) sørge for at spissen treffer på samme punkt på skjermen. I 3D vil det da se ut som om objektet går fra dette punktet i visningsplanet og innover i bildet. På bildet nedenfor er dette vist med fargeseparasjon mellom de to bildene (Playback -> viewing method -> Anaglyph red-cyan).

Her benyttes stereoscopic player med stereoscopic multiplexer for å vise video i sanntid fra kameraene. Pilen angir punktet som ligger i visningsplanet. Benytter man separasjon med shutterglass eller polarisasjon vil bildene legges seg over hverandre og være halvgjennomskinnelige. Med LCD skjerm som har lentikulær separasjon kan man ikke justere bildet på denne måten når viewing method er satt til "vertical interlace". Skjermen vil da projisere ulike bilder til venstre og høyre øye. Dersom viewing method settes til "anaglyph red – cyan" kan man stille på kameraene selv om man har en slik 3D-skjerm (Evt kan man endre betraktningvinkel ut over skjermens/linsenes "sweet spot" der 3D effekten forsvinner).



Figur A-16: : Fargeseparert stereoskopisk bilde som viser 0-parallaksen (i ett med lerretplanet)

### Synkronisert opptak

For at bevegelsene fra høyre og venstre kameravinkel skal være i takt med hverandre må man ha en synkroniseringsmarkør på tapen. På profesjonelt utstyr er det mulig å styre tidskoden på det ene kameraet med det andre slik at de korresponderende bildene på de to tapene blir tildelt samme navn. Dette er den mest anbefalte måten å synkronisere kameraene på og gjøres praktisk ved å koble en cox fra "TC out" til TC in", samt stille spillerne som "master - slave". I tillegg kobles de sammen med et videoreferansesignal som angir når kameraene skal begynne "tegningen" av hvert delbilde (framesync).

Dersom man bruker konsumentutstyr er det likevel mulig å synkronisere kameraene.



Man kan ganske enkelt klappe foran kameraet og lese opp hvilken scene, slate (kameraposisjon) og take (tagning) det er i forkant eller etterkant av opptaket.

Klappet bør være enkelt å finne frem til (hold hendene en stund foran kameraet før du klapper) og finne sted innenfor en frame (et enkeltbilde i videosekvensen). Når man etterpå skal spille inn klippene på PC for redigering kan klippene begynne med dette klappet for høyre og venstre kamera og dermed er begge bildene enkelt å få i takt.

HDV kameraene til NTNU er semiprofesjonelle og har tildels de samme mulighetene som profesjonelt utstyr. Her synkroniseres kameraene ved at man kobler sammen firewire/i-link terminalene. På kameramenyen velger man in/out rec -> ext rec ctrl -> rec ctrl mode -> synchronous. Velg stby command og sett recording mode til rec pause. Nå vil starttidspunktet av opptak/pause og verdien på tidskoden være den samme på de to kameraene.

Dersom man ønsker å bruke firewire utgangen til monitoring kan det være hensiktsmessig å heller starte eller stoppe de samtidige opptakene med en fjernkontroll.

(Eller eventuelt kjøpe en firewire-split slik at man kan synkronisere mellom kameraene med den ene firewire kabelen og se på opptaket i 3D med den andre firewire-kabelen)

For å synkronisere tidskodene må den settes til å løpe med tiden fremfor rammenummer i opptak (fjernkontrollen kan ikke styre kameraene rammenøyaktig). Dette gjøres på kameramenyen tc/ub set -> tc run -> free run. Det må også angis lik verdi på tidskodestart for de to kameraene under tc preset. Tc make stettes til regenerate. Selv om de to tidskodene ikke begynner å løpe samtidig vil samtlige klipp være synkronisert med en fast offset som enkelt kompenseres for i redigeringsprogrammet.

### **Logging**

Etter hvert opptak bør det skrives en log, hvor tidskoden til innpunktet og utpunktet, samt scene/slate/take og en kort beskrivelse og kommentar bør noteres. Denne loggen brukes til å gjøre etterarbeidet enklere. I etterarbeidet kan man for eksempel bare ta inn de klippene som er notert at er gode.

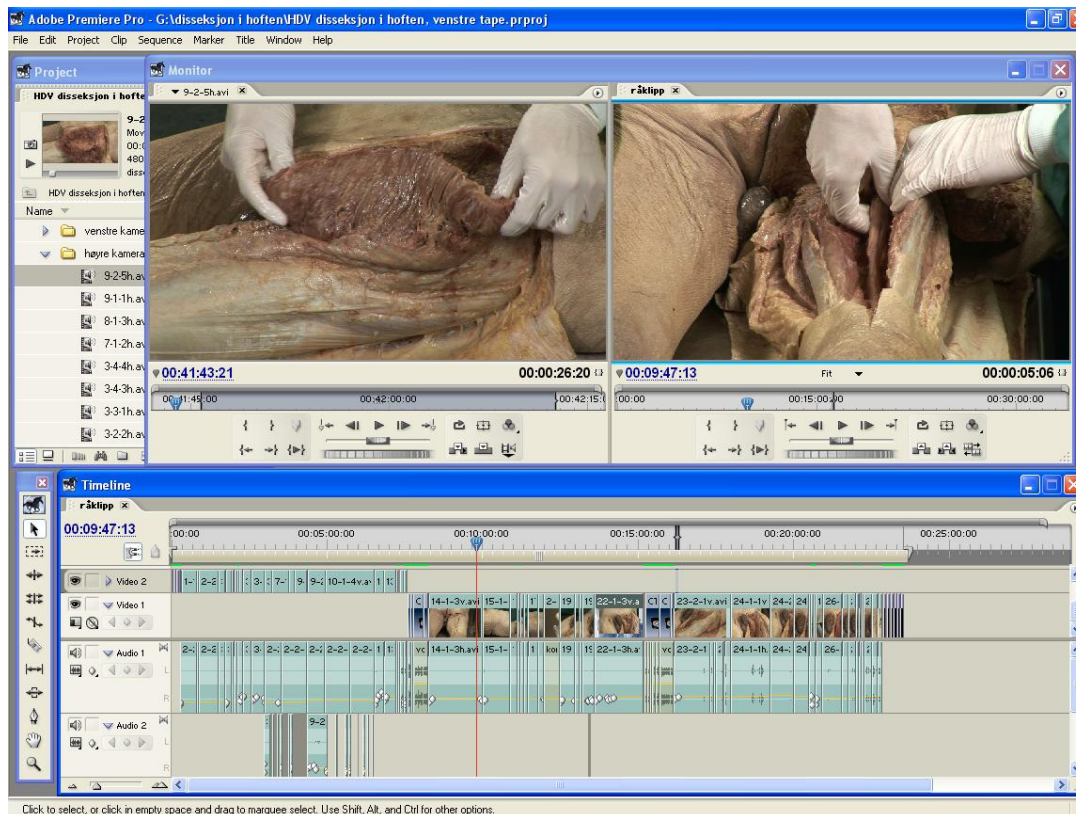
## 17.7 Stereoskopisk etterarbeid

Den første delen av etterarbeidet kan gjøres med helt vanlig redigeringsprogram for video. Når man klipper stereoskopisk video må begge kilder klippes likt ellers er alt som for vanlig video. Utfordringen i etterarbeidet består i å få begge kilder vist samtidig. Til dette brukes vanligvis Adobe After Effects eller Stereo movie maker, der bildene legges over hverandre med ulike fargefiltre (fargeseparasjon) eller side om side som også er egnet for enkel toskjermsløsning (Alternativt over hverandre, above/below, for høyoppløselige bilder som skal komprimeres i windowsmedia encoder).

### 17.7.1 Redigering

Redigeringen er som for vanlig video med unntak av at man må klippe på to videospor samtidig. Menyvalg er forskjellig i de ulike redigeringsprogrammene men prinsippene er stort sett de samme. Vi har her brukt Adobe Premiere (Merk at denne må oppgraderes til versjon 1.5.1 for å ha støtte for HDV).

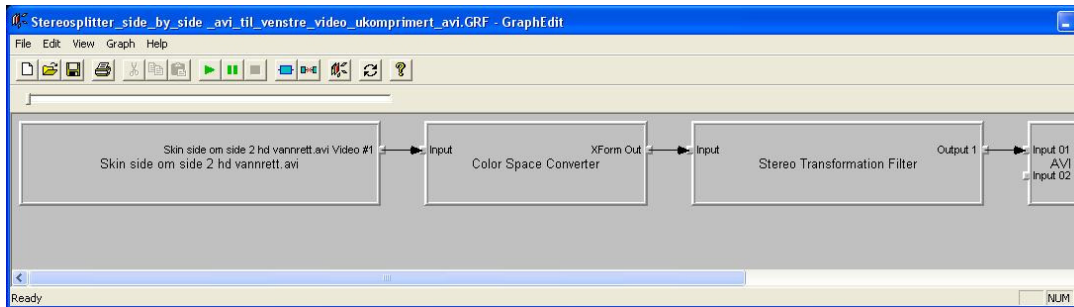
Først logger man tapene, dvs man angir inn og utpunktet for hvert klipp (Dersom de ikke er synkronisert med tidskode, er klippet ditt innpunkt). Innpunkt og utpunkt bør være notert i en log. Når man har logget det ene kamera, kan man for å spare tid kopiere denne loggen, forandre navn på hvert klipp fra for eksempel "h" til "v", og forandre navn på tape. Da gjelder denne loggen også for det andre kameraet. Dette er selvfølgelig under forutsetning av at de to kameraene er synkronisert, dvs har samme tidskode. Når alle klippene er logget skal de spilles inn fra tapen (Batch capture). Deretter legger man klippene fra de to kameraene ned på tidslinja i hvert sitt videospor. Når man klipper på sporene er det viktig at man klipper likt på begge sporene.



Figur A-17: : Stereoskopisk redigering i Adobe Premiere

### 17.7.2 Konvertering av stereovideo med graph edit (demultipleksing)

Stereovideoer er som regel side om side. Dette kan deles i to separate strømmer og integreres med øvrig materiale. Ved å sette sammen directshow filterene AVI/wav file source, stereoscopic transformation filter (properties: input=side by side, left first, output=mono left/right) , AVI mux og file writer kan man hente ut venstre og høyre video enkeltvis og legge de inn sammen med videoen i Adobe-premiere. Merk at det siste leddet, file writer, må erstattes for å lagre til ny fil med et nytt navn(Graph -> import filter-> direct show-> file writer). Ønsker man å demultiplekse en annen stereostrøm byttes det første leddet ut med en AVI/wav file source (Graph -> import filter-> direct show-> AVI/wav file source). Det er og mulig å gjøre opptak direkte fra DV-kameraene til en stereoskopisk ukomprimert videofil (side by side) med et ”stereo. mux. filter”.



Figur A-18: : Filterstruktur i Graph edit

### 17.7.3 Eksportering av video fra Adobe Premiere

Når man skal lage de to videoene til venstre og høyre ”øye” må man aktivere viewing-symbolet på hhv høyre og venstre videospor og eksportere tidslinja til hver sin tape (File -> eksport -> eksport to tape) eller fil (File -> eksport -> eksport to file). Eksportformatet bør være AVI og videocodec Indeo video 5.1 (1440x1080 dersom materialet er i HDV og 720x576 dersom materialet er i DV) med deinterlacing og uten keyframes (mest mulig ukomprimert). Merk at dette stemmer hvis man skal inn i After Effects. Hvis man skal bruke Stereo movie maker trenger man kun kjøre til tape. Da vil det lagres to midlertidige .m2t filer av Adobe Premiere. Disse bør omdøpes til .m2v, og kan da brukes videre. Lyden må i det sistnevnte tilfellet eksporteres ut for seg selv da Stereo Movie maker ikke kan hente lyden fra m2v-fila. Dette gjøres ved å markere ut viewing-symbolet på videoene og eksportere lyden som wav i AVI.

### 17.7.4 Komposisjon

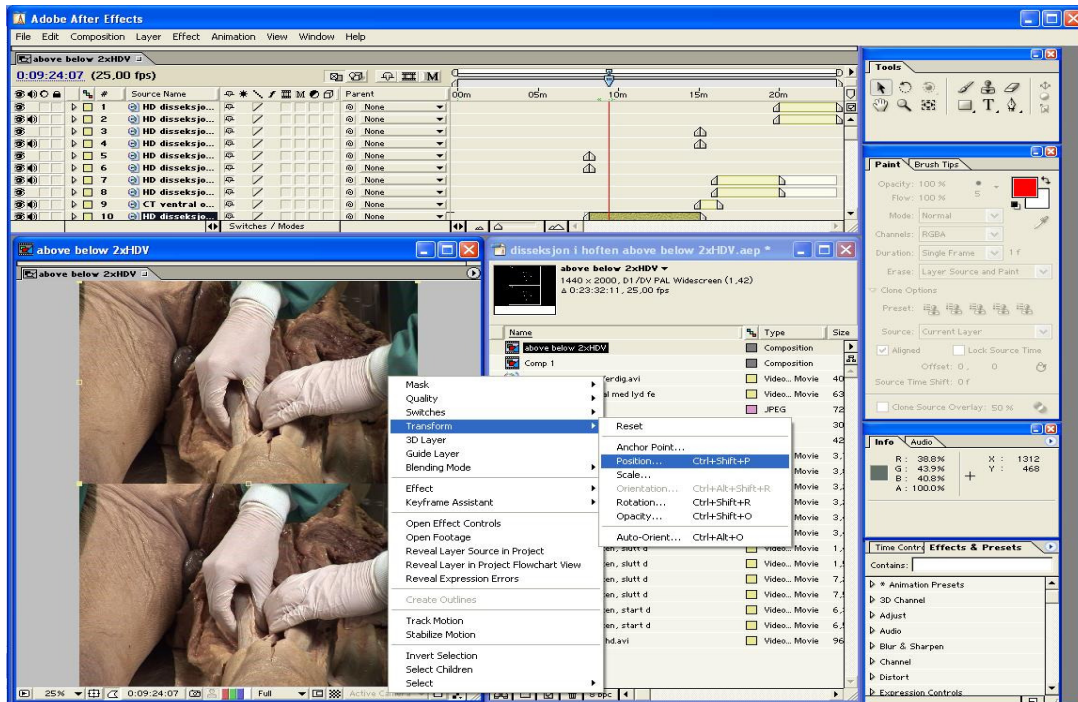
Det er i komposisjonen at stereo/3D-videoen blir laget. En rom-multipleksing sørger for at venstre og høyre videostrøm kan vises samtidig og synkront. Når man skal sette sammen bildene side om side (lavoppløselig video og toskjermsløsning) eller over hverandre (nødvendig for høyoppløselig video dersom denne skal komprimeres med programmet windows media encoder) kan man benytte Adobe After Effects eller Stereo movie maker.

**I After Effects:** Her lager man først et prosjekt, importerer videoene i prosjektvinduet (meny på høyre museknapp) og åpner et komposisjonsvindu (Composition -> New composition). Her settes størrelse 1440x576, DV PAL aspectratio og framerate på 25 dersom man skal sette sammen to DV-strømmer, eller størrelse 1440x2000 (reduksjon i høyden skyldes begrensning i codecen som brukes ved eksportering), 1,42 aspectratio og framerate på 25 dersom man skal sette sammen to HDV-strømmer. Fra prosjektvinduet trekker man videofilene over i komposisjonsvinduet slik at de står side om side dersom kilden er DV eller over hverandre



dersom kilden er HDV. I det sistnevnte formatet må bildene krympes i høyderetningen og strekkes i bredden for å tilpasses vinduets størrelse (HDV har en annen aspect ratio på 1,33, men dette kan man ikke velges i After Effects). Det kan her være praktisk å posisjonere venstre og høyre bilde i midten av hhv øvre og nedre halvdel før man strekker det ut. Ved å høyreklikke i komposisjonsvinduet og velge transform -> position kan bildene sentreres til hhv (720,500) og (720x1500).

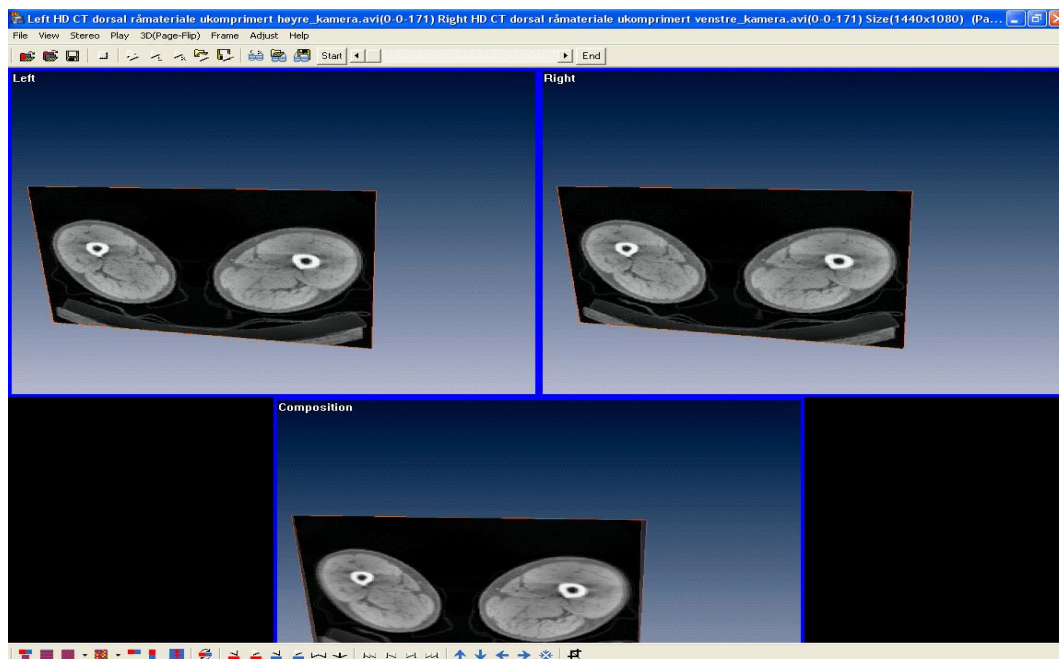
Nå gjenstår det bare å eksportere stereovideoen til et anvendelig format. Det gjøres ved å velge file -> eksport -> AVI og skrive inn et filnavn samt sette video codecen til none (ukomprimert) eller indeo 4.1 dersom du har begrenset plass på harddisken. I det sistnevnte tilfellet bør man ikke bruke keyframes og tillate en høyest mulig bitrate.



Figur A-19: : Stereoskopisk komposisjon i After effects.

**I Stereo movie maker:** Velg file -> open left/right movies, og velg først venstre så høyre .m2v eller .vob fil. Huk av for deinterlace. For å lagre stereovideoen velges File -> Save stereo movie. Huk av for above-below, og skriv inn et filnavn. Her finnes også muligheten resize, hvis det ønskes en mindre videostørrelse (f eks 2xDV i 1440x576, som kan settes side by side. Merk at resize innstillingen er relatert til størrelsen på kilden og i dette tilfellet derfor må være 720x576). Dette går utover kvaliteten men blir lettere å spille av. Dersom videoen

skal spilles av på Nvidia sine shutter briller (på skjermkortet Ti4200) må kildematerialet størrelse begrenses til 1024x768. Denne størrelsen spiller også jevt på visningsutstyret til institutt for laboratoriemedisin (Wildcard kort og cyviz projektorsystem). Trykk ok, og som kompressor velges Microsoft windows media video 9 (standardinnstillingene for denne er bra nok for lokal visning). Dersom man vil lage en stereovideo som skal streames på nett anbefales det at man heller kjører ut videoen mest mulig ukomprimert (velg kompressor "uncompressed" dersom man har plass på disken eller indeo 5.11). Legg merke til at Stereo Movie Maker ikke henter ut lyden fra .m2t.fil (må omdøpes til .m2v eller .vob) som ble mellomlagret under eksportering til tape. Stereovideoen som lagres deles også opp i 2GB store segmenter. Disse må settes sammen med lyden i Virtual Dub. Det sistnevnte kan unngås dersom man istedet for å eksportere til tape lager en DVD i adobe premiere. VOB-filene som ligger på DVD'en kan Stereo Movie Maker lese lyden fra og lyden legges i venstre eller høyre lydkanal avhengig av om det var DVD for venstre eller høyre kamera den ble hentet fra.



Figur A-20: : Stereoskopisk komposisjon i Stereo Movie Maker

Stereo movie maker er enklere og mindre tidkrevende (programmet er også gratis i motsetning til After Effects).

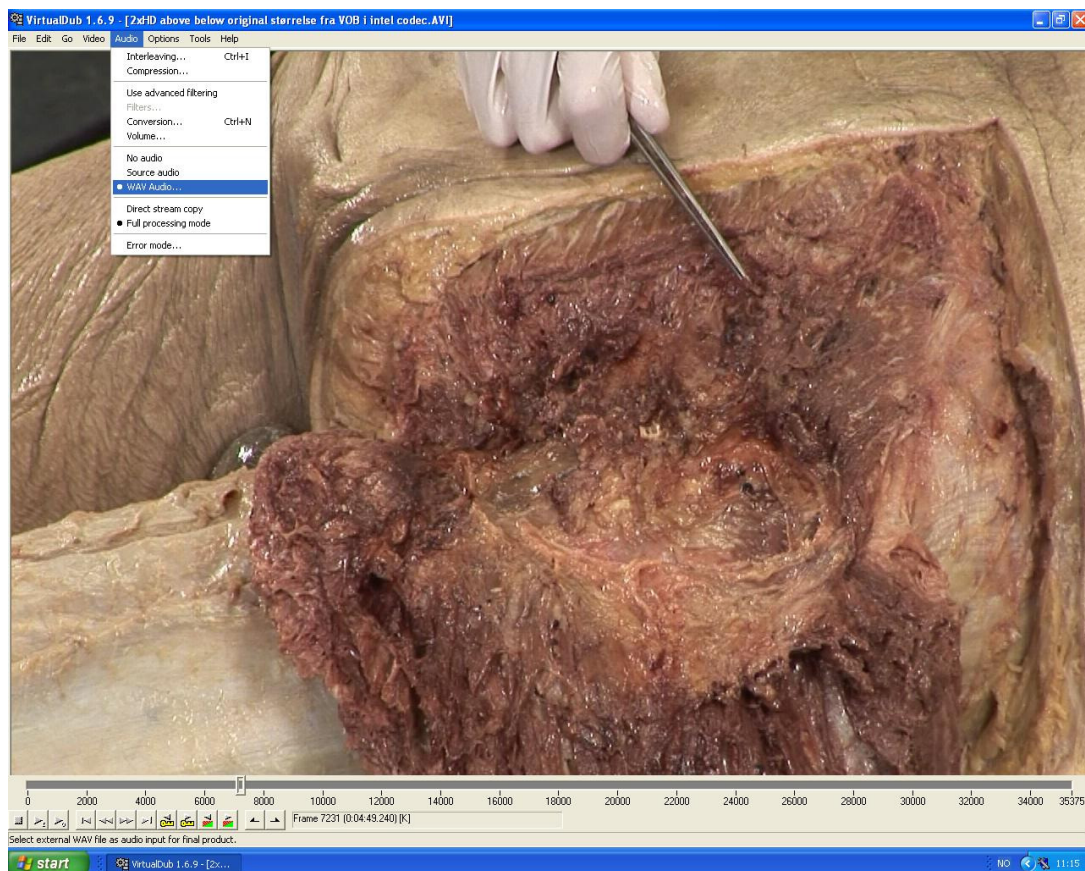
### 17.7.5 Sammensetning og kompresjon

Dersom du benyttet Stereo movie maker kan stereo videoen nå være delt opp i segmenter med 2GB størrelse hver. Lyden må da settes sammen med disse i en komprimert strøm som skal kunne spilles av jevnt på en PC. Pålegging av lyd er ikke nødvendig dersom man henter inn materialet fra en brent DVD (evt VOB-fil) der Stereo movie maker kan lese lyden direkte. Komprimering kan utføres enten av Stereo movie maker, Virtual dub eller programmet Windows media encoder. Man må ha installert Windows media VCN for at windows media 9 codecen skal være tilgjengelig fra menyen i programmene Virtual dub og Stereo movie Maker. Komprimering i Stereo moviemaker anbefales dersom filen skal spilles lokalt (stor fil, variabel bitrate) og komprimering i Virtual dub eller Windowsmedia encoder anbefales dersom filen skal kunne streames (Med konstant bitrate, CBR-modus, er det lettere å overføre videoen jevnt på nett). Oppløsning må reduseres til 1440x2000 i Windows Media encoder programmet, men dette er ikke nødvendig i Windowsmedia plugin-modulen, som ligger i virtual DUB. Dersom den endelige komprimeringen ikke utføres i Stereo movie maker, bør videoen hentes ut mest mulig ukomprimert fra dette programmet etter komposisjon (Anbefaler å bruke codecen indeo video 5.11). Sammensetning må uansett utføres av Virtual dub (Programmet kan lastes ned gratis fra nett) og dersom man heller ikke ønsker å utføre den endelige komprimeringen her lages en sammenhengende video av segmentene samt at man legger til lyd som forklart under. Videoen kan da komprimeres i programmet windowsmedia encoder hvor man har full kontroll over WM9 sine codec-innstillinger.

**Sammensetning og kompresjon med Virtual dub:** Man henter inn første videosegmentet ved å velge file -> open video file -> <segment med lavest nummerering>. Man henter inn det neste segmentet ved å velge -> Append AVI Segment -> [segment med nest lavest nummerering]. Dersom "autodetect additional segments by filename" er huket av vil den finne de neste av seg selv og legge disse etter hverandre.

For å legge på lyd (Ikke nødvendig dersom råmaterialet var en DVD i Stereo Movie Maker), velg Audio -> WAV audio -> <filnavn på lyden som ble eksportert fra premiere>. Velg audio -> direct stream copy. Velg også video -> direct stream copy dersom komprimeringen allerede er utført i Stereo Movie Maker eller man ønsker å utføre komprimeringen senere med f eks windows media encoder programmet. (Setter kun sammen segmentene og lyden uten å endre innholdet. Går derfor relativt fort). Dersom man ønsker en video som skal være streambar velger man her Video -> compression -> Microsoft Windows media 9. For å lagre

filen velges File -> Save AVI. Nå settes lyden (gjelder ikke for DVD/VOB-kilder) og videosegmentene sammen i en ny AVI-fil.

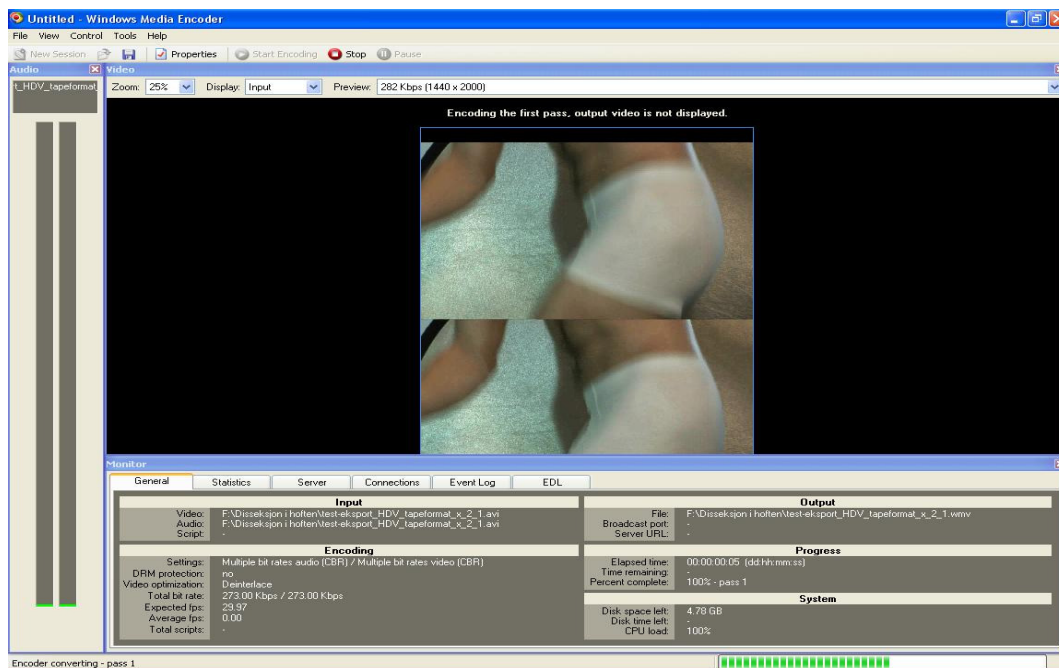


**Figur A-21: : Dubbing i Virtual dub**

Dersom du benyttet After effects er videoen ferdig satt sammen med alle de nødvendige elementene. På grunn av den høye bitraten i den eksporterte stereovideoen fra After Effects er det vanskelig å spille den av jevnt på selv en rask PC. Det er heller ikke mulig å streame denne videoen som kan være ønskelig dersom man vil at flere skal ha tilgang på den. Med Windows Media encoder kan man lage en stereovideo med noe redusert fargekvalitet, men oppløsningen er tilnærmet den samme som råmateriale selv når bitraten reduseres til 5 Mbit/sek.

(Windows Media Encoder programmet begrenser bildestørrelsen til 2000x2000. En HDV strøm er 1440x1080 som satt over hverandre må begrenses til 1440x1000, totalt 1440x2000. Dette kan allerede være gjort i After Effects på grunn av begrensninger i Indeo 4.1-codecen).

**Kompresjon med Windows Media Encoder:** I "new session"-vinduet velger du convert a file, henter inn stereofilen, spesifiserer kontribusjonsform windows media server (streaming), high definition video, CD quality audio og trykker ut "begin converting when i click finish". Før du begynner komprimeringen må du spesifisere video size i properties (1440x2000 for HDV og 1440x575 for DV-video) og processing -> deinterlace dersom dette ikke allerede er gjort med råmaterialet du tok inn i After Effects eller Stereo Movie Maker (ikke nødvendig dersom denne instruksjonen er fulgt hele veien). Trykk start encoding og etterhvert blir det mulig å spille av stereovideoen jevnt på en stereoskopisk player.



Figur A-22: : Komprimering med Windows media encoder

## 17.8 Opplasting og streaming

For at videoen skal være tilgjengelig på flere plasser i nettet samtidig bør man legge videofilen på en streamingserver. Det kan enkelt gjøres ved å laste opp filen til serveren med en FTP-klient og legge ut en link til denne fila på en offentlig nettside.

Adressen til denne er [openchannel.item.ntnu.no](http://openchannel.item.ntnu.no). Her kan man bruke en FTP-klient med et brukernavn og passord som kan tilsendes på forespørsel fra Erik Vold på [adm@openchannel.no](mailto:adm@openchannel.no). Når filen ligger på serveren er den streambar med følgende link:

<mms://wmstrem.openchannel.no/<orgnavn>/<filnavn.wmv>>



### 17.9 Stereoskopisk visning

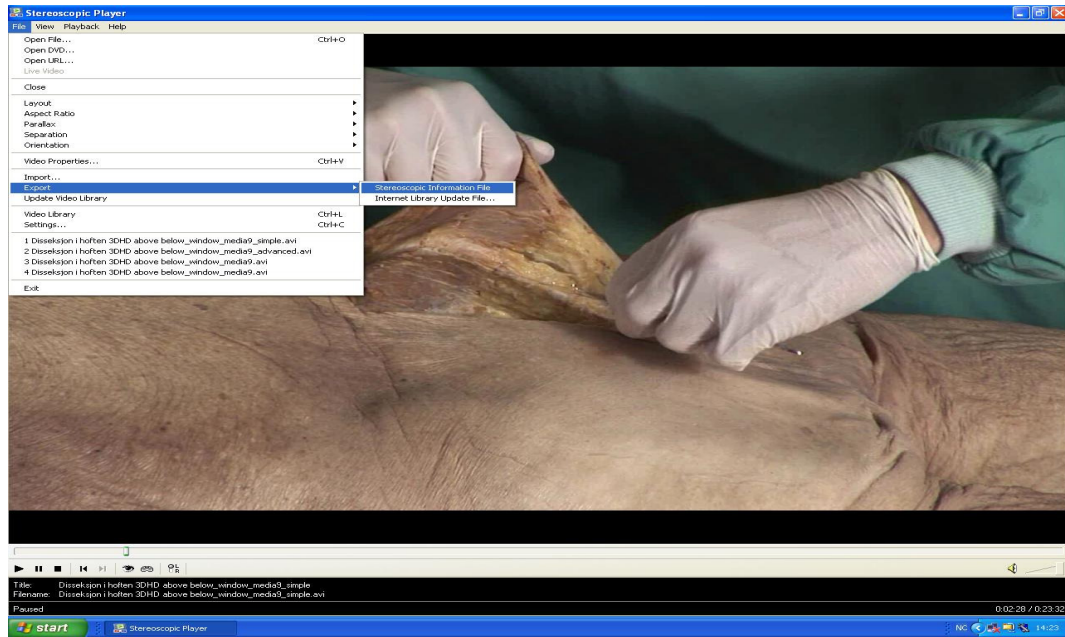
For visning benyttes et toskjermkort som splitter et videosignal til to projektorer. For å lage en synkronisert dual-strøm av den stereoskopiske videoen spilles den av i et program som heter Stereoscopic player. Programmet kan lastes ned gratis fra denne siden:

<http://mitglied.lycos.de/stereo3d/>

De Stereoskopiske bildene ligger side om side eller over hverandre, så file -> layout (kun tilgjengelig på menyen der den stereoskopiske filen er åpnet) må settes til "side by side, left image first" (aspect ratio 4:3 er mest vanlig når kilden er lavoppløselig DV og aspect ratio 16:9 er mest vanlig når kilden er høyoppløselig HDV). eller "over/under, left image on top" dersom stereovideoen ble komprimert med programmet windowsmedia encoder. View -> Viewing method i Stereoscopic player settes til "Dual screen output".

(Dersom man ønsker å se stereovideoen på shutterbriller er anbefalt oppløsning 1024x 768 på Nvidia Ti4200. Viewing method må da settes til Nvidia stereo driver. Fullskjermssbilde får man ved å dobbelklikke på videovinduet. Det hender at playeren her kjører ut feil bilde først og at dybden derfor blir feil. Trykk i så fall på F7 for å bytte på rekkefølgen av venstre og høyre frem til du får et 3D-bilde som er behagelig å se på.)

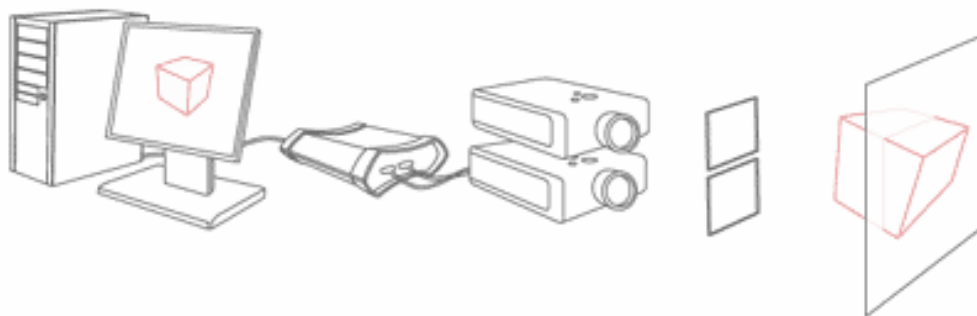
Merk at i Stereoscopic player har man muligheten til å lagre alle innstillinger, ved å velge file -> export -> stereo information file. Når man trykker på denne vil videoen senere åpnes med alle de rette innstillingene med unntak av viewing method som må stilles manuelt (Denne tilpasser playeren til visningsutstyret som vanligvis er fast i motsetning til innstillinger relatert til kilden).



**Figur A-23: : Visning med Stereoscopic player**

Skjermkortet (tegnet inn som ekstern boks på figuren nedenfor) henter ut bildene fra hvert kamera i venstre og høyre halvdel av stereobildet (alternativt øvre og nedre halvdel) og spiller av disse synkront på hver sin utgang. På de to utgangene kobles det til projektorer som dermed viser bildet fra venstre eller høyre kamera. Foran projektorene er det plassert to ulike filtre som slipper gjennom lys polarisert i hver sin retning.

De motsatt polariserte bildene korresponderer med hvert sitt polariseringsfilter i 3D-brillen. På denne måten sperres bildet fra den ene prosjektoren mens bildet fra den andre vises (motsatt for hvert brilleglass). Bildene fra venstre og høyre kamera blir med dette separert når de når frem til øyet og vi kan få en dybdepersepsjon. (Forutsetter at lerretet er av typen ”silverscreen” som bevarer polariseringen).



**Figur A-24: : Stereoskopsk visning med polarisasjonsfiltre, prinsippskisse**

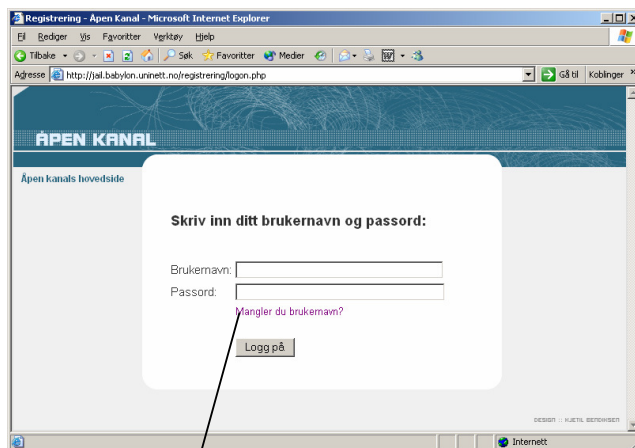
## 18 B - Appendiks: Veiledning i bruk av Åpen kanal

### 18.1 Bruk av opplastingssystemet

Opplastingssystemet til Åpen kanal består av en filtjener og en nettjeneste. Filtjenesten tilbyr muligheten for å laste opp innslag mens nettjenesten tilbyr et grensesnitt hvor du kan legge inn informasjon om innslagene du har lastet opp. Du kan også se på ferdigkomprimerte innslag som du har lastet opp samt innslag som andre organisasjoner har publisert.

For å få en brukerkonto i systemet må du som redaktør registrere deg i systemet. I forbindelse med denne registreringen vil det bli sanket inn informasjon om deg og den organisasjonen du arbeider for før du får tilgang til ytterligere funksjonalitet. Når du har registrert deg ferdig vil det dukke opp flere menyvalg og en katalog vil bli laget for din organisasjon på filtjeneren.

#### 18.1.1 Lage ny bruker



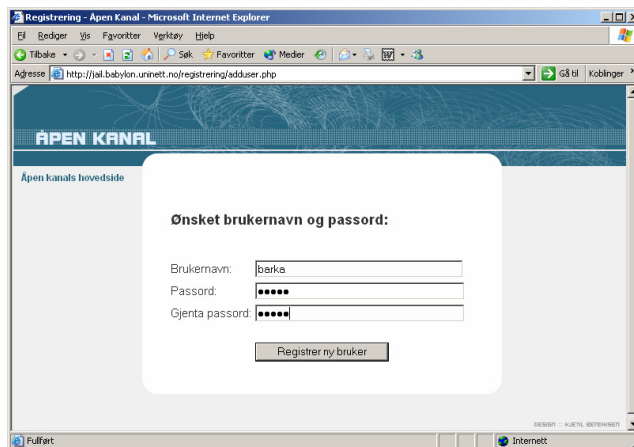
Figur B-1: Grensesnitt for pålogging

Klikk på "Mangler du brukernavn?"

Dersom du ikke er registrert med et brukernavn og passord må du trykke her. Dette for å få opprettet en bruker som umiddelbart vil gi deg tilgang til systemet.

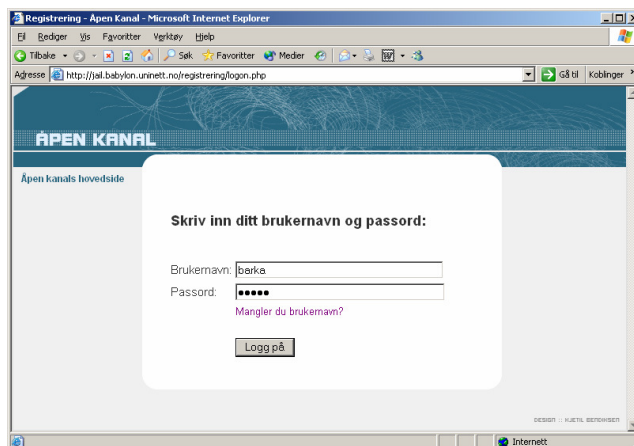
Dersom du allerede har en bruker med et brukernavn og passord kan du logge deg inn direkte.





Figur B-2: Grensesnitt for registrering av ny bruker

Skriv inn ditt brukernavn og ønsket passord. Brukernavnet ditt er bare et kallenavn så velg noe du husker lett og som ikke krever mye inntastning. Du kan kun bruke alfanumeriske tegn samt bindestrek og understrek.



Figur B-3: Grensesnitt for pålogging av bruker etter registrering

Logg deg på!

## 18.1.2 Registrere deg og din organisasjon

The screenshot shows a web browser window titled "Registrering - Åpen Kanal - Microsoft Internet Explorer". The address bar shows a URL from labyfon.uninett.no. The page content includes a sidebar with navigation links like "Åpen kanals hovedside", "Registrering", "Skjema", "Ende personlig info", "Tilbakemeldinger", and "Logg ut". The main content area is titled "Registreringsskjema (side 1):" and contains a form for "Redaktør (deg)".

**Registreringsskjema (side 1):**

**Redaktør (deg)**

Fullt navn:

Gateadr. eller postboks:

Postnummer og sted:

Telefon jobb:

Telefaks jobb:

Telefon hjem:

Telefon mobil:

Elektronisk postadresse:

Føller merket med \* er obligatoriske.

Figur B-4: Grensesnitt for registrering av redaktør

Ved første gangs pålogging vil du bli presentert for et registreringsskjema. Fyll ut og send inn. Som et minimum må du fylle ut ditt fulle navn, ditt personnummer, jobbtelefonnummer samt en av dine elektroniske postadresser.

The screenshot shows the same web browser window, but the main content area is titled "Registreringsskjema (side 2):" and contains a form for "Organisasjon".

**Registreringsskjema (side 2):**

**Organisasjon**

Navn på organisasjon:

Organisasjonsnummer:

Gateadr. eller postboks:

Postnummer og sted:

Webadresse:

Telefon sentralbord:

Har lønnet personell?  Ja  Nei

**Produksjon:**

Har org. fjernsyns- / videoproduksjon i dag?  Ja  Nei

Hvor mye ønsker dere å sende? (timen/uke)

Forventet sendestart:

Innhold ønskes formidlet på?  Nett  Fjernsyn og nett (Fjernsyn medierer høyere medlemsavgift.)

Figur B-5: Grensesnitt for registrering av organisasjon

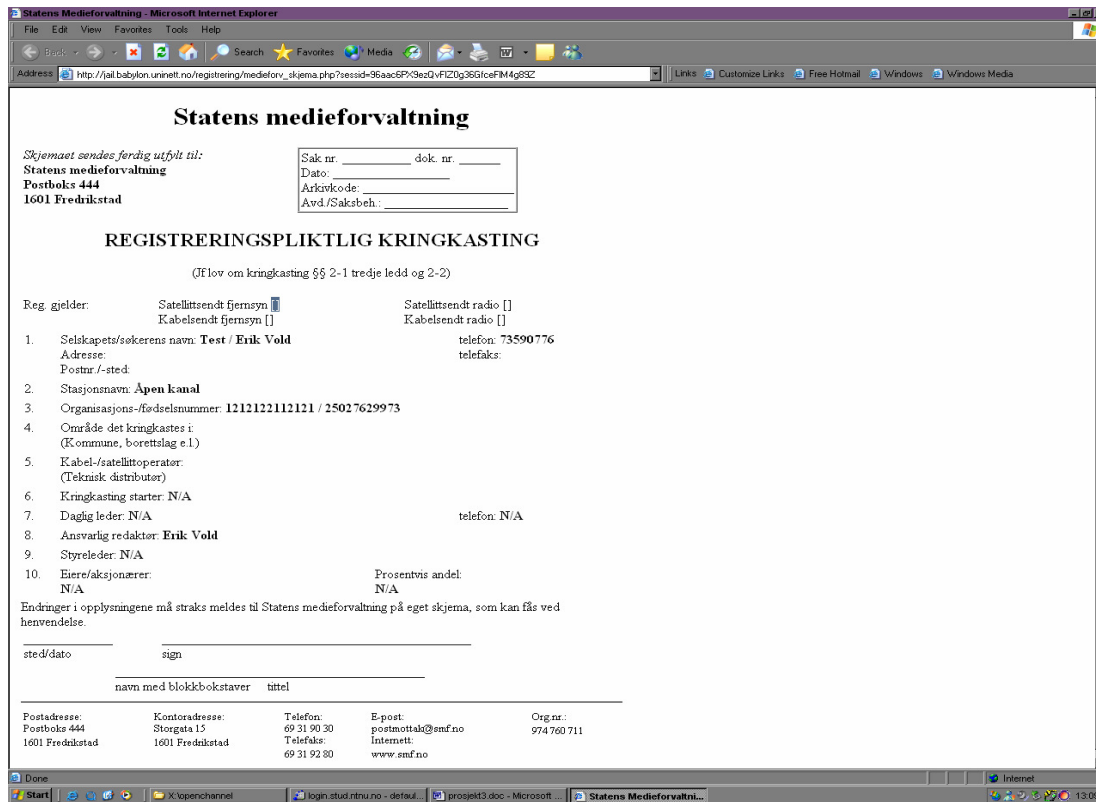
Du vil så bli presentert med et skjema for å registrere informasjon om din organisasjon. Skjemaet må fylles inn i sin helhet.



Figur B-6: Grensesnitt på systemets velkomstsider

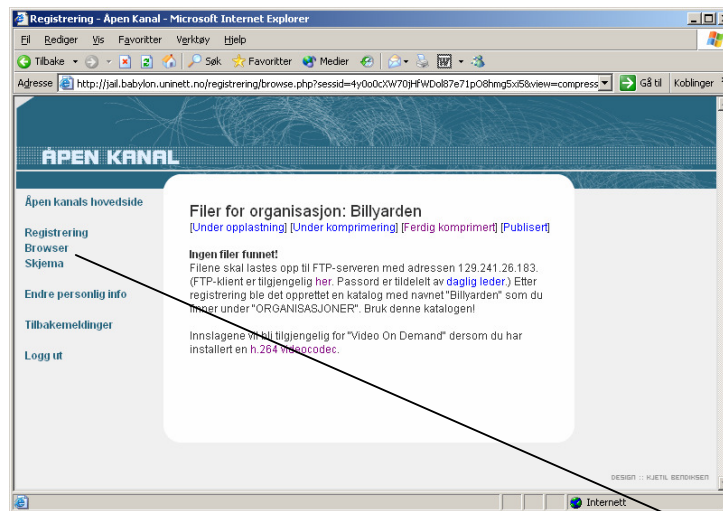
Du er nå pålogget. Dette er din private side der du etter hvert vil finne all funksjonalitet tilknyttet bruken av Åpen kanal.

Når du er ferdig registrert vil du få tilgang på et ferdig utfylt skjema som skal sendes til Statens medieforvaltning for å registrere deg som redaktør ved en evt. satellitt/kabel sending. (Trykk "skjema" for å få opp denne. Det trengs ikke konsesjon for satellitt. Når det blir aktuelt trenger redaktøren kun å sende inn dette skjemaet signert til Statens medieforvaltning)



Figur B-7: Skjema til Statens medieforvaltning

### 18.1.3 Se på videoarkiv



Figur B-8: Menyvalg for tilgang til videoarkiv

For å kikke gjennom innslagene din organisasjon har lastet opp, klikk på "Browse" på menyen på venstre side. Her følges fila gjennom de ulike tilstandene den kan i systemet. Øverst i browse-menyen finner du lenker til filer i hver enkelt tilstand:

- ”Under opplasting”:
- Filer som enda ikke er lastet opp, men der opplastingen har begynt.
- ”Under komprimering”:
- Filer som er ferdig lastet opp, men enten venter på å bli komprimert eller er i ferd med å komprimeres.
- ”Ferdig komprimert”:
- Filer som er ferdig komprimert og klar til å streames.
- ”Publisert”:
- Filer der det er krysset av for ”vederlagsfritt for Åpen kanal”  
(Disse filene blir også tilgjengelig under valgt tema i det offentlige videoarkivet på nettsiden [www.openchannel.no](http://www.openchannel.no))

Dersom filen er merket med et rødt kryss betyr det at den streambare fila ikke er tilgjengelig. Dvs. det finnes kun en DV-fil og ikke en visbar utgave (MPEG-4) av denne selvom den en gang ble laget.

Med opplastede og ferdig komprimerte videoer ser det interne videoarkivet slik ut i web-grensesnittet:



Figur B-9: Internt videoarkiv

### 18.1.4 Laste opp innslag

For å kunne laste opp innslag må du kunne benytte deg av en ftp-klient.

Ftp står for "file transfer protocol" og gjør nettopp det – overfører filer. Det finnes et utall varianter av både klienter og tjenere, alt fra helt enkle uten noe grafikk, bare kommandolinje til superavanserte med funksjoner som opprinnelig ikke var tiltenkt filoverføringsprotokollen i utgangspunktet.

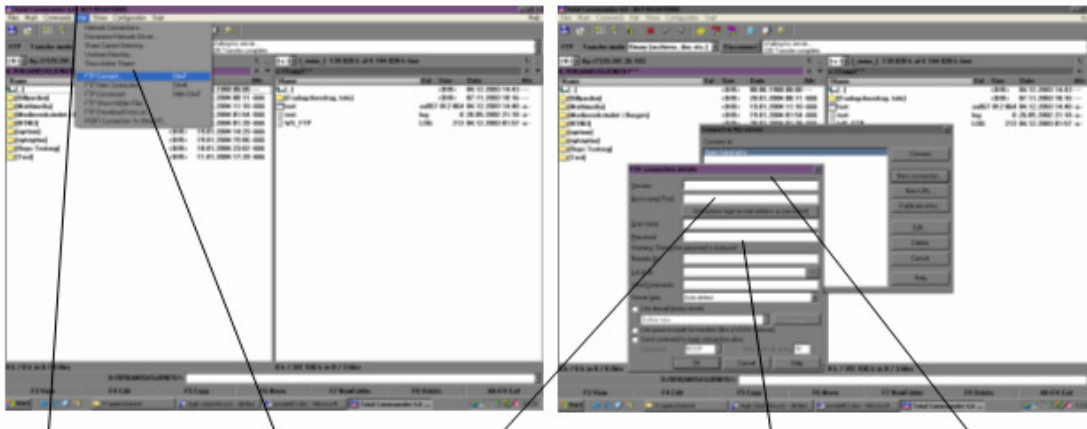
Dersom du ikke vet eller har liten kjennskap til ftp-klient-programmer, har vi en anbefaling til deg, nemlig "Total Commander". Det kan hentes på følgende adresse:

<http://jail.babylon.uninett.no:8080/akanal/Arkiv/FileLibrary5/Files/total%20commander%20FTP%20klient.exe>

Vertsadresse for FTP-serveren er: ftp.openchannel.no

Brukernavn og passord: Fåes ved henvendelse til daglig leder ([adm@openchannel.no](mailto:adm@openchannel.no))

Filområde er: /ORGANISASJONER/<organisasjonsnavn>

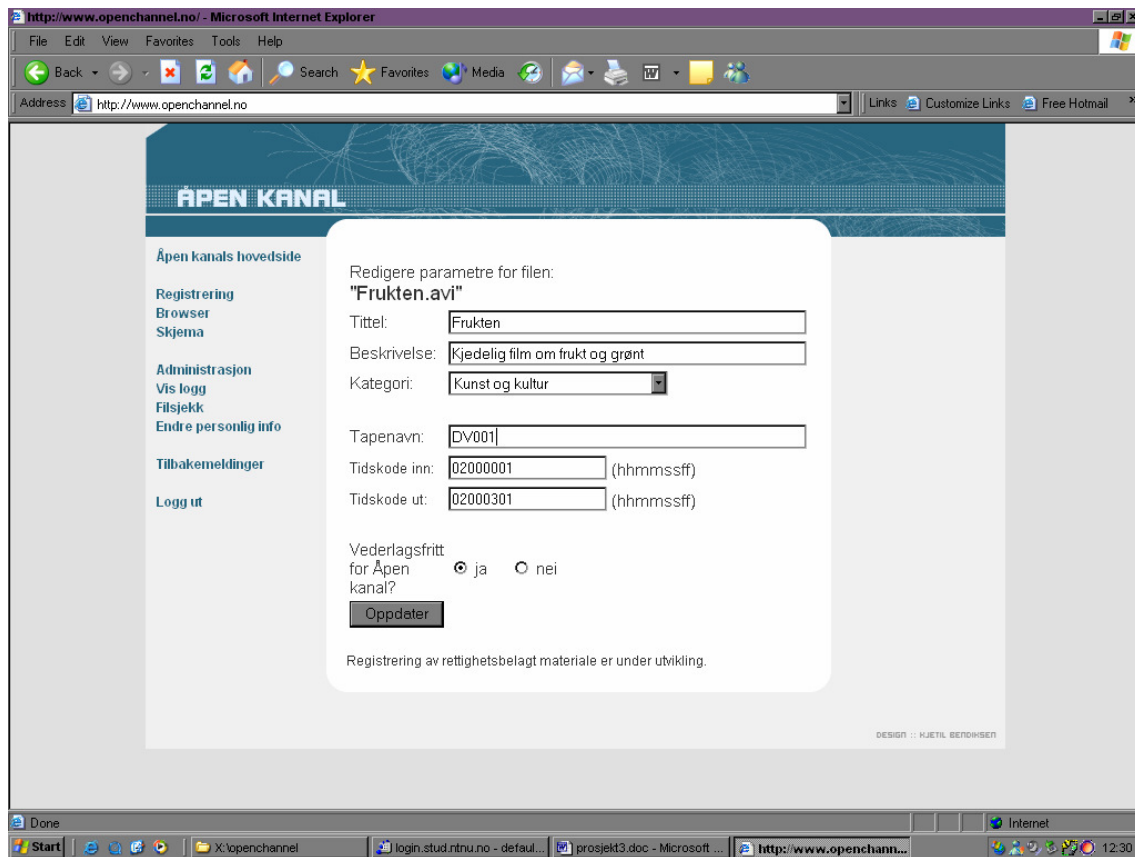


Velg FTP -> FTP Connect ->Fyll inn host/verstnavn ->brukernavn,passord ->Sessionsnavn

**Figur B-10: Grensesnitt for opplasting av filer**

Klienten lagrer alle innstillinger i det midterste vinduet i vinduet til høyre under sesjonsnavnet du fylte inn. Etter dette er det bare "drag and drop" fra ditt lokalområde til din organisasjonsmappe. Her kan det lastes opp filer av typen "avi", "mov" og "mpg". Systemet selv vil innlemme filer under opplasting slik at det umiddelbart kan legges inn informasjon om innslaget. Komprimering (MPEG-4) vil foregå automatisk i bakgrunnen og innslaget vil kunne streames ved å trykke på filnavnet.

## 18.1.5 Legge inn informasjon om innslag



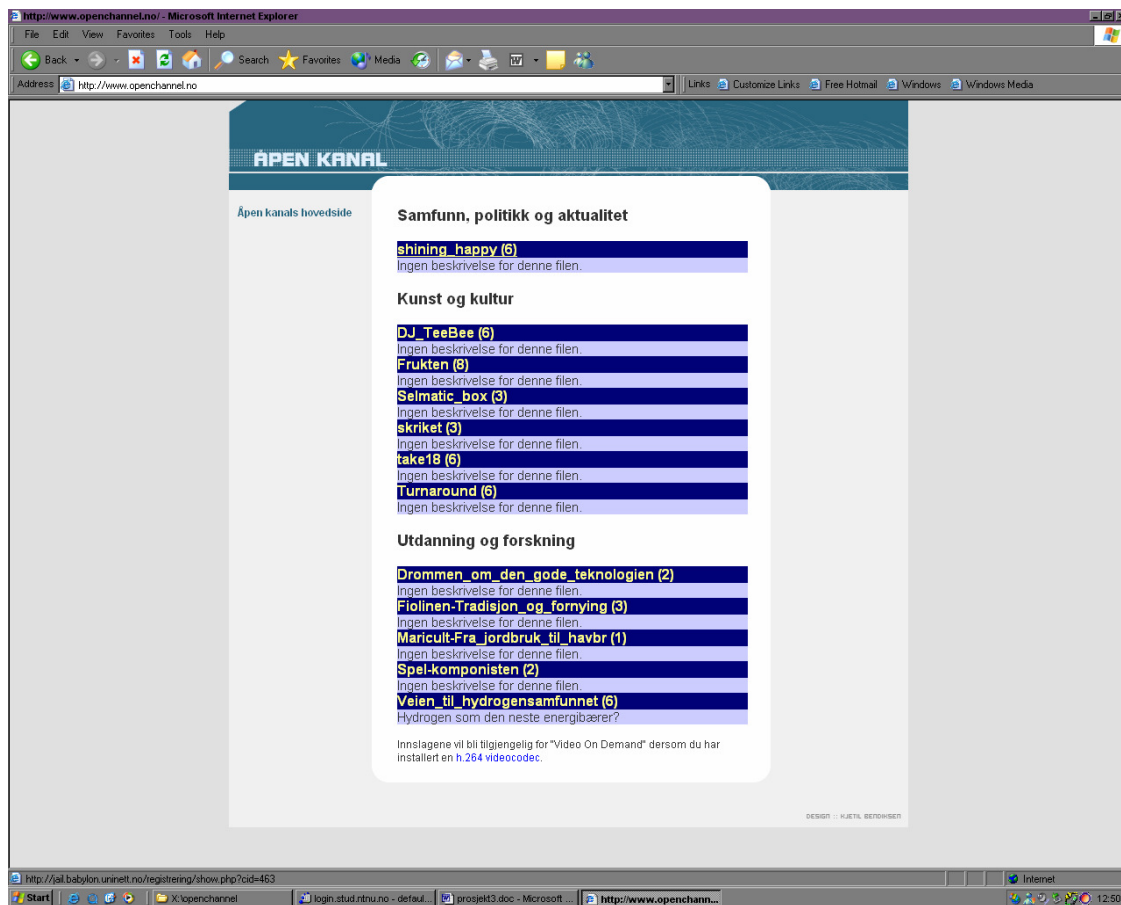
The screenshot shows a Microsoft Internet Explorer browser window displaying the website <http://www.openchannel.no>. The page title is "ÅPEN KANAL". On the left side, there is a navigation menu with the following items: "Åpen kanals hovedside", "Registrering", "Browser Skjema", "Administrasjon", "Vis logg", "Filsjekk", "Endre personlig info", "Tilbakemeldinger", and "Logg ut". The main content area is titled "Redigere parametre for filen: 'Frukten.avi'". It contains several input fields: "Tittel:" with the value "Frukten", "Beskrivelse:" with the value "Kjædelig film om frukt og grønt", "Kategori:" with a dropdown menu showing "Kunst og kultur", "Tapenavn:" with the value "DV001", "Tidskode inn:" with the value "02000001" and "(hhmmssff)", and "Tidskode ut:" with the value "02000301" and "(hhmmssff)". Below these fields, there is a section for "Vederlagsfritt for Åpen kanal?" with radio buttons for "ja" (selected) and "nei". There is an "Oppdater" button. At the bottom of the form, it says "Registrering av rettighetsbelagt materiale er under utvikling." The browser's status bar at the bottom shows "Done" and "Internet". The taskbar at the very bottom shows the Start button and several open applications, including "X:\openchannel", "login.stud.ntnu.no - default...", "prosjekt3.doc - Microsoft ...", and "http://www.openchann...". The system clock shows "12:30".

Figur B-11: Innmating av informasjon om innslag

Ved å trykke "rediger" på en av filene du får opp i browseren kommer du til vinduet i figuren over. Her kan du registrere informasjon om innslaget. Det er kun beskrivelsen og tittelen (isteden for filnavnet) som synes i browseren. Trykker man på mer om i browseren får man opp øvrig informasjon. Det er viktig å registrere hvilken tape og hvor på tapen innslaget kom fra. (hvis den kom fra en tape) Dette fordi at filene kan forsvinne og da blir det lettere å digitalisere og laste opp materiale igjen. På denne siden vil det også komme opp felt for å registrere bruk av rettighetsbelagt musikk (Rapporteres inn til Tono)

## 18.1.6 Publisere innslag

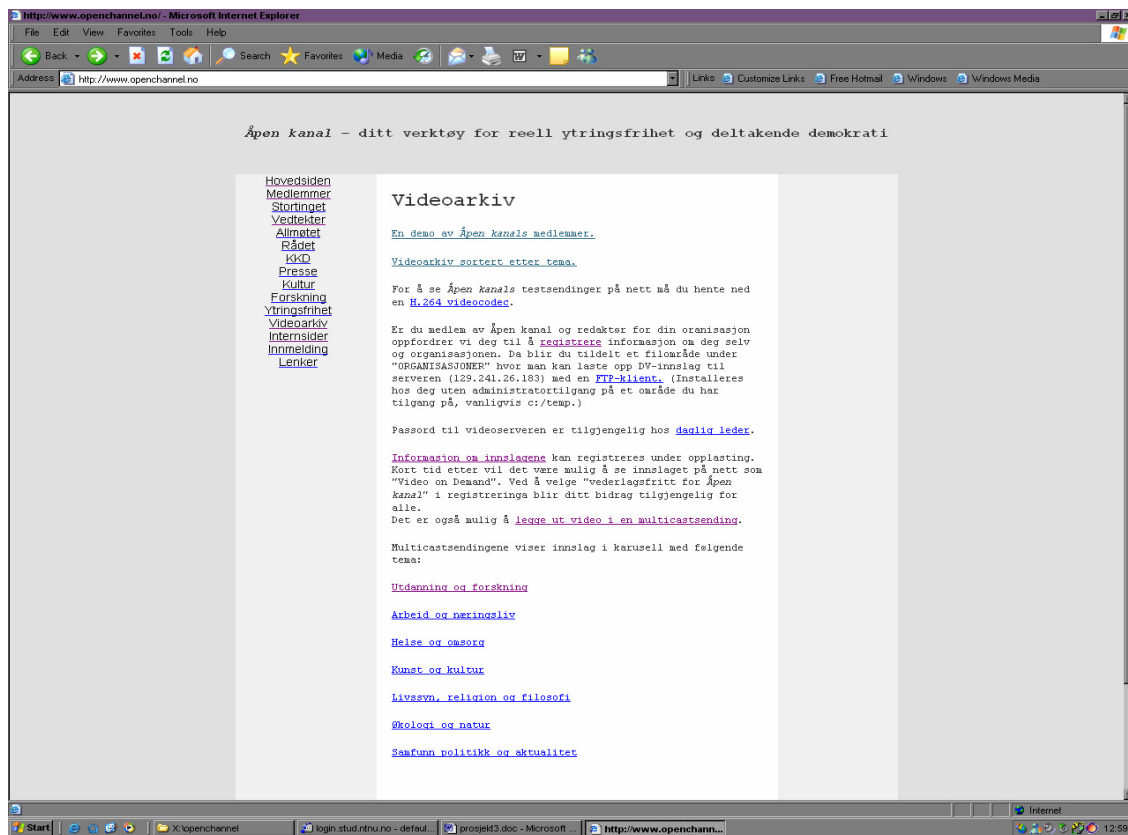
Ved å velge ja på "Vederlagsfritt for åpen kanal" under registreringen blir innslaget tilgjengelig på den offentlige delen av nettsiden [www.openchannel.no](http://www.openchannel.no) både som Video On Demand der du laster ned og ser den når du selv vil, eller som multicast der den går i en karusell sammen med andre innslag under samme tema. (Schedulering av multicastsending gjøres manuelt)



Figur B-12: Offentlig tilgjengelig videoarkiv

De publiserte innslagene i figuren over er offentlig tilgjengelig som Video On Demand (til enhver tid) fra: <http://jail.babylon.uninett.no/registrering/published.php>





Figur B-13: Videolenker for multicast sending

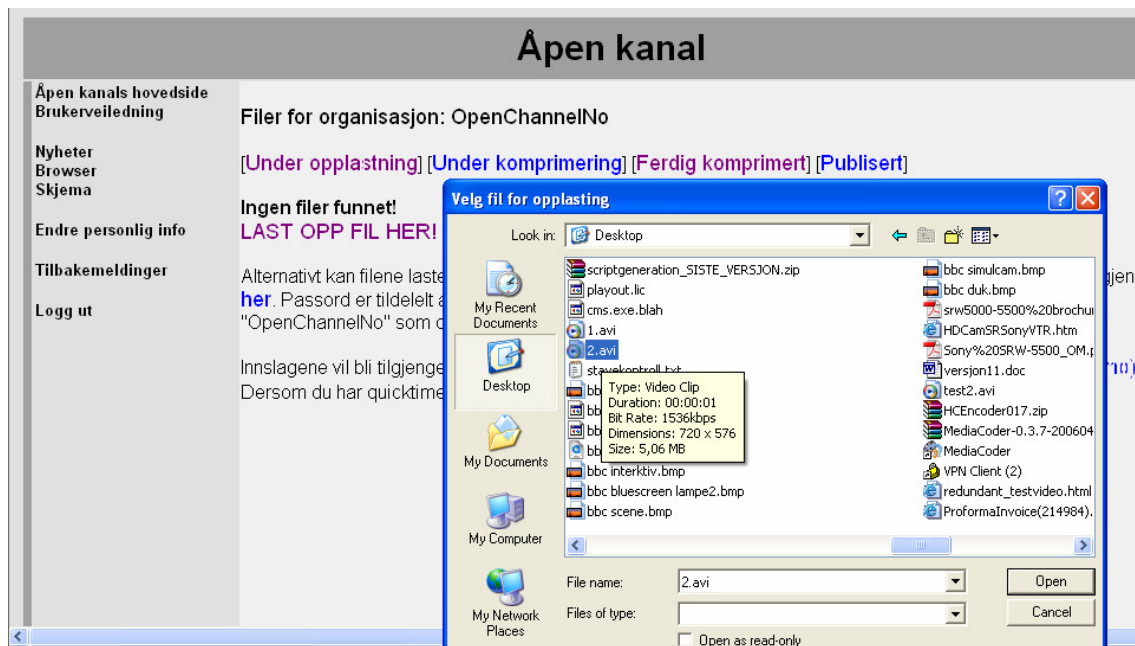
De publiserte innslagene i figuren over er offentlig tilgjengelig på aksessnett med støtte for multicast: (Uninett og NextGenTel) <http://jail.babylon.uninett.no:8080/akanal/videoarkiv.html>

## 19 C-Appendiks: Kildekode med forklaring for opplastingsklient og tjener

### 19.1 Opplastingsklienten

Denne applikasjonen installeres på klient-PC'er tilhørende Åpen kanals brukere. Den skal gjøre opplastingen av innhold til Åpen kanal så enkel som mulig. Første gang .exe-filen eksekveres påbegynnes en installasjon av programmet som fungerer som en tjeneste. Tjenesten starter opp når vedkommende logger seg på automatisk og håndterer bakenforliggende prosesser uten at de er synlige for brukeren. Når man aksesserer filen etter at den er installert vil man kun få opp en browser der man velger filen man vil laste opp. For at det sistnevnte skal fungere på den interne registreringssiden må man utføre visse endringer i operativsystemets register ved første installasjon. Dette forutsetter at vedkommende som utfører installasjonen av den bakenforliggende applikasjonen har administratortilgang.

Lokale egenskaper som IP-adresse samt PC-ens brukernavn og passord vil bli benyttet i installasjonsscriptet.



Figur C-1: Opplastingsklienten aksessert fra Åpen kanal sin nettside

#### 19.1.1 Endringer som utføres i registeret

Denne delen vil utføres av et installasjonsscript, men forutsetter administrator-tilgang.

Klientsystemet trenger en registerendring for å tillate klikk på link i en nettleser:

```
[HKEY_CLASSES_ROOT\OCHAN]
@="URL:Open Channel Uploader"
"URL Protocol"=""
"Animation"="dxmasf.dll,150"
"Source Filter"="{6B6D0800-9ADA-11d0-A520-00A0D10129C0}"

[HKEY_CLASSES_ROOT\OCHAN\DefaultIcon]
@="C:\\Programfiler\\Windows Media Player\\wmplayer.exe,-120"

[HKEY_CLASSES_ROOT\OCHAN\shell]

[HKEY_CLASSES_ROOT\OCHAN\shell\open]

[HKEY_CLASSES_ROOT\OCHAN\shell\open\command]
@="\"C:\\Documents and Settings\\Navn\\Mine dokumenter\\Visual
Studio
2005\\Projects\\contentmanagementserver\\debug\\silenthelper.exe\"
\"%L\""
```

Klientsystemet trenger også et oppstartsvalg i registeret:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
"Ochan Silent Helper App"="c:\\ocserver\\silenthelper.exe <ip> <port>
<user> <password>"
```

hvor følgende må byttes ut med ekte vare:

<ip> må byttes med ip-adressen til tjenerapplikasjonen  
<port> må byttes med portnummeret tjenerapplikasjonen kjører på (2000)  
<user> må byttes med opplastende bruker sitt brukernavn  
<password> og ditto passord

Nå starter systemet opp sammen med systemet og reagerer på klikk fra link på en nettside.

Følgende linje på en nettside vil kalle opp dialogen for valg av fil som skal lastes opp:

```
<a href="OCHAN:selectfile">Velg fil for opplasting</a>
```

Systemet tar seg så av resten.

## 19.1.2 "shnetfile.h"

Denne deklarerer variable som benyttes i netverktilkobling og overføring av filer.

```

/*
 * This class handles networking and file transfer
 */

#include <winsock2.h> //already included in windows.h
#include <iostream>
#include <fstream>

using namespace std;

//Helper functions to make temporar strings permanent
//TODO: clean up afterwards
static char *createstring(char *instring);          //Copy a string to a
permanent location
static int getszlength(char *instring);           //Tell the size of a
string, counts until 0x00 or 0x0a

/*
 * The class itself
 */

class shnetfile{

private:
    bool initialized;          //is networking is initialized?
    bool sendingfile;         //is there a request to send a file?
    bool fileopened;         //is the file to send opened?
    bool loggedon;           //are we logged on to the server?
    bool connected;          //are we connected to the server?
    char *serveraddress;
    char *serverport;
    char *username;
    char *password;
    char *filename;
    char *filepathname;
    long filesize;
    WSADATA w;
    int error;
    SOCKET s;
    sockaddr_in target;
    ifstream *ifs;
    char *buffer;

public:
    shnetfile();
    ~shnetfile();

    bool shdoinitialize();          //initialize network subsystem
    bool shdodeinitialize();
    bool shdoconnect();             //connect to a server
    bool shdodisconnect();
    void shsetfilename(char *fn);   //set the name of the file to
send
    void shsetfilepathname(char *fpn); //set the name of the
file to send
    char *shgetfilename();          //return it
    void shsetusername(char *un);
    void shsetpassword(char *pwd);
    void shsetserveraddress(char *sa);
    void shsetserverport(char *p);
    char *shgetserveraddress();
    char *shgetserverport();

```

```
void shdosendfile();           //request to send a file
bool issendingfile();         //test if a file is being sent
bool shdosenddata();          //sends a portion of the file for each
call
bool shdoopenfile();          //opens a file for reading
bool shdologon();             //logs on to a connected server
};
```

### 19.1.3 "shnetfile.cpp"

Denne håndterer netverktilkobling og overføring av filer.

```
. #include "shnetfile.h"

//Constructor
shnetfile::shnetfile(){
    serveraddress = 0;
    filename = 0;
    username = 0;
    password = 0;
    filesize = 0;
    connected = false;
    initialized = false;
    sendingfile = false;
    fileopened = false;
    loggedon = false;
    error = 0;
    ifs = 0;
    buffer = 0;
}

//Destructor
shnetfile::~shnetfile(){
    shdodeinitialize();
}

//Initialize network subsystem
bool shnetfile::shdoinitialize(){
    if (!initialized){
        error = WSASStartup (0x0202, &w);

        if (error){
            return false;
        }

        initialized = true;

        if (w.wVersion != 0x0202){
            shdodeinitialize();
            return false;
        }
    }

    return true;
}

//Deinitialize the network subsystem
bool shnetfile::shdodeinitialize(){
```

```
        if (connected) shdodisconnect();
        if (initialized){
            WSACleanup ();
            initialized = false;
        }
        return true;          //This will "never" fail (it doesn't matter
anyway)
    }

//Connect to the server
bool shnetfile::shdoconnect(){
    if (!connected){

        if (!initialized){
            if (!shdoinitialize()) return false;
        }

        s = socket (AF_INET, SOCK_STREAM, 0);

        target.sin_family = AF_INET;
        target.sin_port = htons (atoi(shgetserverport())); //Port
number to connect to server
//
        target.sin_addr.s_addr = inet_addr ("129.241.208.139");
        target.sin_addr.s_addr = inet_addr (shgetserveraddress());
        if (connect(s, (SOCKADDR *) &target, sizeof(target)) ==
SOCKET_ERROR){
            shdodeinitialize();
            return false;
        }

        connected = true;
    }

    return true;
}

//Disconnect from the server
bool shnetfile::shdodisconnect(){
    //Check for logon and file open
    if (connected){
        shutdown (s, SD_SEND);
        closesocket (s);
        connected = false;
    }

    return true;
}

void shnetfile::shsetfilename(char *fn){
    filename = createstring(fn);
}

void shnetfile::shsetfilepathname(char *fpn){
    filepathname = createstring(fpn);
}

void shnetfile::shsetserveraddress(char *sa){
    serveraddress = createstring(sa);
}

void shnetfile::shsetserverport(char *p){
    serverport = createstring(p);
}

char *shnetfile::shgetserveraddress(){
    return serveraddress;
}

char *shnetfile::shgetserverport(){
```

```
        return serverport;
    }

    char *shnetfile::shgetfilename(){
        return filename;
    }

    void shnetfile::shsetusername(char *un){
        username = createstring(un);
    }

    void shnetfile::shsetpassword(char *pwd){
        password = createstring(pwd);
    }

    //Signal that a file is selected and is ready for transfer
    void shnetfile::shdosendfile(){
        if ((filename != 0) && (filepathname != 0) && (username != 0) &&
            (password !=0) && (serveraddress != 0) && (serverport != 0)){
            sendingfile = true;
        }
    }

    //Check if a file is flagged as "sendable"
    bool shnetfile::issendingfile(){
        return sendingfile;
    }

    //Log on to the connected server
    bool shnetfile::shdologon(){
        if (!connected){
            if (!shdoconnect()) return false;
        }

        //TODO: proper error checking
        if (!loggedon){
            send (s, username, getszlength(username), 0);
            Sleep(250);
            send (s, password, getszlength(password), 0);
            Sleep(250);
            send (s, filepathname, getszlength(filepathname), 0);
            Sleep(250);
            send (s, "0\n", 2, 0);
            Sleep(250);
            loggedon = true;
        }

        return true;
    }

    //Open the file to send
    bool shnetfile::shdoopenfile(){
        if (!fileopened){
            ifs = new ifstream(filename, ios_base::binary);
            buffer = new char[1024];

            fileopened = true;
        }

        return true;
    }

    //Send a portion of a file
    bool shnetfile::shdosenddata(){
        if (!sendingfile){
            return false;
        }

        if (!loggedon){
```

```

        if (!shdologon()) return false;
    }

    if (!fileopened){
        if (!shdoopenfile()) return false;
    }

    //No else here! This is on purpose! Think about it!

    if (fileopened){
        if (!ifs->good()){
            //TODO: tidy up this section
            loggedon = false; //dologoff()
            sendingfile = false; //dontsendfile()
            fileopened = false;
            delete buffer;
            buffer = 0;
            delete filename;
            filename = 0;
            delete filepathname;
            filepathname = 0;
            shdodisconnect();
            ifs->close();
            ifs = 0;
            return false;
        }

        ifs->read(buffer, 1024);
        if (send (s, buffer, 1024, 0) == SOCKET_ERROR) MessageBox
(NULL, "SOCKET_ERROR", TEXT("Socket error"), MB_OK);
        // Sleep(100);

    }

    return true;
}

/*****
* Method: getszlength
* Returns the length of the given zero terminated string
* including the zero itself, 0x0a is converted to 0x00
*****/

static int getszlength(char *instring){
    //In case of null pointer input
    if (instring == 0) return 0;

    //Search from the beginnig of the string
    int count = 0;

    //Until first occurence of a null character
    char ch = 0x00;
    do{
        ch = *(instring+(count++));
    } while ((ch != '\0') && (ch != 0x0a));

    *((instring+(count))-1) = 0x00;

    //return the found count (plus one)
    return count+1;
}

/*****
* Method: createszstring
* Creates a dynamic string from a constant
* TODO: Replace this with proper string handling
*****/

```



```
static char *createstring(char *instring){
    //Get the input (constant) string's length
    int length = getszlength(instring);

    //Create room for the dynamic string
    char *tempstring = new char[length];

    //Copy the content of the constant string to the dynamic
    for (int i = 0; i<length; i++){
        tempstring[i] = instring[i];
    }

    //Return the pointer to the newly created string
    return tempstring;
}
```

### 19.1.4 "shmain.cpp"

Dette er kjernen i klientapplikasjonen som starter opp først.

```
#include "shnetfile.h"
#include <windows.h>

#define OFN_DONTADDTORECENT 0x02000000 //Missing from the framework,
included here

const char *szWindowClass = "NOCSilentHelperApp";
const char *szWindowName = "NOChan Silent Helper v0.1";
const int RF_SHOW_FILE_DIALOG = 0xA123;

static LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam,
LPARAM lParam);

shnetfile *netfile; //Class that manages network connections
and file transfers

/*
 * getword() picks one and one word from a space-separated list of words
 */

int getword(char *cmdline, int clcnt, char *clbuf, int clbufsize){

    if (cmdline == 0) return -1;

    if (*(cmdline+clcnt) == 0x00) return -1;

    int cnt = 0;
    int ocnt = 0;
    char ch = 0x00;

    while (((cnt-clcnt)<(clbufsize-1)) && ((ch =
*(cmdline+clcnt+cnt))!=0x00)){
        if (ch != 0x20){
            *(clbuf+ocnt++) = *(cmdline+clcnt+cnt);
        }
        else{
            *(clbuf+ocnt) = 0x00;
            return clcnt+cnt+1;
        }
        cnt++;
    }
}
```

```
        *(clbuf+ocnt) = 0x00;
        return clcnt+cnt;
    }

    /*
    * Entry point: WinMain()
    */

int WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine,
int nCmdShow){
    /*
    * Make sure the application has only one running background
instance
    * - further instances of this app is communications to the first
one
    */

    //Create a mutex
HANDLE mutex=CreateMutex(0, true, "mutex_nocshv01_check2xstart");

    if (mutex != 0){

        //Does the mutex already exist? Then there is already an
instance running
        if (GetLastError() == ERROR_ALREADY_EXISTS){

            //Find the window handle to the existant instance
HWND hwnd = FindWindow(szWindowClass, 0);
            if (hwnd != 0){
                //Send it a message to display a file
dialog
                SendMessage(hwnd, RF_SHOW_FILE_DIALOG, 0,
0);
            }
        }
        else { //This is the first instance of the
application

            //Create a class instance to communicate over net
and send a file
            netfile = new shnetfile();

            //Fail if the network subsystem will not initialize
            if (!netfile->shdoinitialize()){
                MessageBox (NULL, TEXT("Could not
initialize network subsystem\nfor this application."), TEXT("ERROR!"),
MB_OK);
                return 0;
            }

            //Parse the command line
            //TODO: make it more robust
            int clcnt = 0;
            int clwcnt = 0;
            char *clbuf = new char[1024];

            //Get parameter 1
            clcnt = getword(lpCmdLine, clcnt, clbuf, 1024);
            if (clcnt == -1) return 0; //TODO: display error
message
            netfile->shsetserveraddress(clbuf);

            //Get parameter 2
            clcnt = getword(lpCmdLine, clcnt, clbuf, 1024);
            if (clcnt == -1) return 0; //TODO: display error
message
            netfile->shsetserverport(clbuf);
        }
    }
}
```

```

//Get parameter 3
clcnt = getword(lpCmdLine, clcnt, clbuf, 1024);
if (clcnt == -1) return 0; //TODO: display error
message
netfile->shsetusername(clbuf);

//Get parameter 4
clcnt = getword(lpCmdLine, clcnt, clbuf, 1024);
if (clcnt == -1) return 0; //TODO: display error
message
netfile->shsetpassword(clbuf);

//Confirm that there are no fifth parameter
clcnt = getword(lpCmdLine, clcnt, clbuf, 1024);
if (clcnt != -1) return 0; //TODO: display error
message

delete clbuf;

//Register an instance of the window class
WNDCLASSEX wndClassex;

if (!hPrevInstance){
    wndClassex.cbSize = sizeof(WNDCLASSEX);
    wndClassex.style = CS_DBLCLKS | CS_OWNDC;
    wndClassex.lpfnWndProc = (WNDPROC)
WindowProc;

    wndClassex.cbClsExtra = 0;
    wndClassex.cbWndExtra = 0;
    wndClassex.hInstance = hInstance;
    wndClassex.hIcon = 0;
    wndClassex.hCursor = 0;
    wndClassex.hbrBackground = (HBRUSH)
COLOR_BACKGROUND;

    wndClassex.lpszMenuName = 0;
    wndClassex.lpszClassName = szWindowClass;
    wndClassex.hIconSm = 0;

    RegisterClassEx(&wndClassex);
    //TODO: test if this fails and end the program (should rarely happen)
}

//Create a window (although invisible) for this
application)
HWND hWnd = CreateWindowEx(WS_EX_OVERLAPPEDWINDOW |
WS_EX_CLIENTEDGE, szWindowClass, szWindowName,
WS_OVERLAPPEDWINDOW | WS_BORDER |
WS_CAPTION, 100, 100, 500, 400, 0, 0, hInstance,0);

//Enter the message loop
MSG msg; //to get messages
bool action = false; //To track activity (if
no activity, Sleep() to relieve system)
bool quit = false; //loop until quit
while(!quit){
    //Check if there are messages in the queue
and get it if so
    if (PeekMessage(&msg, hWnd, 0, 0,
PM_REMOVE) != 0){
        something has happened this run
        action = true; //Set that

        //Check for important messages
        (that is WM_QUIT)
        switch (msg.message){
            case WM_QUIT:
                quit = true;
                break;

```

```

                                default:
                                    break;
                                }
                                //Send the message further
                                TranslateMessage(&msg);
                                DispatchMessage(&msg);
                            }

                                //Send data from file if there are any
                                if (netfile->shdosenddata()){
                                    action = true; //Set that
something has happened this run
                                }

                                //Did something happen?
                                if (!action) {
Sleep()
                                    Sleep(100); //no, let's
run
                                }
                                action = false; //Prepare for next
                                }

                                //Release mutex to allow for a new main instance
after this one closes
                                CloseHandle(mutex);

                                //Exit program
                                return msg.wParam;
                            }
                        }

                                //Exit program
                                return 0;
                    }

static LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam,
LPARAM lParam){
    char *fname; //make global?
    OPENFILENAME ofn; //make global?
    PAINTSTRUCT ps;
    HDC hdc;

    switch (uMsg){
        case WM_CREATE:
empty)
                                //Initialize the form (we don't need any, therefore
                                return 0;

        case RF_SHOW_FILE_DIALOG: //Select and upload file
already sending a file
                                if (!netfile->issendingfile()){ //If not

                                    //Prepare the file dialog
                                    ZeroMemory(&ofn, sizeof(OPENFILENAME));
                                    ofn.lStructSize = sizeof(OPENFILENAME);
                                    fname = new char[1024];
                                    fname[0] = 0x00;
                                    ofn.lpstrFile = fname;
                                    ofn.nMaxFile = 1024;
                                    ofn.lpstrInitialDir = "c:\\";
                                    ofn.lpstrTitle = "Velg fil for opplasting";
                                    ofn.Flags = OFN_DONTADDTORECENT |
OFN_FILEMUSTEXIST | OFN_LONGNAMES | OFN_NONETWORKBUTTON |
OFN_PATHMUSTEXIST;

                                    //Get a file name from the file dialog

```

```
        if (GetOpenFileName(&ofn)){
            netfile->shsetfilename(fname);
//Tell the net and file system the file name
            netfile-
>shsetfilepathname((fname+ofn.nFileOffset)); //Tell the net and file
system the file name
            netfile->shdosendfile();
//Tell the net and file system to send it
        }

        delete fname;
    }

    return 0;

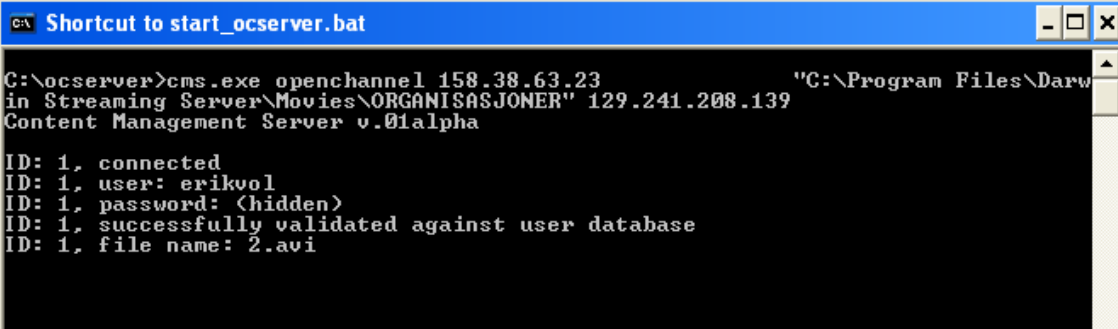
case WM_PAINT: // validate the window
    hdc = BeginPaint(hwnd, &ps); //Dummy
    EndPaint(hwnd, &ps); //Dummy
    return 0;

case WM_CLOSE:
    //Fallthrough
case WM_DESTROY:
    PostQuitMessage(0); //Tell the application
to quit
    return 0;

    default;;
}
return DefWindowProc(hwnd, uMsg, wParam, lParam);
}
```

## 19.2 Opplastingstjeneren

Denne applikasjonen settes opp på en sentralt plassert server. Den lytter til en port og mottar pakker som lagres i en lokal fil. Når filen er lagret sender den en melding til databasen om at filen befinner seg på FTP-serveren. Da kan komprimeringssystemet hente den og til slutt legge ut en streaming-link til fila på Åpen kanal sin offentlige nettside (publisering).



```

C:\> Shortcut to start_ocserver.bat
C:\ocserver>cms.exe openchannel 158.38.63.23          "C:\Program Files\Darw
in Streaming Server\Movies\ORGANISASJONER" 129.241.208.139
Content Management Server v.01alpha

ID: 1, connected
ID: 1, user: erikvol
ID: 1, password: (hidden)
ID: 1, successfully validated against user database
ID: 1, file name: 2.avi
  
```

Figur C-2: Kvittering fra opplastingstjener

### 19.2.1 Parametere som benyttes i installasjon av opplastingstjeneren

Server-installasjonen skal ideelt sett kun foretas en gang. Det kan likevel hende at serveren går ned og da fungerer en reinstallasjon som en ny oppstart av systemet. Det er laget et enkelt bat-script for å utføre installasjonen med nødvendige parametere. Av sikkerhetsmessige årsaker er denne kun tilgjengelig på skrivebordet til serveren. Strukturen av innholdet er beskrevet her.

Tjenersystemet startes opp med noen kommandolinjeparapetere:  
 cms.exe <db\_name> <db\_ip\_address> <db\_user\_name> <db\_password>  
 <basepath> <bind\_ip>

db\_name: navnet på databasen som skal brukes  
 db\_ip\_adress: ip-adressen til databaseserveren (ingen betbiosnavn eller uris)  
 db\_user\_name: brukernavn for tilkobling til databaseserver  
 db\_password: passord for bruker som skal koble seg til database  
 basepath: basis-sti for hvor filer skal lagres på server ved opplasting (uten "trailing backslash")  
 bind\_ip: ip-adresse serveren skal lytte på. Denne kan IKKE være 127.0.0.1 (localhost) og må oppgis for å spesifiserer hvilket nic som serveren skal lytte etter brukere på

Tjenersystemet vil så være selvgående.

## 19.2.2 "cmsusers.h"

Denne håndterer brukerne i systemet

```
#include <fstream>

using namespace std;

typedef struct user{
    int id;
    char *name;
    char *password;
    char *savefilename;
    char *filename;
    char *orgname;
    char *orgid;
    ofstream *ofs;
    int fpos;
    long filesize;
    int logonstep;
    bool connected;
    bool validated;
    user *nextuser;
    char *fileid;
    bool closed;
} USER, *LPUSER;

class cmsusers{

private:
    LPUSER firstuser;
    LPUSER lastuser;
    int nextuserid;
    LPUSER userlist;
    char *basepath;

    LPUSER finduser(int id);
    int getszlength(char *instring);

public:

    cmsusers();
    ~cmsusers();
    int adduser();
    int adduser(int id);
    int setname(int id, char *name);
    char *getname(int id);
    int setpassword(int id, char *password);
    char *getpassword(int id);
    char *createstring(char *instring);
    bool writefile(int id, char *buffer, int *size);
    bool userexists(int id);
    int setfilename(int id, char *name);
    char *getfilename(int id);
    char *getdbfilename(int id);
    void setvalidated(int id, bool validated);
    int getlogonstep(int id);
    void inclogonstep(int id);
    int setorgname(int id, char *name);
    int addtext(char *buffer, char *text2add, int pos);
    void endtext(char *buffer, int pos);
    char *getorgname(int id);
    void setbasepath(char *name);
    int setorgid(int id, char *name);
    char *getorgid(int id);
    void removeuser(int id);
```

```
int setfileid(int id, char *name);
char *getfileid(int id);
void setclosed(int id);
bool getclosed(int id);
};
```

### 19.2.3 "cmsdbfunc.ccp"

Funksjoner for å kommunisere mot databasen.

```
#include "cmsdbfunc.h"

cmsdbfunc::cmsdbfunc(){
    orgid = 0;
    orgname = 0;
}

cmsdbfunc::~cmsdbfunc(){
    if (orgid != 0){
        delete orgid;
        orgid = 0;
    }

    if (orgname != 0){
        delete orgname;
        orgname = 0;
    }
}

//Check if user with given password does exist
bool cmsdbfunc::userlogin(char *username, char *passwd){
    if (orgid != 0){
        delete orgid;
        orgid = 0;
    }

    if (orgname != 0){
        delete orgname;
        orgname = 0;
    }

    //TODO: Error checking

    /*
    db = "openchannel";
    dbhost = "localhost";
    dbuser = "root";
    dbpasswd = "quadrant123";
    */

    mysqlpp::Connection con(db, dbhost, dbuser, dbpasswd);

    mysqlpp::Query query = con.query();

    query << "select * from users where username='" << username << "'
and password=password('" << passwd << "')";

    mysqlpp::Result res = query.store();

    if (res){
        mysqlpp::Row row;
//        mysqlpp::Row::size_type i;

        if (res.size() == 1){
```



```
        //Organisation name must also be fetched
        row = res.at(0);

        orgid = createstring((char *)row["orgid"].c_str());
        mysqlpp::Query query2 = con.query();

        query2 << "select orgnavn from organisasjoner where
orgid='" << orgid << "'";
        mysqlpp::Result res2 = query2.store();

        if (res2){
            mysqlpp::Row row2;

            if (res2.size() == 1){
                row2 = res2.at(0);
                orgname = createstring((char
*)row2["orgnavn"].c_str());
            }
            else return false;
        }
        else return false;

        return true;
    }
    /*
        i = 0;
        while (i<res.size()){
            row = res.at(i);
            cout << row["name"] << "\t" << row["password"] <<
"\t" << "\n";
            i++;
        }
    */
    }

    return false;
}

//Add an incoming file to the database
char *cmsdbfunc::addfile(char *orgid, char *filename){
    mysqlpp::Connection con(db, dbhost, dbuser, dbpasswd);

    mysqlpp::Query query = con.query();

    query << "insert into content (filename, orgid) values ('" <<
filename << "', '" << orgid << "')";

    mysqlpp::Result res = query.store();

    res.purge();

    query.reset();

    query << "select LAST_INSERT_ID() from content";

    res = query.store();

    if (res.columns() == 1){
        mysqlpp::Row row = res.fetch_row();
        return createstring((char
*)row["LAST_INSERT_ID()"].c_str());
    }

    return 0;
}

//Tell the database the file is uploaded
bool cmsdbfunc::setuploaded(char *media_id){
```

```
        mysqlpp::Connection con(db, dbhost, dbuser, dbpasswd);
        mysqlpp::Query query = con.query();
        query << "update content set fin_upload='yes' where contentid='" <<
media_id << "'";
        mysqlpp::Result res = query.store();
        return true;
}

void cmsdbfunc::setdb(char *sdb){
    db=sdb;
}

void cmsdbfunc::setdbhost(char *sdbhost){
    dbhost=sdbhost;
}

void cmsdbfunc::setdbuser(char *sdbuser){
    dbuser=sdbuser;
}

void cmsdbfunc::setdbpasswd(char *sdbpasswd){
    dbpasswd=sdbpasswd;
}

void cmsdbfunc::setorgname(char *sorgname){
    orgname=sorgname;
}

char *cmsdbfunc::getorgname(){
    return orgname;
}

char *cmsdbfunc::getorgid(){
    return orgid;
}

/*****
 * Method: getszlength
 * Returns the length of the given zero terminated string
 * including the zero itself, 0x0a is converted to 0x00
 *****/

static int getszlength(char *instring){
    //In case of null pointer input
    if (instring == 0) return 0;

    //Search from the beginnig of the string
    int count = 0;

    //Until first occurence of a null character
    char ch = 0x00;
    do{
        ch = *(instring+(count++));
    } while ((ch != '\0') && (ch != 0x0a));

    *((instring+(count))-1) = 0x00;

    //return the found count (plus one)
    return count+1;
}

/*****
 * Method: createszstring
 * Creates a dynamic string from a constant
 * TODO: Replace this with proper string handling
 *****/
```

```
*****/
static char *createstring(char *instring){
    //Get the input (constant) string's length
    int length = getszlength(instring);

    //Create room for the dynamic string
    char *tempstring = new char[length];

    //Copy the content of the constant string to the dynamic
    for (int i = 0; i<length; i++){
        tempstring[i] = instring[i];
    }

    //Return the pointer to the newly created string
    return tempstring;
}
```

## 19.2.4 "cmsdbfunc.h"

Deklarering av variable for komminisering mot databasen.

```
#include <iostream>
#include <mysql++.h>

//Helper functions to make temporar strings permanent
//TODO: clean up afterwards
static char *createstring(char *instring);          //Copy a string to a
permanent location
static int getszlength(char *instring);            //Tell the size of a
string, counts until 0x00 or 0x0a

class cmsdbfunc{
private:
    char *db;                //Do not delete the db... members
    char *dbhost;           // they are the exact versions as from main
    char *dbuser;           // ...
    char *dbpasswd;         // ...
    char *orgname;
    char *orgid;

public:
    cmsdbfunc();
    ~cmsdbfunc();
    bool userlogin(char *username, char *passwd);
    void setdb(char *db);
    void setdbhost(char *dbhost);
    void setdbuser(char *dbuser);
    void setdbpasswd(char *dbpasswd);
    void setorgname(char *sorgname);
    char *getorgname();
    char *getorgid();
    char *addfile(char *orgid, char *filename);
    bool setuploaded(char *media_id);
};
```

## 19.2.5 "cmsmain.ccp"

Dette er serverapplikasjonens kjerne som starter opp først.

```

/*
 *
 * Description:                Main part of the server for the file upload
system
 *
 */

/*
 * To compile:
 *
 * Set up the following
 *
 * - Platform SDK
 * - MySQL development files
 * - MySQL++
 * - Winsock2
 *
 * Set directory paths to point to the respective include and library
files
 *
 */

#include <iostream>
#include "cmsusers.h"
#include "cmsdbfunc.h"
#include "cmsserverclass.h"
#include "cmscommandqueue.h"

using namespace std;

int main (int argc, char *argv[]){
    /*****
    * Present a welcome screen when starting server
    *****/

    cout << "Content Management Server v.01alpha\n\n";

    /*****
    * Test the number of arguments used to call this application
    *****/

    if (argc != 7){
        cout << "cms.exe <dbname> <dbipaddress> <dbuser> <passwd>
<basefilepath> <iplisten>\n";
        return 0;
    }

    /*****
    * Make server listen for connections
    *****/
    //TODO: Check for errors, put inside while loop, try to restore
    // listening state in error conditions, make a "state-machine"
    int received = 0;
    char *recvbuf = new char[1026];
    char *tmpstr = 0;
    cmsserverclass *cmssc = new cmsserverclass(2000);

    if (cmssc->lerror() != 0){
        cout << "Error starting server subsystem\n";
        return 0;
    }

    if (cmssc->connect(argv[6]) != 0){

```

```
        cout << "Error starting server subsystem\n";
        return 0;
    }

    //Create a list of connected users
    cmsusers *cmsu = new cmsusers();
    cmsu->setbasepath(argv[5]);

    //The server loop
    int socket_id = 0;
    bool action = false;
    while (1){ //Make so that an admin can shutdown remotely

        /*****
        *    Check for new connections, remove finished
connections,
        *    update the table of connected users
        *****/

        socket_id = cmssc->checkforconnection();
        if (socket_id != 0){
            if (!cmsu->userexists(socket_id)){
                action = true;
                cout << "ID: " << socket_id << ",
connected\n";
                cmsu->adduser(socket_id);
            }
        }

        /*****
        *    Fetch data from users if there are data waiting in
buffers,
        *    respond correspondingly
        *****/

        received = 1025;
        socket_id = cmssc->readConnection(recvbuf, &received);

        if (received == -1){
            //Tell system that file is successfully uploaded
            //Bug: Might actually not be the case (some other
error)
            //This will cause the file to be wrongly tagged as
a finished upload
            if (!cmsu->getclosed(socket_id)){
                cmsu->setclosed(socket_id);

                //Connect to the database
                cmsdbfunc *dbf = new cmsdbfunc();

                dbf->setdb(argv[1]);
                dbf->setdbhost(argv[2]);
                dbf->setdbuser(argv[3]);
                dbf->setdbpasswd(argv[4]);

                dbf->setuploaded(cmsu-
>getfileid(socket_id));
            }

            //Prevent the system to handle this node further
            socket_id = 0;
            received = 0;
        }

        //trick to prevent buffer overflow and to detect correct
```

```
sizes of buffers
    if (received<1025){
        *(recvbuf+received) = 0x00;
    } else {
        *(recvbuf+1026) = 0x00;
    }

    //There is communication pending from the current socket
    if (socket_id != 0){
        action = true;

        //Is entered username pending?
        int logonstep = cmsu->getlogonstep(socket_id);
        if (logonstep == 0){ //Logon name
            tmpstr = cmsu->createstring(recvbuf);
            cout << "ID: " << socket_id << ", user: "
<< tmpstr << "\n";

            cmsu->setname(socket_id, tmpstr);
            cmsu->inclogonstep(socket_id);
        }

        //What about password?
        else if (logonstep == 1){ //Logon password
            tmpstr = cmsu->createstring(recvbuf);
            cout << "ID: " << socket_id << ", password:
(hidden)\n"; //<< tmpstr << "\n";
            cmsu->setpassword(socket_id, tmpstr);
            cmsu->inclogonstep(socket_id);

            //compare this user to the mysql user
database
            cmsdbfunc *dbf = new cmsdbfunc();

            dbf->setdb(argv[1]);
            dbf->setdbhost(argv[2]);
            dbf->setdbuser(argv[3]);
            dbf->setdbpasswd(argv[4]);

            if (dbf->userlogin(cmsu-
>getname(socket_id), cmsu->getpassword(socket_id)){
                cout << "ID: " << socket_id << ",
successfully validated against user database" << "\n";
                cmsu->setvalidated(socket_id,
true);
                cmsu->setorgname(socket_id, dbf-
>getorgname());
                cmsu->setorgid(socket_id, dbf-
>getorgid());
            } else {
                cout << "ID: " << socket_id << ",
could not validate against user database" << "\n";
                //TODO: Should force socket close
here
            }
            delete dbf;
        }

        //The file name?
        else if (logonstep == 2){ //logon transfer
filename
            tmpstr = cmsu->createstring(recvbuf);
            cout << "ID: " << socket_id << ", file
name: " << tmpstr << "\n";

            cmsu->setfilename(socket_id, tmpstr);
            cmsu->inclogonstep(socket_id);
        }

        //File size?
        else if (logonstep == 3){ //logon transfer
```

```

filename
                                tmpstr = cmsu->createstring(recvbuf);
                                cout << "ID: " << socket_id << ", file
size: " << tmpstr << "\n";
                                cmsu->inclogonstep(socket_id);

                                //Connect to the database
                                cmsdbfunc *dbf = new cmsdbfunc();

                                dbf->setdb(argv[1]);
                                dbf->setdbhost(argv[2]);
                                dbf->setdbuser(argv[3]);
                                dbf->setdbpasswd(argv[4]);

                                //TODO: optimize all calls to same
userobject
                                cmsu->setfileid(socket_id, dbf-
>addfile(cmsu->getorgid(socket_id), cmsu->getdbfilename(socket_id));

                                delete dbf;
                                }

                                //Receive file from the current connection
                                else if (logonstep != -1){ //logged on: do
receive
                                if (received > 0){
                                action = true;
                                //TODO: remove commented code? (For
debug)
                                //cout << "ID: " << socket_id << ",
read count: " << received;
                                //if (!cmsu->writefile(socket_id,
recvbuf, &received)) cout << " (refused)";
                                //cout << "\n";

                                cmsu->writefile(socket_id, recvbuf,
&received);
                                }
                                }

                                }

/*****
*      If no communication this turn, wait to relieve the
system
*****/
                                if (!action) {Sleep(100);/* cout << ".";*/};
                                action = false;
                                }

                                //Cleanup
                                delete cmsu;

                                //Exit program
                                return 0;
                                }

```

## 19.2.6 "cmsserverclass.cpp"

### Socket-håndtering på serversiden

```
#include <winsock2.h>
#include "cmsserverclass.h"

//#include <iostream>

//using namespace std;

//Adds a receiving socket to the list
int cmsserverclass::addsocket(SOCKET socket){
    //Create new socket-element
    CLIENT_SOCKETS *cs = new CLIENT_SOCKETS;
    cs->socket_id = next_socket_id++;
    cs->socket = socket;
    cs->dummy = 1;          //Dummy variable for the ioctlsocket()-
command
    ioctlsocket(cs->socket, FIONBIO, (u_long *) &cs->dummy);
    //Make socket non-blockable
    cs->next_client_socket = 0;
    cs->active = true;

    //Insert it into the list
    if ( first_client_socket == 0 ){
        first_client_socket = cs;
        client_sockets = cs;
        last_client_socket = cs;
    } else {
        last_client_socket->next_client_socket = cs;
        last_client_socket = cs;
    }

    return cs->socket_id;
}

//Removes a receiving socket from the list
void cmsserverclass::remsocket(int socket_id){
    //Remove a connection from the list
    /*
    CLIENT_SOCKETS *ctmpsock = first_client_socket;
    CLIENT_SOCKETS *clastsock = 0;

    while (ctmpsock != 0){
        if (ctmpsock->socket_id == socket_id){
            if (clastsock == 0){
                first_client_socket = first_client_socket-
>next_client_socket;
                if (first_client_socket == 0){
                    client_sockets = 0;
                    last_client_socket = 0;
                }
                delete ctmpsock;
                return;
            }
            else{
                clastsock->next_client_socket = ctmpsock-
>next_client_socket;
                if (client_sockets == ctmpsock)
client_sockets = client_sockets->next_client_socket;
                delete ctmpsock;
                if (client_sockets == 0) last_client_socket
= clastsock;
                return;
            }
        }

        clastsock = ctmpsock;
        ctmpsock = ctmpsock->next_client_socket;
    }
    */
}
```



```
//hack to fix a problem temporarily: MEMLEAK!!
first_client_socket = 0;          //"forget" existing sockets
last_client_socket = 0;
client_sockets = 0;

}

cmsserverclass::cmsserverclass(){
    cmsserverclass::listen_port = 2000;
    cmsserverclass::last_error = 0;
    cmsserverclass::last_error = WSASStartup( MAKEWORD(2,2),
&(cmsserverclass::wsadata));
    cmsserverclass::client_sockets = NULL;
    cmsserverclass::last_client_socket = NULL;
    cmsserverclass::first_client_socket = 0;
    cmsserverclass::next_socket_id = 1;
}

cmsserverclass::cmsserverclass(int listen_port){
    cmsserverclass::listen_port = listen_port;
    cmsserverclass::last_error = 0;
    cmsserverclass::last_error = WSASStartup( MAKEWORD(2,2),
&(cmsserverclass::wsadata));
    cmsserverclass::client_sockets = NULL;
    cmsserverclass::last_client_socket = NULL;
    cmsserverclass::first_client_socket = 0;
    cmsserverclass::next_socket_id = 1;
}

//Create a listening socket for the server
int cmsserverclass::connect(char *listenip){
    server_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);

    dummy = 1;
    ioctlsocket(server_socket, FIONBIO, (u_long *) &dummy);

    server_service.sin_family = AF_INET;
    server_service.sin_addr.s_addr = inet_addr(listenip);
    server_service.sin_port = htons(listen_port);

    if (bind( server_socket, (SOCKADDR*) &(server_service),
sizeof(server_service) ) == SOCKET_ERROR){
        last_error = WSAGetLastError();
        return last_error;
    }

    if (listen(server_socket, 1) == SOCKET_ERROR){
        last_error = WSAGetLastError();
        return last_error;
    }

    return 0;
}

//Check if there is any pending incoming sockets
int cmsserverclass::checkforconnection(){
    SOCKET AcceptSocket = SOCKET_ERROR;          //This is probably old
remains not in use

    //Make sure accept won't block
/*
    fd_set fds;
    fds.fd_array[0] = server_socket;
    fds.fd_count = 1;
    timeval t;
    t.tv_sec = 0;
    t.tv_usec = 0;
    int res = select(0, &fds, 0, 0, &t);

    //Accept won't block inside here
```

```

        if ((res != 0) && (res != SOCKET_ERROR)){
*/
            AcceptSocket = accept( server_socket, NULL, NULL );
            last_error = WSAGetLastError();
            if (AcceptSocket != SOCKET_ERROR){
                return addsocket(AcceptSocket);
            }
//
        }

        return 0;
    }

//Check if there is data to be received in the incoming buffers for the
current socket
int cmsserverclass::readConnection(char *buffer, int *buffer_size){

//
    if (client_sockets == 0) client_sockets = first_client_socket;
//
    else client_sockets = client_sockets->next_client_socket;

    client_sockets = first_client_socket;

//Are there sockets to check?
if (client_sockets != 0){

    /*
        //Make sure recv won't block
        fd_set fds;
        fds.fd_array[0] = client_sockets->socket;
        fds.fd_count = 1;
        timeval t;
        t.tv_sec = 0;
        t.tv_usec = 0;
        int res = select(0, &fds, 0, 0, &t);

        //recv won't block inside here
        if ((res != 0) && (res != SOCKET_ERROR)){
*/
            *buffer_size = recv( client_sockets->socket,
buffer, *buffer_size, 0 );
            if (*buffer_size == 0){
                int id = client_sockets->socket_id;
                shutdown(client_sockets->socket, SD_BOTH);
                closesocket(client_sockets->socket);
                remsocket(client_sockets->socket_id);
                *buffer_size = -1;
                return id;
            }
/*
        } else {
            *buffer_size = 0;
            return 0;
        }
*/
    }
    else{
        *buffer_size = 0;
        return 0;
    }

    if (*buffer_size > 0) return client_sockets->socket_id;

    *buffer_size = 0;
    return 0;
}

//Search for a particular socket
CLIENT_SOCKETS *cmsserverclass::findSocket(int socket_id){
    CLIENT_SOCKETS *csock = first_client_socket;

    while (csock != 0){
        if (csock->socket_id == socket_id) return csock;
        csock = csock->next_client_socket;
    }
}

```

```
        return 0;
    }

    //Write to a socket
    int cmsserverclass::writeConnection(int socket_id, char *buffer, int
buffer_size){

        CLIENT_SOCKETS *csock = findSocket(socket_id);

        if (csock != NULL){
            send(csock->socket, buffer, buffer_size, 0);
        }

        return 0;
    }

    //Return the last error value
    int cmsserverclass::lerror(){
        return last_error;
    }
}
```

## 19.2.7 "cmsserverclass.h"

Deklarering for server-sockets

```
/*
 * Owner:                Åpen kanal (C) 2005
 *
 * Description:          Class to manage multiple network connections
(header file)
 *
 */

typedef struct tag_client_sockets{
    int socket_id;
    int last_error;
    SOCKET socket;
    int state;
    bool active;
    int dummy;

    tag_client_sockets *next_client_socket;
} CLIENT_SOCKETS, *LPCLIENT_SOCKETS;

class cmsserverclass{
private:
    int listen_port;
    int last_error;
    WSADATA wsadata;
    SOCKET server_socket;
    sockaddr_in server_service;
    int dummy;

    int next_socket_id;
    CLIENT_SOCKETS *client_sockets;
    CLIENT_SOCKETS *first_client_socket;
    CLIENT_SOCKETS *last_client_socket;

    int addsocket(SOCKET socket);
    CLIENT_SOCKETS *findSocket(int socket_id);

public:
```

---

```
cmserverclass();  
cmserverclass(int listen_port);  
int connect(char *listenip);  
int checkforconnection();  
int readConnection(char *buffer, int *buffer_size);  
int writeConnection(int socket_id, char *buffer, int buffer_size);  
void remsocket(int socket_id);  
int lerror();  
};
```

## 19.2.8 "cmsusers.cpp"

Håndtering av tilkoblede brukere.

```

/*****
 * Class cmsusers:
 *   Manage currently connected users
 *****/

#include "cmsusers.h"
//#include <iostream>

//using namespace std;

/*****
 * Constructor:
 *   Sets the initial state for a new instance of this class
 *****/

cmsusers::cmsusers(){
    nextuserid = 1;
    userlist = 0;
    firstuser = 0;
    lastuser = 0;
    basepath = 0;
}

/*****
 * Destructor:
 *   Cleans up when destroying this class
 *****/

cmsusers::~cmsusers(){
    //As long as there are users
    while (firstuser != 0){
        //Prepare the next user while the pointer is valid
        lastuser = firstuser->nextuser;

        //Does the current user have a set name?
        if (firstuser->name != 0){
            //Delete the users name
            delete firstuser->name;
        }
        //Delete the current user
        delete firstuser;

        //Process the next user on the next pass
        firstuser = lastuser;
    }
}

/*****
 *****/

** Private method implementations
 *****/

/*****
 * Method: finduser
 *   Searches a list of users for a given id, and
 *   returns the user object
 *****/

```

```
LPUSER cmsusers::finduser(int id){
    //Start search from the first user in the list
    LPUSER tempuser = firstuser;

    //As long as there are more users to search
    while (tempuser != 0){
        //Check if the right users ID is found (and return the user
object)
        if (tempuser->id == id) return tempuser;

        //No match: prepare for the next user to check
        tempuser = tempuser->nextuser;
    }

    //No user with that ID was found
    return 0;
}

bool cmsusers::userexists(int id){
    //Start search from the first user in the list
    LPUSER tempuser = firstuser;

    //As long as there are more users to search
    while (tempuser != 0){
        //Check if the right users ID is found (and return the user
object)
        if (tempuser->id == id) return true;

        //No match: prepare for the next user to check
        tempuser = tempuser->nextuser;
    }

    //No user with that ID was found
    return false;
}

/*****
 * Method: getszlength
 * Returns the length of the given zero terminated string
 * including the zero itself
 *****/

int cmsusers::getszlength(char *instring){
    //In case of null pointer input
    if (instring == 0) return 0;

    //Search from the beginnig of the string
    int count = 0;

    //Until first occurence of a null character
    char ch = 0x00;
    do{
        ch = *(instring+(count++));
    } while ((ch != '\0') && (ch != 0x0a));

    *((instring+(count))-1) = 0x00;

    //return the found count (plus one)
    return count+1;
}

/*****
 *****/
** Public method implementations
*****/
```

```
/*
 * Method: adduser
 *   Adds a user to the list of connected users
 *   returning an ID for the created object
 */
int cmsusers::adduser(){
    //Create the new object
    LPUSER newuser = new USER;

    //Assign it initial values
    newuser->id = nextuserid++;
    newuser->name = 0;
    newuser->password = 0;
    newuser->savefilename = 0;
    newuser->filename = 0;
    newuser->orgname = 0;
    newuser->orgid = 0;
    newuser->fpos = -1;
    newuser->ofs = 0;
    newuser->filesize = 0;
    newuser->logonstep = 0;
    newuser->connected = false;
    newuser->validated = false;
    newuser->nextuser = 0;
    newuser->closed = false;

    //Add it to the list
    if (firstuser == 0){
        firstuser = newuser;
        lastuser = newuser;
    } else {
        lastuser->nextuser = newuser;
        newuser->nextuser = firstuser;
        lastuser = newuser;
    }

    //Return the ID
    return newuser->id;
}

int cmsusers::adduser(int id){
    //Create the new object
    LPUSER newuser = new USER;

    //Assign it initial values
    newuser->id = id;
    newuser->name = 0;
    newuser->password = 0;
    newuser->savefilename = 0;
    newuser->filename = 0;
    newuser->orgname = 0;
    newuser->orgid = 0;
    newuser->fpos = -1;
    newuser->ofs = 0;
    newuser->filesize = 0;
    newuser->logonstep = 0;
    newuser->connected = false;
    newuser->validated = false;
    newuser->nextuser = 0;
    newuser->closed = false;

    //Add it to the list
    if (firstuser == 0){
        firstuser = newuser;
        lastuser = newuser;
    } else {
        lastuser->nextuser = newuser;
        newuser->nextuser = firstuser;
    }
}
```

```
        lastuser = newuser;
    }

    //Return the ID
    return newuser->id;
}

void cmsusers::removeuser(int id){
    if (firstuser == 0) return;

    if (firstuser->id == id){
        if (firstuser->nextuser == firstuser){
            delete firstuser;
            firstuser = 0;
            lastuser = 0;
            return;
        }

        LPUSER tempuser = firstuser;
        firstuser = firstuser->nextuser;
        lastuser->nextuser = firstuser;
        delete tempuser;

        return;
    }

    LPUSER tempuser = firstuser;
    while (tempuser->nextuser->id != id){
        tempuser = tempuser->nextuser;
    }

    if (tempuser->nextuser == lastuser){
        delete lastuser;
        lastuser = tempuser->nextuser;
        tempuser->nextuser = firstuser;
    }

}

}

/*****
 * Method: setname
 *   Names a user by his or her ID
 *   returning the ID for the user modified,
 *   or 0 if the user was not found
 *****/

int cmsusers::setname(int id, char *name){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){

        //Is the user already named?
        if (tempuser->name != 0){
            //Remove the old name
            delete tempuser->name;
            tempuser->name = 0;
        }

        //Set the user's name
        tempuser->name = name;
        //Return the user's ID
        return (tempuser->id);
    }

    //User was not found
    return 0;
}
```



```
}

void cmsusers::setbasepath(char *name){
    basepath = name;
}

int cmsusers::setfileid(int id, char *name){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){

        //Is the user already named?
        if (tempuser->fileid != 0){
            //Remove the old name
            delete tempuser->fileid;
            tempuser->fileid = 0;
        }

        //Set the user's name
        tempuser->fileid = createstring(name);
        //Return the user's ID
        return (tempuser->id);
    }

    //User was not found
    return 0;
}

char *cmsusers::getfileid(int id){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){
        //Return the user's ID
        return (tempuser->fileid);
    }

    //User was not found
    return 0;
}

int cmsusers::addtext(char *buffer, char *text2add, int pos){
    int cnt = 0;

    while (*(text2add+cnt) != 0x00){
        *(buffer+pos+cnt) = *(text2add+cnt);
        cnt++;
    }

    return pos+cnt;
}

void cmsusers::endtext(char *buffer, int pos){
    *(buffer+pos) = 0x00;
}

int cmsusers::setfilename(int id, char *name){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){

        tempuser->filename = name;

        int cnt = 0;
    }
}
```

```
handling      char *buffer = new char[2000]; //TODO: fix unsafe buffer

              cnt = addtext(buffer, basepath, cnt);
              cnt = addtext(buffer, "\\ ", cnt);
              cnt = addtext(buffer, tempuser->orgname, cnt);
              cnt = addtext(buffer, "\\ ", cnt);
              cnt = addtext(buffer, name, cnt);
              endtext(buffer, cnt);
              char *tempname=createstring(buffer);
              delete buffer;

              //Is the user already named?
              if (tempuser->savefilename != 0){
                  //Remove the old name
                  delete tempuser->savefilename;
                  tempuser->savefilename = 0;
              }

              //Set the user's name
              tempuser->savefilename = tempname;
              //Return the user's ID
              return (tempuser->id);
          }

          //User was not found
          return 0;
      }

int cmsusers::setorgname(int id, char *name){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){

        //Is the user already named?
        if (tempuser->orgname != 0){
            //Remove the old name
            delete tempuser->orgname;
            tempuser->orgname = 0;
        }

        //Set the user's name
        tempuser->orgname = createstring(name);
        //Return the user's ID
        return (tempuser->id);
    }

    //User was not found
    return 0;
}

int cmsusers::setorgid(int id, char *name){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){

        //Is the user already named?
        if (tempuser->orgid != 0){
            //Remove the old name
            delete tempuser->orgid;
            tempuser->orgid = 0;
        }

        //Set the user's name
        tempuser->orgid = createstring(name);
        //Return the user's ID
```

```
        return (tempuser->id);
    }

    //User was not found
    return 0;
}

char *cmsusers::getorgname(int id){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){
        return tempuser->orgname;
    }

    //User was not found
    return 0;
}

char *cmsusers::getfilename(int id){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){
        return tempuser->savefilename;
    }

    //User was not found
    return 0;
}

char *cmsusers::getdbfilename(int id){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){
        return tempuser->filename;
    }

    //User was not found
    return 0;
}

char *cmsusers::getorgid(int id){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){
        return tempuser->orgid;
    }

    //User was not found
    return 0;
}

int cmsusers::getlogonstep(int id){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){
        return tempuser->logonstep;
    }

    //User was not found
```

```

        return -1;
    }

void cmsusers::inclogonstep(int id){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){
        tempuser->logonstep++;
    }
}

void cmsusers::setvalidated(int id, bool validated){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){
        tempuser->validated = validated;
    }
}

/*****
 * Method: setpassword
 *   Sets the password for a user given his or her ID
 *   returning the ID for the user modified,
 *   or 0 if the user was not found
 *****/

int cmsusers::setpassword(int id, char *password){
    //Find the user to set password
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){

        //Has the user already given a password?
        if (tempuser->password != 0){
            //Remove the old password
            delete tempuser->password;
            tempuser->password = 0;
        }

        //Set the user's password
        tempuser->password = password;
        //Return the user's ID
        return (tempuser->id);
    }

    //User was not found
    return 0;
}

/*****
 * Method: getname
 *   Returns the name of a user given his or her ID
 *   or 0 if the user was not found
 *****/

char *cmsusers::getname(int id){
    //Find the user
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){
        //Return his or her name
        return tempuser->name;
    }
}

```

```
        //User was not found
        return 0;
    }

/*****
 * Method: getpassword
 * Returns the password of a user given his or her ID
 * or 0 if the user was not found
 *****/

char *cmsusers::getpassword(int id){
    //Find the user
    LPUSER tempuser = finduser(id);

    //User was found
    if (tempuser != 0){
        //Return his or her name
        return tempuser->password;
    }

    //User was not found
    return 0;
}

/*****
 * Method: createszstring
 * Creates a dynamic string from a constant
 * TODO: Replace this with proper string handling
 *****/

char *cmsusers::createstring(char *instring){
    //Get the input (constant) string's length
    int length = getszlength(instring);

    //Create room for the dynamic string
    char *tempstring = new char[length];

    //Copy the content of the constant string to the dynamic
    for (int i = 0; i<length; i++){
        tempstring[i] = instring[i];
    }

    //Return the pointer to the newly created string
    return tempstring;
}

bool cmsusers::writefile(int id, char *buffer, int *size){
    USER *user = finduser(id);

    if ((user != 0) && user->validated && (user->savefilename != 0)){
        if (user->ofs == 0){
            user->ofs = new ofstream(user->savefilename,
ios_base::binary);
        }

        user->ofs->write(buffer, *size);

        return true;
    }

    return false;
}

void cmsusers::setclosed(int id){
    //Find the user to name
    LPUSER tempuser = finduser(id);

    //User was found
```

```
        if (tempuser != 0){
            tempuser->closed = true;
        }
    }
bool cmsusers::getclosed(int id){
    //Find the user to name
    LPUSER tempuser = finduser(id);

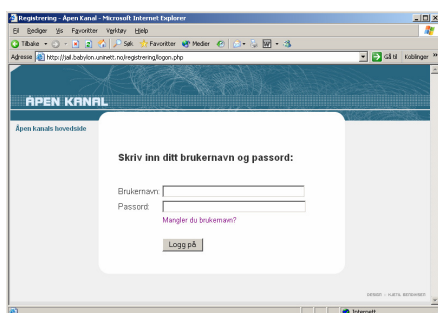
    //User was found
    if (tempuser != 0){
        return tempuser->closed;
    }
    return true;
}
```

## 20 D - Appendiks: Kildekode med forklaring av komprimering- og publiseringssystemet

### 20.1 Registreringssystemet

Systemet skal la bruker logge seg på Åpen Kanal sine tjenester samt ta i mot informasjon om nye brukere og organisasjoner.

#### 20.1.1 "logon.php"



Figur D-1: Logon.php

Påloggingsskjerm for registrering og administrasjon av brukerkonti samt for å få tilgang til *Åpen kanal* sitt system. Dette scriptet henter brukernavn og passord for å validere dette mot databasen. En sesjonsid blir generert og knyttet opp mot brukeren sammen med brukerens ip-adresse. Videre kommunikasjon mellom tjener og bruker foregår utelukkende på sesjonsid og ip-adresse. Dette gjøres for å unngå å sende brukernavn og passord i klartekst mer enn nødvendig. Https er under planlegging for å øke sikkerheten. En ny sesjonsid for hver side ville ha økt sikkerheten ytterligere, men tanken er foreløpig forlatt siden det vil "forkludre" en brukers mulighet for å kunne bruke tilbakeknappen på sin nettleser.

```
<?php
// Dersom bruker er paalogget, omdirigerer til hovedsiden;
// dersom ikke, presenterer paaloggingsbildet.

// -----
// --- Initialisasjon: ---
// -----

include_once("_designlib_.php");
include_once("_dblib_.php");
```

```
include_once("_htmllib_.php");
include_once("_contentlib_.php");
include_once("_funklib_.php");

// Hent brukerens ip adresse
$ip=getenv("REMOTE_ADDR");

// Hent sesjonsidentifikatoren om den finnes
$sessionid=$_GET["sessionid"];

// Hent inntastet brukernavn og passord om det eksisterer
$username=$_POST["username"];
$password=$_POST["password"];
$action=$_POST["action"];

// Valider alle brukerinntastede parametre
$username=validate($username);
$password=validate($password);
$sessionid=validate($sessionid);
$action=validate($action);

// -----
// --- Test for tilgang: ---
// -----

// Koble til database
connect();
selectdb("openchannel");

if ($action=="logon"){
    //Spør etter bruker
    $result = mysql_query("select userid, username, password
, password(\"$password\") from users where username=\"$username\"")
    or die("Could not get user info : " . mysql_error());

    $row = mysql_fetch_assoc($result);

    mysql_free_result($result)
    or die("could not free result");

    //Dersom spørring returnerte et resultat saa finnes brukeren
    if ($row) {
        //Dobbeltsjekk resultatet mot paaloggingskriterier
        if (($row["username"]==$username) &&
($row["password(\"$password\")"]==$row["password"])){
            //Bruker godkjent, generer sessionid og oppdater denne og ip i
            databasen
            $sessionid=getrandsessionid();
            mysql_query("update users set sessionid=\"$sessionid\" where userid=\"$
$row["userid"] . "\");");
            mysql_query("update users set lastip=\"$ip\" where userid=\"$
$row["userid"] . "\");");

            dblog($row["userid"], $ip, "Successfully logged on");
        } else {
            //Adgang ikke tillatt, sessionid satt til ingenting, ingen endring
            i databasen

            dblog(0, $ip, "Unsuccessful logon: Error in password", $username . ",
" . $password);
        }
        } else {
            if (!(($username=="") && ($password==""))) dblog(0, $ip, "Unsuccessful logon:
Bad user name", $username . ", " . $password);
        }
    }

    // Adgang ikke tillatt uten videre
    $accepted = false;
}
```



```

// Sjekke om bruker har tilgang
$accepted = validate_user($sessid, $ip);

//TODO: freeresult

// Koble fra databasen igjen
disconnect();

//Dersom bruker ikke har tilgang, presenter paaloggingsbildet
if (!$accepted) {
    designtop();

    echo "    <form action=\"logon.php\" method=\"post\">\n";
    echo "    <table>\n";
    echo "        <tr>\n";
    echo "            <td colspan=\"2\"><br><br><h3>Skriv inn ditt brukernavn og"
        " passord:</h3><br></td>\n";
    echo "        </tr>\n";
    echo "        <tr>\n";
    echo "            <td>Brukernavn:</td>\n";
    echo "            <td>\" . itext("bnavn", "40") . "\"</td>\n";
    echo "        </tr>\n";
    echo "        <tr>\n";
    echo "            <td>Passord:</td>\n";
    echo "            <td>\" . ipwd("pwd", "40") . "\"</td>\n";
    echo "        </tr>\n";
    echo "        <tr>\n";
    echo "            <td>&nbsp;</td>\n";
    echo "            <td align=\"right\"><font size=\"2\"><a
href=\"adduser.php\">Mangler du" .
        " brukernavn?</a></font></td>\n";
    echo "        </tr>\n";
    echo "    \n";
    echo "    <tr><td>&nbsp;</td><td>&nbsp;</td></tr>\n";
    echo "    \n";
    echo "    <tr>\n";
    echo "        <td>&nbsp;</td>\n";
    echo "        <td align=\"right\">\" . isubmit("Logg p&aring;") .
"<br><br></td>\n";
    echo "    </tr>\n";
    echo "    </table>\n";
    echo "    \" . ihidden("action","logon") . "\n";
    echo "    </form>\n";

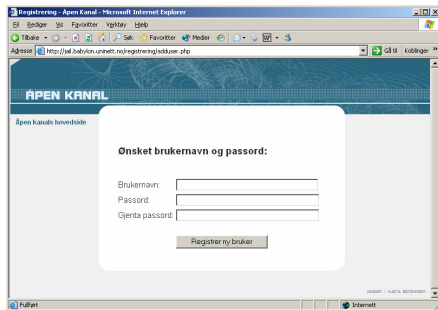
    //Dersom bruker hadde vaert validert ville ikke scriptet kjoert her.
    //Siden action variabelen har verdien logon saa har brukeren forsoekt
aa
    //logge seg paa. - Gi beskjed om at noe er galt!
    if ($action=="logon"){
        echo "<font color=\"ff0000\"><h3>Feil i brukernavn eller
passord!</h3></font>";
    }

    designmiddle();
    contentmenu();
    designbottom();
}
//Dersom bruker har tilgang, omdiriger til hovedside
else {
    header("Location:
http://jail.babylon.uninett.no/registrering/registrer.php?sessid=$sessid");
}

?>

```

## 20.1.2 "adduser.php"



Figur D-2: adduser.php

Bilde for å registrere ny brukerkonto. Ny bruker blir opprettet i databasen slik at brukeren kan logge seg på for å registrere seg og sin organisasjon. Funksjonalitet som har vært til vurdering er at bruker ikke får tilgang til mer enn å kun registrere seg og sin organisasjon før han/henne har sendt inn et signert skjema og blitt godkjent av en administrator. Dette har blitt forkastet siden dette vil gjøre det vanskelig for en bruker å kunne benytte seg av systemet umiddelbart. Åpenhet har blitt prioritert over sikkerhet på dette punktet.

```
<?php
include_once("_designlib_.php");
include_once("_contentlib_.php");
include_once("_htmllib_.php");
include_once("_dblib_.php");
include_once("_funklib_.php");

//Hent og valider brukerinntede parametre
$action=validate($_POST["action"]);
$bnavn=validate($_POST["bnavn"]);
$pwd1=validate($_POST["pwd1"]);
$pwd2=validate($_POST["pwd2"]);

connect();
selectdb("openchannel");

//Kun alfanumeriske tegn med noe attaatt
$nameaccepted=isalphanum($bnavn);
//Passord maa vaere like
$passwordmatch=$pwd1==$pwd2;
//Passord maa vaere av en viss kvalitet
$pwdok=goodpass($pwd1);
//Bruker maa ikke finnes fra foer av
$userexists=usrexist($bnavn);

if (($action=="adduser") && $nameaccepted && $passwordmatch &&
($bnavn!="") && $pwdok && !$userexists){
    mysql_query("insert into users (username, password) values (\\"$bnavn\\",
password(\\"$pwd1\\"))");
```

```

    header("Location:
http://jail.babylon.uninett.no/registrering/logon.php");
    exit;
}

designtop();

echo "    <form action=\"adduser.php\" method=\"post\">\n";
echo "    <table>\n";
echo "        <tr>\n";
echo "            <td colspan=\"2\"><br><br><h3>&Oslash;nsket brukernavn og" .
"    " passord:</h3><br></td>\n";
echo "        </tr>\n";
echo "        <tr>\n";
echo "            <td>Brukernavn:</td>\n";
echo "            <td>" . itext("bnavn", "40") . "</td>\n";
echo "        </tr>\n";
echo "        <tr>\n";
echo "            <td>Passord:</td>\n";
echo "            <td>" . ipwd("pwd1", "40") . "</td>\n";
echo "        </tr>\n";
echo "        <tr>\n";
echo "            <td>Gjenta passord:</td>\n";
echo "            <td>" . ipwd("pwd2", "40") . "</td>\n";
echo "        </tr>\n";
echo "\n";
echo "        <tr><td>&nbsp;</td><td>&nbsp;</td></tr>\n";
echo "\n";
echo "        <tr>\n";
echo "            <td>&nbsp;</td>\n";
echo "            <td align=\"right\"><input type=\"submit\" value=\"Registrer
ny" .
"    " bruker\"><br><br></td>\n";
echo "        </tr>\n";
echo "    </table>\n";
echo "<input type=\"hidden\" name=\"action\" value=\"adduser\">\n";
echo "    </form>\n";

//Vis feilmeldinger (kun en selv om det er flere som er begaatt)
if ($action=="adduser"){
    if (!$nameaccepted) echo "<font color=\"ff0000\"><h3>Kun alfanumeriske
tegn i brukernavn!</h3></font><br>\n";
    else if ($userexists) echo "<font color=\"ff0000\"><h3>Brukernavnet er
i bruk!</h3></font><br>\n";
    else if ($bnavn=="") echo "<font color=\"ff0000\"><h3>Du kan ikke
registrere blankt brukernavn!</h3></font><br>\n";
    else if (!$passwordmatch) echo "<font color=\"ff0000\"><h3>Passordene
matcher ikke!</h3></font><br>\n";
    else if (!$pwdok) echo "<font color=\"ff0000\"><h3>Du m&aring; nok
finne et bedre passord!</h3></font><br>\n";
}
    designmiddle();
    contentmenu();
    designbottom();
?>

```

## 20.1.3 "registrer.php"



Figur D-3: register.php

Dette er scriptet som genererer sidene hvor en ny bruker kan registrere informasjon om seg og sin organisasjon. Hvis dette scriptet blir kalt uten at en bruker er validert, vil brukeren diskret bli omdirigert tilbake til påloggingsscriptet. Ellers vil brukeren måtte fullføre registreringen for å kunne få full tilgang til systemet.

```
<?php
include_once("_designlib_.php");
include_once("_contentlib_.php");

//Set default access to not allowed
$accepted=false;

//Get remote IP
$ip=getenv("REMOTE_ADDR");

$ssid=$_GET["ssid"];

$link = mysql_connect("localhost", "defaultuser", "")
    or die("Could not connect : " . mysql_error());

mysql_select_db("openchannel",$link)
    or die("Could not select db : " . mysql_error());

$result = mysql_query("select username, lastip, step from users where
ssid=\"\$ssid\" and lastip=\"\$ip\"")
    or die("Could not get user info : " . mysql_error());

$row = mysql_fetch_assoc($result);

// Debug code:
// echo $row["username"] . "<br>" . $row["password(\"$pwd\)");
// echo "<br>" . $pwd . "<br>" . $row["password"] . "<br>";

mysql_close($link)
    or die("Could not close db : " . mysql_error());

if ($row) {
    if ($row["lastip"]== $ip) {
        $accepted=true;
    }
}

if (!$accepted){ //Bruker ikke akseptert
    header("Location:
http://jail.babylon.uninett.no/registrering/logon.php");
    exit; //Just in case...
```

```

} else { //Bruker akseptert

    if ($row["step"]==0){ //Tvungen reg av seg selv
        designtop();
        contentpage1();
        designmiddle();
        contentmenu();
        designbottom();
    } else if ($row["step"]==1){ //Tvungen reg av org
        designtop();
        contentpage2();
        designmiddle();
        contentmenu();
        designbottom();
    } else if ($row["step"]==2){ //Tvungen reg av
        designtop();
        designmiddle();
        contentmenu();
        designbottom();
    } else {
        designtop();
        echo "<h3>Velkommen!</h3>\n";
        echo "<b>Dette skal v&aelig;re et system hvor du som
bruker skal kunne ".
            "gj&oslash;re alt du trenger fra et
sted.</b><br><br>";
        echo "Systemet er under utvikling. For opplasting brukes
i mellomtiden en".
        " <a
href=\"http://jail.babylon.uninett.no:8080/akanal/Arkiv/FileLibrary5/Files/
FTP-program.exe\" target=\"_blank\">FTP-klient</a>.".
        " For programmering av sendplan brukes <a
href=\"http://129.241.26.184:1220\" target=\"_blank\">Envivio sitt".
        " web-grensesnitt</a>.".
        " Automatisk komprimering vil fungere innen onsdag.";
        echo "<br>Inntil videre er det bare &aring; sette i
gang &aring; bruke systemet";
        echo " med de funksjonene som alt er ferdig.";
        echo "<br><br><b>&lt;====</b> Velg p&aring; menyen
til venstre hva du vil gj&oslash;re.";
        echo "<br><br>Kom gjerne med tilbakemeldinger!";
        designmiddle();
        contentmenu();
        designbottom();
    }
}

?>

```

#### 20.1.4 "addred.php"

Registrerer personlig informasjon.

Som alltid må en bruker være validert for å kunne eksekvere scripts. Det gjelder også for dette scriptet. Brukerinformasjonen som ble fylt inn via "registrering.php" sin side 1 blir lagret i databasen og knyttet opp mot den påloggede bruker. Feltet "step" som er knyttet mot brukeren i databasen økes med en, hvilket forteller "registrering.php" at side 2 skal

presenteres. Dette scriptet genererer ikke noe tekst til nettleseren, men omdirigerer tilbake til ”registrering.php” når det er utført. Dette vil presentere brukeren for side 2.

```
<?php
include_once("_dblib_.php");

//Get remote IP
$ip=getenv("REMOTE_ADDR");

//Get session id if it exists
$ssid=$_GET["ssid"];

//SQL stuff
connect();
selectdb("openchannel");

$accepted = false;
$accepted = validate_user($ssid, $ip);

//If any results returned...
if ($accepted) {
    $redfullnavn=$_POST["redfullnavn"];
    $redpernum=$_POST["redpernum"];
    $redadr1=$_POST["redadr1"];
    $redadr2=$_POST["redadr2"];
    $redtlfjobb=$_POST["redtlfjobb"];
    $redtlfhjem=$_POST["redtlfhjem"];
    $redtlfmob=$_POST["redtlfmob"];
    $redepost=$_POST["redepost"];
    $redtlffax=$_POST["redtlffax"];

    $qry="insert into redaktorer (fullnavn, " .
        "adressed, adresse2, " .
        "jobbtelefon, hjemtelefon, mobiltelefon, " .
        "telefaks, epost) values (\\"$redfullnavn\\", " .
        "\\"$redadr1\\", " .
        "\\"$redadr2\\", "\\"$redtlfjobb\\", " .
        "\\"$redtlfhjem\\", "\\"$redtlfmob\\", " .
        "\\"$redtlffax\\", "\\"$redepost\\")";
    mysql_query($qry);

    $qry="select redid from redaktorer where personnummer=\\"$redpernum\\"";
    $result=mysql_query($qry);

    if ($result) {
        $row=mysql_fetch_assoc($result);
        $redid=$row["redid"];
        $qry="select userid from users where ssid=\\"$ssid\\"";
        $result=mysql_query($qry);
        if ($result){
            $row=mysql_fetch_assoc($result);
            $userid=$row["userid"];
            $qry="update users set redid=\\"$redid\\" where userid=\\"$userid\\"";
            mysql_query($qry);
            $qry="update users set step=\\\"1\\\" where userid=\\"$userid\\"";
            mysql_query($qry);
        }
    }
}

disconnect();

header("Location:
http://jail.babylon.uninett.no/registrering/registrer.php?ssid=$ssid");
?>
```

## 20.1.5 "addorg.php"

Registerer informasjon om organisasjon.

Informasjon om organisasjon som ble fylt inn via "registrering.php" sin side 2 blir lagret i databasen og knyttet opp mot den påloggede bruker. Feltet "step" som er knyttet mot brukeren i databasen økes med en, hvilket forteller "registrering.php" at side 3 skal presenteres. Dette scriptet genererer ikke noe tekst til nettleseren, men omdirigerer tilbake til "registrering.php" når oppgaven er utført. Dette vil presentere brukeren for side 3.

```
<?php
include_once ("_browselib_.php");

//Set default access to not allowed
$accepted=false;

//Get remote IP
$ip=getenv("REMOTE_ADDR");

//Debug:
$fop=$_POST["fop"];
if (isset($fop)){ echo "Finn organisasjon trykket!"; }

$ssid=$_GET["ssid"];

$link = mysql_connect("localhost", "defaultuser", "")
or die("Could not connect : " . mysql_error());

mysql_select_db("openchannel",$link)
or die("Could not select db : " . mysql_error());

$result = mysql_query("select username, lastip from users where
ssid=\"".$ssid.\" and lastip=\".$ip.\"")
or die("Could not get user info : " . mysql_error());

$row = mysql_fetch_assoc($result);

if ($row) {
    if ($row["lastip"]== $ip) {
        $accepted=true;

        $orgnavn=$_POST["orgnavn"];
        $orgnum=$_POST["orgnum"];
        $orgtlf=$_POST["orgtlf"];
        $orglonpers=$_POST["orglonpers"];
        $prodidag=$_POST["prodidag"];
        $prodfq=$_POST["prodfq"];
        $prodsendstart=$_POST["prodsendstart"];
        $prodkunnet=$_POST["prodkunnet"];
        $orgweb=$_POST["orgweb"];

        $orgadr1=$_POST["orgadr1"];
        $orgadr2=$_POST["orgadr2"];
```

```
$qry="insert into organisasjoner (orgnavn, " .
"orgnummer, telefon, lonnpersonell, " .
"fjernsynvideoprod, timerperuke, " .
"forvsendstart, " .
"formidlesmedium, webaddress, " .
"adressed, adresse2) " .
"values (\ "$orgnavn\ ", " .
"\ "$orgnum\ ", "\ "$orgtlf\ ", " .
"\ "$orglonpers\ ", "\ "$prodidag\ ", " .
"\ "$prodfq\ ", " .
"\ "$prodsendstart\ ", " .
"\ "$prodkunnet\ ", " .
"\ "$orgweb\ ", "\ "$orgadr1\ ", "\ "$orgadr2\ ")";

mysql_query($qry);

//tilegg for adresse
//
//          $qry2="insert into redaktorer (adressed, " .
//          "adresse2) " .
//          "values (\ "$redadr1\ ", " .
//          "\ "$redadr2\ ")";
//
//          mysql_query($qry2);

$orgnummer="\ "$orgnum\ ";
$orgnummer="\ "$orgnum\ ";
$result=mysql_query($qry);

if ($result) {
    $row=mysql_fetch_assoc($result);
    $orgid=$row["orgid"];
    $qry="select userid from users where " .
        "sessid=\ "$sessid\ ";
    $result=mysql_query($qry);
    if ($result){
        $row=mysql_fetch_assoc($result);
        $userid=$row["userid"];
        $qry="update users set orgid=\ "$orgid\ " .
            "where userid=\ "$userid\ ";
        mysql_query($qry);

        if ($prodkunnet=="nett"){
            $qry="update users set step=\ "99\ " where
userid=\ "$userid\ ";
        } else {
            $qry="update users set
step=\ "2\ " where userid=\ "$userid\ ";
        }

        mysql_query($qry);
    }
}

ftp_addorg($orgnavn);

}

mysql_close($link);

header("Location:
http://jail.babylon.uninett.no/registrering/registrer.php?sessid=$sessid");

?>
```



### 20.1.6 "index.php"

"index.html" eller "index.php" er standardfila som blir lastet ned til nettleseren når det ikke er spesifikt oppgitt et filnavn. "jail.babylon.uninett.no" er et eksempel hvor intet filnavn er oppgitt. "/index.html" er et motsatt eksempel. Enkelte webtjenere viser innholdet av en katalog dersom det ikke finnes en fil med navn "index.\*". Dette scriptets funksjon er å omdirigere til riktig side dersom ingen fil er spesifisert for dette filområdet.

```
<?php
    header("Location: http://jail.babylon.uninett.no/registrering/logon.php");
?>
```

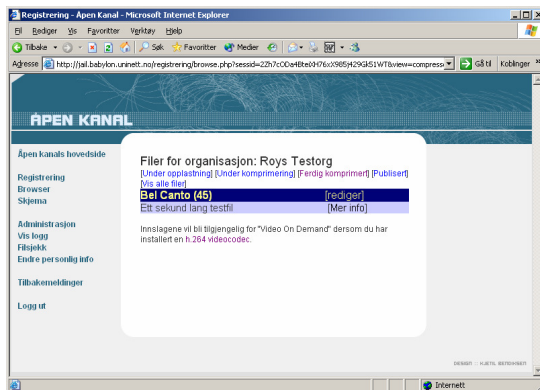
Dette konkluderer dokumentasjonen på registreringssystemet.

## 20.2 Opplastingssystemet

Systemet skal la bruker (organisasjon) laste opp innslag, håndtere disse innslagene samt se på egne og andre organisasjoners innslag. Systemet skal også utføre oppgaver som er usynlige for brukeren som for eksempel automatisk (re-)komprimering fra opplastet format til andre formater som er nødvendige.

I denne versjonen av systemet foregår opplasting av innslag gjennom en ftp server. Når en bruker har lastet opp et innslag og logger seg på Åpen Kanal sine tjenester, vil brukeren kunne se sin organisasjon sine filer gjennom nettleseren sin. Når nye filer er lastet opp, må brukeren foreløpig trykke på en link som utfører et søk etter nye filer som legges inn i databasen slik at de er tilgjengelige via nettleseren for innlegging av informasjon om innslag og rettigheter.

### 20.2.1 "browse.php"



Figur D-4: browse.php

Viser en liste over filer som pålogget organisasjon har i sin "stall". "browse.php" er meget spesifikt knyttet opp mot organisasjon i denne versjonen. En ny og mer generell versjon som tillater å se andre organisasjoners filer samt mulighet for alternativ sortering er under utarbeidelse (og faktisk kommet godt på vei.)

```
<?php
include_once "_designlib_.php";
include_once "_contentlib_.php";
include_once "_browselib_.php";
include_once "_dblib_.php";
```

```
//Get remote IP
$ip=getenv("REMOTE_ADDR");

//Get session id if it exists
$sessionid = $_GET["sessionid"];

connect();
selectdb("openchannel");

$accepted = false;
$accepted = validate_user($sessionid, $ip);

if ($accepted){
    design_top();
    browseview();
    designmiddle();
    contentmenu();
    designbottom();
} else {
    header("Location: http://jail.babylon.uninett.no/registrering");
}

?>
```

## 20.2.2 "editcontent.php"

Legger inn eller oppdaterer informasjon om innslag som er lastet opp.

```
<?php
//Load needed libraries for this script
include_once ("_browselib_.php");
include_once ("_dblib_.php");
include_once ("_designlib_.php");
include_once ("_contentlib_.php");
include_once ("_htmllib_.php");

//Set default access to not allowed
$accepted=false;

//Get remote IP
$ip=getenv("REMOTE_ADDR");

$sessionid=$_GET["sessionid"];

connect();
selectdb("openchannel");

$oid=getorgid($sessionid);
$oname=getorgname($oid);
$fileid=$_GET["fileid"];
$fname=getdbitem("filename", "content", "contentid", $fileid);
$fdesc=getdbitem("description", "content", "contentid", $fileid);
$ftitle=getdbitem("title", "content", "contentid", $fileid);
$ftapename=getdbitem("tapename", "content", "contentid", $fileid);
$ftcin=getdbitem("tc_in", "content", "contentid", $fileid);
$ftcout=getdbitem("tc_out", "content", "contentid", $fileid);
$flength=getdbitem("length", "content", "contentid", $fileid);
$fcsel=getdbitem("categoryid", "content", "contentid", $fileid);
$fvfri=getdbitem("vfri", "content", "contentid", $fileid);

design_top();
```



### 20.2.3 "editcontent\_alter.php"

Denne funksjonen utfører endringene som er gjort på siden som er generert av "editcontent.php".

```
<?php
//Load needed libraries for this script
include_once ("_browselib_.php");
include_once ("_dblib_.php");
include_once ("_designlib_.php");
include_once ("_contentlib_.php");

//Set default access to not allowed
$accepted=false;

//Get remote IP
$ip=getenv("REMOTE_ADDR");

$ssid=$_GET["ssid"];

connect();
selectdb("openchannel");

$oid=getorgid($ssid);
$oname=getorgname($oid);
$fid=$_POST["fid"];
$fname=getdbitem("filename", "content", $fid);
$fdesc=$_POST["desc"];
$ftitle=$_POST["title"];
$ftname=$_POST["tname"];
$ftcin=$_POST["tcin"];
$ftcout=$_POST["tcout"];
$vfri=$_POST["vfri"];
$catid=$_POST["catid"];

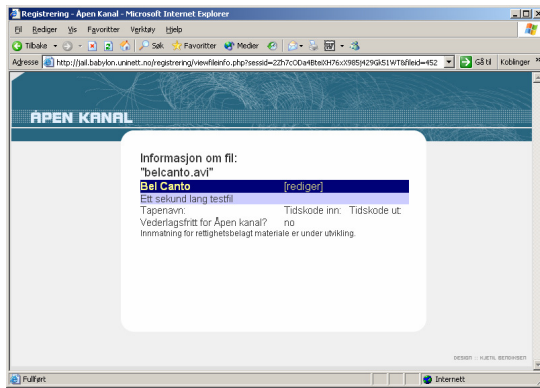
mysql_query("update content set description=\"\$fdesc\" where
contentid=\"\$fid\"");
mysql_query("update content set title=\"\$ftitle\" where
contentid=\"\$fid\"");
mysql_query("update content set tapename=\"\$ftname\" where
contentid=\"\$fid\"");
mysql_query("update content set tc_in=\"\$ftcin\" where
contentid=\"\$fid\"");
mysql_query("update content set tc_out=\"\$ftcout\" where
contentid=\"\$fid\"");
mysql_query("update content set vfri=\"\$vfri\" where
contentid=\"\$fid\"");
mysql_query("update content set categoryid=\"\$catid\" where
contentid=\"\$fid\"");

disconnect();

header("Location:
http://jail.babylon.uninett.no/registrering/browse.php?ssid=$ssid");

?>
```

### 20.2.4 "viewfileinfo.php"



Figur D-5: viewfileinfo.php

Dette scriptet viser “mer informasjon” om et innslag. Når en bruker ser på oversikten over filer er det ønskelig å kun presentere essensiell informasjon. Dersom en bruker trenger mer informasjon om et innslag, så kan han eller henne nå dette via en knapp i filtreet.

```
<?php
    include_once("_dblib_.php");
    include_once("_designlib_.php");
    include_once("_contentlib_.php");

    $sessid=$_GET["sessid"];
    connect();
    selectdb("openchannel");

    designtop();

//    $soid = getorgid($sessid);
//    $oname = getorgname($soid);
    $fid = $_GET["fileid"];
    $fname = getdbitem("filename", "content", "contentid", $fid);
    echo "<font size=\"4\">Informasjon om fil:<br> &quot;" . $fname .
    "&quot;</font><br>";

    $content=mysql_query("select * from content where contentid=\"$fid\"");
    $row=mysql_fetch_assoc($content);

    if (!$row) echo "<br>Ingen filer funnet!<br>";

    echo "<table width=\"400\" cellspacing=\"0\" cellpadding=\"0\"
border=\"0\">\n";

    while ($row){
        //Show title where exists, show filename if not
        $disptitle = $row["title"];
        if (!$disptitle) $disptitle=$row["filename"];

        echo "<tr>\n";
        echo "<tr><td class=\"tablehead\" colspan=\"4\"><b>" . $disptitle .
            "</b></td><td class=\"tablehead\" align=\"right\" colspan=\"2\">" .
            "[<a class=\"tablehead\"
href=\"editcontent.php?sessid=$sessid&fileid=" . $row["contentid"] .
"\">rediger</a></td>";
        echo "<tr><td colspan=\"6\" bgcolor=\"#ccccff\">";
        $descript=$row["description"];

```

```
        if (!$descript){ echo "Ingen beskrivelse for denne fila, trykk
p&aring; rediger for &aring; endre."; }
        else echo $descript;
        echo "</td></tr>";
        echo "<tr><td colspan=\"4\">Tapenavn:</td><td>Tidskode
inn:</td><td>Tidskode ut:</td>";
        echo "<tr><td colspan=\"4\">" . $row["tapename"] .
        "</td><td>" . $row["tc_in"] .
        "</td><td>" . $row["tc_out"] . "</td></tr>\n";
        echo "<tr><td colspan=\"4\">Vederlagsfritt for &Aring;pen
kanal?</td><td colspan=\"2\">" . $row["vfri"] . "</td></tr>";
        $row=mysql_fetch_assoc($content);
    }
    echo "</table>\n";
    echo "Innmatning for rettighetsbelagt materiale er under utvikling.\n";

designmiddle();
designbottom();
?>
```

## 20.3 Administrasjonsfunksjoner

### 20.3.1 "administer.php"

Presenterer en meny med administrative funksjoner for de som har administrative rettigheter.

```
<?php
//Load needed libraries for this script
include_once ("_browselib_.php");
include_once ("_dblib_.php");
include_once ("_designlib_.php");
include_once ("_contentlib_.php");

//Set default access to not allowed
$accepted=false;

//Get remote IP
$ip=getenv("REMOTE_ADDR");

$ssid=$_GET["ssid"];

designtop();

echo "<h3>Denne siden er ikke ferdig</h3><br>";
echo "<h4><b>Administrasjon av &aring;pen kanal sine sider:</b></h4>\n";

echo "<a href=\"_admusers.php?ssid=$ssid\">Brukere</a><br>\n";
echo "<a href=\"_\">Organisasjoner</a><br>\n";
echo "<a href=\"_category.php?ssid=$ssid\">Temaer</a><br>\n";
echo "<a
href=\"_admaccgrpnames.php?ssid=$ssid\">Rettigheter</a><br>\n";
// echo "<a href=\"_\"></a><br>\n";

designmiddle();
contentmenu();
designbottom();
```

```
disconnect();  
  
// header("Location:  
http://jail.babylon.uninett.no/registrering/browse.php?sessid=$sessid");  
  
?>
```

## 20.3.2 "admusers.php

Script for å administrere brukere. I denne versjonen kan man kun se på informasjonen som ligger i databasen.

```
<?php  
include_once("_dblib_.php");  
include_once("_designlib_.php");  
include_once("_contentlib_.php");  
  
//Get remote IP  
$ip=getenv("REMOTE_ADDR");  
  
//Get session id if it exists  
$sessid=$_GET["sessid"];  
  
//SQL stuff  
connect();  
selectdb("openchannel");  
  
$accepted = false;  
$accepted = validate_user($sessid, $ip);  
  
//If any results returned...  
if ($accepted) {  
    $qry="select * from users";  
    $result=mysql_query($qry);  
  
    if ($result) {  
        $row=mysql_fetch_assoc($result);  
  
        designtop();  
  
        echo "<table border=\"1\" cellspacing=\"0\" cellpadding=\"2\">\n";  
  
        while ($row){  
            $uid=$row["userid"];  
            $oid=$row["orgid"];  
            $rid=$row["redid"];  
            $uname=$row["username"];  
  
            $result2=mysql_query("select * from redaktorer where  
redid=\"".$rid."");  
            if (!$result2){  
                $rname="N/A";  
                $repost="N/A";  
                $radr1="N/A";  
                $radr2="N/A";  
                $rtjobb="N/A";  
                $rthjem="N/A";  
                $rtmob="N/A";  
                $rtfaks="N/A";  
            } else {
```



```

row2=mysql_fetch_assoc($result2);

$name=$row2["fullnavn"];
$repost=$row2["epost"];
$radr1=$row2["adresse1"];
$radr2=$row2["adresse2"];
$rtjobb=$row2["jobbtelefon"];
$rthjem=$row2["hjemtelefon"];
$rtmob=$row2["mobiltelefon"];
$rtfaks=$row2["telefaks"];
}

$result2=mysql_query("select * from organisasjoner where
orgid=\"$oid\"");
if (!$result2){
    $name="N/A";
    $num="N/A";
    $adr1="N/A";
    $adr2="N/A";
    $web="N/A";
} else {
    row2=mysql_fetch_assoc($result2);

    $name=$row2["orgnavn"];
    $num=$row2["orgnummer"];
    $adr1=$row2["adresse1"];
    $adr2=$row2["adresse2"];
    $web=$row2["webadresse"];
}

echo " <tr>\n";
echo " <td class=\"tablehead\" colspan=\"3\"><b> . $name . " (
. $name . " )&nbsp;  </b></td>\n";
echo " <td class=\"tablehead\" colspan=\"2\"><b> . $repost .
"&nbsp;  </b></td>\n";
echo " </tr>\n";
echo " <tr>\n";
echo " <td class=\"tablecontent\" colspan=\"3\"> . $adr1 .
"&nbsp;  </td>\n";
echo " <td class=\"tablecontent\">Hjem:&nbsp;  </td>\n";
echo " <td class=\"tablecontent\"> . $rthjem . "&nbsp;  </td>\n";
echo " </tr>\n";
echo " <tr>\n";
echo " <td class=\"tablecontent\"> . $adr2 . "&nbsp;  </td>\n";
echo " <td class=\"tablecontent\">&nbsp;  </td>\n";
echo " <td class=\"tablecontent\">&nbsp;  </td>\n";
echo " <td class=\"tablecontent\">Mobil:</td>\n";
echo " <td class=\"tablecontent\"> . $rtmob . "&nbsp;  </td>\n";
echo " </tr>\n";
echo " <tr>\n";
echo " <td class=\"tablecontent\">&nbsp;  </td>\n";
echo " <td class=\"tablecontent\">&nbsp;  </td>\n";
echo " <td class=\"tablecontent\">&nbsp;  </td>\n";
echo " <td class=\"tablecontent\">Jobb:</td>\n";
echo " <td class=\"tablecontent\"> . $rtjobb . "&nbsp;  </td>\n";
echo " </tr>\n";
echo " <tr>\n";
echo " <td class=\"tablecontent\" colspan=\"3\"> . $name .
"&nbsp;  </td>\n";
echo " <td class=\"tablecontent\">Faks:</td>\n";
echo " <td class=\"tablecontent\"> . $rtfaks . "&nbsp;  </td>\n";
echo " </tr>\n";

//
echo " </tr>\n";
row=mysql_fetch_assoc($result);
}

echo "</table>\n";

```

```
        designmiddle();
        contentmenu();
        designbottom();
    }
} else {
    header("Location:
http://jail.babylon.uninett.no/registrering/logon.php");
}

    disconnect();

?>
```

### 20.3.3 "category.php"

Administrasjon av kategorier. Kan foreløpig kun se på og legge til kategorier.

```
<?php
include_once("_dblib_.php");
include_once("_designlib_.php");
include_once("_contentlib_.php");
include_once("_funklib_.php");
include_once("_htmllib_.php");

//Get remote IP
$ip=getenv("REMOTE_ADDR");

//Validate user input
$action=validate($_POST["action"]);
$catname=validate($_POST["catname"]);
$cocom=validate($_POST["cocom"]);
$sessionid=validate($_GET["sessionid"]);

//SQL stuff
connect();
selectdb("openchannel");

$accepted = false;
$accepted = validate_user($sessionid, $ip);

//If any results returned...
if ($accepted) {
    designtop();

    if ($action=="add"){
        mysql_query("insert into category (name, comment) values
(\\"$catname\\",\\"$cocom\\")");
        echo "<font color=\\\"#ff0000\\\"><h3>Kategori er lagt
til!</h3></font>\n";
    }

    $result=mysql_query("select * from category");
    $row=mysql_fetch_assoc($result);
    echo "<table>\n";
    echo " <tr>\n";
    while ($row){
        echo " <tr>\n";
        echo " <td>" . $row["categoryid"] . "</td>\n";
        echo " <td>" . $row["name"] . "</td>\n";
        echo " <td>" . $row["comment"] . "</td>\n";
        echo " </tr>\n";
    }
}
```

```
        $row=mysql_fetch_assoc($result);
    }
    echo "</table>\n";

    echo "Redigere kategorier:<br>\n";
    echo "<form action=\"category.php?sessid=$sessid\" method=\"POST\">\n";
//    echo "Navn p&aring; kategori: <input type=\"text\" name=\"catname\"
size=\"40\"><br>\n";
    echo "Navn p&aring;kategori: <input type=\"text\" name=\"ccom\"
size=\"40\"><br>\n";
    echo ihidden("action", "add") . "\n";
    echo isubmit("Legg til kategori") . "\n";
    echo "</form>\n";

    designmiddle();
    contentmenu();
    designbottom();
} else {
    header("Location:
http://jail.babylon.uninett.no/registrering/registrer.php?sessid=$sessid");
}

?>
```

### 20.3.4 "changeowninfo.php"

Script for å gi brukerne en mulighet for å endre på sine egne registreringer. Foreløpig ikke noen funksjonalitet her, kun en side som forteller at det kommer en gang i fremtiden.

```
<?php
//Load needed libraries for this script
include_once ("_browselib_.php");
include_once ("_dblib_.php");
include_once ("_designlib_.php");
include_once ("_contentlib_.php");

//Set default access to not allowed
$accepted=false;

//Get remote IP
$ip=getenv("REMOTE_ADDR");

$ssid=$_GET["sessid"];

designtop();

echo "<b>Endring av egen info kommer snart!</b>";

designmiddle();
contentmenu();
designbottom();

disconnect();

// header("Location:
http://jail.babylon.uninett.no/registrering/browse.php?sessid=$sessid");

?>
```

## 20.3.5 "fcheck.php"

Et automatisert script som sjekker etter nye filer på ftp-serveren. Nye filer legges inn og filer hvor filstørrelsen ikke har endret seg fra forrige sjekk får status "ferdig lastet opp". Det er en cron-jobb som skal sørge for at dette scriptet kjører.

```
<?php
    include_once ("dblib.php");

    function checkfiles($ftpstream, $oname="", $oid="", $fext="*."){
        $ftpcontent=ftp_nlist($ftpstream, "/ORGANISASJONER/" . $oname . "/" .
        $fext);

        if ($ftpcontent){

            foreach ($ftpcontent as $entry) {
                $fname=$entry;
                $fsize=ftp_size($ftpstream, "/ORGANISASJONER/" . $oname . "/" .
                $fname);

                $result=mysql_query("select * from content where
filename=\"\$fname\" and orgid=\"\$oid\"");
                $row=mysql_fetch_assoc($result);
                if (!$row) {
                    echo "Lagt til i databasen: " . $fname . "<br>\n";
                    mysql_query("insert into content (filename, orgid, filesize)
values (\$fname\", \"\$oid\", \"\$fsize\")");
                } else {
                    if ($row["filesize"]==$fsize) {
                        $result2=mysql_query("select fin_upload from content where
contentid=\"\" . $row["contentid"] . "\"");

                        if ($result2){
                            $row2=mysql_fetch_assoc($result2);

                            if ($row2["fin_upload"]=="no"){
                                echo "Ferdig opplastet: " . $fname . "<br>\n";
                                mysql_query("update content set fin_upload=\"yes\" where
contentid=\"\" . $row["contentid"] . "\"");
                            }

                        } else {
                            echo "fcheck.php - ftp: file is done, but not found in db
error";
                        }
                    } else {
                        echo "Fortsatt under opplastning: " . $fname . "<br>\n";
                    }
                }
            }
        }

        connect();
        selectdb("openchannel");

        $ftpstream=ftp_connect("129.241.26.183");
        if (!$ftpstream){
            echo "fcheck.php - Could not connect to ftp server error<br>\r\n";
        } else {
```

```
ftp_login($ftpstream, "Open_Channel", "publicaccess!");
ftp_pasv($ftpstream, TRUE);

$result=mysql_query("select * from organisasjoner");
if ($result) {
    $row=mysql_fetch_assoc($result);

    while ($row) {
        $oid=$row["orgid"];
        $oname=$row["orgnavn"];
        echo $oid . " - " . $oname . "<br>\n";

        checkfiles($ftpstream, $oname, $oid, "*.avi");
        checkfiles($ftpstream, $oname, $oid, "*.mov");
        checkfiles($ftpstream, $oname, $oid, "*.mpg");

        $row=mysql_fetch_assoc($result);
    }

    ftp_close($ftpstream);
} else {
    echo "fcheck.php - could not connect to db error<br>\r\n";
}
}
?>
```

### 20.3.6 "feedback.php"

Gir brukeren en mulighet for å komme med tilbakemeldinger.

```
<?php
include_once("_dblib_.php");
include_once("_contentlib_.php");
include_once("_designlib_.php");
include_once("_htmllib_.php");
include_once("_funklib_.php");

//Get remote IP
$ip=getenv("REMOTE_ADDR");

//Get session id if it exists
$ssid=$_GET["ssid"];

//valider
$comment=validate($_POST["com"]);
$ssid=validate($ssid);
$action=validate($_POST["action"]);

//SQL stuff
connect();
selectdb("openchannel");

$accepted = false;
$accepted = validate_user($ssid, $ip);

//If any results returned...
if ($accepted) {

    designtop();
```

```
    if ($action=="send"){
        dblog(0, $ip,$comment,$sessid);
        echo "<font color=\"ff0000\"><h3>Vi takker for din
tilbakemelding!</h3></font>";

    } else {

        echo "    <form action=\"feedback.php?sessid=$sessid\"
method=\"post\">\n";
        echo "    <table>\n";
        echo "    <tr>\n";
        echo "    <td><br><br><h3>Gi oss
tilbakemeldinger!</h3><br></td>\n";
        echo "    </tr>\n";
        echo "    <tr><td>Tilbakemelding:</td></tr>\n";
        echo "    <tr><td> . itext("com", "60") . "</td></tr>\n";
        echo "\n";
        echo "    <tr><td>&nbsp;</td></tr>\n";
        echo "\n";
        echo "    <tr>\n";
        echo "    <td align=\"right\">" . isubmit("Send melding") .
"<br><br></td>\n";
        echo "    </tr>\n";
        echo "    </table>\n";
        echo ihidden("action", "send");
        echo "    </form>\n";

    }

    designmiddle();
    contentmenu();
    designbottom();

} else {
    header("Location:
http://jail.babylon.uninett.no/registrering/logon.php");
}

disconnect();

?>
```

### 20.3.7 "log.php"

Et script som viser innholdet av loggen.

```
<?php
include_once("_dblib_.php");

//Get remote IP
$ip=getenv("REMOTE_ADDR");

//Get session id if it exists
$sessid=$_GET["sessid"];

//SQL stuff
connect();
selectdb("openchannel");

$accepted = false;
$accepted = validate_user($sessid, $ip);
```

```
//If any results returned...
if ($accepted) {
    $qry="select * from log";
    $result=mysql_query($qry);

    if ($result) {
        $row=mysql_fetch_assoc($result);
        echo "<table border=\"1\" cellspacing=\"0\" cellpadding=\"2\">\n";
        echo " <tr>\n";
        echo " <td>log id</td>\n";
        echo " <td>user id</td>\n";
        echo " <td>ip address</td>\n";
        echo " <td>timestamp</td>\n";
        echo " <td>comment</td>\n";
        echo " <td>additional info</td>\n";
        echo " </tr>\n";

        while ($row){
            echo " <tr>\n";
            echo " <td>" . $row["logid"] . "&nbsp;</td>\n";
            echo " <td>" . $row["userid"] . "&nbsp;</td>\n";
            echo " <td>" . $row["ip"] . "&nbsp;</td>\n";
            echo " <td>" . $row["dt"] . "&nbsp;</td>\n";
            echo " <td>" . $row["comment"] . "&nbsp;</td>\n";
            echo " <td>" . $row["additional"] . "&nbsp;</td>\n";
            echo " </tr>\n";
            $row=mysql_fetch_assoc($result);
        }

        echo "</table>\n";
    }
    else {
        header("Location:
http://jail.babylon.uninett.no/registrering/logon.php");
    }

    disconnect();
}
?>
```

## 20.3.8 "logout.php"

Logger en bruker ut av systemet.

```
<?php
//Load needed libraries for this script
include_once ("_dblib_.php");

//Set default access to not allowed
$accepted=false;

//Get remote IP
$ip=getenv("REMOTE_ADDR");

$ssid=$_GET["ssid"];

connect();
selectdb("openchannel");

mysql_query("update users set lastip=\"0\" where ssid=\"".$ssid."");
```

```

disconnect();

header("Location: http://jail.babylon.uninett.no/registrering");

?>

```

### 20.3.9 "published.php"

Viser en liste over publiserte innslag offentlig.

```

<?php
include_once("_designlib_.php");
include_once("_contentlib_.php");
include_once("_browselib_.php");
include_once("_dblib_.php");
include_once("_funklib_.php");

connect();
selectdb("openchannel");

    designtop();

connect();
selectdb("openchannel");

$content=getcontent(0, "published");
$row=mysql_fetch_assoc($content);

if (!$row) echo "<br><b>Ingen filer funnet!</b><br>";

echo "<table width=\"400\" cellspacing=\"0\" cellpadding=\"0\"
border=\"0\">\n";

    while ($row){
        //Show title where exists, show filename if not
        $disptitle = $row["title"];
        if (!$disptitle) $disptitle=remext($row["filename"]);

        $fname=ruce($row["filename"]);
        $url="e-rtsp://129.241.26.183/Open_Channel/ORGANISASJONER/" .
ruce($oname) . "/" . $fname;
        echo "<tr><td class=\"tablehead\" colspan=\"4\"><b>";

            if ($row["fexists"]=="no") $disptitle="&nbsp;<font
color=\"#ff0000\">X</font> " . $disptitle;
            $disptitle=$disptitle . " (" . $row["number_played"] . ")";
            $catid=getdbitem("name", "category", "categoryid",
$row["categoryid"]);

            $fin_comp=getdbitem("fin_compress", "content", "contentid",
$row["contentid"]);
            if ($fin_comp=="yes"){
                echo "<a class=\"tablehead\" href=\"show.php?sessid=$sessid&cid=" .
$row["contentid"] . "\" target=\"_blank\"> " . $disptitle . "</a>";
            } else {
                echo "$disptitle";
            }
            echo "&nbsp;</b> ";
            echo "</td><td class=\"tablehead\" align=\"right\" colspan=\"2\">";
            echo "&nbsp;</td></tr>";
            echo "<tr><td colspan=\"5\" bgcolor=\"#ccccff\">";
            $descript=$row["description"];

```



```
    if (!$descript){ echo "Ingen beskrivelse for denne filen."; }
    else echo $descript;
    echo "</td><td bgcolor=\`#ccccff\`" align=\`right\`"> " .
        "&nbsp;</td></tr>";
    $row=mysql_fetch_assoc($content);
}

echo "</table>\n";

echo "<br>Innslagene vil bli tilgjengelig for \`Video On Demand\`
dersom du har installert en <a
href=\`http://jail.babylon.uninett.no:8080/akanal/Arkiv/FileLibrary5/Files/
EnvivioTV-1.5.exe\`">h.264 videocodec</a>.";
disconnect();

designmiddle();
contentmenu();
designbottom();

?>
```

### 20.3.10 "show.php"

Øker en teller i databasen for hvor mange ganger et innslag er åpnet. Redirigerer også nettleseren slik at innslaget som er bestilt åpnes i standard avspillingsprogram.

```
<?php
include_once("_dblib_.php");
include_once("_funklib_.php");
include_once("_browselib_.php");

//Get remote IP
$ip=getenv("REMOTE_ADDR");

//Hent brukerparametre og valider
$ssid=validate($_GET["ssid"]);
$cid=validate($_GET["cid"]);

//SQL stuff
connect();
selectdb("openchannel");

$accepted = false;
$accepted = validate_user($ssid, $ip);

//If any results returned...
if ($accepted) {
    $fname=getdbitem("filename", "content", "contentid", $cid);
    $oid=getdbitem("orgid", "content", "contentid", $cid);
    $oname=getdbitem("orgnavn", "organisasjoner", "orgid", $oid);
    $fexists=ftp_fileexists("/ORGANISASJONER/" . $oname . "/" .
remext($fname) . ".mp4");

    if ($fexists){
        mysql_query("update content set number_played=number_played+1 where
contentid=\`$cid\`");
        mysql_query("update content set fexists=\`yes\` where
contentid=\`$cid\`");
        $dest="e-rtsp://129.241.26.183/Open_Channel/ORGANISASJONER/" .
ruce($oname) . "/" . remext($fname) . ".mp4";
        echo "<META HTTP-EQUIV=\`Refresh\` Content=\`0; URL=$dest\`>";
    }
}
```

```
//      header("Location: " . $dest);
      exit;

      } else {
        mysql_query("update content set fexists=\"no\" where
contentid=\"$cid\"");
        echo "Beklager men vi kunne ikke finne filen p&aring;
filomr&aring;det!";
      }
    }

    disconnect();

?>
```

## 20.4 Biblioteker

### 20.4.1 “\_designlib\_.php”

Åpen Kanal sitt design.

```
<?php

// __DESIGNLIB__
//
// Funksjoner for layout av Open Channel NO
// Design: Kjetil Bendiksen, www.epicdesign.net

// Funksjonene designtop(), designmiddle() og designbottom() kalles med
// innhold mellom top og middle og meny mellom middle og bottom.

//-----
function designtop(){ //Kalles foer innhold (aller foerst)
    echo "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN\" .
        "\"http://www.w3.org/TR/html4/strict.dtd\">\n";
    echo "<html>\n";

    echo "<head>\n";
    echo " <title>Registrering - &Aring;pen Kanal</title>\n";
    echo " <META http-equiv=\"Content-Type\" content=\"text/html;
charset=ISO-8859-1\"/>\n";
    echo " <META name=\"description\" content=\"Web pages for Open Channel
Norway, a public TV concept\"/>\n";
    echo " <META name=\"keywords\" content=\"public access tv, open
channel\"/>\n";
    echo " <link rel=\"stylesheet\" href=\"openkanal.css\"
type=\"text/css\">\n";
    echo " <link rel=\"stylesheet\" href=\"design.css\"
type=\"text/css\">\n";
    echo "</head>\n";
    echo "<body bgcolor=\"#e1e0e0\" text=\"#000000\" leftmargin=\"0\"
topmargin=\"0\">\n";
    echo "marginwidth=\"0\" marginheight=\"0\">\n";
    echo "\n";
    echo "<table width=\"751\" border=\"0\" cellspacing=\"0\"
cellpadding=\"0\" height=\"100%\" align=\"center\">\n";
    echo " <tr>\n";
    echo " <td colspan=\"3\" height=\"88\"><img src=\"graphics/top.gif\"
width=\"775\">\n";
    echo "height=\"101\"></td>\n";
    echo " </tr>\n";
    echo " <tr>\n";
    echo " <td width=\"163\" align=\"left\" valign=\"top\" height=\"5\"
.
    " bgcolor=\"#F1F0F0\"><img src=\"graphics/left.gif\" width=\"163\"
height=\"7\"></td>\n";
    echo " <td rowspan=\"2\" height=\"5\" bgcolor=\"#FFFFFF\"
valign=\"top\" align=\"right\">\n";
    echo " <table width=\"396\" border=\"0\" cellspacing=\"0\"
cellpadding=\"0\">
    " align=\"center\" height=\"81\">\n";
    echo " <tr>\n";
    echo " <td height=\"99\">&nbsp;&nbsp;&nbsp;</td>\n";
    echo " <td height=\"99\"><span class=\"main\"><br>\n";
}
```

```
//-----
function designmiddle(){ //Kalles etter innhold, men foer meny
    echo "        </span></td>\n";
    echo "        </tr>\n";
    echo "        </table>\n";
    echo "        </td>\n";
    echo "        <td width=\"158\" valign=\"top\" align=\"right\"
height=\"5\"\" .
        \" bgcolor=\"F1F0F0\"><img src=\"graphics/right.gif\" width=\"157\"
height=\"7\"></td>\n";
    echo "        </tr>\n";
    echo "        <tr>\n";
    echo "        <td width=\"163\" bgcolor=\"F1F0F0\" align=\"right\"
height=\"254\"\" .
        \" valign=\"top\">\n";
    echo "        <table width=\"143\" border=\"0\" cellspacing=\"0\"
cellpadding=\"0\"\" .
        \" valign=\"top\">\n";
    }

//-----
function designbottom(){ //Kalles etter meny (helt til sist)
    echo "        </table>\n";
    echo "        </td>\n";
    echo "        <td width=\"158\" bgcolor=\"F1F0F0\"
height=\"254\">&nbsp;</td>\n";
    echo "        </tr>\n";
    echo "        <tr>\n";
    echo "        <td width=\"163\" bgcolor=\"F1F0F0\">&nbsp;</td>\n";
    echo "        <td width=\"455\" valign=\"top\" bgcolor=\"F1F0F0\"><img\" .
        \" src=\"graphics/bottom.gif\" width=\"455\" height=\"51\"
border=\"0\"></td>\n";
    echo "        <td width=\"158\" bgcolor=\"F1F0F0\">&nbsp;</td>\n";
    echo "        </tr>\n";
    echo "        <tr>\n";
    echo "        <td width=\"163\" bgcolor=\"F1F0F0\"
height=\"20\">&nbsp;</td>\n";
    echo "        <td width=\"455\" bgcolor=\"F1F0F0\"
height=\"20\">&nbsp;</td>\n";
    echo "        <td width=\"158\" bgcolor=\"F1F0F0\" height=\"20\"
valign=\"bottom\">\n";
    echo "        <div align=\"right\"><img src=\"graphics/bottomlink.gif\"
width=\"123\"\" .
        \" height=\"17\" usemap=\"#Map\" border=\"0\">\n";
    echo "        <map name=\"Map\">\n";
    echo "        <area shape=\"rect\" coords=\"42,2,120,14\"\" .
        \" href=\"http://www.epicdesign.net/kjetil\" target=\"_blank\"
alt=\"&copy; 2002\"\" .
        \" title=\"&copy; 2002\">\n";
    echo "        </map>\n";
    echo "        </div>\n";
    echo "        </td>\n";
    echo "        </tr>\n";
    echo "</table>\n";
    echo "\n";
    echo "</body>\n";
    echo "</html>\n";
}
?>
```

## 20.4.2 “\_contentlib\_.php”

Innhold til selve nettstedet. Dette virket opprinnelig som en god idé, men etter hvert som fila vokser vil php prosessoren måtte gjøre mye unødvendig jobb. Planen for denne fila fremover vil være å splitte den opp i flere mindre filer eller til og med inkludere det i selve koden hver enkelt del hører til. Tanken var opprinnelig å skille innhold fra kode slik at innholdet ikke skulle overskygge den overordnede gangen i et script, men det viser seg at det er lettere å holde oversikt når all vesentlig kode befinner seg i samme script. Det at redigering av innhold ble lettere med en slik fil er underordnet den tapte oversikten man taper med en slik løsning.

```
<?php
// __CONTENTLIB__
//
// Funksjoner for selve innholdet av registreringssidene til Open Channel
NO

function contentmenu(){ //Meny
    include_once("_dblib_.php");
    connect();
    selectdb("openchannel");

    //TODO: Lage ordentlig funksjonalitet... :)
    echo "        <tr>\n";
    echo "            <td width=\"10\">&nbsp;</td>\n";
    echo "            <td class=\"openkanal\" width=\"133\"><a\" .
    " href=\"http://www.openchannel.no\"><font\" .
    " color=\"28657F\">&Aring;pen kanals hovedside</font></a></td>\n";
    echo "        </tr>\n";

//    global $sessid;
    $sessid=$_GET["sessid"];
    if (isset($sessid)){

        echo "            <tr>\n";
        echo "                <td width=\"10\">&nbsp;</td>\n";
        echo "                <td class=\"openkanal\" width=\"133\">&nbsp;</td>\n";
        echo "            </tr>\n";

        echo "            <tr>\n";
        echo "                <td width=\"10\">&nbsp;</td>\n";
        echo "                <td class=\"openkanal\" width=\"133\"><a\" .
        " href=\"registrer.php?sessid=$sessid\"><font\" .
        " color=\"28657F\">Registrering</font></a></td>\n";
        echo "            </tr>\n";

        echo "            <tr>\n";
        echo "                <td width=\"10\">&nbsp;</td>\n";
        echo "                <td class=\"openkanal\" width=\"133\"><a\" .
        " href=\"browse.php?sessid=$sessid&view=compressed\"><font\" .
        " color=\"28657F\">Browser</font></a></td>\n";
        echo "            </tr>\n";

        echo "            <tr>\n";
        echo "                <td width=\"10\">&nbsp;</td>\n";
        echo "                <td class=\"openkanal\" width=\"133\"><a\" .
        " href=\"medieforv_skjema.php?sessid=$sessid\"
target=\"_blank\"><font\" .
        " color=\"28657f\">Skjema</font></a></td>\n";
        echo "            </tr>\n";

        echo "            <tr>\n";
        echo "                <td width=\"10\">&nbsp;</td>\n";
        echo "                <td class=\"openkanal\" width=\"133\">&nbsp;</td>\n";
```

```

echo "          </tr>\n";

if (access_granted($sessid, "all rights")){

    echo "          <tr>\n";
    echo "              <td width=\"10\">&nbsp;</td>\n";
    echo "              <td class=\"openkanal\" width=\"133\"><a\" .
    " href=\"administer.php?sessid=$sessid\"><font\" .
    " color=\"28657f\">Administrasjon</font></a></td>\n";
    echo "          </tr>\n";

    echo "          <tr>\n";
    echo "              <td width=\"10\">&nbsp;</td>\n";
    echo "              <td class=\"openkanal\" width=\"133\"><a\" .
    " href=\"log.php?sessid=$sessid\"><font\" .
    " color=\"28657f\">Vis logg</font></a></td>\n";
    echo "          </tr>\n";

    echo "          <tr>\n";
    echo "              <td width=\"10\">&nbsp;</td>\n";
    echo "              <td class=\"openkanal\" width=\"133\"><a\" .
    " href=\"fcheck.php?sessid=$sessid\" target=\"_blank\"><font\" .
    " color=\"28657f\">Filsjekk</font></a></td>\n";
    echo "          </tr>\n";

}

echo "          <tr>\n";
echo "              <td width=\"10\">&nbsp;</td>\n";
echo "              <td class=\"openkanal\" width=\"133\"><a\" .
    " href=\"changeowninfo.php?sessid=$sessid\"><font\" .
    " color=\"28657f\">Endre personlig info</font></a></td>\n";
echo "          </tr>\n";

echo "          <tr>\n";
echo "              <td width=\"10\">&nbsp;</td>\n";
echo "              <td class=\"openkanal\" width=\"133\">&nbsp;</td>\n";
echo "          </tr>\n";

echo "          <tr>\n";
echo "              <td width=\"10\">&nbsp;</td>\n";
echo "              <td class=\"openkanal\" width=\"133\"><a\" .
    " href=\"feedback.php?sessid=$sessid\"><font\" .
    " color=\"28657f\">Tilbakemeldinger</font></a></td>\n";
echo "          </tr>\n";

echo "          <tr>\n";
echo "              <td width=\"10\">&nbsp;</td>\n";
echo "              <td class=\"openkanal\" width=\"133\">&nbsp;</td>\n";
echo "          </tr>\n";

echo "          <tr>\n";
echo "              <td width=\"10\">&nbsp;</td>\n";
echo "              <td class=\"openkanal\" width=\"133\"><a\" .
    " href=\"logout.php?sessid=$sessid\"><font\" .
    " color=\"28657f\">Logg ut</font></a></td>\n";
echo "          </tr>\n";

}
}

function contentpage1(){ //Registrering av redaktoer
    include_once("_htmlllib_.php");
    global $sessid;

    echo "<h3>Registreringsskjema (side 1):</h3>\n";
    echo "\n";
    echo "    <form action=\"adred.php?sessid=$sessid\"
method=\"post\">\n";

```

```

echo "      <table border=\"0\" cellspacing=\"2\" cellpadding=\"3\">\n";
echo "      <tr>\n";
echo "      <td colspan=\"2\"><h4>Redakt&oslash;r (deg!)</h4></td>\n";
echo "      </tr>\n";
echo "      <tr>\n";
echo "      <td>Fullt navn:</td>\n";
echo "      <td>\" . itext(\"redfullnavn\", \"40\") . \"*\n";
echo "      </tr>\n";
/*      echo "      <tr>\n";
echo "      <td>Personnummer:</td>\n";
echo "      <td>\" . itext(\"redpernum\", \"40\") . \"*\n";
echo "      </tr>\n";
*/      echo "      <tr>\n";
echo "      <td>Gateadr. eller postboks:</td>\n";
echo "      <td>\" . itext(\"redadr1\", \"40\") . \"</td>\n";
echo "      </tr>\n";
echo "      <tr>\n";
echo "      <td>Postnummer og sted:</td>\n";
echo "      <td>\" . itext(\"redadr2\", \"40\") . \"</td>\n";
echo "      </tr>\n";
echo "      <tr>\n";
echo "      <td>Telefon jobb:</td>\n";
echo "      <td>\" . itext(\"redtlfjobb\", \"40\") . \"*\n";
echo "      </tr>\n";
echo "      <tr>\n";
echo "      <td>Telefaks jobb:</td>\n";
echo "      <td>\" . itext(\"redtlffax\", \"40\") . \"</td>\n";
echo "      </tr>\n";
echo "      <tr>\n";
echo "      <td>Telefon hjem:</td>\n";
echo "      <td>\" . itext(\"redtlfhjem\", \"40\") . \"</td>\n";
echo "      </tr>\n";
echo "      <tr>\n";
echo "      <td>Telefon mobil:</td>\n";
echo "      <td>\" . itext(\"redtlfmob\", \"40\") . \"</td>\n";
echo "      </tr>\n";
echo "      <tr>\n";
echo "      <td>Elektronisk postadresse:</td>\n";
echo "      <td>\" . itext(\"redepst\", \"40\") . \"*\n";
echo "      </tr>\n";
echo "      <tr><td colspan=\"2\" class=\"commentright\">Felter merket
med * er obligatoriske.</td></tr>\n";
//      echo "\n";
//      echo "      <tr><td>&nbsp;</td><td>&nbsp;</td></tr>\n";
//      echo "\n";
echo "      <tr>\n";
echo "      <td>&nbsp;</td>\n";
echo "      <td class=\"right\">\" . isubmit(\"Send skjema\") . \"</td>\n";
echo "      </tr>\n";
echo "      </table>\n";
echo "      </form>\n";
}

function contentpage2(){ //Registrering av organisasjon
include_once(\"_htmllib_.php\");
global $sessid;

echo "<h3>Registreringsskjema (side 2):</h3>\n";
echo "\n";
echo "      <form action=\"addorg.php?sessid=$sessid\"
method=\"post\">\n";
echo "      <table border=\"0\" cellspacing=\"2\" cellpadding=\"3\">\n";
echo "      <tr>\n";
echo "      <td colspan=\"2\"><h2>Organisasjon</td>\n";
echo "      </tr>\n";
echo "      <tr>\n";
echo "      <td>Navn p&aring; organisasjon:</td>\n";
echo "      <td>\" . itext(\"orgnavn\", \"40\") . \"</td>\n";
echo "      </tr>\n";

```





```
echo "      </td>\n";
echo "\n";
echo "\n";
echo "      <tr><td>&nbsp;</td><td>&nbsp;</td></tr>\n";
echo "\n";
echo "      <tr>\n";
echo "      <td>&nbsp;</td>\n";
echo "      <td align=\"right\">\" . isubmit("Send skjema", "addorg") .
"</td>\n";
echo "      </tr>\n";
echo "    </table>\n";
echo "  </form>\n";

}

?>
```

### 20.4.3 “\_dblib\_.php”

Databasefunksjoner. Inneholder funksjoner for å koble seg opp mot databasen samt funksjoner for å lette spørringer. Det er gjort en del jobb med å generalisere funksjonalitet, men det gjenstår fortsatt litt jobb for å få dette biblioteket ”perfekt”.

```
<?php
    $connected = false;
    $link;
    $accepted = false;

    function connect(){
        global $connected, $link;
        if (!$connected){
            //Connect to db
            $link = mysql_connect("localhost", "defaultuser", "")
                or die("Could not connect : " . mysql_error());
            $connected = true;
        }
    }

    function selectdb($database){
        global $connected, $link;

        if ($connected){
            //Select db to use
            mysql_select_db($database,$link)
                or die("Could not select db : " . mysql_error());
        }
    }

    function validate_user($session, $ipaddress){
        global $accepted, $connected;
        $accepted = false;
        if ($connected){
            //Get username and ip address from this session and ip
            $result = mysql_query("select username, lastip from users
where sessid=\"\$session\" and lastip=\"\$ipaddress\"")
                or die("Could not get user info : " .
mysql_error());

            //Extract rows
            $row = mysql_fetch_assoc($result);
```

```
        if ($row) {
            $accepted = true;
            return true;
        }
    }
    return false;
}

function disconnect(){
    global $link;
    if ($connected){
        mysql_close($link);
        $connected = false;
    }
}

function getorgid($sid=""){
    global $connected;
    $the_org=-1;

    if ($connected){
        $result=mysql_query("select orgid from users where
sessid=\"\$sid\"");
        if (!$result){
            echo "Feil: kunne ikke finne organisasjon for din
sesjonsid<br>";
        } else {
            $row=mysql_fetch_assoc($result);
            $the_org=$row["orgid"];
        }
    }
    return $the_org;
}

function getorgname($oid=""){
    global $connected;
    $the_org_name="";

    if ($connected){
        $result=mysql_query("select ornavn from organisasjoner where
orgid=\"\$oid\"");
        if (!$result){
            echo "ERROR: could not select ornavn";
        } else {
            $row=mysql_fetch_assoc($result);
            $the_org_name=$row["orgnavn"];
        }
    }
    return $the_org_name;
}

function getcontent($oid = -1, $view="compressed"){
    global $connected, $sessid;
    $result=0;

    if (access_granted($sessid, "all rights")) $admin=true; else
$admin=false;

    if ($connected){
        $qry="";
        if ($view=="new") $qry="select * from content where
orgid=\"\$oid\" and fin_compress=\"no\" order by filename";
        else if ($view=="compressed") $qry="select * from content where
orgid=\"\$oid\" and fin_compress=\"yes\" order by title";
        else if ($view=="uploaded") $qry="select * from content where
orgid=\"\$oid\" and fin_upload=\"yes\"";
        else if ($view=="uploading") $qry="select * from content where
orgid=\"\$oid\" and fin_upload=\"no\"";
    }
}
```

```
        else if ($view=="compressing") $qry="select * from content
where orgid=\"\$oid\" and fin_upload=\"yes\" and fin_compress=\"no\"";
        else if ($view=="published") $qry="select * from content where
fin_compress=\"yes\" and vfri=\"yes\"";
        else if (($view=="all") and $admin) $qry="select * from content
order by fin_compress, title, filename";
        else $qry="select * from content where orgid=\"\$oid\"";
        $result=mysql_query($qry);
    }
    return $result;
}

function getdbitem($itemname = "", $tablename = "", $dependence = "",
$value = -1){
    global $connected;
    $item = "";

    if ($connected){
        $result = mysql_query("select $itemname from $tablename where
$dependence=\"\$value\"");
        if ($result){
            $row = mysql_fetch_assoc($result);
            $item = $row[$itemname];
        }
    }
    return $item;
}

function dblog($uid, $ip, $comment, $additional = ""){
    global $connected;
    if ($connected){
        mysql_query("insert into log (userid, ip, comment, additional) values
(\"\$uid\", \"\$ip\", \"\$comment\", \"\$additional\")");
    }
}

function access_granted($sessid, $key){
    global $connected;

    if ($connected){
        $gid=getdbitem("privid", "users", "sessid", $sessid);
        $keyvalue=getdbitem("keyvalue", "priv", "privid", $gid);
        if ($keyvalue=="yes") return true;
    }

    return false;
}

function usrexist($nick){
    global $connected;

    if ($connected){
        $result=mysql_query("select * from users where username=\"\$nick\"");
        $row=mysql_fetch_assoc($result);
        if ($row) return true;
    } else return true;

    return false;
}
?>
```

#### 20.4.4 “\_htmlib\_.php”

Bibliotek for å gjøre enkelte html operasjoner enklere og programmeringsmessig kjappere. Resultatet er mindre kode å skrive samt lettere lesbar kode.

```
<?php
function itext($name, $size="40", $value=""){
    $txtstr="<input type=\"text\" name=\"$name\" size=\"$size\"";
    if ($value!=""){ $txtstr=$txtstr . " value=\"$value\""; }
    $txtstr=$txtstr . ">";
    return $txtstr;
}

function ipwd($name, $size="40"){
    $txtstr="<input type=\"password\" name=\"$name\" size=\"$size\">";
    return $txtstr;
}

function isubmit($value, $name=""){
    $txtstr="<input type=\"submit\" value=\"$value\"";
    if ($name!=""){ $txtstr=$txtstr . " name=\"$name\""; }
    $txtstr=$txtstr . ">";
    return $txtstr;
}

function iradio($name, $value, $checked){
    $txtstr="<input type=\"radio\" name=\"$name\" value=\"$value\"";
    if ($checked) $txtstr.=" checked";
    $txtstr.=">";

    return $txtstr;
}

function ick($name, $value){
    $txtstr="<input type=\"checkbox\" name=\"$name\" value=\"$value\">";
    return $txtstr;
}

function ihidden($name, $value){
    $txtstr="<input type=\"hidden\" name=\"$name\" value=\"$value\">";
    return $txtstr;
}
?>
```

#### 20.4.5 “\_browselib\_.php”

Opprinnelig funksjoner for filvisningssystemet, men har blitt til et bibliotek for filfunksjoner.

```
<?php
include_once("_dblib_.php");
include_once("_funklib_.php");

function browseview(){
//    echo "Innhold:<br>\n";

    $sessid=$_GET["sessid"];
    $view=$_GET["view"];

    connect();
```

```
selectdb("openchannel");

$oid = getorgid($sessid);
$name = getorgname($oid);
echo "<font size=\"4\">Filer for organisasjon: " . $name .
"</font><br>";

echo "[<a href=\"browse.php?sessid=$sessid&view=uploading\"> " .
"Under opplasting</a>] [<a
href=\"browse.php?sessid=$sessid&view=compressing\">Under
komprimering</a>]" .
" [<a href=\"browse.php?sessid=$sessid&view=compressed\">Ferdig
komprimert</a>] " .
" [<a
href=\"browse.php?sessid=$sessid&view=published\">Publisert</a>]";
if (access_granted($sessid, "all rights")){
echo " [<a href=\"browse.php?sessid=$sessid&view=all\">Vis alle
filer</a>]";
}
echo "<br>\n";

$content=getcontent($oid, $view);
$row=mysql_fetch_assoc($content);

if (!$row) echo "<br><b>Ingen filer funnet!</b><br>Filene skal lastes
opp ".
"til FTP-serveren med adressen 129.241.26.183. (FTP-klient er
tilgjengelig <a
href=\"http://jail.babylon.uninett.no:8080/akanal/Arkiv/FileLibrary5/Files/
FTP-program.exe\">her</a>. Passord er tildelelt av ".
"<a href=\"mailto:adm@openchannel.no\">daglig leder</a>)." .
"Etter registrering ble det opprettet en katalog med navnet &quot;
. $name . &quot; som du finner under \"ORGANISASJONER\". Bruk denne
katalogen!<br>";

echo "<table width=\"400\" cellspacing=\"0\" cellpadding=\"0\"
border=\"0\">\n";

while ($row){
//Show title where exists, show filename if not
$disptitle = $row["title"];
if (!$disptitle) $disptitle=remext($row["filename"]);

// echo "<tr>\n";
$fname=ruce($row["filename"]);
$url="e-rtsp://129.241.26.183/Open_Channel/ORGANISASJONER/" .
ruce($name) . "/" . $fname;
echo "<tr><td class=\"tablehead\" colspan=\"4\"><b>";

if ($row["fexists"]=="no") $disptitle="&nbsp;<font
color=\"#ff0000\">X</font> " . $disptitle;
$disptitle=$disptitle . " (" . $row["number_played"] . ")";

$fin_comp=getdbitem("fin_compress", "content", "contentid",
$row["contentid"]);
if ($fin_comp=="yes"){
echo "<a class=\"tablehead\" href=\"show.php?sessid=$sessid&cid=" .
$row["contentid"] . "\" target=\"_blank\"> " . $disptitle . "</a>";
} else {
echo "$disptitle";
}
echo "&nbsp;</b> ";
// echo "Uploaded: "; if ($row["fin_upload"]=="yes") echo "Y, "; else
echo "N, ";
// echo "Compressed: "; if ($row["fin_compress"]=="yes") echo "Y"; else
echo "N";
echo "</td><td class=\"tablehead\" align=\"right\" colspan=\"2\">";
if ($oid==$row["orgid"] || (access_granted($sessid, "all rights"))){
```

```

        echo "[<a class=\"tablehead\"
href=\"editcontent.php?sessid=$sessid&fileid=" . $row["contentid"] .
"\>rediger</a>]";
    }
    echo "&nbsp;</td></tr>";
    echo "<tr><td colspan=\"5\" bgcolor=\"#ccccff\">";
    $descript=$row["description"];
    if (!$descript){ echo "Ingen beskrivelse for denne fila, trykk
&aring; rediger for &aring; andre."; }
    else echo $descript;
    echo "</td><td bgcolor=\"#ccccff\" align=\"right\"><span
class=\"right\">[<a class=\"tablecontent\"
href=\"viewfileinfo.php?sessid=$sessid&fileid=" .
    $row["contentid"] . "\>Mer info</a>]</span></td></tr>";
    $row=mysql_fetch_assoc($content);
}

echo "</table>\n";

//    echo "<br><a href=\"updatefile.php?sessid=$sessid\">S&oslash;k etter
nye filer.</a><br>";

    echo "<br>Innslagene vil bli tilgjengelig for \"Video On Demand\"
dersom du har installert en <a
href=\"http://jail.babylon.uninett.no:8080/akanal/Arkiv/FileLibrary5/Files/
EnvivioTV-1.5.exe\">h.264 videocodec</a>.";
    disconnect();
}

function ftp_addorg($orgnavn){
    $ftpstream=ftp_connect("129.241.26.183");
    ftp_login($ftpstream, "Open_Channel", "publicaccess!");
    ftp_pasv($ftpstream, TRUE);
    ftp_mkdir($ftpstream, "/ORGANISASJONER/$orgnavn");
    ftp_close($ftpstream);
}

function ftp_fileexists($filename){
    $fexists=false;
    $ftpstream=ftp_connect("129.241.26.183");
    ftp_login($ftpstream, "Open_Channel", "publicaccess!");
    ftp_pasv($ftpstream, TRUE);
    if (ftp_size($ftpstream, $filename)!=-1) $fexists=true;
    ftp_close($ftpstream);
    return $fexists;
}

?>

```

## 20.4.6 "\_funklib\_.php"

Forskjellige småfunksjoner som ikke helt passer inn i noen større kategori.

```

<?php
    //Create a session identifier
    function getrandsessid(){
        $sessid="";
        for ($lp=0; $lp<32; $lp++) {
            $thingy=rand(0,2);
            if ($thingy==0){

```

```
        $sessid=$sessid . chr(rand(0,25)+ord("a"));
    } else if ($thingy==1){
        $sessid=$sessid . chr(rand(0,25)+ord("A"));
    } else {
        $sessid=$sessid . chr(rand(0,9)+ord("0"));
    }
}
return $sessid;
}

function validate($parameter){
    $validated=$parameter;

    if (!get_magic_quotes_gpc()){
        $validated=addslashes($validated);
    }

    $validated=strip_tags($validated);

    return $validated;
}

function ruce($txt){
    $tmpstr="";
    for ($lp=0; $lp<strlen($txt); $lp++){
        if ($txt{$lp}!=" ") $tmpstr.=$txt{$lp}; else $tmpstr.="2520";
    }
    return ($tmpstr);
}

function isalphanum($txt){
    for ($lp=0; $lp<strlen($txt); $lp++){
        if (!(($txt{$lp}>="a") && ($txt{$lp}<="z")))
            if (!(($txt{$lp}>="A") && ($txt{$lp}<="Z")))
                if (!(($txt{$lp}>="0") && ($txt{$lp}<="9"))){
                    $ok_list="-_!";

                    $match=false;
                    for ($lp2=0; $lp2<strlen($ok_list); $lp2++){
                        if ($txt{$lp}==$ok_list{$lp2}) $match=true;
                    }

                    if (!$match) return false;
                }
    };
}

return true;
}

function goodpass($pwd){
    if ($pwd=="") return false;
    return true;
}

function remext($fname){
    $tmpstr="";
    $dot=false;
    for ($lp=strlen($fname); $lp>=0; $lp--){
        if ($dot) $tmpstr=$fname{$lp} . $tmpstr;
        if ($fname{$lp}==".") $dot=true;
    }
    return $tmpstr;
}
?>
```





```
    if (!$accepted) {
        header("Location:
http://jail.babylon.uninett.no/registrering/registrer.php?sessid=$sessid");
        exit;
    }
?>
```

```
<STYLE TYPE="text/css">
<!--
TD { vertical-align: top; text-align: left; }
-->
</STYLE>
```

```
<html>
```

```
<head>
<title>Statens Medieforvaltning</title>
</head>
```

```
<body>
<table width="700">
<tr><td>
```

```
<h1 align="center">Statens medieforvaltning</h1>
```

```
<table width="700">
<tr>
<td>
<td>
<i>Skjemaet sendes ferdig utfylt til:</i><br>
<b>Statens medieforvaltning<br>
Postboks 444<br>
1601 Fredrikstad</b>
</td>
<td></td>
<td><table border="1" cellpadding="2" cellspacing="0">
<tr>
<td>
<td>
Sak nr. _____ dok. nr. _____<br>
Dato: _____<br>
Arkivkode: _____<br>
Avd./Saksbeh.: _____<br>
</td>
</tr>
</table></td>
</tr>
</table>
```

```
<h2 align="center">REGISTRERINGSPLIKTLIG KRINGKASTING</h2>
```

```
<p align="center">(Jf lov om kringkasting &#167;&#167; 2-1 tredje ledd og 2-2)</p>
```

```
<table width="700">
<tr>
<td>Reg. gjelder:</td>
<td>Satellittsendt fjernsyn []<br>
Kabelsendt fjernsyn []</td>
<td>Satellittsendt radio []<br>
Kabelsendt radio []</td>
</tr>
</table>
```

```
<table width="700" cellpadding="2">
<tr>
<td>1.</td>
<td colspan="5">
```

```
Selskapets/serens navn: <b><?php echo $oname . " / " .
$fullname;?></b><br>
Adresse: <b><?php echo $adrl;?></b><br>
Postnr./-sted: <b><?php echo $adr2;?></b></td>
<td>
telefon: <b><?php echo $tlfj;?></b><br>
telefaks: <b><?php echo $tlff;?></b></td>
</tr>

<tr>
<td>2.</td>
<td colspan="6">Stasjonsnavn: <b>&aring;pen kanal</b></td>
</tr>

<tr>
<td>3.</td>
<td colspan="6">Organisasjons-/f&oslash;snummer: <b><?php echo
$onum . " / " . $pnum;?></b></td>
</tr>

<tr>
<td>4.</td>
<td colspan="6">Omr&aring;de det kringkastes i: <br>(Kommune,
borettslag e.l.)</td>
</tr>

<tr>
<td>5.</td>
<td colspan="6">Kabel-/satellittoperat&oslash;r: <br>(Teknisk
distribut&oslash;r)</td>
</tr>

<tr>
<td>6.</td>
<td colspan="6">Kringkasting starter: <b>N/A</b></td>
</tr>

<tr>
<td>7.</td>
<td colspan="5">Daglig leder: <b>N/A</b></td>
<td>telefon: <b>N/A</b></td>
</tr>

<tr>
<td>8.</td>
<td colspan="6">Ansvarlig redakt&oslash;r: <b><?php echo $fullname;
?></b></td>
</tr>

<tr>
<td>9.</td>
<td colspan="6">Styreleder: <b>N/A</b></td>
</tr>

<tr>
<td>10.</td>
<td colspan="4">Eiere/aksjon&aelig;rer:<br><b>N/A</b></td>
<td colspan="2">Prosentvis andel:<br><b>N/A</b></td>
</tr>
</table>
```

Endringer i opplysningene m&aring; straks meldes til Statens medieforvaltning p&aring; eget skjema, som kan f&aring;s ved henvendelse.<br>



Dette nøkkelordet sørger for å hente en variabel som er ute av "scope". Ved kall på funksjoner vil funksjonene ha sitt eget navnområde (namespace). Ved hjelp av global kan man referere til variabler som er utenfor funksjonens navnområde. (Eksempelvis det globale navneområdet.)

*mysql\_fetch\_assoc();*

Denne metoden tar et resultat etter en databasespørring og returnerer et array som man kan referere til etter databasetabellens feltnavn i stedet for å returnere et array som kun kan referere til et tall etter rekkefølgen feltene i tabellen ble laget. Dette er praktisk siden felter senere kan legges til, noe som kan endre rekkefølgen på felter i en tabell og dermed fremme et behov for å endre koden. Returnerer null dersom det ikke er noe (flere) rader.

*getenv("")*

Henter verdien til en miljøvariabel. For eksempel ip-adressen til en fjern bruker.

*\$\_POST[""]; \$\_GET[""];*

Henter data som er sendt til webserveren via henholdsvis post- eller get- metoden. Hovedsakelig om dataene kommer fra et skjema i selve html-dokumentet (post) eller om det kommer fra den kallende linken (get).

*mysql\_connect();*

Kobler til en sql-serveren og returnerer en ressurshåndterer.

*mysql\_select\_db();*

Velger hvilken database som skal brukes.

*mysql\_query();*

Utfører en spørring.

*mysql\_result();*

Henter resultatet fra en spørring.

*mysql\_fetch\_assoc();*

Returnerer et assosiativt array med resultatene. Eksempel på assosiative arrays versus numeriske indisier: `$row["userid"]` mot `$row[0]`. Hvilken måte å referere på er mest praktisk?

`mysql_free_result();`

Frigjør minnet som er reservert for resultatet av en spørring.

`mysql_close();`

Frigjør ressurser som er allokert for databasetilknytningen.

`mysql_error();`

Returnerer resultatet (feilkoden) av siste utførte mysql-kommando.

`foreach()`

Type for-løkke som itererer over et array. Et praktisk alternativ til vanlige for-løkker.

`header("");`

Sender headerinformasjon til web-browsersen.

`...or die("");`

Dersom en operasjon feiler (returnerer null) så vil scriptet "dø" med feilmeldingen i anførselstegn som begrunnelse mot brukeren.

`$<variabelnavn>`

Alle variabler i php skrives med \$ foran. Php bruker noe som heter "type-juggling" hvilket betyr at php ikke er spesielt kresen med hvilke variabeltyper som brukes mot hverandre. Det er med andre ord ikke så viktig å deklare variabler før bruk selv om det eksisterer mulighet for dette dersom det ved sjeldne tilfeller skulle være nødvendig å ha full kontroll.

`ftp_connect();`

Åpner en ftp-tilknytning. Returnerer en ressurshåndterer.

`ftp_login();`

Logger på en ftp-tilknytning.

*ftp\_pasv()*;

Setter passiv modus. For maskiner bak nat, firewalls, etc.

*ftp\_chdir()*;

Bytt til et annet filområde.

*ftp\_size()*;

Hent størrelsen på en fil.

*ftp\_nlist()*;

List alle filer i en katalog.

*ftp\_close()*;

Steng en ftp-tilknytning

*include\_once("")*;

Inkluderer en ekstern fil maks 1 gang.

*chr()*;

Returnerer ascii-tegnet for tallverdien som puttes inn i funksjonen.

*rand([min, max])*;

Returnerer et "tilfeldig" tall mellom *min* og *max*.

*ord()*;

Returnerer tallverdien for tegnet som puttes inn i funksjonen.

*\$string{n}*

Refererer til den n'te bokstaven i strengen \$string.

## 21 E - Appendiks: Komprimeringssystemet

Dette systemet er ikke synlig for brukeren. Alt skjer automatisk ved hjelp av script som genereres for hver fil som skal komprimeres. Hensikten med å komprimere en fil er for å gjøre den streambar slik at man kan se den på nett. I dette tilfelle lages det kun MPEG-4 filer. Tilsvarende script kan også brukes for å lage MPEG-2 filer til TV-sendng.

### 21.1 Generering av script

Scriptgenerering består av en samling med småprogrammer skrevet i Pascal. Programsnittene heter gscript, gsstart, gsstop, gswait og gsclean. Når en fil får status som ferdig opplastet i databasen vil gscript lage scriptfilene som trengs for å utføre komprimeringsjobben. "sgen.pas" er laget i forkant av diplomoppgaven og erstatter "gscript.pas". Begrunnelsen for dette finnes i kodeforklaringen lenger ned.

Scriptene som blir generert er en bat-fil, to txt-filer og en xml-fil. Bat-filen er den som skal kjøres, txt-filene er script for ftp-klienten for å hente ukomprimerte innslag og sende det ferdig komprimerte innslaget tilbake igjen. Xml-fila er en konfigurasjonsfil som encoder-programvaren skal benytte seg av. Det første bat-fila gjør er å kalle på gsstart som setter et flagg i databasen at komprimering er igangsatt. Så vil det bli kalt på et ftp-program for å hente ned innslaget lokalt, komprimering vil bli igangsatt og når det er ferdig kalles det på ftp-programmet igjen for å sende det komprimerte innslaget opp igjen på streamingsserveren. Til slutt kalles det på gsstop som sier i fra i databasen at filen er ferdig komprimert (og opplastet), altså at den kan vises på streamingsserveren.

Disse scriptene som blir generert kan ikke kjøre av seg selv. Derfor er det laget en overordnet bat-fil som kjører i løkke og som sjekker (og kjører) de genererte scriptene. I tillegg kjøres det herfra en opprydningsjobb som sletter scripts som ikke lenger skal brukes, gsclean. Gswait brukes for å ha en delay mellom hvert forsøk på å finne kjørbare scripts. Gswait venter til det blir en endring på minuttene i tid.

#### 21.1.1 "autoscript.bat"

Dett er den overodna bat-fila.

```
@echo off
d:
cd \users\open_channel\scripting\scripts

:loop

..\gswait

for %%n in (*.bat) do call %%n

..\gsclean

goto loop
```

### 21.1.2 "gscript.pas"

Dette Pascal programmet brukes til å lage scripts for nedlasting komprimering og opplasting. Parametere til komprimeringen er hentet fra en XML-fil som Envivio Encoding Station (4Coder) genererer. I denne versjonen limes parametrene inn i Pascal koden, men en senere versjon vil referere til mal- og konfigureringsfiler slik at det blir lettere å utføre endringer med hensyn til innstillinger og eventuelle endringer i passord, ftp-serveradresse eller plassering av lokale kataloger.

```
program generatescript;

uses
  mysql, sysutils;

const
  DataBase : Pchar = 'openchannel';

  ftp_in_ip_address='129.241.26.183';
  ftp_in_login_name='Open_Channel';
  ftp_in_password='publicaccess!';
  ftp_in_default_path='ORGANISASJONER';

  ftp_out_ip_address='129.241.26.183';
  ftp_out_login_name='Open_Channel';
  ftp_out_password='publicaccess!';
  ftp_out_default_path='ORGANISASJONER';

  local_path='d:\Users\Open_Channel\scripting';
  local_path_to_store_scripts='scripts';
  local_path_for_files_in='innfiler';
  local_path_for_files_out='utfiler';

function randname():string;
var
  lp:integer;
  tmpstr:string;
  bt:byte;
begin
  tmpstr:='';

  for lp:=1 to 16 do
```



```

begin
  bt:=trunc(random()*26);
  tmpstr:=tmpstr+chr(bt+ord('a'));
end;
randname:=tmpstr;
end;

procedure generate(orgname:string; filename:string; randscriptfile:string;
cid:string);
var
  textfile:text;
  result:integer;
begin
  {Script 1}
  assign (textfile,
local_path+'\'+local_path_to_store_scripts+'\'+orgname+'_'+randscriptfile+'
.bat');
  {$i-}
  rewrite(textfile);
  {$i+}
  result:=ioresult;

  {$i-}
  writeln(textfile, 'gsstart '+cid+' '+randscriptfile);
  writeln(textfile, 'FTP -s:"'+orgname+'_'+randscriptfile+'_in.txt"');
  writeln(textfile, 'c:');
  writeln(textfile, 'cd\');
  writeln(textfile, 'cd Program Files\Envivio\Envivio Encoding Station
2.5');
  writeln(textfile, 'ees_cmd -job
"'+local_path+'\'+local_path_to_store_scripts+'\'+orgname+'_'+randscriptfil
e+'.xml" -v');
  writeln(textfile, 'd:');
  writeln(textfile, 'cd\');
  writeln(textfile, 'cd users\Open_Channel\scripting');
  writeln(textfile, 'FTP -s:"'+orgname+'_'+randscriptfile+'_out.txt"');
  writeln(textfile, 'gsstop '+cid);
  {i+}
  result:=ioresult;
  close(textfile);

  {script 2}
  assign (textfile,
local_path+'\'+local_path_to_store_scripts+'\'+orgname+'_'+randscriptfile+'
_in.txt');
  {$i-}
  rewrite(textfile);
  {$i+}
  result:=ioresult;

  {$i-}
  writeln(textfile, 'open '+ftp_in_ip_address);
  writeln(textfile, ftp_in_login_name);
  writeln(textfile, ftp_in_password);
  writeln(textfile, 'cd "'+ftp_in_default_path+'");
  writeln(textfile, 'cd "'+orgname+'");
  writeln(textfile, 'lcd "'+local_path+'\'+local_path_for_files_in+'");
  writeln(textfile, 'hash');
  writeln(textfile, 'get "'+filename+'");
  writeln(textfile, 'quit');
  {i+}
  result:=ioresult;
  close(textfile);

  {script 3}
  assign (textfile,
local_path+'\'+local_path_to_store_scripts+'\'+orgname+'_'+randscriptfile+'
_out.txt');

```

```

{$i-}
rewrite(textfile);
{$i+}
result:=ioresult;

{$i-}
writeln(textfile, 'open '+ftp_out_ip_address);
writeln(textfile, ftp_out_login_name);
writeln(textfile, ftp_out_password);
writeln(textfile, 'lcd "'+local_path+'\'+local_path_for_files_out+'");
writeln(textfile, 'hash');
writeln(textfile, 'cd "'+ftp_out_default_path+'");
writeln(textfile, 'cd "'+orgname+'");
writeln(textfile, 'put "'+orgname+'_'+randscriptfile+'.mp4");
writeln(textfile, '!');
{i+}
result:=ioresult;
close(textfile);

{script 3}
assign (textfile,
local_path+'\'+local_path_to_store_scripts+'\'+orgname+'_'+randscriptfile+'.xml');
{$i-}
rewrite(textfile);
{$i+}
result:=ioresult;

{$i-}
writeln(textfile, '<?xml version="1.0" encoding="UTF-8"?>');
writeln(textfile, '<job>');
writeln(textfile, '    <version build="270" stage="2.5"/>');
writeln(textfile, '    <input>');
writeln(textfile, '        <file avilib="DirectShow"
name="' + local_path + '\'+local_path_for_files_in + '\'+orgname + '_' + randscriptfi
le + '.avi"/>');
writeln(textfile, '    </input>');
writeln(textfile, '    <output>');
writeln(textfile, '        <file
name="' + local_path + '\'+local_path_for_files_out + '\'+orgname + '_' + randscriptf
ile + '.mp4"/>');
writeln(textfile, '    </output>');
writeln(textfile, '    <parameters>');
writeln(textfile, '        <export>');
writeln(textfile, '            <parameter id="MTUSize" value="1448"/>');
writeln(textfile, '            <parameter id="exportType" value="mp4"/>');
writeln(textfile, '            <parameter id="hinted" value="true"/>');
writeln(textfile, '            <parameter id="progressiveDownload"
value="true"/>');
writeln(textfile, '        </export>');
writeln(textfile, '    <presetProperties>');
writeln(textfile, '        <parameter id="comments" value=""/>');
writeln(textfile, '        <parameter id="summary" value="..."/>');
writeln(textfile, '    </presetProperties>');
writeln(textfile, '    <videoEncoder>');
writeln(textfile, '        <parameter id="bitRate"
value="2000000"/>');
writeln(textfile, '        <parameter id="interlaced"
value="false"/>');
writeln(textfile, '        <parameter id="keyFramePeriodInMs"
value="10000"/>');
writeln(textfile, '        <parameter id="motionEstimationMethod"
value="normal"/>');
writeln(textfile, '        <parameter id="numberOfPass" value="2"/>');
writeln(textfile, '        <parameter id="quantizationType"
value="1"/>');

```



```

procedure findnewfiles();
var
  count,num : longint;
  code : integer;
  sock : PMYSQL;
  qmysql : TMYSQL;
  qbuf : string [160];
  rowbuf : TMYSQL_ROW;
  rowbuf2 : TMYSQL_ROW;
  dummy : string;
  recbuf : PMYSQL_RES;
  recbuf2 : PMYSQL_RES;
  Query : Pchar;
  orgname : string;
  qstr : string;
begin
  orgname:='';
  { Write ('Connecting to MySQL...');}
  sock :=
mysql_connect(PMysql(@qmysql),'158.38.63.23','outsider','aB!2u(8c#');
  if sock=nil then
    begin
      Writeln (stderr,'Couldn't connect to MySQL. ');
      Writeln (stderr,mysql_error(@qmysql));
      halt(1);
    end;
  { Writeln ('Done. ');}

  { Writeln ('Connection data:');}
  writeln ('Host info : ',mysql_get_host_info(sock));
  writeln ('Server info : ',mysql_stat(sock));
  writeln ('Client info : ',mysql_get_client_info);}

  { Writeln ('Selecting Database ',DataBase,'...');}
  if mysql_select_db(sock,DataBase) < 0 then
    begin
      Writeln (stderr,'Couldn't select database ',Database);
      Writeln (stderr,mysql_error(sock));
      halt (1);
    end;

  Query:='select orgid, filename, contentid from content where
fin_upload="yes" and gen_script="no"';
  { writeln ('Executing query : ',Query,'...');}
  if (mysql_query(sock,Query) < 0) then
    begin
      Writeln (stderr,'Query failed ');
      writeln (stderr,mysql_error(sock));
      Halt(1);
    end;

  recbuf := mysql_store_result(sock);
  if RecBuf=nil then
    begin
      Writeln ('Query returned nil result. ');
      mysql_close(sock);
      halt (1);
    end;
  { Writeln ('Number of records returned : ',mysql_num_rows (recbuf));
  Writeln ('Number of fields per record : ',mysql_num_fields(recbuf));}

  rowbuf := mysql_fetch_row(recbuf);
  while (rowbuf <>nil) do
    begin
      qstr:='select orgnavn from organisasjoner where
orgid="'+rowbuf[0]+'";

```

```
strcpy(Query,qstr);

orgname:='';
if (mysql_query(sock, Query)<0) then
begin
  writeln('2nd lvl q failed');
end else
begin
  recbuf2:=mysql_store_result(sock);
  if recbuf2<>nil then
  begin
    rowbuf2:=mysql_fetch_row(recbuf2);
    orgname:=rowbuf2[0];
  end;
end;

{
  Write ('(orgid: ', rowbuf[0]);
  Write (' ', orgname: ', orgname);
  Write (' ', filename: ', rowbuf[1]);
  writeln;}

if orgname<>' ' then
begin
  writeln ('Script for '+orgname+', file '+rowbuf[1]+' created!');
  generate(orgname, rowbuf[1], randname(), rowbuf[2]);
  qstr:='update content set gen_script="yes" where
contentid="'+rowbuf[2]+'";
  strcpy(Query,qstr);
  if (mysql_query(sock, Query)<0) then
  begin
    writeln ('Failed!');
  end;
end;

  rowbuf := mysql_fetch_row(recbuf);

end;
{ Writeln ('Freeing memory occupied by result set...');}
mysql_free_result (recbuf);

{ Writeln ('Closing connection with MySQL.')}
mysql_close(sock);

end;

begin
  randomize();
  findnewfiles();
end.
```

### 21.1.3 "sgen.pas"

Dette programmet ble skrevet under oppsetningen av infrastrukturen demonstrert i diplomoppgaven. Etter gjentatte hacker angrep var det ønskelig å lage et program som genererte scripts fra IP-adresse, passord og brukernavn i en tekstfil. Tekstfilen er enklere å endre når maskiner går ned og man må flytte programmet til alternative enheter. I "gscript.pas" måtte disse parametrene skrives inn i koden som deretter måtte kompileres på nytt. Denne er i sin helhet erstattet av "sgen.pas".

```

{*****}
*
* sgen - script generator
*
* sgen (c) 2004 av Erik Vold,
* laget av Erik Vold for Åpen Kanal Norge i 2006
*
*****}

//{$DEFINE Debug}

program sgen;

uses
  crt, mysql, sysutils;

{*****}
*
* K O N F I G U R A S J O N S D A T A
*
*****}

type
  configt=record
    verbose:string;
    local_path_relative:string;
    local_root_to_work_dir:string;
    path_for_storing_scripts:string;
    ip_for_db:string;
    name_for_db:string;
    login_name_for_db:string;
    pwd_for_db:string;
    ip_for_ftp_in:string;
    login_name_for_ftp_in:string;
    pwd_for_ftp_in:string;
    local_path_for_ftp_in:string;
    ip_for_ftp_out:string;
    login_name_for_ftp_out:string;
    pwd_for_ftp_out:string;
    local_path_for_ftp_out:string;

    def_path_to_store_files_on_remote_ftp:string;
    def_path_to_get_files_on_remote_ftp:string;

    number_of_bitrates:integer;
    bitrates:array[0..9] of string;

    local_path_xml_templates:string;
    number_of_xml_templates:integer;
    xml_template:array[0..9] of string;

    org_name:string;
    file_name:string;
    content_id:string;
  end;

{*****}
*
* G L O B A L E   V A R I A B L E R
*
*****}

var
  line_count:integer; //For å indikere hvilken linje i konfig fil noe
  feiler

```

```

{*****}
*
* initconfig()
*   inndata: en "record" med konfigureringsdata
*
* Brukes for å initialisere konfigureringsstrukturen.
*
*****}

procedure initconfig(var cfg:configt);
var
  lp:integer;
begin
  with (cfg) do
  begin
    verbose:='';
    local_path_relative:='';
    local_root_to_work_dir:='';
    path_for_storing_scripts:='';
    ip_for_db:='';
    pwd_for_db:='';
    ip_for_ftp_in:='';
    login_name_for_ftp_in:='';
    pwd_for_ftp_in:='';
    local_path_for_ftp_in:='';
    ip_for_ftp_out:='';
    login_name_for_ftp_out:='';
    pwd_for_ftp_out:='';
    local_path_for_ftp_out:='';
    name_for_db:='';
    login_name_for_db:='';

    def_path_to_store_files_on_remote_ftp:='';
    def_path_to_get_files_on_remote_ftp:='';

    number_of_bitrates:=0;
    for lp:=0 to 9 do
    begin
      bitrates[lp]:='';
    end;

    local_path_xml_templates:='';
    number_of_xml_templates:=0;
    for lp:=0 to 9 do
    begin
      xml_template[lp]:='';
    end;

    org_name:='';
    file_name:='';
    content_id:='';
  end;
end;

{*****}
*
* stripext()
*   inndata: et filnavn
*   utdata : Samme filnavn uten extention
*
* Bruker for å fjerne filextention
*
*****}

function stripext(file_name:string):string;
var
  tmpstr:string;
  lp:integer;
  dotfound:boolean;

```

```
begin
  tmpstr:='';
  dotfound:=false;

  if (length(file_name)>1) then
  begin
    for lp:=length(file_name) downto 1 do
    begin
      if (dotfound) then tmpstr:=file_name[lp]+tmpstr;

      if (file_name[lp]='.') then dotfound:=true;
    end;
  end;

  stripext:=tmpstr;
end;

{*****
 *
 * get_number_of_tokens()
 *   inndata: en linje med tekst
 *   utdata : antall "ord" i en setning.
 *
 * Brukes for å returnere antall tokens i en streng.
 *
 *****}

function get_number_of_tokens(line:string):integer;
//Returnerer antall "ord" i en setning (tekststreng)
var
  last_space:boolean; //Siste tegnet lest er mellomrom
  lp:integer; //Teller for løkke
  cnt:integer; //Antall tokens
  quote:boolean; //Vi er inne i en "quote", mellomrom skal behandles
  annerledes
begin
  cnt := 0;
  last_space := true;
  quote:=false;

  //Hopp over linjer med kommentarer eller tomme linjer
  if (line[1] <> '#') and (line <> '') then
  begin

    //Tell ord
    for lp:=1 to length(line) do
    begin

      //Bytt oppførsel for hvert anførselstegn som er funnet,
      //mellomrom som er mellom et "quote"-par skal ikke regnes som skille
mellom "tokens"
      if (line[lp] = '"') then
      begin
        if (quote) then quote:=false else quote:=true;
      end;

      //Dersom vi har noe annet enn et mellomrom og at forrige tegn var
mellomrom
      if (line[lp] <> ' ') and (last_space) then
      begin
        //og vi ikke befinner oss i en "quote",
        if (not quote) then
        begin
          //så har vi et nytt token
          inc (cnt);
          last_space := false; //Dette tegnet var ikke mellomrom
        end;
      end;
    end;
  end;
end;
```



```
        end else
        begin
            if (line[lp] = ' ') then last_space := true; //Dette tegnet var
mellomrom
            end;
        end;
    end;

    get_number_of_tokens := cnt;
end;

{*****
*
* get_token()
*   inndata: en linje med tekst,
*           en indeks for hvilket token i tekst som skal returneres
*   utdata : "ordet" som er hentet ved indeks n
*
* Brukes for å hente et spesifikt "ord" fra en streng
*
*****}

function get_token(line:string; index:integer):string;
//Henter et "ord" fra en setning (tekststreng)
var
    last_space:boolean; //Siste tegnet lest er mellomrom
    lp:integer; //Teller for løkke
    cnt:integer; //Antall tokens
    tmp_token:string; //Mellomlager for hentet "token"
    quote:boolean; //Vi er inne i en "quote", mellomrom skal behandles
annerledes
begin
    quote:=false;
    tmp_token:='';

    cnt := 0;
    last_space := true;

    //Hopp over linjer med kommentarer eller tomme linjer
    if (line[1] <> '#') and (line <> '') then
    begin

        for lp:=1 to length(line) do
        begin

            //Bytt oppførsel for hvert anførselstegn som er funnet
            if (line[lp] ='"') then
            begin
                if quote then quote:=false else quote:=true;
            end;

            //Dersom vi har noe annet enn et mellomrom, eller vi befinner oss
inne i en "quote"
            if (line[lp] <> ' ') or (quote) then
            begin
                //og forrige tegnet var mellomrom
                if (last_space) then
                begin
                    //Så har vi et nytt "ord" eller "token"
                    inc (cnt);
                    last_space := false; //Dette tegnet var ikke mellomrom
                end;

                //Nå har vi funnet "ordet" som vi ønsker å hente, ta vare på det!
                if (cnt = index) and (line[lp]<>' "') then tmp_token := tmp_token +
line[lp];
            end else
            begin
```

```
        if (line[lp] = ' ') then last_space := true; //Dette tegnet var
mellomrom
        end;
        end;
        end;

        get_token := tmp_token;
end;

{*****
*
* translate()
*   inndata: en linje med tekst,
*           konfigureringsstruktur som skal fylles med data fra linje med
tekst
*
* Brukes for å oversette linjer fra konfigureringsfil for å fylle en
konfigureringsstruktur
*
*****}

procedure translate(line:string; var cfg:configt);
//Parser konfigureringsfila
var
    number_of_tokens:integer;
    token1, token2, token3:string;
    num:integer;
    err:word;
begin
    number_of_tokens := get_number_of_tokens(line);

    //Initialiser tokens
    token1:='';
    token2:='';
    token3:='';

    //Sikre at det er minst to tokens før vi gidder å gjøre noe
    if (number_of_tokens>1) then
    begin
        with (cfg) do
        begin
            if (number_of_tokens>=1) then token1 := get_token(line, 1);
            if (number_of_tokens>=2) then token2 := get_token(line, 2);
            if (number_of_tokens>=3) then token3 := get_token(line, 3);

            {$IFDEF DEBUG}
            //***** For debugging: *****
            writeln (number_of_tokens, ' tokens: ', token1, ', ', token2, ', ',
            token3);
            {$ENDIF}

            //Parametre som utgjør 2 tokens
            if (token1='verbose') then verbose := token2 else
            if (token1='local_path_relative') then local_path_relative := token2
        else
            if (token1='local_root_to_work_dir') then local_root_to_work_dir :=
            token2 else
            if (token1='path_for_storing_scripts') then path_for_storing_scripts
            := token2 else
            if (token1='ip_for_db') then ip_for_db := token2 else
            if (token1='pwd_for_db') then pwd_for_db := token2 else
            if (token1='ip_for_ftp_in') then ip_for_ftp_in := token2 else
            if (token1='login_name_for_ftp_in') then login_name_for_ftp_in :=
            token2 else
            if (token1='pwd_for_ftp_in') then pwd_for_ftp_in := token2 else
            if (token1='local_path_for_ftp_in') then local_path_for_ftp_in :=
            token2 else
```

```
    if (token1='ip_for_ftp_out') then ip_for_ftp_out := token2 else
    if (token1='login_name_for_ftp_out') then login_name_for_ftp_out :=
token2 else
    if (token1='pwd_for_ftp_out') then pwd_for_ftp_out := token2 else
    if (token1='local_path_for_ftp_out') then local_path_for_ftp_out :=
token2 else
    if (token1='name_for_db') then name_for_db := token2 else
    if (token1='login_name_for_db') then login_name_for_db := token2 else
    if (token1='local_path_xml_templates') then
local_path_xml_templates:= token2 else
    if (token1='def_path_to_store_files_on_remote_ftp') then
def_path_to_store_files_on_remote_ftp:=token2 else
    if (token1='def_path_to_get_files_on_remote_ftp') then
def_path_to_get_files_on_remote_ftp:=token2 else

    if (number_of_tokens>2) then
    begin

    if (token1='bitrate') then
    begin
        //Oversett tekst til nummer
        val(token2, num, err);

        //Dersom oversettelsen feilet
        if (err<>0) then
        begin
            //Informer om at token2 ikke kunne oversettes til et tall
            writeln('Linje ', line_count, ', translate(), feil, token2 er
ikke et nummer');
        end else
        begin
            //Oversettelsen gikk bra

            //Ta vare på info om bitrate
            bitrates[number_of_bitrates]:=token3;

            //reflekter antallet ved å øke telleren
            inc (number_of_bitrates);
        end;

    end else

    if (token1='xml_template') then
    begin
        //Oversett tekst til nummer
        val(token2, num, err);

        //Dersom oversettelsen feilet
        if (err<>0) then
        begin
            //Informer om at token2 ikke kunne oversettes til et tall
            writeln('Linje ', line_count, ', translate(), feil, token2 er
ikke et nummer');
        end else
        begin
            //Oversettelsen gikk bra

            //Ta vare på info om filnavn til XML maler
            xml_template[number_of_xml_templates]:=token3;

            //reflekter antallet ved å øke telleren
            inc (number_of_xml_templates);
        end;
    end else

    begin
        //Informer om at det er parametre i konfigurasjonsfila som ikke
ble forstått
```

```
        writeln('Linje ', line_count, ', translate(), feil, forstår ikke
', token1);
        end;

        end else
        begin
        //Informer om at det er parametre i konfigurasjonsfila som ikke ble
forstått
        writeln('Linje ', line_count, ', translate(), feil, forstår ikke ',
token1);
        end;
        end;

        end;
end;

{*****
*
* randname()
*   utdata : en 16 tegn lang tilfeldig alfanumerisk streng
*
* Brukes for å generere del av et tilfeldig filnavn
*
*****}

function randname():string;
var
    lp:integer;
    tmpstr:string;
    bt:byte;
begin
    tmpstr:='';

    for lp:=1 to 16 do
    begin
        bt:=trunc(random()*26);
        tmpstr:=tmpstr+chr(bt+ord('a'));
    end;
    randname:=tmpstr;
end;

{*****
*
* read_cfg_file()
*   inndata: et filnavn til en konfigureringsfil,
*           struktur for å ta i mot konfigureringsdata
*   utdata : feilkode. 0=ingen feil
*
* Brukes for å lese inn en konfigureringsfil.
*
*****}

function read_cfg_file(cfg_file_name:string; var cfg:configt):integer;
//Leser konfigureringsfila linje for linje
var
    line:string; //Holder en og en lest linje
    cfg_file:text; //Filehåndterer for konfigureringsfil
    result:integer; //Resultat av i/u-kommandoer
begin
    read_cfg_file := 0; //Initielt så har lesing gått greit
    initconfig(cfg); //initialiser konfigurasjonsstruktur

    assign(cfg_file, cfg_file_name);
    {$i-} //Slå av i/u-feilsjekking
    reset(cfg_file); //Åpne fil for lesing
    {$i+} //Slå på i/u-feilsjekking

    result := ioreult;
    if (result=0) then //Åpning av fil ok
```

```
begin
  while (not eof(cfg_file)) do //Så lenge det er konfigurasjonsparametre
å lese
  begin
    //Hold orden på hvilken linje i fil som blir prossesert i øyeblikket
    inc (line_count);

    {$i-}
    readln(cfg_file, line); //Les en linje
    {$i+}

    result := ioreult;
    if (ioreult=0) then
    begin
      translate(line, cfg) //Oversett og fyll inn i konfig-objektet
    end else
    begin
      //Lesing fra fil feilet av en eller annen mystisk grunn.
      //Dette vil sannsynligvis være en meget sjelden hendelse, men
      //det bør håndteres.
      read_cfg_file := result; //Feilkode som skal returneres
      break; //Bryt ut av "while"-løkke.
    end;
  end;

  close(cfg_file); //Steng konfigurasjonsfil
end;

read_cfg_File := result; //Returner resultat av siste I/U-operasjon
end;

{*****
*
* verify_config()
*   inndata: konfigurasjonsdata
*
* Kontroll av de viktigste parametrene
*
*****}

procedure verify_config(cfg:configt);
begin
  with (cfg) do
  begin

    if (number_of_bitrates=0) then
    begin
      //Informer om at det ikke er definert noen bitrater
      writeln('FEIL: Ingen bitrater er definert!');
      halt(0); //Ingen vits i å fortsette når man ikke kan gjøre noe
    end else

    if (number_of_bitrates<>number_of_xml_templates) then
    begin
      //Informer om at det ikke er definert tilsvarende templates for
      bitrater
      writeln('FEIL: #bitrates != #templates');
      halt(0); //Ingen vits i å fortsette når man ikke kan gjøre noe
    end;

  end;
end;

{*****
*
* gen_file_ftp_in()
*   inndata:
*
* Genererer script for mottak av filer over ftp
*****}
```

```

*
*****}

procedure gen_file_ftp_in(rnd_str:string; cfg:configt);
var
  textfile:text;
  result:integer;
  local_path:string;
  local_file_name:string;
begin
  with (cfg) do
  begin

    //Genererer filnavn
    local_file_name:=org_name + '_' + rnd_str + '_in.txt';

    if (local_path_relative='true') then
    begin
      //Lokale stier er relative
      local_path:=local_root_to_work_dir + '\';
    end else
    if (local_path_relative='false') then
    begin
      //Lokale stier er absolutte
      local_path:='';
    end else
    begin
      //Feil verdi i variabel local_path_relative
      writeln('ERROR: local_path_relative inneholder en ulovlig verdi');
      halt(0);
    end;

    assign (textfile,
local_path+path_for_storing_scripts+'\'+local_file_name);
    {$i-}
    rewrite(textfile);
    {$i+}
    result:=ioresult;

    //TODO: Feilsjekking

    {$i-}
    writeln(textfile, 'open ' + ip_for_ftp_in);
    writeln(textfile, login_name_for_ftp_in);
    writeln(textfile, pwd_for_ftp_in);
    writeln(textfile, 'cd "' + def_path_to_get_files_on_remote_ftp + '"');
    writeln(textfile, 'cd "' + org_name + '"');
    writeln(textfile, 'lcd "' + local_path + local_path_for_ftp_in + '"');

    {$IFDEF DEBUG}
    //***** For debugging: *****
    writeln(textfile, 'hash');
    {$ENDIF}

    writeln(textfile, 'get "' + file_name + '"');
    writeln(textfile, 'quit');
    {$i+}

    //TODO: Feilsjekking

    //Vi er ferdige med fila nå
    close(textfile);
  end;
end;

{*****}
*
* gen_file_ftp_out()
* inndata:

```

```

*
* Genererer script for sending av filer over ftp
*
*****}

procedure gen_file_ftp_out(lp:integer; rnd_str:string; cfg:configt);
var
  textfile:text;
  result:integer;
  local_path:string;
  local_file_name:string;
begin
  with (cfg) do
  begin

    //Genererer filnavn
    local_file_name:=org_name + '_' + rnd_str + '_' + bitrates[lp]
+ '_out.txt';

    if (local_path_relative='true') then
    begin
      //Lokale stier er relative
      local_path:=local_root_to_work_dir + '\';
    end else
    if (local_path_relative='false') then
    begin
      //Lokale stier er absolutte
      local_path:='';
    end else
    begin
      //Feil verdi i variabel local_path_relative
      writeln('ERROR: local_path_relative inneholder en ulovlig verdi');
      halt(0);
    end;

    assign (textfile,
local_path+path_for_storing_scripts+'\'+local_file_name);
    {$i-}
    rewrite(textfile);
    {$i+}
    result:=ioresult;

    {$i-}
    writeln(textfile, 'open '+ip_for_ftp_out);
    writeln(textfile, login_name_for_ftp_out);
    writeln(textfile, pwd_for_ftp_out);

    {$IFDEF DEBUG}
    //***** For debugging: *****
    writeln(textfile, 'hash');
    {$ENDIF}

    writeln(textfile, 'cd "'+def_path_to_store_files_on_remote_ftp+'");
    writeln(textfile, 'cd "'+org_name+'");
    writeln(textfile, 'cd "'+bitrates[lp]+'");
    writeln(textfile, 'lcd "'+local_path+local_path_for_ftp_out+'");
    writeln(textfile, 'put "'+stripext(file_name)+'.mp4"');
    writeln(textfile, 'cd ..');
    writeln(textfile, 'bye');
    {$i+}
    result:=ioresult;
    close(textfile);
  end;
end;

{*****
*
* gen_file_bat()
* inndata:

```

```

*
* Genererer script for utførelse av jobber
*
*****}

procedure gen_file_bat(rnd_str:string; cfg:configt);
var
  textfile:text;
  result:integer;
  local_path:string;
  lp:integer;
  input_file, output_file, preset_file:string;
begin
  with (cfg) do
    begin
      if (local_path_relative='true') then
        begin
          //Lokale stier er relative
          local_path:=local_root_to_work_dir + '\';
        end else
          if (local_path_relative='false') then
            begin
              //Lokale stier er absolutte
              local_path:='';
            end else
              begin
                //Feil verdi i variabel local_path_relative
                writeln('ERROR: local_path_relative inneholder en ulovlig verdi');
                halt(0);
              end;

            assign (textfile,
local_path+'\'+path_for_storing_scripts+'\'+org_name+'_'+rnd_str+'.bat');
            {$i-}
            rewrite(textfile);
            {$i+}
            result:=ioresult;

            {$i-}
            writeln(textfile, 'd:');
            writeln(textfile, 'cd ' + local_root_to_work_dir);
            writeln(textfile, local_root_to_work_dir+'\gsstart ' + content_id + ' '
+ rnd_str);
            writeln(textfile, 'cd ' + local_root_to_work_dir+ '\ ' +
path_for_storing_scripts);
            writeln(textfile, 'FTP -s:"'+org_name+'_'+rnd_str+'_in.txt");
            writeln(textfile, 'del "'+org_name+'_'+rnd_str+'_in.txt");

            //Generer utførelse av encoding for alle bitrater
            for lp:=0 to (number_of_bitrates-1) do
              begin
                //Ordne filnavn
                input_file:=local_path+local_path_for_ftp_in+'\'+file_name;

                output_file:=local_path+local_path_for_ftp_out+'\'+stripext(file_name)+'_mp
4';

                preset_file:=local_path+local_path_xml_templates+'\'+xml_template[lp];

                writeln(textfile, 'c:');
                writeln(textfile, 'cd "\Program Files\Envivio\Envivio Encoding
Station 2.5");
                writeln(textfile, 'ees_cmd -i "' + input_file + '" -o "' +
output_file + '" -p "' + preset_file + '"');
                writeln(textfile, 'd:');
                writeln(textfile, 'cd ' + local_root_to_work_dir+ '\ ' +
path_for_storing_scripts);

```



```

        writeln(textfile, 'FTP -
s: '"+org_name+'_'+rnd_str+'_'+bitrates[lp]+'_out.txt");
//        writeln(textfile, 'del
"' +local_path+path_for_storing_scripts+'\'+org_name+'_'+rnd_str+ '_' +
bitrates[lp] +'.xml"');
        writeln(textfile, 'del
"' +org_name+'_'+rnd_str+'_'+bitrates[lp]+'_out.txt");
        end;

        writeln(textfile, local_root_to_work_dir+'\gsstop '+content_id);
        {i+}
        result:=ioresult;
        close(textfile);
    end;
end;

{*****
*
* process_new_files()
* inndata: struktur for å ta i mot konfigureringsdata
*
* Generer script ut i fra nye filer som er lastet opp
*
* TODO: Analysere og se om denne kan "plukkes" fra hverandre og kortes
ned
*
*****}

procedure process_new_files(cfg:configt);
var
    sock:pmysql;
    qmysql:tmysql;
    recbuf, recbuf2:pmysql_res;
    rowbuf, rowbuf2:tmysql_row;

    ip_address:pchar;
    login_name:pchar;
    pass_word:pchar;
    db_name:pchar;

    query:string;
    pquery:pchar;

    lp:integer;
    rnd_str:string;
begin
    {$IFDEF DEBUG}
    //***** For debugging: *****
    Write ('Kobler til mySQL: ');
    {$ENDIF}

    //Konverter fra string til pchar
    ip_address:=nil;
    login_name:=nil;
    pass_word:=nil;
    db_name:=nil;

    //Alloker buffer til pchar
    ip_address:=stralloc(250);
    login_name:=stralloc(250);
    pass_word:=stralloc(250);
    db_name:=stralloc(250);

    //Sikker konvertering, max 250 tegn
    strplcopy(ip_address, cfg.ip_for_db, 250);
    strplcopy(login_name, cfg.login_name_for_db, 250);
    strplcopy(pass_word, cfg.pwd_for_db, 250);
    strplcopy(db_name, cfg.name_for_db, 250);

```

```
//Koble til databaseserver
sock := mysql_connect(pmysql(@qmysql), ip_address, login_name,
pass_word);

if (sock=nil) then
begin
  writeln(stderr, 'Tilkobling til mySQL feilet. ');
  writeln(stderr, mysql_error(@qmysql));
  halt(0);
end;

{$IFDEF DEBUG}
//***** For debugging: *****
  writeln('OK');
  writeln();
{$ENDIF}

{$IFDEF DEBUG}
//***** For debugging: *****
  writeln ('Host info      : ', mysql_get_host_info(sock));
  writeln ('Server info     : ', mysql_stat(sock));
  writeln ('Client info      : ', mysql_get_client_info);
{$ENDIF}

{$IFDEF DEBUG}
//***** For debugging: *****
  write ('Velger database: ');
{$ENDIF}

//Velger database
if (mysql_select_db(sock, db_name)<0) then
begin
  writeln(stderr, 'Klarte ikke å velge database ', db_name);
  writeln(stderr, mysql_error(sock));
  halt (0);
end;

{$IFDEF DEBUG}
//***** For debugging: *****
  writeln('OK');
  writeln();
{$ENDIF}

//Utfører spørring etter nye filer
query:='select orgid, filename, contentid from content where
fin_upload="yes" and gen_script="no"';

//konverterer fra string til pchar
pquery:=nil;
pquery:=stralloc(250);
strplcopy(pquery, query, 250);

{$IFDEF DEBUG}
//***** For debugging: *****
  write('Utfører spørring etter nye filer: ');
{$ENDIF}

if (mysql_query(sock,pquery) < 0) then
begin
  writeln(stderr, 'Spørring etter filliste feilet');
  writeln(stderr, mysql_error(sock));
  halt(0);
end;

{$IFDEF DEBUG}
//***** For debugging: *****
  writeln('OK');
  writeln();

```

```
{ $ENDIF }

{ $IFDEF DEBUG }
//***** For debugging: *****
write("Parser" spørring: ');
{ $ENDIF }

//Lagre resultatet av spørring i buffer
recbuf:=mysql_store_result(sock);
if recbuf=nil then
begin
writeln ('Ingen nye filer funnet!');
mysql_close(sock);
halt(0);
end;

{ $IFDEF DEBUG }
//***** For debugging: *****
writeln('OK');
writeln();
{ $ENDIF }

{ $IFDEF DEBUG }
//***** For debugging: *****
writeln('Antall rader returnert: ', mysql_num_rows(recbuf));
writeln('Antall felter per rad : ', mysql_num_fields(recbuf));
writeln();
{ $ENDIF }

{ $IFDEF DEBUG }
//***** For debugging: *****
writeln ('Liste over nye filer:');
{ $ENDIF }

//Hent en rad
rowbuf := mysql_fetch_row(recbuf);

//Så lenge det er rader, prosesser filer
while (rowbuf<>nil) do
begin

//Spørring for å få tak i organisasjonsnavn (eier av denne fila)
query:='select orgnavn from organisasjoner where
orgid="'+rowbuf[0]+'";

//Konverter fra string til pchar - allokering er allerede gjort
strncpy(pquery, query, 250);

{ $IFDEF DEBUG }
//***** For debugging: *****
write('Utfører spørring etter nye organisasjonsnavn: ');
{ $ENDIF }

if (mysql_query(sock, pquery)<0) then
begin
//Spørring feilet
writeln('Spørring etter organisasjon med id ', rowbuf[0], ' feilet');
writeln();

//Resultatet dersom dette skjer er at scripts ikke blir generert for
denne fila,
// nytt forsøk neste gang.
//TODO: Kanskje logge dette på en eller annen måte? Er DB
tilgjengelig i og med at spørring feilet?
end else
begin

{ $IFDEF DEBUG }
//***** For debugging: *****
```

```
writeln('OK');
{$ENDIF}

// "parse" resultatet av spørringen
recbuf2:=mysql_store_result(sock);

// Dersom spørringen ledet til noe resultat
if (recbuf2<>nil) then
begin
  rowbuf2:=mysql_fetch_row(recbuf2);

// Dersom rowbuf ikke returnerer blankt
if (rowbuf2<>nil) then
begin
  // Ta vare på organisasjonsnavnet
  cfg.org_name:=rowbuf2[0];

  // Ta vare på content id
  cfg.content_id:=rowbuf2[2];

  // Frigjør ressurser som ikke lenger behøves
  mysql_free_result(recbuf2);

/**
/** Her har vi all informasjonen som vi trenger for å generere
/** de filene vi har behov for.
/**
/** *****
/** *****
/** *****
/** *****
/** *****
/** *****
/** *****
/** *****

  // Genererer tilfeldig del av et filnavn
  rnd_str:=randname();

  // Tar vare på filnavn fra databasen
  cfg.file_name:=rowbuf2[1];

  // Genererer jobbfil
  gen_file_bat(rnd_str, cfg);

  // Generer fil for nedlasting over FTP
  gen_file_ftp_in(rnd_str, cfg);

{$IFDEF DEBUG}
/** ***** For debugging: *****
  writeln('Tilfeldig del av filnavn: ', rnd_str);
{$ENDIF}

  // Vi vet helt sikkert at det eksisterer bitrater i listen
  // For hver bitrate som er definert
  for lp:=0 to (cfg.number_of_bitrates-1) do
  begin
    // Generer fil for opplasting over FTP
    gen_file_ftp_out(lp, rnd_str, cfg);
  end;

  // Sett flagg at script for denne fila er generert
  // Spørring for å få tak i organisasjonsnavn (eier av denne fila)
  query:='update content set gen_script="yes" where contentid="' +
  cfg.content_id + '"';

  // Konverter fra string til pchar - allokering er allerede gjort
  strplcopy(pquery, query, 250);

  if (mysql_query(sock, pquery)<0) then
```

```
begin
  //Spørring feilet
  writeln('kunne ikke sette status "gen_script=yes"');
end;

//*****
//*****
//*****
//*****
//*****
//*****
//*****
end else
begin
  //Det finnes ikke noe organisasjonsnavn - inkonsistent database!

  //Dette er en ignoreringstillstand.
  //TODO: Skrive i en logg at det er en organisasjon som er blitt
slettet, men at det er data
  //      som fortsatt peker på den slettede organisasjonen.

{$IFDEF DEBUG}
//***** For debugging: *****
  writeln('Manglende organisasjonsnavn for denne fila.');
```

```

{$ENDIF}

  end;

{$IFDEF DEBUG}
//***** For debugging: *****
  //Vis info om denne fila
  writeln ('(orgid, contentid, file, org)', rowbuf[0], ', ', rowbuf[2],
', ', rowbuf[1], ', ', cfg.org_name);
  writeln();
{$ENDIF}

  end else
  begin
    writeln('Organisasjonsnavn for orgid ', rowbuf[0], ' ikke
funnet!');
    writeln();
  end;
  end;
  //Hent en rad til
  rowbuf:=mysql_fetch_row(recbuf);

end; //while

//Frigjør ressurser som ikke lenger behøves
mysql_free_result(recbuf);

{$IFDEF DEBUG}
//***** For debugging: *****
  writeln('Kobler ned forbindelsen med mySQL-serveren.');
```

```

{$ENDIF}

  //Koble fra mySQL server
  mysql_close(sock);

end;

var
  cfg_file_name:string; //Filnavn for konfigureringsfil
  cfg:configt; //Konfigurasjonsparametrene
  result:integer; //Resultat av i/u-operasjoner
begin

  //Initialiser linjetelleren
  line_count := 0;
```

```
//Generere nytt "random seed" til bruk for å generere tilfeldige filnavn
randomize();

// Teste om riktig antall parametere er gitt
// Tar kun ett parameter: filnavn for konfigureringsfil
if (paramcount()=1) then

begin //Riktig antall parametere, les inn konfigureringsfil
  cfg_file_name := paramstr(1); //Hent filnavn fra brukerinmatning

{$IFDEF DEBUG}
//***** For debugging: *****
  writeln('Leser konfigureringsfil med navn "' + cfg_file_name + '".');
{$ENDIF}

  //Les inn konfigureringsfil
  result := read_cfg_file(cfg_file_name, cfg);

  if (result=0) then
  begin

{$IFDEF DEBUG}
//***** For debugging: *****
  writeln('Fil er lest.');
```

```

{$ENDIF}

{$IFDEF DEBUG}
//***** For debugging: *****
  write('Kontrollerer konfigurasjonsdata: ');
{$ENDIF}

  //Kontrollere de konfigurasjonsdata som er hentet inn
  verify_config(cfg);

{$IFDEF DEBUG}
//***** For debugging: *****
  writeln('Konfigurasjonsdata er ok');
{$ENDIF}

  //Sjekk om det er lastet opp nye filer og generer skript for disse
  filene
  process_new_files(cfg);

  end else
  begin //Kunne ikke lese konfigureringsfil
    writeln ('Feil under åpning av konfigureringsfil,');
    write (' feilkode som ble returnert er: ');
    writeln (result);
  end;
end else //Feil i antall parametre, forklar brukeren hvordan
begin
  writeln('sgen v0.1 - script generator for "Envivio 4coder");
  writeln('Bruk av programmet:');
  writeln(' sgen <config file name>');
end;
end.
```

#### 21.1.4 "gsstart.pas"

Dette programmet setter status på et innslag som "under komprimering".

```
program start_script;

uses
  mysql, sysutils;

const
  DataBase : Pchar = 'openchannel';

procedure setstart(conid:string;randfile:string);
var
  count,num : longint;
  code : integer;
  sock : PMYSQL;
  qmysql : TMYSQL;
  qbuf : string [160];
  rowbuf : TMYSQL_ROW;
  rowbuf2 : TMYSQL_ROW;
  dummy : string;
  recbuf : PMYSQL_RES;
  recbuf2 : PMYSQL_RES;
  Query : Pchar;
  orgname : string;
  qstr : string;
begin
  orgname:='';
  sock :=
mysql_connect(PMysql(@qmysql), '158.38.63.23', 'outsider', 'aB!2u(8c#');
  if sock=nil then
    begin
      Writeln (stderr, 'Couldn't connect to MySQL. ');
      Writeln (stderr, mysql_error(@qmysql));
      halt(1);
    end;

    if mysql_select_db(sock, DataBase) < 0 then
      begin
        Writeln (stderr, 'Couldn't select database ', Database);
        Writeln (stderr, mysql_error(sock));
        halt (1);
      end;

      qstr:='update content set begin_compress="yes" where
contentid="'+conid+'";
      strcpy(Query, qstr);
      if (mysql_query(sock, Query) < 0) then
        begin
          Writeln (stderr, 'Query failed ');
          writeln (stderr, mysql_error(sock));
          Halt(1);
        end;

        qstr:='update content set randfilename="'+randfile+' " where
contentid="'+conid+'";
        strcpy(Query, qstr);
        if (mysql_query(sock, Query) < 0) then
          begin
            Writeln (stderr, 'Query failed ');
            writeln (stderr, mysql_error(sock));
            Halt(1);
          end;

          recbuf := mysql_store_result(sock);
          if RecBuf=nil then
            begin
              Writeln ('Query returned nil result. ');
              mysql_close(sock);
              halt (1);
            end;
          end;
```

```
mysql_free_result (recbuf);
mysql_close(sock);
end;

begin
  if (paramcount=2) then
    begin
      setstart(paramstr(1), paramstr(2));
    end;
  end.
end.
```

### 21.1.5 "gsstop.pas"

Dette programmet setter status på innslag til "ferdig komprimert".

```
program finito_script;

uses
  mysql, sysutils;

const
  DataBase : Pchar = 'openchannel';

procedure setstart(conid:string);
var
  count,num : longint;
  code : integer;
  sock : PMYSQL;
  qmysql : TMYSQL;
  qbuf : string [160];
  rowbuf : TMYSQL_ROW;
  rowbuf2 : TMYSQL_ROW;
  dummy : string;
  recbuf : PMYSQL_RES;
  recbuf2 : PMYSQL_RES;
  Query : Pchar;
  orgname : string;
  qstr : string;
begin
  orgname:='';
  sock :=
mysql_connect(PMysql(@qmysql),'158.38.63.23','outsider','aB!2u(8c#');
  if sock=nil then
    begin
      Writeln (stderr,'Couldn't connect to MySQL. ');
      Writeln (stderr,mysql_error(@qmysql));
      halt(1);
    end;

  if mysql_select_db(sock,DataBase) < 0 then
    begin
      Writeln (stderr,'Couldn't select database ',Database);
      Writeln (stderr,mysql_error(sock));
      halt (1);
    end;

  qstr:='update content set fin_compress="yes" where
contentid="'+conid+'";
  strpcopy(Query,qstr);
  if (mysql_query(sock,Query) < 0) then
    begin
      Writeln (stderr,'Query failed ');
      writeln (stderr,mysql_error(sock));
    end;
  end.
```



```
    Halt(1);
    end;

    recbuf := mysql_store_result(sock);
    if RecBuf=nil then
    begin
        Writeln ('Query returned nil result. ');
        mysql_close(sock);
        halt (1);
    end;

    mysql_free_result (recbuf);
    mysql_close(sock);
end;

begin
    if (paramcount=1) then
    begin
        setstart(paramstr(1));
    end;
end.
```

## 21.1.6 "gsclean.pas"

Sletter scriptfiler etter ferdige jobber.

```
program generatescript;

uses
    crt, mysql, sysutils;

const
    DataBase : Pchar = 'openchannel';

    ftp_in_ip_address='129.241.26.183';
    ftp_in_login_name='Open_Channel';
    ftp_in_password='publicaccess!';
    ftp_in_default_path='ORGANISASJONER';

    ftp_out_ip_address='129.241.26.183';
    ftp_out_login_name='Open_Channel';
    ftp_out_password='publicaccess!';
    ftp_out_default_path='ORGANISASJONER';

    local_path='d:\Users\Open_Channel\scripting';
    local_path_to_store_scripts='scripts';
    local_path_for_files_in='innfiler';
    local_path_for_files_out='utfiler';

procedure clean();
var
    count,num : longint;
    code : integer;
    sock : PMYSQL;
    qmysql : TMYSQL;
    qbuf : string [160];
    rowbuf : TMYSQL_ROW;
    rowbuf2 : TMYSQL_ROW;
    dummy : string;
    recbuf : PMYSQL_RES;
    recbuf2 : PMYSQL_RES;
    Query : Pchar;
```

```
orgname : string;
qstr : string;
rfile:string;
begin
orgname:='';
sock :=
mysql_connect(PMysql(@mysql),'158.38.63.23','outsider','aB!2u(8c#');
if sock=nil then
begin
Writeln(stderr,'Couldn't connect to MySQL. ');
Writeln(stderr,mysql_error(@mysql));
halt(1);
end;

if mysql_select_db(sock,DataBase) < 0 then
begin
Writeln(stderr,'Couldn't select database ',Database);
Writeln(stderr,mysql_error(sock));
halt(1);
end;

Query:='';
qstr:='select orgid, randfilename from content where
fin_compress="yes"';
strcpy(Query,qstr);
if (mysql_query(sock,Query) < 0) then
begin
Writeln(stderr,'Query failed ');
writeln(stderr,mysql_error(sock));
Halt(1);
end;

recbuf := mysql_store_result(sock);
if RecBuf=nil then
begin
Writeln('Query returned nil result. ');
mysql_close(sock);
halt(1);
end;

rowbuf := mysql_fetch_row(recbuf);
while (rowbuf <>nil) do
begin
writeln('q2');
qstr:='select orgnavn from organisasjoner where
orgid="'+rowbuf[0]+'";
strcpy(Query,qstr);

orgname:='';
if (mysql_query(sock, Query)<0) then
begin
writeln('2nd lvl q failed');
end else
begin
recbuf2:=mysql_store_result(sock);
if recbuf2<>nil then
begin
rowbuf2:=mysql_fetch_row(recbuf2);
orgname:=rowbuf2[0];
end;
end;

if (orgname<>'') then
begin
rfile:=local_path+'\''+local_path_to_store_scripts+'\''+orgname+'_'+rowbuf[1]
;
deletefile(rfile+'.bat');
```

```
    deletefile(rfile+'.xml');
    deletefile(rfile+'_in.txt');
    deletefile(rfile+'_out.txt');
end;

rowbuf := mysql_fetch_row(recbuf);

end;

mysql_free_result (recbuf);
mysql_close(sock);
end;

begin
    randomize();

    clean();
end.
```

### 21.1.7 "gswait.pas"

Venter til minuttene endres. (Maks 1 minutt.) Skal kun lage en kort pause mellom hver økt i komprimeringssystemet.

```
program start_script;

uses
    crt, sysutils;

var
    otid, tid:string;
begin
    tid:=timetostr(time);
    otid:=tid;

    while (otid=tid) do
    begin
        otid:=tid;
        tid:=timetostr(time);
        delay(1000);
        write('.');
    end;

    writeln;
end.
```

## 21.2 Forslag til utbedringer

Https bør implementeres og et sertifikat kan mulig skaffes. Https tilbyr autentisering av nettsted samt kryptering. Autentisering er ikke egentlig noe vi trenger, men kryptering er vi avhengige av. Uten sertifikat vil brukeren få opp en dialogboks med beskjed om at nettstedet "ikke er godkjent". Dette er ikke noe problem for erfarne brukere, men med et sertifikat slipper brukeren å måtte ta stilling til dette.

Databasedesignet er ikke optimalt. Dette er grunnet usikkerhet om hva som er tenkt lagret i databasen og hvor dette skal lagres. En mindre endring i databasen vil være nødvendig for å få et bedre design.

Feilhåndtering er så og si ikke-eksisterende. Det er ikke sikkert at det er så mye som skal gjøres ved feilsituasjoner, problemet går mer på å unngå at feilmeldinger unødig vises for brukeren. (Det er ikke så mye et php script kan gjøre dersom en sql eller ftp server feiler.)

Sikkerhet: teksteditoren joe som all kode er skrevet i genererer arbeidsfiler med samme navn som filene det jobbes på med en tilde (~) etter. Dersom tildefilene (arbeidsfiler) åpnes i en nettleser eksponeres all php kode for det scriptet inklusive passord som måtte finnes i filen. Disse filene MÅ slettes etter at filer er redigert. Dette er bare et punkt å huske, altså ikke noe som må implementeres.

Det med passord i klartekst i php-koden er også en uting dersom php skulle feile og webserveren skulle i vanvare vise php koden i klartekst. Det er fortsatt en usikkerhet hvorvidt passord skal lagres i databasen eller i egne filer utenfor offisielle tilgjengelige filområder, men det siste ses som mest sannsynlig.

Administrasjonssider og rettighetssystem er et behov for å administrere systemet. En god strategi er lagt og systemet er delvis implementert. Overordnet fortalt baserer systemet seg på "hardkodete" nøkkelord samlet i grupper ala "redaktør" og "administrator" som må matche i databasen. Tilhørende nøkkelordet er også en verdi. Kan på en måte sammenlignes med "registry" i MS Windows.

Det er et mål å få koden 100% html strict. Det er bare snakk om å erstatte en del tags med et tilsvarende css entry samt avslutte enkelt-tags på en ordentlig måte. (Eksempel: ikke <br> men <br />.)

Validering av all brukerinntmatning er noe som må implementeres så snart som mulig og må være på plass før systemet settes i drift. Dette må være på plass på grunn av at en bruker som et eksempel kan legge inn ytterligere sql eller html kode istedenfor brukerinformasjon. Det finnes fine og ferdige funksjoner for dette i php.

## 22 F - Appendiks: "Drag and drop" komprimeringscript

I forbindelse med demonstrasjonen for Leif Arne Rønningen ble det laget scrip som sørget for filhåndtering og komprimering av flere formater. Formatene dannet et sammenligningsgrunnlag for en diskusjon rundt fremtidige komprimeringsalgoritmer i TV-nettet. Scriptene som beskrives her vil startes av tjenerapplikasjonen som håndterer filopplasting. Det er ikke ønskelig å inkludere de i allerede eksisterende komprimeringsscript selv om dette ville vært svært enkelt. Åpen kanal ønsker å ha mulighet til å utføre kompresjonen på en maskin de alltid har tilgang på i tillegg til den vi låner av Q2S-senteret (Envivio 4Coder). Dette fordi man ønsker en redundans og forutsigbarhet i systemet, samt at man ikke vil være avhengig samarbeidet med Q2S-senteret.

### 22.1 Komprimering til Windows Media 9

Dette scriptet deinterlaser videoen og komprimerer til Windows media 9. Filen kopieres også til en Windows media streaming server som Åpen kanal har midlertidig tilgang til gjennom NTNU. Her er den streambar med linken `mms://<server>/<filnavn>` For at all video skal samles på et sted overføres fila med FTP til åpen kanal sin FTP-server. Parametrene som kalles i ftp-overføringene er vanligvis fra en tekstfil etter `-s:` Her benytter man i stedet `%0` som betyr at scriptet kaller på seg selv der ftp serveren finner parametrene og ignorerer de linjene den ikke forstår. `%1` er filnavnet på videoen som blir sluppet over bat-fila (eksekvering med 1.parameter) Windows media encoder er avhengig av fila `Cscript.exe` for å kunne hente metodene til commandolinje kjøringen. Denne finnes i en installasjonsfil på nettet som heter `scripten.exe`. Microsoft sitt `WMcmd.vbs`-fil må også lastes ned der metodene er beskrevet.

```
CScript.exe wmcmd.vbs -input %1 -output %1.wmv -a_mode 1 -a_setting
128_48_2 -a_codec WMA9STD -v_preproc 1 -v_mode 1 -v_framerate 25 -v_bitrate
1000000 -v_codec WMV9
ftp -s:%0
goto done
open openchannel.item.ntnu.no
Open_Channel
[skjult passord]
lcd "C:\Program Files\Darwin Streaming
Server\Movies\ORGANISASJONER\OpenChannelNo"
cd ORGANISASJONER
cd OpenChannelNo
cd WindowsMedia
mput *.wmv
yes
disconnect
bye
:done
copy %1.wmv Y:\ORGANISASJONER\WindowsMedia
del %1.wmv
```

## 22.2 Komprimering til MPEG2 over MPEG2TS

Dette scriptet er laget for komprimering av materiale som skal spilles av på playout serveren der videoen forblir interlaced. Her benytter jeg VLC til å utføre MPEG-komprimeringen og pakke materialet inn i en MPEG2 transport strøm (TS). Filen blir overført med FTP til Åpen kanal sin FTP-server der den er tilgjengelig for cardinal console som igjen laster opp til playout-serveren samt registrerer innslaget i avviklings-databasen.

```
vlc -vvv %1
:sout=#transcode{vcodec=mp2v,vb=3500,scale=1,acodec=mpga,ab=192,channels=2}
:duplicate{dst=std{access=file,mux=ts,url=%1.ts}} vlc:quit
ftp -s:%0
goto done
open openchannel.item.ntnu.no
Open_Channel
[skjult passord]
lcd "C:\Program Files\Darwin Streaming
Server\Movies\ORGANISASJONER\OpenChannelNo"
cd ORGANISASJONER
cd OpenChannelNo
cd MPEG2TS
mput *.ts
yes
disconnect
bye
:done
del %1.ts
```

## 22.3 Komprimering til H264 over MPEG2TS

Dette scriptet er laget for komprimering av materiale som skal spilles av på playoutserveren. VLC benytter her en mer biteffektiv codec som er aktuell i det nye digitale bakkenettet. Det finnes ikke støttte for codecen H264 i set-top-boksen vi har så denne fikk vi ikke verifisert annet enn på en PC. I sammenligningstesten ble H264-kodet materiale i stedet hintet og lagt ut på en streaming server. Fremgangsmåten er beskrevet i scriptet nedenfor. Ellers er alt som for MPEG2 over MPEG2TS med unntak av at dette scriptet også foretar en FTP-overføring til en alternativ (redundant) server med FTP-parametre gitt av en tekstfil.

```
vlc -vvv %1
:sout=#transcode{vcodec=h264,vb=1024,scale=1,acodec=aac,ab=192,channels=2}:
duplicate{dst=std{access=file,mux=ts,url=%1.ts}} vlc:quit
ftp -s:%0
goto done
open openchannel.item.ntnu.no
Open_Channel
[skjult passord]
lcd "C:\Program Files\Darwin Streaming
Server\Movies\ORGANISASJONER\OpenChannelNo"
cd ORGANISASJONER
cd OpenChannelNo
```

```
cd H264_over_MPEG2TS
mput *.ts
yes
disconnect
bye
:done
ftp -s:"C:\Program Files\Darwin Streaming
Server\Movies\ORGANISASJONER\OpenChannelNo\ftp_kommandoer_for_redundans_TS.
txt"
copy %1.ts Z:\Open_Channel\ORGANISASJONER\OpenChannelNo\H264_over_MPEG2TS
del %1.ts
```

## 22.4 Komprimering til H264 i en streambar MPEG4 fil

Dette scriptet bruker VLC til å komprimere til H264 og legge innholdet i en MPEG4-fil. For å gjøre filen streambar må hver track (her nummer 1 og 2) hintes. Dette utføres av MP4Creator som er å finne i MPEG4IP sine kompilasjoner. Filen overføres til Åpen kanal sin FTP-server samt en alternativ (redundant) server med FTP-parametere hentet fra en tekstfil. Filen kan streames fra begge servere (Darwin QT streaming server er installert) med linken e-rtsp://<server>/<filnavn>

```
vlc -vvv %1
:sout=#transcode{vcodec=h264,vb=1024,scale=1,acodec=mp4a,ab=192,channels=2}
:duplicate{dst=std{access=file,mux=mp4,url=%1.mp4}} vlc:quit --sout-
transcode-deinterlace
mp4creator -list %1.mp4
mp4creator -hint=1 %1.mp4
mp4creator -hint=2 %1.mp4
ftp -s:%0
goto done
open openchannel.item.ntnu.no
Open_Channel
[skjult passord]
lcd "C:\Program Files\Darwin Streaming
Server\Movies\ORGANISASJONER\OpenChannelNo"
cd ORGANISASJONER
cd OpenChannelNo
cd H264
mput *.mp4
yes
disconnect
bye
:done
ftp -s:"C:\Program Files\Darwin Streaming
Server\Movies\ORGANISASJONER\OpenChannelNo\ftp_kommandoer_for_redundans_H26
4.txt"
copy %1.mp4 Z:\Open_Channel\ORGANISASJONER\OpenChannelNo\H264
del %1.mp4
```

## Vedlegg 1 – Samarbeid med Uninett

Åpen Kanal  
v/Erik Vold  
Tidemandsgate 20

7030 Trondheim

Deres ref.:

Vår ref.: BU105/02

Trondheim, 11/04/2006

Your ref.:

Our ref.: BU105/02

### Samarbeidet mellom UNINETT og Åpen Kanal

Åpen Kanal har signalisert et ønske om å koble opp sine medlemmer til UNINETT på prosjektbasis. Dette stiller UNINETT seg meget positive til, og ser dette i sammenheng med Midgard Medialab og utbredelse av multicast teknologi. UNINETT er også forespurt om å overta drift av Midgard Medialabs Avid Triligent system, dette er dog ikke avklart ennå selv om vi har stilt oss positive til dette.

Bakgrunnen for at vi stiller oss positive er begrunnet med at dette er en nasjonal ressurs for streaming med krav til høy kvalitet og kapasitet. Midgard er en nøkkelressurs for "Åpen kanal", et initiativ for å gi allmenheten ikke-kommersiell adgang til fjernsynsmediet. UNINETT er spesielt interessert i Åpen kanal av den grunn at dette initiativet er en utmerket anledning til å prøve ut bruken av multicast i en større sammenheng.



UNINETT ser på Åpen Kanal som en glimrende brekkstang for å tvinge inn åpne og nye standarder, som MPEG4 og multicast, og mener vi her har en glimrende mulighet for å teste og utvikle nye nettjenester.

UNINETT har bredbåndstilknytning utover det meste av Norge, med 2.5Gbps forbindelse Oslo-Trondheim samt Oslo-Stockholm og Oslo-København via NORDUnet samarbeidet. Videre er det minimum 34mbps forbindelser til alle noder som er direkte koblet til nettet, og ofte mer. Internasjonalt har NORDUnet veier inn til GEANT og DANTE, samt tilknytning opp mot Internet2 i USA. Generell internet tilknytning er også godt dekket opp via Norwegian Internet Exchange (NIX1 og NIX2) i Oslo, og tilsvarende enheter rundt om i andre land.

UNINETTs nettverk har full støtte for multicast og er åpent for sperringer, hvilket betyr at streaming over RTP/RTSP, HTTP, UDP også videre ikke er noe problem. Per i dag har vi omtrent 10Mbps med kontinuerlig videostreaming fra vår tjener i Oslo. UNINETT ønsker å få større utbredelse av multicast og QoS teknologier i nettet – og spesielt generelt utover internet, samt at UNINETT vil jobbe mot åpne standarder for internet bruk. Dette har resultert i at vi nå blant annet benytter MPEG4 streaming over RTSP for streaming til egne formål.

Med den generelle lasten og kapasiteten UNINETT har kan vi ikke se noen problemer med å videreformidle Åpen Kanals produksjoner – og nettverket burde uten problemer også håndtere HDTV og tilgang til mediainnhold for digital kinematografi.

UNINETT har omtrent 40 ansatt, hvor av to jobber fulltid med multimedia tjenester og utvikling. I tillegg har vi flere prosjekter ute på universiteten på forskning og testing av multimedia tjenester – noe som har resultert i blant annet ”Radio On Demand”, digital DAB streaming av NRK1, samt multicasting av all radio og TV innhold levert av NRK.

Med vennlig hilsen

Geir Olav Jensen

## Vedlegg 2 – Åpen kanals vedtekter

### 1. Navn og rettslig stilling

Åpen kanal er en selvstendig forening dannet i Oslo, 24. februar 2002. Organisasjonsnr. 887420882.

### 2. Formål

Foreningen Åpen kanal tilrettelegger Norges åpen kanal som det deltakende demokratiet og ytringsfrihetens fjernsynskanal.

### 3. Virkemåte Norges åpen kanal

- I. Alle som ønsker det skal få sendeflate på Norges åpen kanal.
- II. Sendingene skal være ikke-kommersielle. Reklame, snikreklame, produktplassing, pengeinnsamling, sponsoreksponering eller virksomhet som tar sikte på økonomisk gevinst, er utelukket.

### 4. Begrensning

- I. Politiske partier kan ikke delta.
- II. Ingen deltaker får mer enn én halvtimes daglig sending til bestemt tid eller mer enn fem timer i løpet av én måned.

### 5. Organisasjon

Åpen kanals medlemmer er registrerte deltakere. Åpen kanals øverste ledelse er Allmøtet som består av registrerte deltakere. Det velges et styre på syv personer og til sammen tre vararepresentanter.

### 6. Deltakeren

- I. registrerer seg over nettet.
- II. betaler registreringsavgift.
- III. melder seg ut skriftlig til styreleder.
- IV. tar selvstendig redaktøransvar for egne sendinger.

- V. plikter å sette seg grundig inn i redaktøransvaret i henhold til norsk lov, samt vedtektene.
- VI. gjør det tydelig hvem som sender og er redaktør.
- VII. deltar i allmøtet med én stemme.
- VIII. melder inn saker til allmøtet.
- IX. melder framtidig sendebehov på elektronisk skjema.
- X. kan kreve uttalelse i saker som vedgår dem før styret fatter beslutning.

## **7. Allmøtet**

- I. oppdaterer vedtektene med tre fjerdedels flertall unntatt pkt. 2 og 3 som krever enstemmighet.
- II. fastsetter registreringsavgiften.
- III. velger styre og styreleder blant deltakerne, for to år av gangen.
- IV. godkjenner regnskap, budsjett og årsmelding.
- V. vedtar sendeflatestruktur med tre fjerdedels flertall.
- VI. behandler styrets vedtak om utestenginger.
- VII. fastsetter tid og sted for neste allmøte.

## **8. Styret**

- I. melder til deltakerne.
- II. fatter vedtak med minst fire stemmer for.
- III. er underlagt vanlige habilitetsregler.
- IV. opprettholder en prototyp for drift.
- V. veileder om opplasting/avvikling og rettslige forhold.
- VI. oppdaterer hjemmesiden.
- VII. fører regnskap og utarbeider årsmelding.
- VIII. forbereder, minner om og sender ut saksliste, sendebhovskjema, budsjettforsalg m.m. henholdsvis 4 uker og 1 uke før allmøter og styremøter.
- IX. kan innkalle til ekstraordinært allmøte.
- X. kan tilpasse sendeflaten mellom allmøtene så fremt dette lar seg gjøre uten skade for noen.
- XI. kan fram til neste allmøte enstemmig stenge ute et medlem som bevisst har omgått vedtektene.

## **9. Oppløsning**

Oppløsning regnes for vedtektsendring og krever innkalt allmøte. Foreningens midler skal anvendes slik at foreningens formål blir tilgodesett best mulig.

## **Vedlegg 3 - Oversetting av reglement gjeldende for åpne kanaler i Tyskland**

Denne oversettelsen er utført av Åpen kanal uten en autorisert tolk. Oversettelsen er kun til veiledning i det juridiske arbeidet. De tyske åpne kanalene er en mal for realiseringen av Åpen kanal i Norge. Her fordrer de redaksjonelle modellene en reell ytringsrett.

Anmerkning til oversettelsen:

Mediananstalt tilsvarener Norges *Medietilsyn*

Medienrates tilsvarener Kultur(-/Medie-)departementet

### **Staatsvertrag über die Zusammenarbeit zwischen Berlin und Brandenburg im Bereich des Rundfunks (MStV)**

#### **Präambel**

Die Länder Berlin und Brandenburg wollen mit diesem Staatsvertrag die Grundlage für eine gemeinsame Medienordnung schaffen die den engen kulturellen, wirtschaftlichen und gesellschaftlichen Verflechtungen innerhalb der Region Rechnung trägt. Ihr Ziel ist es, ein leistungsfähiges öffentlich-rechtliches und privates Rundfunkwesen zu entwickeln, das den Bürgern der Region ein qualitativ gutes, vielfältiges Programmangebot bietet. Im Bereich des öffentlich-rechtlichen Rundfunks erfordert dies eine Kooperation der beiden Landesrundfunkanstalten.

Die Länder erwarten, daß der SFB und der ORB in ihrem eigenen und in gemeinsam gestalteten Programmen die kulturelle Vielfalt ebenso wie die Gemeinsamkeiten innerhalb der Region widerspiegeln und die Bedeutung der Region als Sitz der deutschen Hauptstadt und Brücke zu Osteuropa durch die Zusammenarbeit auch innerhalb der ARD angemessen darstellen. Durch gegenseitige Absprachen und Kooperationen auch in allen übrigen Bereichen sollen die verfügbaren Ressourcen so effizient wie möglich genutzt werden. Das Zusammenwachsen der Region darf nicht durch den Aufbau unnötiger Doppelstrukturen im Rundfunk behindert werden. Die beiden Länder sind sich in dem Ziel einig, bei der Entwicklung der Region Berlin-Brandenburg als Medienstandort von nationaler und europäischer Bedeutung zusammenzuarbeiten. Beide Länder werden sich um die Erarbeitung eines gemeinsamen, arbeitsteiligen medienwirtschaftlichen Standortkonzepts in Berlin und Brandenburg bemühen. Berlin beteiligt sich am Aufbau Babelsbergs als einem Schwerpunkt der Film- und Fernsehproduktion in der Bundesrepublik Deutschland.

## **Statskontrakt om samarbeid mellom Berlin og Brandenburg i kringkastingsfeltet**

### **Fortale**

Statene Berlin og Brandenburg vil med denne statskontrakt skaffe et fundament for en felles medieordning som tar hensyn til de nære kulturelle økonomiske og samfunnsmessige relasjoner innenfor regionen. Målet med denne er å utvikle et produktivt offentligrettlig og privat kringkastingsvesen som gir borgerne i regionen et kvalitativt godt og mangfoldig programtilbud. Innen feltet til den offentligrettslige kringkastingen fordrer dette samarbeid fra begge statskringkastingsinstitusjoner.

Statene forventer at SFB og ORBs egne og i samarbeid lagete programmer gjenspeiler kulturell mangfold og felles kultur i regionen og på en passende måte representerer betydningen av regionen som tysk hovedstad og bro til Øst-Europa gjennom samarbeid også innenfor ARD. Gjennom gjensidige avtaler og samarbeidsområder også i alle øvrige områder skal de tilgjengelige ressurser bli utnyttet så effektivt som mulig. Samkjøringen av regionen må ikke bli hindret av oppbyggingen av dobbeltstrukturer i kringkasting. Begge statene er enige i målet om å samarbeide om utviklingen av regionen Berlin-Brandenburg som mediested av nasjonal og europeisk betydning. Begge stater vil arbeide for en felles arbeidsdelende medieøkonomisk stedskonsept i Berlin og Brandenburg. Berlin deltar i oppbygging av Babelsbergs som en betydelig del av film- og fjernsynsproduksjon i forbundsrepublikken Tyskland.

## **Erster Abschnitt**

### **Allgemeine Vorschriften**

#### § 1 Anwendungsbereich

(1) Dieser Staatsvertrag regelt

1. die Zuordnung von Übertragungsmöglichkeiten für den öffentlich-rechtlichen und den privaten Rundfunk,
2. die Veranstaltung von Rundfunk durch private Veranstalter,
3. die Verbreitung von Rundfunkprogrammen und Mediendiensten,
4. offene Kanäle,
5. die Entwicklung und Nutzung der durch neue Techniken und neue Nutzungsformen eröffneten weiteren Möglichkeiten für die Veranstaltung und Verbreitung von Rundfunk und Mediendiensten.

#### § 14 Aufgaben des Direktors

(1) Der Direktor vertritt die Medienanstalt gerichtlich und außergerichtlich; er führt die laufenden Geschäfte der Medienanstalt, bereitet die Entscheidungen des Medienrates vor und vollzieht dessen Beschlüsse.

(2) Über die Einstellung und Entlassung der Bediensteten der Medienanstalt entscheidet der Direktor, bei Referenten und bei Leitern eines offenen Kanals mit Zustimmung des Medienrates.

#### § 40 Pflichten der Kabelanlagenbetreiber

(1) Kabelanlagen sind so zu errichten und zu betreiben, daß

1. allen Veranstaltern von Rundfunk Zugang zu chancengleichen, angemessenen und diskriminierungsfreien Bedingungen und Entgelten ermöglicht wird,
2. sie dem Bedarf und dem Stand der Übertragungstechnik entsprechen und
3. Entscheidungen über die Belegung der Kanäle unter Berücksichtigung der regionalen oder lokalen Besonderheiten der jeweiligen Kabelanlage getroffen werden können.

## **Første avsnitt**

### **Allmenne forskrifter**

#### § 1 Anvendelsesområde

(1) Denne statskontrakt bestemmer

1. tilordning av overføringsmuligheter for den offentlig-rettslig og privat kringkasting.

2. Møter vedrørende kringkasting gjennom private organiseringer.

3. Spredning av kringkastningsprogrammer og medietjenester.

4. åpne kanaler

5. utviklingen og benyttelse av nye teknikker og nye benyttelsesmåter åpner nye muligheter for utbredelse av kringkasting og medietjenester

#### § 14 Direktørens oppgaver

(1) Direktøren representerer Medietilsynet rettslig og utenomrettslig; han leder løpende forretninger av [Medietilsynet], forbereder avgjørelser av [Mediedepartementet] og fullfører disse avgjørelsene.

(2) Direktøren bestemmer over ansettelse og avskjedelser når det gjelder [Medietilsynets] personal, ved referendum og av ledere til en åpen kanal må i tillegg [Mediedepartementet] samtykke.

#### § 40 Plikter for de som driver kabelanlegget

(1) Kabelopplegg må opprettes og drives slik at,

1. samtlige operatører av kringkasting har like muligheter for tilgang, slik at det finnes vilkår fri for diskriminering

2. disse tilsvarer behovet og nivået for overføringsteknikk

3. Bestemmelser over bruk av kanalene kan bli gjort med hensyn til regionale og lokale særpreg



(2) Unter Berücksichtigung der technischen und wirtschaftlichen Entwicklung kann der Betreiber einer Kabelanlage verpflichtet werden, die digitale Verbreitung regionaler oder lokaler Programme durch Bereitstellen der erforderlichen technischen Dienstleistungen zu gewährleisten. Die Medienanstalt kann den Betreiber weiter verpflichten, unter Berücksichtigung der Wirtschaftlichkeit der Kabelanlage Kapazitäten für innovative Anwendungsformen zur Verfügung zu stellen.

(3) Der Betreiber einer Kabelanlage mit mehr als 15 Fernsehkanälen, an die mehr als 50000 Haushalte angeschlossen sind, kann durch Beschluß des Medienrates verpflichtet werden, einen Fernsehkanal unentgeltlich für die Nutzung als offenen Kanal zur Verfügung zu stellen; entsprechendes gilt für die Nutzung eines Hörfunkkanals, wenn in der Kabelanlage mehr als 20 Hörfunkkanäle genutzt werden können. Die Medienanstalt wird ermächtigt, nähere Einzelheiten durch Satzung zu regeln.

(4) Auf Beschluß des Medienrates kann unter Berücksichtigung der Vielfalt des Programmangebotes und der Nachfrage nach Übertragungskapazitäten ein Teil der Übertragungskapazität in Kabelanlagen, an die mehr als 50000 Haushalte angeschlossen sind, für die zeitlich aufgeteilte Nutzung durch voneinander unabhängige Veranstalter zur Verfügung gestellt werden (Mischkanäle). Der Medienrat kann für Zwecke des Mischkanals und des offenen Kanals einen Kanal zur gemeinsamen Nutzung vorsehen.

#### § 41a Rechtsvorschriften für digitalisierte Kabelanlagen

(1) Soweit der Betreiber von digitalisierten Kabelanlagen Fernsehprogramme oder Mediendienste verbreitet, hat er sicherzustellen, dass

1. die erforderlichen Übertragungskapazitäten für die für Berlin und Brandenburg gesetzlich bestimmten Fernsehprogramme des öffentlich-rechtlichen Rundfunks einschließlich seiner Programmbouquets zur Verfügung stehen,

2. die Übertragungskapazität eines analogen Fernsehkanals für die in Berlin oder Brandenburg zugelassenen regionalen und lokalen Fernsehprogramme sowie dem Offenen Kanal in Berlin zur Verfügung steht; soweit diese Übertragungskapazität danach nicht ausgeschöpft ist, richtet sich die Belegung nach §§ 41 und 42; § 40 Abs. 3 und 4 bleibt unberührt,

3. die technischen Übertragungskapazitäten nach Nummern 1 und 2 im Verhältnis zu anderen digitalen Kanälen technisch gleichwertig sind,

(2) med hensyn til den tekniske og økonomisk utviklingen kan den som driver kabelanlegget pålegges å sikre den digitale utbredelsen av regionale og lokale programmer gjennom nødvendig tekniske tjenester.

(3) Den som driver et kabelanlegg med flere enn 15 tv-kanaler, som er tilkoblet flere enn 50000 husholdninger, kan gjennom beslutning av [Mediedepartementet] bli forpliktet til å stille en tv-kanal til disposisjon uten vederlag; tilsvarende gjelder for benyttelse av en radiokanal, dersom kabelanlegget gjør flere enn 20 radiokanaler tilgjengelig. [Medietilsynet] har rett til å regulere nærmere detaljer etter statutter.

(4) På bestemmelse av [Mediedepartementet] kan under hensyntaken av mangfoldet i programtilbudet og etterspørsel etter overføringskapasitet en del av overføringskapasiteten i kabelanlegget som er tilkoblet flere enn 50000 husholdninger for tidsmessig fordelt benyttelse av seg imellom uavhengige operatører stilt til disposisjon (blandingskanaler). [Mediedepartementet] kan stille en kanal til disposisjon for blandingskanaler og åpen kanal til felles benyttelse.

#### § 41a Rettsforskrifter for digitaliserte kabelanlegg

(1) I den graden de som driver digitaliserte kabelanlegg utbrer fjernsynsprogrammer eller medietjenester, bør de sikre at,

1. den nødvendige overføringskapasiteten for de for Berlin og Brandenburg lovlig bestemte fjernsynsprogrammer av offentligrettslig kringkasting inklusive deres programpalett står til disposisjon

2. overføringskapasiteten til analoge tv-kanaler for de i Berlin eller Brandenburg tillatte regionale og lokale tv-programmer samt åpen kanal i Berlin er disponibel; så vidt denne overføringskapasiteten deretter ikke er benyttet?, retter seg belegg etter §§ 41 og 42; § 43 Abs. 3 og 4 blir uberørt,

3. den tekniske overføringskapasiteten etter nummer 1 og 2 er teknisk likeverdig i forhold til andre digitale kanaler,

4. Entgelte und Tarife für die Programme nach Nummern 1 und 2 offen gelegt werden; Entgelte und Tarife sind im Rahmen des Telekommunikationsgesetzes so zu gestalten, dass auch regionale und lokale Angebote zu angemessenen und chancengleichen Bedingungen verbreitet werden können; § 40 Abs. 3 und 5 bleibt unberührt.

#### § 43 Offene Kanäle

(1) Offene Kanäle geben ihren Nutzern Gelegenheit zur Darstellung ihrer Anliegen und Meinungen durch selbstgestaltete Beiträge. Die Medienanstalt schafft die organisatorischen und personellen Voraussetzungen der offenen Kanäle. Die Medienanstalt berät die Nutzer in organisatorischer Hinsicht und stellt Produktionsmittel zur Verfügung.

(2) Auf Beschluß des Medienrates wird eine UKW-Frequenz ganz oder teilweise für die Nutzung durch den Offenen Kanal vorgesehen, wenn die Kapazitätssituation dies erlaubt und die Kosten aus den dem Offenen Kanal zur Verfügung gestellten Mitteln übernommen werden.

(3) Ein offener Kanal darf nicht zur Erzielung von Einnahmen benutzt werden. Werbung ist ausgeschlossen.

(4) Der Zugang zu den offenen Kanälen wird von der Medienanstalt durch eine vom Medienrat zu erlassende Satzung geregelt, die insbesondere die Chancengleichheit des Zugangs und der Nutzung gewährleistet sowie das Verfahren und die Sanktionen bei Mißbrauch regelt. Für Beiträge, die sich durch gemeinsame Merkmale auszeichnen, kann die Zusammenfassung von Sendezeiten vorgesehen werden. Die Satzung kann bestimmen, daß das Nutzungsverhältnis privatrechtlich ausgestaltet wird. Die Satzung ist zu veröffentlichen.

(5) Die Verantwortung für die Beiträge im offenen Kanal obliegt ausschließlich dem jeweiligen Nutzer. Er trägt dafür Sorge, daß seine Beiträge Rechte Dritter, insbesondere urheberrechtlicher Art, nicht verletzen. Sämtliche mit dem Programm zusammenhängenden Ansprüche, insbesondere Ansprüche auf Unterlassung, Widerruf, Schadensersatz oder Gegendarstellung, sind gegen den Nutzer geltend zu machen. Die Medienanstalt gewährleistet die Verbreitung der Gegendarstellung.

(6) Im übrigen gelten die Regelungen für die Veranstaltung von Rundfunk entsprechend.

4. Vederlag og tariffer for programmene etter nummer 1 og 2 blir tilgjengelig; vederlag og tariffer bør utformes innenfor rammene av telekommunikasjonslovverket, slik at også regionale og lokale tilbud

kan bli sendt ut under tilpassete og like betingelser; § 40 Abs. 3 og 5 forblir urørt.

#### § 43 Åpne kanaler

(1) Åpne kanaler gir deres brukere anledning til presentasjon av deres meninger gjennom egenproduserte bidrag. [Medietilsynet] skaper organisatoriske og personellmessige forutsetninger for de åpne kanaler. [Medietilsynet] gir råd til brukerne mht organisatoriske spørsmål og stiller produksjonsmidler til disposisjon.

(2) Etter bestemmelse fra [Mediedepartementet] blir en ultrakortbølgefrequens helt eller delvis gitt til benyttelse av åpen kanal, dersom kapasitetssituasjonen tillater dette og kostnadene kan bli tatt fra midlene til åpen kanal.

(3) En åpen kanal kan ikke bli benyttet for inntektsformål. Reklame er utelukket.

(4) Tilgang til åpen kanal blir regulert gjennom et reglement??? av [Medietilsynet].

Reglementet som bestemmes av [Mediedepartementet] skal sikre like muligheter for tilgang og bruk, samt prosedyre og sanksjoner ved misbruk. Bidrag som kjennetegnes av felles egenskaper, kan sammenfall av sendetid bli vurdert. Reglementet kan bestemme at benyttelsesgrad blir reglet privatrettslig. Reglementet må offentliggjøres.

(5) Ansvar for bidragene i åpen kanal tilligger utelukkende den respektive bruker. Brukeren sørger for at bidragene ikke bryter med rettigheter til tredjeparter, ikke minst copyright. Samtlige evt. krav ift skadeerstatning eller kontrainnlegg, må rettes til bruker. [Medietilsynet] tillater offentliggjøringen av et kontrainnlegg.

(6) I det øvrige gjelder respektive reglementer for bruk av kringkasting