

Complex nr. 1/2009

Odd Randgaard Kleiva

ÅPEN PROGRAMVARE

-noen rettslige problemstillinger

Senter for rettsinformatikk/Avdeling for forvaltningsinformatikk
Postboks 6706 St Olavs plass
0130 Oslo

Henveldeiser om denne bok kan gjøres til:

Senter for rettsinformatikk
Postboks 6706 St. Olavs plass
0130 Oslo
Tlf. 22 85 01 01
www.jus.uio.no/iri/

ISBN 978-82-7226-120-6
ISSN 0806-1912

Utgitt i samarbeid med Unipub AS
Trykk: e-dit AiT AS
Omslagsdesign Kitty Ensby

FORORD

Denne boken er et resultat av masteroppgaven jeg skrev høsten 2007 og våren 2008 mens jeg arbeidet som vitenskapelig assistent ved Senter for rettsinformatikk, som er en del av Institutt for privatrett, ved Universitetet i Oslo.

Boken er skrevet primært for jurister, men kan også ha en viss interesse for lesere med informatikkbakgrunn eller andre med interesse for åpen programvare.

Boken er forsøkt å være ajourført per mai 2008. Jeg har prøvd å fremstille gjeldende rett så korrekt som mulig, i den grad det går an å bruke ord som «korrekt» på et felt hvor det til tider er så sparsomt med rettskilder som her. Allikevel er det nærmest uunngåelig at en tekst av denne lengden inneholder feil eller utelatelser. Hvis leseren skulle finne slike setter jeg pris på en e-post.

Jeg vil gjerne takke alle de som har vært involvert i oppgaven min. Jeg vil spesielt takke min inspirerende og innsiktsfulle veileder professor Jon Bing for gode kommentarer og innspill. Jeg vil også takke sensorene mine, professor Olav Torvund og Terje Lundby Michaelsen, for nyttige og kritiske kommentarer. Jeg vil gjerne takke professor Dag Wiese Schartum for mange morsomme arbeidsoppgaver utenom skrivingen og bibliotekarene Anne Gunn Bekken og Frøydis Asklund for å ha hjulpet meg med å finne kilder. Takk for-tjener også Helge Sønneland for samtaler om opphavsrett. Jeg vil også takke Gro Halvorsen og alle dere andre ved Senter for rettsinformatikk og Avdeling for forvaltningsinformatikk som har gjort oppgaveskriving til en lærerik og hyggelig opplevelse. Utenfor det juridiske fakultet vil jeg takke professor Stein Krogdahl for å ha hjulpet meg med tekniske spørsmål og Thomas Gramstad for å ha satt meg i kontakt med åpen kildekodeentusiaster. Til sist vil jeg takke venner og familie for å ha støttet meg under skrivingen.

Boken er også tilgjengelig på <http://fpbok.kleiva.com>.

Odd Randgaard Kleiva

INNHold

1	Innledning.....	9
2	Introduksjon til åpen kildekode.....	11
2.1	En kort introduksjon til programmer og kildekode.....	11
2.1.1	<i>Fra idé til program</i>	11
2.1.2	<i>Noen perspektiver</i>	12
2.2	Historien til åpen kildekode	13
2.2.1	<i>I begynnelsen... - teknologien, programvareindustrien og delekulturen</i>	14
2.2.2	<i>Opphavsrett</i>	15
2.2.3	<i>Åpen kildekode – en motreaksjon?</i>	17
2.2.4	<i>I dag</i>	20
3	Rettslig beskyttelse av åpen programvare	23
3.1	Introduksjon til opphavsrett	23
3.2	Når får en opphavsrett?	24
3.3	Hva innebærer vernet?	25
3.3.1	<i>Økonomiske interesser</i>	25
3.3.2	<i>Ideelle rettigheter og deres økte betydning i åpen kildekodeprosjekter</i>	27
3.4	Lisenser og overføring av opphavsrett.....	28
3.5	Begrunnelsen for opphavsrett og forholdet til åpen kildekode.....	29
3.6	Kravet til frembringelse.....	30
3.7	Verkets type – Hva er et «datamaskinprogram»?	30
3.8	Kravet til originalitet.....	31
3.8.1	<i>Særlig om vern av programmets «dypstruktur»</i>	35
3.9	Subjektet – hvem er «opphavsmannen»?.....	36
3.9.1	<i>Ansettelsesforhold</i>	36
3.9.2	<i>Konsulenter og oppdragsforhold</i>	36
3.10	Fellesverk	37
3.10.1	<i>(U)selvstendige verk</i>	37
3.10.2	<i>Samarbeid</i>	39
3.11	Sammensatte verk	40
3.12	Samleverk	40
3.13	Avledede verk (endringer og bearbeidelser) og nye selvstendige verk	41
3.13.1	<i>Skillet mellom avledede verk og nye selvstendige verk</i>	42
3.13.2	<i>Avledede verk ved relisensiering av programmer</i>	42

3.14	Særlig om forholdet mellom kildekode og programmet på objektform	43
3.15	Særlig om kombinasjon av moduler og bruk av grensesnitt	43
3.15.1	<i>Dynamisk linking</i>	46
3.16	Friere enn fri? «The public domain»	47
4	Åpen kildekode lisenser vs. proprietære lisenser	49
4.1	Kjennetegn for proprietære lisenser	49
4.2	Mylderet av lisenser	49
4.3	Kjennetegn for åpen kildekode lisenser	50
4.3.1	<i>Tilgang til kildekode</i>	50
4.3.2	<i>Mulighet for endring</i>	51
4.3.3	<i>Liberal rett til eksemplar fremstilling og tilgjengeliggjøring</i>	51
4.3.4	<i>Liberal rett til bruk</i>	52
4.3.5	<i>Gratis?</i>	52
4.3.6	<i>Forgreninger - «forking»</i>	53
5	Lovvalg og jurisdiksjon	55
5.1	Jurisdiksjon	55
5.2	Lovvalg	56
5.2.1	<i>Lovvalg for krav basert på opphavsrett (opphavsrettsstatuttet)</i>	57
5.2.2	<i>Lovvalg for krav basert på kontrakt (kontraktsrettsstatuttet)</i>	58
5.2.3	<i>Lovvalg for erstatningskrav</i>	59
5.3	Risiko	59
6	Tolkning av åpen kildekode avtaler	61
6.1	Generelt om tolkning av åpen kildekode lisenser	61
6.2	For- og etterarbeider til avtalene	63
6.3	Bransje praksis	63
7	Copyleft	65
7.1	Introduksjon	65
7.2	Svak copyleft	66
7.3	Sterk copyleft etter GPL	67
7.3.1	<i>Hva blir omfattet av copyleft etter GPL?</i>	67
7.3.2	<i>«any work that... is derived from the Program»</i>	67
7.3.3	<i>«any work that ... in whole or in part contains ... the Program»</i>	68
7.3.4	<i>Avgrensning av copyleft mot tilgjengeliggjorte programmer</i>	74
7.3.5	<i>Programvare som tjeneste (SaaS) - «The ASP loophole»</i>	76
7.3.6	<i>Diff-er og patch-er</i>	77
7.3.7	<i>GPLv3</i>	80
7.4	<i>Copyleft's avhengighet av opphavsretten</i>	83
7.5	<i>Hybridlisensiering</i>	83

7.5.1	Dobbel- og multi-lisensiering.....	83
7.5.2	«Proff»-versjoner og proprietær deler	84
7.5.3	Forholdet til bidragsyttere ved dobbel lisensiering	84
8	Nærmere om ulike former for lisensiering	87
8.1	Sublisensiering	87
8.1.1	Sublisensiering i åpen kildekodeavtaler (MIT-lisensen)	89
8.2	Direkte nedstrømslisenser	89
8.2.1	GPLs nedstrømslisensbestemmelse	90
8.2.2	Kontraktinggåelsen ved nedstrømslisenser.....	91
8.2.3	Direkte nedstrømslisenser i BSD-lisensen	92
8.2.4	Konsekvensene av direkte nedstrømslisenser	93
8.3	Lisensmodellens betydning for opphavsrettslige krav	93
9	Nærmere om skillet mellom opphavsrettslig og kontraktsrettslig virkende vilkår i lisensene	95
9.1	Copyleft.....	96
9.2	Opphørsklausuler	96
10	Vanhjemmel og sanksjoner	99
10.1	Introduksjon	99
10.1.1	Forskjellige typer vanhjemmel i kontrakt.....	99
10.1.2	Vindikasjon og ekstinksjon ved overføring av opphavsrett	100
10.2	Årsaker til vanhjemmel	102
10.2.1	Privat reguleringsvillfarelse.....	102
10.2.2	Rettsvillfarelse og bevisste lovbrudd	103
10.2.3	Direkte kopi og plagiat.....	103
10.2.4	Dobbeltsuksesjon	104
10.2.5	Selvplagiat	105
10.3	Sanksjoner ved vanhjemmel	106
10.4	Erstatning og restitusjon av berikelse	106
10.5	Krav basert på krenkelse av opphavsretten	107
10.5.1	Erstatningskrav	107
10.5.2	Krav basert på urettmessig berikelse.....	110
10.5.3	Bruk av programmer etter oppdaget vanhjemmel.....	112
10.6	Krav basert på kontrakt	113
10.6.1	De lege ferenda – Hvordan bør ansvaret mellom lisensgiver og lisenstaker bli fordelt?	113
10.6.2	Hva tilsier bakgrunnsretten?.....	114
10.6.3	Hva tilsier avtalene?	117
10.6.4	Kan ansvarsfraskrivelsene opprettholdes? - Sensur	119
10.6.5	Konklusjon.....	122

11 Litteraturliste	123
12 Dommer.....	131
13 Forarbeider	133
14 Forkortelser	135
15 TIDLIGERE UTGITT I COMPLEX-SERIEN	137

1 INNLEDNING

De første åpen kildekode-lisensene kom for nesten 20 år siden. Fra sin spede begynnelse har konseptet vokst frem til å totalt forandre måten mange tenker på programvare på.

En kan jo lure på hvorfor dette temaet er interessant å skrive om nesten 20 år senere? Åpen kildekode-lisenser er fremdeles lite utforsket i norsk rett, og i løpet av disse årene har det skjedd en del forandringer. Ikke minst har bruken av åpen kildekode vokst i omfang, men det er også kommet flere nye typer av standardlisenser. Åpen kildekode og de tilhørende lisensene er interessante å utforske juridisk både fordi de har en så sentral plass i IT-industrien, og dermed i samfunnet vårt, og fordi lisensene strekker opphavsretten til det ytterste. Ekstreme samarbeidsprosjekter om åndsverk, som åpen kildekodeprosjekter ofte er, er det relativt få eksempler på ellers.

Åpen kildekode kan defineres som et sett med prinsipper og praksis for å skrive programvare og for å tilgjengeliggjøre den, hvor det viktigste prinsippet er at kildekoden er åpent tilgjengelig for liberal utnyttelse. Denne boken vil dreie seg om en liten del av åpen kildekodeuniverset. Den vil dreie seg om en del av det juridiske rammeverket for disse prosjektene, lisensene.

Når jeg skriver denne innledningen har Open Source¹ Initiative registrert ca. 70 forskjellige åpen kildekode-lisenser². Det finnes i tillegg mange andre relevante lisenser som enten noen ikke har søkt om å få registrere eller som ikke har sluppet gjennom. Lisensene reiser en rekke juridiske spørsmål, både avtalerettslige, obligasjonsrettslige, erstatningsrettslige, prosessrettslig, opphavsrettslige (inkludert flere tilgrensende rettigheter som databasevern og vern av tekniske beskyttelsessystemer), patentrettslige... Listen går videre. Jeg må derfor gjøre et utvalg. Jeg vil i denne boken fokusere på opphavsrettslig vanhjemmel, og de mest praktiske konsekvensene av denne, dvs. erstatning og restitusjon av urettmessig berikelse. Siden åpen kildekodeprogrammer har en så global natur vil jeg også kort omtale noen spørsmål i forhold til internasjonal privatrett.

Jeg vil gå relativt grundig inn på en del av de grunnleggende opphavsrettslige problemstillingene. Forhåpentligvis følger de mer avanserte problemene lettere når grunnlaget er på plass. Siden mange av problemene i denne oppgaven er relativt lite utforsket av lovgiver og i rettspraksis, innebærer det, som alltid i slike situasjoner, at konklusjonene blir mer usikre, og, når det gjelder de

1 Dvs. «åpen kildekode»

2 Se «Licenses by Name»

grunnleggende problemstillingene, vil dette kunne forplante seg. Det er derfor særlig viktig å se nøye på de grunnleggende prinsippene.

For å sette åpen kildekode-lisensen i perspektiv har jeg også tatt med et kapittel om historien til åpen programvare.

All *kursivering* av sitater i denne boken er min egen, med mindre annet er sagt.

I denne boken henviser jeg til en del lisenser. Disse er lett tilgjengelig på Internett og er derfor ikke tatt med. <http://www.opensource.org/licenses> anbefales.

2 INTRODUKSJON TIL ÅPEN KILDEKODELISENSER

2.1 En kort introduksjon til programmer og kildekode

For å forstå en del av de juridiske problemstillingene jeg skal skrive om, må en ha en grunnleggende forståelse av teknologien som ligger bak dataprogrammer, hva programmer er og hvordan programmer blir til. Her prøver jeg å gi en enkel oversikt over noen grunnleggende begreper og konsepter. Jeg vil komme tilbake til disse og forklare de nærmere der det er nødvendig. For at dette skal blir forståelig for folk uten databakgrunn må jeg forenkle det litt.

2.1.1 Fra idé til program

Utviklingen av programmer starter vanligvis med en idé. Bortsett fra for trivielle programmer, gjennomfører utviklerne en planleggingsfase. I denne spesifiserer en hva programmet skal gjøre, og hvordan det skal gjøre det. Det blir gjerne utarbeidet spesifikasjoner og modeller.

Når en har planlagt tilstrekkelig, kan en gjennomføre planene, programmet implementeres. Programmer skrives i programmeringsspråk, som Java, C, C++, C#, Perl, Ruby etc. Programmeringsspråkene er kunstige språk som blir bruk for å kontrollere oppførselen til en datamaskin. Programmeringsspråk består, akkurat som naturlige språk som mennesker primært bruker, av syntaktiske og semantiske regler. Reglene for programmeringsspråk er imidlertid mye fastere enn reglene for naturlige språk. Dette sikrer presisjon og entydighet. Det programmereren skriver (koder) i programmeringsspråket kalles kildekode, eller bare kode.

Datamaskinen «forstår» ikke programmeringsspråk direkte. Derfor må kildekoden enten gjøres om (kompileres) til maskinspråk før programmet kjøres eller så må et program (en interpreter) tolke programmet for maskinen mens det kjøres.

Kompilatoren er et program som gjør kildekode om til maskinspråk.³ Resultatet er kode på objektform, også kalt objekter eller maskinkode. Ofte vil det også trenges noen moduler, som gir tilleggsfunksjonalitet til dette

3 En kompilator defineres ofte videre. Den gjør om et kildespråk til et målspråk, men denne nyansen har vi ikke bruk for her.

(hoved)programmet. Disse modulene kan være i form av objekter eller bibliotek (som kan sees på som en samling av objekter). En «linker» vil så sette sammen det kompilerte programmet med disse andre objektene til en fil. Denne sammensettingsprosessen kalles statisk linking. Den skjer som sagt rett etter kompileringen.⁴

Dynamisk linking er en annen form for linking. Ved dynamisk linking linkes programmet med biblioteker i det programmet skal kjøres eller når programmet trenger biblioteket⁵. Bibliotekene er vanligvis i egne filer. Denne teknikken har den fordelen at flere programmer kan dele bibliotek, og bibliotekene kan oppdateres uten at en trenger å oppdatere programmet.

Det finnes også mellomting mellom tolkning og kompilering. Kildekoden kan kompileres til bytekode, som så blir tolket av en interpreter eller blir omgjort til maskinspråk når programmet kjøres (just-in-time kompilering). Denne siste metoden er vanlig for Java-programmer, som er mye brukt for tiden.

Programmeringsspråkene, som Java og C, har et høyt abstraksjonsnivå, dvs. instruksjonene i disse språkene ligger relativt langt unna de instruksjonene maskinen utfører. Dette gjør det enkelt for mennesker å bruke programmeringsspråkene. Disse språkene kalles gjerne høynivåspråk. Det finnes også språk som ligger nært opptil de instruksjonene maskinen utfører, disse kalles lavnivåspråk. Maskinspråk (maskinens faktiske språk) og de forskjellige typene assembly er eksempler på lavnivåspråk. Assembly-språkene er svært tett knyttet opp til de enkelte maskinens arkitektur, men de er allikevel enklere for mennesker å forstå enn maskinens faktiske språk. Assembly-en må bli gjort om til maskinspråk for at maskinen skal skjønne det. Det gjør en assembler.

For å håndtere kildekode brukes gjerne spesialprogrammer som utviklingsmiljø (IDE)⁶ og versjonskontrollsystemer. Utviklingsmiljøet gjør det lett å arbeide med koden og å navigere i den. Versjonskontrollsystemet gjør at mange programmerere kan samarbeide om utviklingen av programmet; det holder styr på flere versjoner av programmet og på hvem som har gjort hva etc.

2.1.2 Noen perspektiver

Veldig mange av programmene som blir utviklet i dag benytter seg av biblioteker som «leverer» funksjonalitet til programmene, f. eks. for å sortere en datastruktur, dekode video på en DVD (MPEG-2) etc. Å bygge på biblioteker gjør gjenbruk mulig, noe som er svært viktig for å spare tid og penger. Mange viktige biblioteker er lisensiert under åpen kildekode lisenser. Ifølge Gartner

4 Altså ved compile-time

5 Altså ved run-time

6 Integrated development environment: Et program for utvikling av programmer.

vil minst 80 % av proprietær programvare innen 2011 inneholde en betydelig mengde åpen kildekode.⁷ Evnen til effektivt og juridisk sikkert å kunne utnytte slike komponenter vil kunne være en viktig konkurransefaktor for leverandører av programvare.

Å lage store programmer og systemer er en relativt komplisert affære som kan representere store verdier og kan involvere mange deltakere. En uraffinert, men praktisk måte å illustrere dette på er å se på antall linjer kildekode.

Debian GNU⁸/Linux er en av de største distribusjonene av operativsystemet GNU/Linux (ofte bare kalt Linux). En slik distribusjon består av Linux kernelen (kjernen), GNU operativsystemkomponenter og et stort antall programvarepakker, alt fra tekstbehandling til webservere. GNU/Linux er som kjent åpen kildekode. Debian v3.1 hadde ca 15.300 pakker med ferdig programvare som var åpen kildekode.⁹ Totalt bestod distribusjonen av ca 230 millioner linje kode.¹⁰ Debian distribusjonen ble da vedlikeholdt av ca 1.400 frivillige over hele verden.¹¹ Bak disse frivillige stod det et hopetall av andre som har utviklet og vedlikeholder de forskjellige prosjektene som er blitt gjort til pakker.

Åpne programmer begynner i dag å bli utbredt også blant den vanlige forbruker. Noen populære eksempler som kan nevnes er Firefox (webleser), Thunderbird (e-postleser) og OpenOffice (kontorprogrammer). Et annet eksempel er operativsystemet Mac OS X, som er bygget ved hjelp av mye åpen kildekode, men som også har mange proprietære komponenter.¹²

2.2 Historien til åpen kildekode

Mange som skriver historien til åpen kildekode begynner med Richard Stallmann og GNU i 1983, men for å få det fulle bildet av opphavet til åpen kildekode må en ta et skritt tilbake og se på hva som skjedde, særlig i USA, før den tiden. Nedenfor vil jeg i korte trekk fortelle historien til utviklingen av teknologien, datasamfunnet og jussen rundt åpen kildekode.

⁷ Se Brodtkin, 2007

⁸ GNU er et rekursivt akronym for Gnu's Not Unix. G-en står altså for GNU.

⁹ Se «Measuring Libre Software», s. 14

¹⁰ Se «Measuring Libre Software», s. 14

¹¹ Se «Measuring Libre Software», s. 13

¹² Se «What is Mac OS X?»

2.2.1 I begynnelsen... - teknologien, programvareindustrien og delekulturen

I begynnelsen, som for denne historien er rund 50-tallet, var dataindustriens fokus på hardware. Software hadde lav prioritet. Bill Atkinson¹³, som ble ansatt av Apple i 1978¹⁴, beskrev det slik: «They hired me as the application software *department*. They saw software sort of as a necessary evil to sell the box, which was really what they were building»¹⁵. Siden dataindustrien var så lite rettet mot software i begynnelsen var det ingen til å drive frem den rettslige utviklingen av opphavsrett for programvare og en delekultur for programmer kunne vokse frem.

I tillegg skjedde mye av den banebrytende utviklingen ved universiteter og forskningsmiljøer, der nettopp det å fritt dele kunnskap var og er et sentralt ideal.

På 60-tallet skiftet en over fra radiorør til transistorer i datamaskiner. Datamaskiner begynte å bli tilgjengelige for en videre gruppe av spesielt interesserte ved universiteter, forskningsinstitutter o.l. Den typiske brukeren hadde altså spesialkompetanse. Dette muliggjorde et teknisk samarbeid mellom mange brukere og produsenter.

ARPA¹⁶ begynte arbeidet på ARPAnet, en forgjenger til dagens Internett. ARPAnet ble åpnet i 1969. ARPAnet og Internett skulle vise seg å bli essensielt for åpen kildekode slik vi kjenner det i dag. ARPAnet var det første transkontinentale høyhastighetsnettet. Det brakte sammen forskere og datainteresserte over hele verden ved universiteter, forskningsinstitutter og leverandørbedrifter til forsvaret. Programmererne gikk fra å være nokså geografisk isolerte grupper til å ha en unik samarbeidsplattform på ARPAnet.¹⁷ Selv om ARPAnet kun var tilgjengelig for en relativt begrenset gruppe, så en begynnelsen av fremveksten av en global kultur for deling av programvare.

Operativsystemet UNIX fra Bell Laboratories (senere AT&T Bell Laboratories) ble mye brukt utover på 70-tallet. De tidligste utgavene av operativsystemet UNIX fra 70-tallet inkluderte kildekode, noe som gjorde at forskere ved flere av universitetene kunne endre og videreutvikle operativsystemet. UNIX vil vise seg å bli en modell bl.a. for Linux og BSD-familien av operativsystemer som vi kommer til senere.

Høynivåprogrammeringsspråk, som var vokst frem, gjorde det enklere å portere programmer fra én datamaskinarkitektur til én annen, fordi en slapp å oversette hele programmer fra et assemblerspråk til et annet. Dette gjorde

13 Som har laget programmer som HyperCard og MacPaint

14 Se Moggridge, 2007, s. 85.

15 Se Moggridge, 2007, intervju med Atkinson på vedlagt DVD

16 Advanced Research Projects Agency, underlagt USA sitt forsvarsdepartement

17 Se Raymond, 1999, s. 20

det enklere å dele programvare. Programmeringsspråket C ble laget i 1972, av Dennis Ritchie and Brian Kernighan hos AT&T¹⁸. Den første kjernen (kernelen) til operativsystemet UNIX skrevet i C kom året etter¹⁹. I løpet av de neste årene ble hele UNIX miljøer portert til mange forskjellige plattformer.

Frem til tidlig 70-tall hadde datamaskiner vanligvis vært store kolosser som ble brukt av spesialiserte teknikere. Mikroprosessoren fikk sitt gjennombrudd rundt midten av 70-tallet og i løpet av 70-tallet ble datamaskiner stadig mer tilgjengelige. Fra midten av 70-tallet vokste BBS-ene²⁰ frem og datainteresserte også utenfor ARPAnet kunne enkelt dele digitale filer.

Det mange betegner som programvareindustriens fødsel kom i 1969 når IBM skilte salget av programvare og maskinvare (unbundling).²¹ Når de «små» mikroprosessorbaserte datamaskinene (PC-ene²²) begynte å bli markedsført fra midten av 70-tallet, vokste det frem et stort marked for programvare.

På dette tidspunktet var det som sagt en utbredt delekultur for programvare, både over nettverk og ved bruk av fysiske lagringsmedier. Som Bill Gates skrev i 1976 var manges innstilling «Hardware must be paid for, but software is something to share.» Sitatet er hentet fra det berømte «Open Letter to Hobbyists» i Homebrew Computer Club Newsletter.

Delekulturen ble etter hvert en fare for inntektene til den voksende programvareindustrien. Konfidensialitetsavtaler ble brukt for å bøte på dette, men de var ikke tilstrekkelige. Industrien rettet derfor oppmerksomheten mot lovgivningen for å sikre seg inntekter ved lisensiering av programvare. Som en respons til dette vokste opphavsrettsbeskyttelsen til dataprogrammer frem.

2.2.2 Opphavsrett

Det var først ved en lovendring i 1980 det ble klart at programmer var opphavsrettsbeskyttet i USA. I perioden før det var det en del usikkerhet om hvorvidt maskinlesbare programmer var beskyttet mot eksemplarframstilling siden

18 Se «Computer programming language :: C -- Britannica Online Encyclopedia»

19 Se «C Programming Language History»

20 Bulletin board system, også kjent som tavlesystemer.

21 Se «IBM Archives: 1969», Grad og Johnson, 2002 og Bing, 2006

Et antitrust søksmål mot IBM var antakelig en viktig faktor som førte til «unbundeling» av programvaren. Se U.S. v. IBM (Anlagt hos U.S. District Court, Southern District of New York den 17 januar 1969. Saken ble trukket tilbake 8 januar 1982. Se «IBM Antitrust Suit Records»)

22 Begrepet «personal computer» fikk nok for alvor sitt gjennomslag da IBM introduserte maskinen «IBM PC» den 12. august 1981, men begrepet «personal computer» hadde vært brukt tidligere. Pressemeldingen er tilgjengelig på <http://www-1.ibm.com/ibm/history/documents/pdf/pcpress.pdf>

programmene ikke var leselige for mennesker²³. Denne usikkerheten skyldtes bl.a. dommen *White-Smith Music Publishing Co v Apolle Co*²⁴ som hadde slått fast at et automatisk piano sin notetrommel ikke kunne ansees som et eksemplar («a copy») av et musikkverk. Dette var fordi notetrommelen ikke var leselig for mennesker (trommelen var ikke «in intelligible notation.»²⁵).

Spørsmålet mange amerikanske jurister antagelig stilte seg var: Hvorfor skulle da programmer på en magnetplate ha rett til vern når magnetplaten var like uleselig som notetrommelen?²⁶ At lagring på magnetplater var eksemplarfremstilling ble imidlertid slått fast ved Copyright Act of 1976 (i kraft 1978)²⁷. Selv etter denne endringen var ikke spørsmålet om programmer var beskyttet av opphavsretten helt klart før National Commission on new Technological Uses of Copyrighted Works (CONTU) utredet spørsmålet og deres endringsforslag ble vedtatt i 1980.²⁸

Før denne tiden var det også en stor diskusjon om programmer skulle beskyttes av opphavsretten eller om det skulle være et eget vern spesielt for programmer (et *sui generis* vern)²⁹. Dette skyldtes, som vi skal se, de mange vanskelige problemene med å sammenligne dataprogrammer med f.eks. litteratur.

I Europa vedtok Council of the European Communities³⁰ i 1991 et direktiv³¹ som fastsatte at programmer er beskyttet av opphavsretten. Opphavsrett på programmer var imidlertid aktuelt i medlemslandene lenge før det. I Norge var opphavsrett på programmer i alle fall fra slutten av 60-tallet antatt i teorien³², selv om det var en god del skepsis å spore. Det er gjerne det «XII nordiska mötet för industriell rättsskydd» i Stockholm i 1968 som bli sett på som startpunktet. I rettspraksis ser en kanskje det første sporet i en midlertidig

23 Se Bing, 1985, s. 24 og Bing, 2006

24 Se 209 U.S. 1 (1908),

Se også Bing, 1985, s. 24 og Bing, 2006

25 Den åpenbare konsekvensen at verket da kan utnyttes fritt ble ikke avgjørende for dommen. Av dommen følger det: «It may be true that the use of these perforated rolls, in the absence of statutory protection, enables the manufacturers thereof to enjoy the use of musical compositions for which they pay no value. But such considerations properly address themselves to the legislative, and not to the judicial, branch of the government.»

26 Se Bing, 1985, s. 25 og Bing, 2006

Se Samuels, 2000, s. 81

27 Se Bing, 1985, s. 25

28 Se Bing, 1985, s. 24-25

Se Samules, 2000, s. 82

29 Se Bing, 1985, s. 21 og Bing, 2006

30 Nå Council of the European Union

31 Se Dir 91/250 (EDB-direktivet)

32 Se f.eks. Lassen, 1969, s 25-27 og Bull, 1971, s 26-32 (Bull, 1973, s. 33-40)

forføyning fra 1976 der retten ikke avviste at et program kunne være beskyttet etter katalogregelen³³. Norge var altså tidlig ute.

I dag er opphavsrettsbeskyttelse av programmer en selvfølgelighet, selv om det fremdeles gjenstår mange spørsmål å løse i nærmere detalj. Det at det tok mange år før opphavsrettsbeskyttelse av dataprogrammer ble en realitet, har antakelig bidratt til fremveksten av delekulturen.

2.2.3 Åpen kildekode – en motreaksjon?

Når programvareindustrien vokste frem la den tekniske og juridiske begrensinger på lisenstakers mulighet til bruk, videreutvikling og kopiering av programmer. Programmer ble sjeldnere gjort tilgjengelig i kildekodeform. I stedet valgte programvareprodusentene vanligvis kun å gi programmet ut i objektform for å hindre at konkurrentene skulle få en fordel.

Delekulturen, som hadde vokst frem for programmer, innebar at mange programmerere tidligere hadde kunnet løst problemer ved programmer de mottok og deretter delt løsningene med andre. Programmererne hadde altså i praksis (men uten formelle lisenser etc.) benyttet en åpen kildekode utviklingsstrategi lenge før konseptene åpen kildekode og fri programvare var kjent³⁴. Kanskje er det derfor betegnende å se på åpen programvare som en motereaksjon til den proprietære programvaren som vokste frem.

I 1983 startet Richard Stallman GNU³⁵. På det tidspunktet jobbet han som forsker ved MIT's Artificial Intelligence Lab (AI Lab)³⁶, noe han hadde gjort siden 1971. På MIT, som i andre universitetsmiljøer, var det en kultur for å dele programvare. Programmererne på AI Lab-en hadde løst mange praktiske problemer med å endre på programvaren de hadde fått fra andre universiteter og eksterne leverandører. Når nå programvareindustrien vokste frem på 70-tallet, skiftet databransjens syn på programvare fra å være et «nødvendig onde» til å være en verdifull ressurs. Rundt 1977 fikk AI Lab-en en avansert printer i gave fra Xerox.³⁷ Problemet var at papiret kjørte seg fast. Stallmann hadde for de eldre printerne løst dette ved at det ble sendt en beskjed til personene som stod i utskriftskøen om problemet, men siden kildekode ikke var tilgjengelig kunne han ikke gjøre det samme for denne nye printerens.³⁸ Stallman så hemmelighold

33 Se Bing, 1985, s. 30

Se Oslo Namsrett Kjennelse 12/1 1976 gjengitt i Bull, 1980, s. 80-176 se særlig s. 104

34 For tilsvarende syn se Raymond, 2005

35 GNU er, som sagt, et rekursivt akronym for Gnu's Not Unix. G-en står altså for GNU.

36 Se også DiBona, Ockman og Stone, 1999, s. 2

37 Se Stallman, 1992

38 Se Williams, 1992, kapittel 1

av kildekode som et alvorlig problem som han ikke kunne leve med³⁹. Den 27. september 1983 annonserte han på Usenet⁴⁰ at:

«Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for Gnu's Not Unix), and give it away free to everyone who can use it. ... I consider that the golden rule requires that if I like a program I must share it with other people who like it. I cannot in good conscience sign a nondisclosure agreement or a software license agreement.

So that I can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free.»⁴¹

Allikevel var det gjennom en programvarelisens, GPL, at fri programvare for alvor fikk sitt gjennomslag.

I mars 1985 publiserte Dr. Dobb's Journal of Software Tools en artikkel av Richard Stallman, «The GNU Manifesto». Artikkelen utdyper synet i Usenet annonseringen og fikk mye oppmerksomhet.

Før åpen kildekode-lisensene ble populære ville opphavsmennene gjerne dedikere programmene til «det fri», «the public domain», når de ønsket at programmet skulle kunne utnyttes liberalt. Ved programvare i «det fri» kunne proprietære programleverandører benytte koden uten selv på bidra til utviklingen av den. Dette anså Stallmann som et problem. GPL-lisensen ble en realitet og GPL versjon 1 er copyrightmerket februar 1989. En av særegenhetene ved GPL-lisensen er at den inneholder copyleft. Copyleft innebærer at i alle fall avledede verk som en lisenstaker lager må lisensieres under GPL-lisensen når lisenstaker (som blir lisensgiver) tilgjengeliggjør det avledede verket. Ideen bak copyleft er altså å sikre at lisenstakerne bidrar og at programvaren forblir fri, dvs. underlagt GPL-lisensen. Copyleft er et viktig tema som jeg vil se nærmere på nedenfor.

Lisenskostnadene for UNIX-systemet hadde vært stadig voksende. Dette var et problem for mange. Mange universitetsmiljøer hadde, som sagt, arbeidet med UNIX og laget utvidelser av systemet, deriblant University of California at Berkeley. Berkeley sine utvidelser hadde blitt særlig populære. På grunn av et økende ønske i miljøet om å få tilgang til nettverkfunksjonalitet som var blitt utviklet av Berkeley uten bruk av vernet UNIX⁴² kode, ble i juni 1989 en

39 Se Williams, 1992, kapittel 1

40 Usenet er en forkortelse av «user network» og er et Internett-basert diskusjonssystem hvor brukeren kan følge med og delta på forskjellige nyhetgrupper.

41 Reformatert og forkortet utgave av original Usenet artikkel. Se Stallman, 1983

42 Se «Berkeley Software Distribution»

del av BSD⁴³ (Berkeley UNIX) systemet, Networking Release 1, sluppet. Denne var lisensiert under den originale BSD-lisensen⁴⁴.

Utviklingen av GNU tok av, men en sentral komponent, kjernen (kernelen), manglet for at en kunne ha et operativsystem bygget på fri programvare. Det var dette Linus Torvalds løste med sitt initiativ til å få en kernel utviklet i 1991⁴⁵.

Miljøet rundt BSD hadde også sett fordelene av å ha et operativsystem under en liberal lisens. Etter Networking Release 1 begynte en å implementere UNIX-like komponenter som skulle danne et nytt operativsystem. Networking Release 2 kom i 1991 og var nesten et komplett operativsystem, men en komplett kernel manglet. Problemer oppstod imidlertid i 1992 da AT&T (Unix System Laboratories) saksøkte Berkeley. Dette bremsset den videre utviklingen. Saken ble forliket året etter. Torvalds har sagt i et intervju at «If 386BSD had been available when I started on Linux, Linux would probably never had happened.»⁴⁶

22. januar 1998 annonserte Netscape at nettleseren deres Mozilla, skulle lanseres under en åpen kildekodelisens⁴⁷. Begrepet «åpen kildekode» ble ikke brukt. De hadde «bold plans to make the source code for the next generation of its highly popular Netscape Communicator client software available for *free licensing* on the Internet.»⁴⁸ Netscape ble senere sluppet under Netscape Public License. Dette ble grunnlaget for dagens Firefox. Annonseringen var viktig fordi den viste at en stor kommersiell aktør var villige til å benytte fri programvare, men annonseringen ble møtt med en del skepsis.

Fri programvare var kontroversielt. Raymond forklarer dette med at «[Stalman's] general attack on intellectual property and the quasi-Marxist flavor of much of his propaganda turned off many hackers⁴⁹ and utterly alienated most software producers and customers outside the hacker culture itself.»⁵⁰

Netscape sin annonsering hadde gjort at fri programvare plutselig hadde fått mye oppmerksomhet. På et strategimøte som ble holdt 3. februar 1998 i Palo Alto, med bl.a. Raymond tilstede, bestemte en gruppe tilhengere av fri

43 Berkeley Software Distribution

44 Se McKusick, 1999, s. 41

45 Se Torvalds, 1991

46 Se Linksvayer, 1993

47 Se «Netscape announces plans to make next-generation communicator source code available free on the net»

48 Se «Netscape announces plans to make next-generation communicator source code available free on the net»

49 Brukes her «not as the term is now abused by journalists to mean a computer criminal, but in its true and original sense of an enthusiast, an artist, a tinkerer, a problem solver, an expert» jfr. Raymond, 2000

50 Se Raymond, 2005

programvare seg for å benytte begrepet åpen kildekode. Begrepet åpen kildekode ble funnet opp av Chris Peterson⁵¹. Ideen var å bli kvitt den negative holdningen til fri programvare, som av noen ble oppfattet som moraliserende og konfronterende, og heller lage et konseptet basert på forretningsmessige argumenter.⁵²

En kan derfor si at det er noe forskjellige filosofi bak bevegelsen for åpen kildekode og bevegelsen for fri programvare, men at lisensieringskonseptene i stor grad er de samme. Stallmann sier det på denne måten: «We [the Free Software movement] disagree [with the Open Source movement] on the basic principles, but agree more or less on the practical recommendations.»⁵³ Avgjørelsen om å innføre begrepet åpen kildekode har senere blitt kritisert for å skape unødvendig forvirring.

The Open Source Initiative ble grunnlagt i slutten av februar 1998 av Eric Raymond og Bruce Perens.⁵⁴ The Open Source Initiative er en forening som promoterer åpen kildekode og har en godkjenningprosess for åpen kildekodelisenser.

2.2.4 I dag

I løpet av 80-tallet hadde hjemme-PC-en blitt en realitet. På midten av 90-tallet begynte Internett å bli allment tilgjengelig. Fremveksten av Internett førte til at åpen kildekode kunne nå stadig nye brukere og utviklere.

I dag ser vi mange store aktører som støtter og bruker åpne kildekode-prosjekter når det gir en strategisk fordel. Å foreta utviklingen av programmer gjennom et globalt samarbeid gir stadig mer mening for mange bedrifter. Dette skyldes bl.a. den økte kompleksiteten ved å utvikle programmer, som ved samarbeid kan takles bedre. Som Raymond sier det: «Given enough eyes, all bugs are shallow». Videre mener en at globalt samarbeid gir mening fordi talentfulle programmerere er spredd over hele verden og dagens Internett kan lett bringe de sammen. I tillegg er det et behov for å integrere programvare fra mange kilder, noe et åpent samarbeid kan muliggjøre.

Til og med stater har vist sin interesse for fri programvare. I 2005 vedtok Venezuela at administrasjonen skal gå over til å fri programvare⁵⁵. Og siden jeg skriver om jus; følger det av direktiv 3.390, artikkel 1:

51 Se Tiemann, 2006

52 Se Tiemann, 2006

53 Se Stallman, 2002, s 57

54 Se Tiemann, 2006

55 Se «Venezuela: Government migration to Open Source makes experts scarce»

«A tales fines, todos los órganos y entes de la Administración Pública Nacional iniciarán los procesos de migración gradual y progresiva de éstos hacia el Software Libre desarrollado con Estándares Abiertos.»

Også EU har med sin European Union Public Licence (EURL) begynt aktivt å bruke åpen kildekode. I Norge har staten også vist stor interesse for fri programvare, gjennom bl.a. støttet til et nasjonalt kompetansesenter for fri programvare.

I dag ser vi også at lisenser, basert på tankegangen utviklet for åpen kildekode, blir brukt på andre åndsverkstyper som illustrasjoner, litteratur etc. Mest kjent er kanskje Creative Commons⁵⁶ og deres lisenser.

56 Se <http://creativecommons.org/>

3 RETTSLIG BESKYTTELSE AV ÅPEN PROGRAMVARE

I dette kapittelet vil jeg se på grunnleggende problemstillinger i forhold til den rettslige beskyttelsen av åpen kildekodeprogrammer.

Programvare består av mange element som kildekode, program i objektform, data som programmet benytter, dokumentasjon etc. Disse blir beskyttet på forskjellige måter. Det er reglene om opphavsrett for programmer som er i fokus i denne fremstillingen.

Programmer kan også bli beskyttet av andre regler enn opphavsretten. Programmer «as such» kan ikke patenteres⁵⁷, jfr. EPC artikkel 52 2. ledd bokstav c og 3. ledd og patl. § 1 1. ledd nr. 3. Programstyrte maskiner og prosesser kan allikevel være patenterbare når programmer inngår i disse, forutsatt at oppfinnelsen har teknisk karakter⁵⁸.

Skjermbilder kan være beskyttet av designrett⁵⁹ i tillegg til opphavsrett. Data som programmet benytter kan være beskyttet som åndsverk. Programmer og databaser kan være beskyttet etter katalogregelen i åvl. § 43 når de ikke er beskyttet av opphavsretten. Dette er kanskje begrepsmessig litt rart, men katalogregelens ordlyd er vid nok til også å kunne omfatte visse typer programmer. Markedsføringsloven kan også gi beskyttelse, f.eks. mot etterlikning av produkter, jfr. §§ 8a og 1.

3.1 Introduksjon til opphavsrett

Programmer blir vanligvis beskyttet av opphavsretten. Opphavsretten ble opprinnelig laget for litteratur, men beskytter i dag alt fra ølrikker til romaner, fra filmer til pantomime og, ikke minst, dataprogrammer. Et regelverk som skal omfatte alt dette må nødvendigvis bli abstrakt. Dette gir oss en del utfordringer

57 Noen mener dette strider med TRIPS Artikkel 27 1. ledd som sier «patents shall be available for any inventions, whether products or processes, *in all fields of technology*». En slik løsning kan vanskelig sies å være akseptert i europeisk rett.

58 Vicom T208/84. EPO bruker følgende eksempel: «... the problem of improving signal strengths between mobile phones is a technical problem, even if it is solved by modifications to the phone software rather than its hardware. Such an invention would obtain a patent, provided that the solution is also novel and inventive.».

Se «Computer-Implemented Inventions»

Se Koktvedgaard, 2005, s. 219

59 Designloven av 14. mars 2003 nr. 15

når det gjelder å trekke klare, praktiske grenser. En amerikansk dommer sa en gang «Applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit.»⁶⁰ Nedenfor kan jeg ikke garantere å få alle brikkene til å passe, men jeg kan i alle fall legge brikkene utover bordet så en kan se hvilket bilde de danner.

Ved første øyekast kan det se ut som om et program bare er tekst. Kildekoden er tekst, selv om den ikke er skrevet i et naturlig språk. I opphavsretten blir programmer i utgangspunktet behandlet som litterære verk; EDB-direktivet beskytter programmer som «literary works within the meaning of the Berne Convention», jfr. artikkel 1 1. ledd.

Programmer er allikevel ikke bare tekst. Teksten, er instruksjoner som er skrevet for at en datamaskin skal kunne gjenskape en viss oppførsel eller funksjonalitet. Tradisjonelle litterære verk, som bøker, har ikke det samme funksjonelle aspektet.

Den funksjonelle siden av programmer har ført til at det i juridisk litteratur har vært argumentert for at programmer kan sees på som maskiner som er konstruert i tekst⁶¹ («myk maskiner»)⁶². Og det var nettopp disse funksjonelle sidene som har gjort at det har blitt argumentert for et eget (sui generis) vern for programmer, og sammenligningen med maskiner har også ført til at det har blitt argumentert for patenter på programmer.⁶³ Den funksjonelle siden til programmer skaper en del utfordringer for den opphavsrettslige vurderingen. Disse vil jeg komme tilbake til nedenfor.

3.2 Når får en opphavsrett?

Beskyttelse etter åndsverksloven oppstår i det verket blir skapt, det er ingen krav til registrering eller merking⁶⁴. Dette betyr at jeg får opphavsrett til denne teksten etter hvert som jeg skriver den og programmereren etter hvert som han

60 Lotus Dev. Corp. v. Borland Int'l, Inc.

61 Se Samuelson, 1994, s. 2323 «Program text is, thus, like steel and plastic, a medium in which other works can be created».

62 Se Bing, 1985, s. 22

63 Se Samuelson, 1994, s. 2323

Se også Bull, 1973, s. 25-33

64 Se Bern konvensjonen Artikkel 5(2)

skriver kildekode. Jeg trenger heller ikke merke denne teksten med ©-tegnet⁶⁵ for at verket skal være beskyttet. I utlandet kan imidlertid registrering eller copyright-merking noen steder gi opphavsmannen en bedre stilling⁶⁶. Å oppgi navn på opphavsmannen til verket kan også gi en bedre rettsstilling i Norge siden åvl. § 7 oppstiller en forutsetning om at den som er påført verket som opphavsmann også er det. For kildekode plasserer en typisk navnet på programmererne i en innledende kommentar i kildekodefilen. For eksekverbare filer kan det vanligvis legges til opphavsrettsinformasjon som metadata. Etter § 7 må retten i utgangspunktet legge til grunn at de som er nevnt i kildekode, i metadata til den eksekverbare filen, i «about box» e.l. er opphavsmennene. Selvfølgelig kan bevis føres for at en annen er den rette opphavsmannen.

For å få opphavsrett må opphavsmannen ha uttrykt et, til en viss grad, originalt og individuelt verk. De nærmere kravene kommer jeg tilbake til.

3.3 Hva innebærer vernet?

Å ha opphavsrett innebærer at opphavsmannen får en tidsbegrenset beskyttelse for både økonomiske og ideelle rettigheter til verket. De økonomiske rettighetene følger av åvl. § 2, mens de ideelle følger av åvl. § 3 og delvis av § 2.

3.3.1 Økonomiske interesser

Av åvl. § 2 følger det at:

«Opphavsretten gir innen de grenser som er angitt i denne lov, enerett til å råde over åndsverket ved å fremstille varig eller midlertidig eksemplar av det og ved å gjøre det tilgjengelig for allmennheten, i opprinnelig eller endret skikkelse, i oversettelse eller bearbeidelse, i annen litteratur- eller kunstart eller i annen teknikk.»

65 Merkingen skjer kort fortalt ved at © og rettighetshaverens navn og året for første utgivelse plasseres på et rimelig synlig sted på hvert eksemplar. Året skrives enten med araber- eller romertall. Merket ble innført som følge av «Verdenskonvensjonen om opphavsrett» («Universal Copyright Convention») som ble undertegnet i Genève i 1952. Se artikkel III § 1. Verdenskonvensjonen har i dag p.g.a. TRIPS og Bern-konvensjonen mindre betydning. Fraser som «all rights reserved» eller «med enerett» har også tidligere blitt krevd, se f.eks, «Buenos Aires Konvensjonen» som ble undertegnet i 1910 artikkel 3.

Se også 17 USC § 401

Se også Lassen (1961) s. 36-62 som nøye gjennomgår copyrightmerking.

66 Se f.eks. 17 USC § 401 (2) og § 412

Opphavsretten gir altså opphavsmannen i utgangspunktet en monopolstilling når det gjelder rett til eksemplarframstilling (det vil upresist si kopiering) og til å gjøre verket tilgjengelig for allmennheten. Opphavsmannen har disse to enerettene både for verket i opprinnelig form og når det er endret eller bearbeidet.

Eksemplarframstilling har en ganske vid definisjon og omfatter både «varig eller midlertidig» eksemplar, jfr. åvl. § 2 og Rdir. 91/205 Art. 4 (a). Dette betyr at duplisering av filer er eksemplarframstilling og at å flytte en programfil mellom medier (noe som innebærer at det midlertidig er to eksemplarer av filen før den gamle slettes) antakelig vil være en (midlertidig) eksemplarframstilling⁶⁷. Når jeg laster et program opp til f. eks. en webserver og en bruker så laster det ned til seg har det altså skjedd minst to⁶⁸ eksemplarframstillinger, en på webserver-en og en hos brukeren. At et program blir lastet opp i hurtigminnet (RAM⁶⁹) er også vanligvis en eksemplarframstilling, jfr. «midlertidig»⁷⁰.

Den som har rett til å bruke programmet har også fått rett til å «fremstille eksemplar av ... programmet i den utstrekning det er nødvendig for å bruke programmet i samsvar med dets formål», jfr. åvl. § 39h. Se også Rdir. 91/205 Art. 5 1. ledd.

Tilgjengeliggjøring for allmennheten inkluderer å by frem programmet til salg, utleie, utlån utenfor det private område, jfr. åvl. § 2 3. ledd lita a. Mer praktisk for åpen kildekodeprosjekter er tilgjengeliggjøring fra en webserver eller gjennom et versjonskontrollsystem. For spørsmålet om hvorvidt tilgjengeliggjøring for allmennheten over nettet har funnet sted, er det antakelig *muligheten* for at allmennheten *kan få tilgang* til verket som er avgjørende, ikke om allmennheten faktisk har tilegnet seg verket⁷¹. Hvis filen på web-siden eller i versjonskontrollsystemet er passordbeskyttet blir altså spørsmålet da om passordet er tilgjengelig utenfor det private område. Er passordet det, er verket normalt tilgjengeliggjort for allmennheten.

Selv om en får en tillatelse til eksemplarframstilling, betyr ikke det at en har rett til å eksemplarframstille endre utgaver av verket. Av åvl. § 39b 1. ledd følger det «Overdragelse av opphavsrett gir ikke rett til å endre verket med mindre annet er avtalt.» Som vi skal se er retten til å tilgjengeliggjøre endrede verk viktig for åpen kildekodetilisens.

67 Se Andersen, 2005, s. 306-309

68 Egentlig er det skjedd flere eksemplarframstillinger siden filen også vil ha vært i hurtigminnet.

69 Random Access Memory. Datamaskiner har ofte også virtuell hukommelse, men jeg går ikke inn på slike «detaljer» i denne fremstillingen.

70 Se Andersen, 2005, s. 308 og Riis, 2001, s. 48

Se Direktiv 2001/29/EC artikkel 2

71 Se Andersen, 2005, s. 313-314 som støtter seg på Wagle og Ødegaard, 1997 s. 183-186.

3.3.2 Ideelle rettigheter og deres økte betydning i åpen kildekodeprosjekter

De ideelle rettighetene⁷² som åvl. § 3 gir er gitt for å beskytte integriteten til verket og ikke de økonomiske interessene⁷³. De ideelle rettighetene kan opphavsmannen i utgangspunktet ikke fraskrive seg (jfr. åvl. § 3 3. ledd) eller overføre (jfr. åvl. § 39). For kunstneriske verk som programmer uttrykker (eks. noen skjermbilder og lyd) er de ideelle rettighetene klart aktuelle. For andre sider av programmer blir ofte de ideelle rettighetene i juridisk litteratur avvist som uinteressante. I motsetning til ved proprietære programmer, har lisenstaker ved åpen kildekodetilisens rett til å endre og tilgjengeliggjøre programmet for allmennheten. Dette gjør at de ideelle rettighetene blir mer aktuelle ved åpen programvare enn ved proprietær programvare.

De ideelle rettighetene innebærer for det første en rett til navngivelse av opphavsmannen slik «god skikk tilsier», jfr. åvl. § 3 1. ledd. Hva som er «god skikk» innenfor IT-bransjen er det lite rettspraksis på og derfor vanskelig å si⁷⁴. I alle fall når det er relativt få programmerere er navngivelse i «about box», dokumentasjon o.l. ganske vanlig.

Navngivelse er viktig for at den programmereren som har laget koden skal kunne høste den anerkjennelse det gir å ha utviklet den. Denne anerkjennelsen blir særlig viktig når programmererne av åpne programmer ikke tar seg betalt. En annen side av navngivelse er at en skal unngå at den opprinnelige programmereren får «skylden» for andres feilaktige endringer. GPLv2 oppstiller bl.a. derfor et krav i artikkel 2 bokstav a om at «You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.» Navngivelse har også betydning for å kunne spore opphavsrettsbrudd.

For det andre oppstiller § 3 2. ledd et krav om respekt for opphavsmannen og verket. Endring og spredning må «ikke skje på en måte eller i en sammenheng som er krenkende for opphavsmannens litterære, vitenskapelige eller kunstneriske anseelse eller egenart, eller for verkets anseelse eller egenart.» Generelt skal det nok mye til for at opphavsmannen til et program blir krenket. Det er også verd å merke åvl. § 46 som verner «en tittel, et dekknavn eller et merke» mot nye verk som kan forveksles med «med tidligere offentliggjort verk eller dets opphavsmann.»

⁷² droits moraux

⁷³ droits patrimoniaux

⁷⁴ Se Andersen, 2005, 329

3.4 Lisenser og overføring av opphavsrett

Opphavsmannen kan selvfølgelig gi andre rett til å utnytte verket, se f. eks. åvl. § 39. Begrepene «overdragelse» og «lisens» er gitt litt forskjellig innhold i forskjellig litteratur⁷⁵. Jeg vil derfor her forklare min bruk. Min bruk er, som hos mange andre norske rettsanvendere, inspirert av angloamerikansk rett.

Å gi en lisens er å gi en tillatelse (en rett). Lisensen vil kunne omfatte en eller flere eller deler av de rettighetene opphavsretten gir opphavsmannen. At lisensen avgrenser i forhold til tid, sted og utnyttelsesmåte er vanlig. En lisens blir gitt fra en lisensgiver⁷⁶ til en lisenstaker⁷⁷ som et element i avtalen mellom dem. Allikevel omtales gjerne hele avtalen som en lisens. Lisensgiver kan i avtalen stille vilkår for at lisenstaker skal få lisensen. Dette gjør at åpen kildekodelisenser fungerer rettslig.

En skiller gjerne mellom eksklusive lisenser og simple (ikke-eksklusive) lisenser⁷⁸. Ved en eksklusiv lisens gis lisenstaker enerett til å utøve rettighetene han har fått (kanskje også ovenfor lisensgiver)⁷⁹. Lisensgiver kan da ikke lovlig gi de samme rettigheter til andre. Ved simple lisenser kan lisensgiver gi samme lisens til andre. Åpen kildekodelisenser bruker simple lisenser. Jeg vil vanligvis bare kalle disse «lisenser» for enkelthets skyld.

Innehaveren av en lisens blir ikke opphavsmann (eller «eier»⁸⁰ av eneretten etter åndsverkloven). For at det skal skje må opphavsretten overdras i sin helhet⁸¹.

Slik jeg bruker begrepene vil både lisensiering og overdragelse i sin helhet innebære en «overdragelse» etter åvl. § 39. Begrepsbruken innen opphavsretten er altså ikke konsistent og er nok i stor grad en smakssak.⁸²

Selv om en har fått en lisens kan en, i utgangspunktet, ikke overdra denne lisensen videre. Dette følger av åvl. § 39c 2. ledd som sier «Retten kan heller ikke overdras videre uten samtykke...». I åpen kildekodelisenser får en ikke rett

75 F.eks. sammenlign Knoph, 1936, s 136-140 med Wagle og Ødegaard, 1997, s 229-230s

76 Engelsk: licensor

77 Engelsk: licensee

78 begrepet simple lisenser kommer fra tysk rett «einfache nutzungsrechte». Simple lisenser kalles også enkle lisenser. Eksklusive lisenser blir ofte kalt enelisenser.

79 Se f.eks. Michaelsen, 2006, s. 41-42 «At opphavsmannen har gitt en eksklusiv tillatelse bør ikke medføre at han opphavsrettslig taper retten til selv å utøve rettigheten slik som tilfellet vil være ved overdragelser.» Situasjonene jeg skal omtale i denne oppgaven gir ikke foranledning til å gå nærmere inn på dette.

80 Begrepet «eier» er kun brukt for illustrasjon. Opphavsretten gir kun noen begrensede rettigheter til opphavsmannen. Det kan derfor diskuteres om begrepet bør brukes her.

81 For åndsverk er det ikke mulig pga. ideelle rettigheter. En må derfor forstå en overdragelse i sin helhet som en overdragelse så langt loven tillater.

82 Se Knoph, 1963, s 142

til å overdra sin egen lisensen til andre⁸³, men, som vi skal se, får lisenstaker rett til å distribuere programmet på samme vilkår som han mottok det. Dette kommer jeg nærmere tilbake til i kapittelet «Nærmere om lisensiering».

Konsekvensen av å ha fått en eksklusiv lisens eller å ha fått opphavsretten overført til seg er antakelig at innehaveren av den kan gå til søksmål hvis andre krenker hans enerett⁸⁴. For simple lisenser kan lisensinnehaver antakelig ikke gå til søksmål.⁸⁵ Dette fordi han ikke har en berettiget forventning om å ha en enerett. Dvs. når det åpne kildekodeprosjektet har fått et bidrag under en simpel lisens vil de antakelig ikke kunne påtale krenkelser mot opphavsretten til verk de har fått lisensiert. Blant annet derfor blir det, av og til, laget egne bidragsyteravtaler («contributor agreements») som overfører opphavsretten til prosjektet.

3.5 Begrunnelsen for opphavsrett og forholdet til åpen kildekode

Opphavsretten kan, som andre immaterielle rettigheter, begrunnes med at «den som har sådd bør høste fruktene av sitt arbeid»⁸⁶. Den økonomiske begrunnelsen for eneretten er at det å nekte ulisensiert utnyttelse skal muliggjøre eller øke potensialet for den økonomisk kompensasjon som opphavsmannen kan oppnå⁸⁷.

Ved bruk av åpen kildekodelisenser velger opphavsmannen i stor grad å gi avkall på eneretten til eksemplarframstilling og til den økonomiske kompensasjonen bruk av denne retten kunne ha medført. Dette er allikevel ikke det samme som at opphavsmannen ikke «høster fruktene av sitt arbeid». Opphavsmannen kan høste fruktene av sitt arbeid med åpen kildekode bl.a. i form av salg av andre tjenester og produkter, samt oppmerksomhet og anerkjennelse. Dette kommer jeg tilbake til.

Hensyn til samfunnets behov for å utnytte åndsverket begrenser eneretten til opphavsmannen. Et eksempel på dette er den ganske generelle retten til eksemplarframstilling til privat bruk etter avl. § 12. Programmer⁸⁸ som er

83 Sublisensiering derimot kan være aktuelt for noen lisenser. Dette kommer jeg tilbake til.

84 Se Rognstad, 1999, s. 89 «Med andre ord innebærer en eksklusiv lisens overdragelse av opphavsmannens forbudsrett innenfor lisensens område»

85 Se f.eks. Michaelsen, 2006, s. 42 -43

86 Se Knoph, 1936, s. 4

87 Se Rognstad, 2004, s. 76

88 Dvs. både på kildekode og objektform, se Ot. prp. nr.84 (1991-1992) s. 18

«maskinlesbare»⁸⁹ er unntatt fra denne retten, jfr. åvl. § 12 2. ledd bokstav b⁹⁰. En har derimot en rett til å ta sikkerhetskopier, jfr. åvl. § 39h annet ledd.

3.6 Kravet til frembringelse

Det er verket opphavsmannen selv har skapt som blir beskyttet, jfr. åvl. § 1 1. ledd. I begrepet «skapt» ligger det et krav om at det frembrakte må ha en ytre realitet.⁹¹ En del land har et krav om fiksering for at verket skal bli vernet⁹². Dette har vi ikke i Norge.⁹³ I begrepet «skapt» ligger det også et krav til originalitet, dette vil jeg behandle nedenfor.

Beskyttelsen etter åvl. § 1 strekker seg så langt at verket er beskyttet også i endret skikkelse når verket kan sies å bevare sin identitet.⁹⁴ Kode på objektform er kildekode i en annen skikkelse. Det betyr at både kildekode og objektkode til samme program beskyttes som samme verk.

3.7 Verkets type – Hva er et «datamaskinprogram»?⁹⁵

Åndsverk blir definert som «litterære, vitenskapelige eller kunstneriske verk av enhver av enhver art og uansett uttrykksmåte og uttrykksform», jfr. åvl. § 1.2. Det innebærer i utgangspunktet at en hver manifestert informasjonsmengde kan beskyttes⁹⁶. Det er derfor klart at programmer både som kildekode og på objektform kan bli vernet. Det vil også spesifikasjoner og dokumentasjon kunne bli. Det er derfor ikke nødvendig å trekke grensene nærmere i forhold til kategoriene i åvl. § 1.

«Datamaskinprogrammer» er nevnt som et av eksemplene på åndsverk, jfr. åvl. § 1 2. ledd nr 12. Begrepet omfatter både kildekode og programmet på

89 Dvs. «en form som er egnet for innlesning i datamaskin», jfr. Ot. prp. nr.84 (1991-1992) s. 18. Kode som står i en lærebok kan derimot eksemplarfremsendes til privat bruk.

90 Dette følger av at Rdir 91/250 mangler et unntak for privat bruk. Hovedregelen i artikkel 4 får derfor anvendelse.

91 Se Rognstad, 2004, s. 21

92 Se f.eks. Storbritannias Copyright, Designs and Patents Act 1988 Section 3 (2)

93 Se Rognstad, 2004, s. 21 og åvl. § 1 2. ledd nr. 2 som omfatter «muntlige foredrag»

94 Se Rognstad, 2004, s. 68

95 Se også Rdir. 91/205 art. 1.1

96 Se Andersen, 2005, s. 289

objektform⁹⁷. EFs direktiv om datamaskinprogrammer, Rdir. 91/205, inkluderer også forberedende designmateriale (spesifikasjoner) i begrepet «datamaskinprogram», jfr. artikkel 1 1. ledd «the term 'computer programs' shall include their preparatory design material»⁹⁸.

Begrepet «datamaskinprogram» i opphavsrettslig forstand er derfor noe annerledes enn i vanlig språkbruk. Jeg bruker ordet «program» som en kortform av «datamaskinprogram».

Programmer bruker ofte og er ofte avhengige av data fra eksterne kilder. F.eks. kan det være en billedfil som ligger utenfor selve programfilen som brukes som velkomstbilde når programmet starter e.l. Slike eksterne filer er nok ikke i utgangspunktet del av «datamaskinprogrammet» ut fra definisjonen over⁹⁹.

Det at data som bilder, tabeller etc. legges inn i programfilen (kalt resurser) gjør ikke at de blir del av «datamaskinprogram» i opphavsrettslig forstand siden definisjonen ikke omfatter dette¹⁰⁰. Koder en inn instruksjoner som f.eks. spiller et musikkstykke, er nok instruksjonene en del av datamaskinprogrammet.

3.8 Kravet til originalitet

Av åndsverkloven § 1 følger det at beskyttelsen gjelder «åndsverk». Programmet må altså kunne betegnes som et (ånds)verk for å være beskyttet av opphavsretten. Det vil si programmet må ha verkshøyde. For å ha verkshøyde må verket «i alle fall i noen grad være uttrykk for original og individuelt preget åndsvirksomhet fra opphavsmannens side»¹⁰¹ Det er denne åndsinnnsatsen loven gir uttrykk for når det står «skaper». Kravet er ikke strengt.

Kravet til originalitet følger også av Programvaredirektivet artikkel 1 3. ledd¹⁰² som sier at «A computer program shall be protected if it is original in

97 Se Ot. prp. nr. 33 (1989-1990) s. 5

98 Se også avsnitt 11 i fortalen «Whereas, for the purpose of this Directive, the term 'computer program' shall include programs in any form, including those which are incorporated into hardware; whereas this term also includes preparatory design work leading to the development of a computer program provided that the nature of the preparatory work is such that a computer program can result from it at a later stage;»

99 Se Andersen. 2005, s 358 og 361. I forhold til f.eks. § 39h kan det være av praktiske hensyn aktuelt å legge en videre forståelse til grunn.

100 Se Andersen, 2005, s. 360

101 Se Knoph, 1936, s. 64

102 Se Rognstad, 2004, s 25

the sense that it is *the author's own intellectual creation*. No other criteria shall be applied to determine its eligibility for protection.»

Opphavsmannen må selvfølgelig ikke ha skapt verket helt alene for å få vern, ingen verk er skapt i et vakuum. Opphavsmannen kan bygge på andres ideer etc., men ikke på andres vernede uttrykk (gjør han det oppstår det en bearbeidelse). Det er denne selvstendige, skapende innsatsen til opphavsmannen som er grunnlaget for vernet. I opphavsretten er det ikke krav om kunstnerisk eller vitenskaplig verdi, kvalitet e.l.

Selv om verkshøydekravet har gyldighet for hele opphavsretten, vil hvor strengt kravet er variere etter verkstypen¹⁰³. Kravet til originalitet for programmer er relativt høyt.¹⁰⁴ De fleste hele programmer vil i praksis ha tilstrekkelig verkshøyde¹⁰⁵.

For å få beskyttelse må verket være originalt, men det trenger ikke være nyskapende. Kravet til originalitet er subjektivt og skiller seg derfor fra patentrettens krav til nyhet som er objektivt. Dvs. hvis to personer på selvstendig måte lager det samme uttrykket får de to opphavsrett til hvert sitt verk (kalt dobbeltfrembringelse). Dette er selvfølgelig ganske upraktisk for hele programmer¹⁰⁶, det blir derimot mer praktisk når en ser på mindre deler av et program. Der hvor risikoen for dobbeltfrembringelse er stor, er det grunn til å anta at «verket» (typisk delen av programmet) ikke har tilstrekkelig verkshøyde til å være vernet¹⁰⁷.

Opphavsretten beskytter ikke de underliggende ideer, prosesser, metoder, prinsipper, algoritmer, tekniske løsninger etc. som sådanne selv om de er en del av programmet¹⁰⁸. Fakta blir heller ikke beskyttet¹⁰⁹. Det er det konkrete uttrykket med individuelt preg som er gjort av opphavsmannen av de abstrakte ideene, algoritmene etc. i programmet som blir beskyttet. For programmer vil det si at det er selve koden, eller programmet i objektform, som kan bli beskyttet av opphavsretten, ikke ideene bak. Ideene bak programmet (som

103 Se Andersen, 2005, s. 293

104 Se Andersen, 2005, s. 293-294

105 Se Bing, 1985, s. 32

106 Det er usannsynlig at to vil frembringe noe likt når størrelsen og kompleksiteten blir stor

107 Se Andersen, 2005, s. 295

108 «Det er et grunnleggende prinsipp at når åndsverkloven beskytter tekniske tegninger, jf § 1 nr 11, gjelder beskyttelsen ikke de tekniske løsninger eller det produkt som tegningen beskriver.», jfr. Rt. 1997 s. 199 «Cirrus» s. 219

«Det er utformingen av verket som er vernet, ikke ideer og prinsipper bak programmet, innholdet i programmet eller metode og teknikk benyttet ved utformingen av verket.», jfr. RG 2000 s. 1516 «Markus Data»

Se også Rdir. 91/250 fortale

Se også Koktvedgaard, 2005, s 78

109 Se f.eks. Bern-konvensjonen Artikkel 2 (8)

algoritmene) vil i en del tilfeller være det mest verdifulle ved programmet. Algoritmene blir, på tross av at de kan være det mest verdifulle, ikke beskyttet, siden en slik beskyttelse ville hemme den tekniske utviklingen. Mange som har unike algoritmer av økonomisk betydning vil nok gjerne beholde dem som forretningshemmeligheter. Antagelig vil en i slike situasjoner unngå åpen kildekode for å holde algoritmene mest mulig hemmelig.

Når en idé og uttrykket av denne ideen er ett, kan en ikke få opphavsrett til uttrykket av ideen. Å gi opphavsrett til dette uttrykket ville være å gi monopol på ideen, noe som opphavsretten ikke gir. Ideen og uttrykket av denne vil være ett når det bare er én eller et svært lite antall tenkelige måter å uttrykke ideen på¹¹⁰. For hele programmer er nok dette sjeldent tilfelle, men for mindre deler er det aktuelt.

Selv om programmet som helhet har tilstrekkelig verkshøyde, er det ikke sikkert at de enkelte delene av det har tilstrekkelig verkshøyde. Hvis en splitter opp (eller partisjonerer) programmet vil en kunne få deler som i seg selv ikke har tilstrekkelig verkshøyde¹¹¹ og som dermed er ubeskyttet¹¹². Parallellen til litterære verk er illustrerende. I en roman vil gjerne hvert enkelt kapittel være et verk. Avsnittene vil typisk også være det. For de enkelte setningene er det derimot verre. Mange vil ofte ikke være tilstrekkelig originale til å være vernet. De enkelte ordene og bokstavene vil ikke være vernet.

Det samme gjelder for programmer, men i motsetning til litterære verk har programmereren til en hvis grad mindre frihet i måten han uttrykker seg på, og dermed kan det være vanskeligere å oppnå tilstrekkelig verkshøyde¹¹³. Dette betyr at størrelsen på delene som blir uten vern ved partisjonering, som over, ofte vil kunne være større for programmer enn for tradisjonell litteratur. Dette skyldes dels at valgfriheten er mindre i programmeringsspråk enn naturlige språk, dels at funksjonen til programmet bestemmer utformingen. Dette poenget må ikke strekkes for lagt da det vanligvis er et stort antall mulige måter å uttrykke seg på for å lage samme funksjonalitet.

110 Dette er kjent som «the merger doctrine» i amerikansk rett, og konseptet er ikke tradisjonelt uttrykt helt på denne måten i norsk rett. I stedet snakker vi gjerne om at funksjonen dikterer uttrykket og at uttrykket derfor ikke er vernet. Konseptet må imidlertid også gjelde i norsk rett (siden vi ikke verner ideer). Jeg viser også til Riis, 2001 s. 37-38, som finner støtte for merger-doktrinen i norsk rett.

111 Altså de minimis

112 Se Bing, 1999, 293

113 Se Bing, 1999, 293 og Andersen, 2005, s. 365

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```

Illustrasjon 1: Eksempel på et trivielt program uten verkshøyde

For et spesifikt problem i et bestemt programmeringsspråk vil det i en del tilfeller finnes én eller et fåtall optimale løsninger. Typiske eksempler er implementeringer av mange av de klassiske, gjennomarbeidede algoritmene. I slike situasjoner, med svært få løsningsmuligheter, vil det være vanskelig å tilføre original skapende åndsinnsett siden ideen dikterer løsningen. I disse tilfellene vil løsningene ikke ha verkshøyde og være uten vern.¹¹⁴ Hvis en f.eks. vil på en enkel måte implementere (programmere) insert algoritmen for en vanlig binær heap ved bruk av en array¹¹⁵ mest mulig effektivt, er en nesten helt bundet til en bestemt utforming. Implementeringen vil derfor ikke innebære individuelt preget åndsvirksomhet og er derfor ikke vernet. (En må selvfølgelig se bort fra at forskjellige løsninger vil ha kosmetiske forskjeller i variabelnavn etc.)

Utbredt praksis for måter en løser visse problemer på, kan også innebære at den enkelte løsningen ikke er original. Ofte finnes det gjerne alminnelige oppfatninger av gode måter å løse problemer på. I disse tilfellene blir fort risikoen for dobbelt frembringelse for stor til at en får opphavsrettsbeskyttelse. Det er i disse tilfellene er det lite trolig at opphavsmannen har tilført noe originalt¹¹⁶. Dette vil typisk gjelde kun begrensede deler av programmet.

En annen side av kravet til individuell skapende åndsinnsett er at det må være et menneske som har skapt åndsverket, dvs. ikke en maskin¹¹⁷. At et menneske bruker en maskin som et verktøy i fremstillingen (f.eks. et IDE¹¹⁸) hindrer ikke vern.

¹¹⁴ Se Bing, 1985, s. 31

¹¹⁵ En binær heap er en enkel datastruktur for prioritetskøer. I prioritetskøer kan en legge til elementer i køen og ta ut elementet med høyest prioritet.

¹¹⁶ Disse situasjonene har mye tilfelles med situasjonen som «scenes à faire»-doktrinen i amerikansk rett omhandler.

¹¹⁷ Se Rognsatts, 2004 s. 20

¹¹⁸ «Integrated development environment»

3.8.1 Særlig om vern av programmets «dypstruktur»

Det er ikke bare selve instruksjonene og deres faktiske rekkefølge / struktur¹¹⁹ i koden som er en del av uttrykket, men også til en viss grad «meningen» bak disse. Dette viser MOCS-dommen¹²⁰, der det ble uttalt at «Lagmannsretten anser også at gitte sekvenser av meningsinnhold i programform vil kunne gis opphavsrettslig vern.» Dette innebærer en vanskelig grensedragnings. Beskyttelsen av uttrykket strekker seg altså til et noe mer abstrakt nivå enn selve den faktiske koden, men vernet av uttrykket kan ikke strekke seg til et så abstrakt nivå at ideer, prosesser, algoritmer etc. blir vernet, siden det ville stride mot grunnleggende opphavsrett og gi monopol på ideer.

Lagmannsretten sier at dypstrukturen er vernet. Dette er et skille som stammer fra lingvistikken der en skiller en mellom overflatestruktur og dypstruktur¹²¹. Vern av «meningsinnholdet» må i alle fall innebære at selv om en foretar skinnendringer i en kode vil koden fremdeles være det samme uttrykket. F.eks. hvis en skifter ut variabelnavn (som ikke har betydning for kodens funksjon) vil det ikke være nok til at et nytt uavhengig uttrykk blir skapt. Heller ikke hvis en skifter ut rekkefølgen på subrutiner i en fil vil det vanligvis ha betydning for programmet, og subrutinene vil noen ganger kunne sammenslås eller deles opp og flyttes til andre klasser uten at det har noen betydning for programmets funksjonalitet¹²². Slike skinnendringer blir i utgangspunktet ikke nok til at det er oppstått et nytt uavhengig uttrykk. Det vanskelige spørsmålet blir da hvor langt strekker opphavsrettsvernet av uttrykket seg utover slike skinnendringer?

Skillet mellom uttrykk og idé, eller at det er tilstrekkelig originalitet om en vil, må vurderes konkret i hver enkelt sak. Dommer Learned Hand har trefende sagt «Nobody has ever been able to fix that boundary, and nobody ever can»¹²³. Ut fra hensynet til at ideer ikke skal kunne monopoliseres strekker nok vernet seg kun til ganske konkrete uttrykk.

119 Se RG 1999 s. 330 (MOCS) og Andersen, 2005, s. 364. Når det står i Sandefjord tingretts dom RG 2002 s. 1516 (Markus data) at «Strukturering av et datamaskinprogram vil ikke være opphavsrettslig vernet» gjelder antakelig det ikke kildekoden, men et menysystem.

120 RG 1999 s. 330

121 For eksempel «Ole har spist bananen», «Bananen har Ole spist» og «Bananen har blitt spist av Ole» har samme dypstruktur, men ikke samme overflatestruktur.

122 Ikke alle programmeringsspråk har konsepter som subrutiner og klasser.

123 Se Nichols v Universal Pictures Corp

3.9 Subjektet – hvem er «opphavsmannen»?

Utgangspunktet i opphavsretten er at den som skaper åndsverket får opphavsrett til det, jfr. åvl. § 1 1. ledd. Det er programmereren som skaper programmet (verket) og derfor er han opphavsmann.

3.9.1 Ansettelsesforhold

I et arbeidsforhold vil åvl. § 1 1. ledd tilsi at den ansatte som lager åndsverket får opphavsretten i utgangspunktet, men her er det viktige unntak.

Hvis en er ansatt for å lage åndsverk (generelt) vil arbeidsgiver få rett til å råde over åndsverket i kraft av tjenesteforholdet, men rådigheten blir begrenset. Arbeidsgiver får den rett «som er nødvendig og rimelig, hvis arbeidsavtalen skal nå sitt formål, men heller ikke mer»¹²⁴, når ikke annet er avtalt¹²⁵.

For ansatte programmerere er det et viktig avvik fra dette utgangspunktet. Av åvl. § 39g¹²⁶ følger det at:

«Opphavsrett til datamaskinprogram som er skapt av en arbeidstaker under utførelsen av oppgaver som omfattes av arbeidsforholdet eller etter arbeidsgivers anvisninger går, med den begrensning som følger av §3, over til arbeidsgiveren, med mindre annet er avtalt.» § 3 gjelder ideelle rettigheter som altså programmereren beholder.

Hvis arbeidstaker bidrar til et åpen kildekodeprosjekt, uten avtale med arbeidsgiveren om dette, og verket arbeidstaker bidrar med faller inn under § 39g, vil det fort bli et brudd på arbeidsgivers opphavsrett. Dette kommer jeg tilbake til nedenfor.

3.9.2 Konsulenter og oppdragsforhold

Når eksterne konsulenter programmerer eller bidrar til *selve utformingen* av programmet vil konsulenten få opphavsrett (eller medopphavsrett) til det han gjør, med mindre noe annet er avtalt. Oppdragsgiver vil imidlertid, uten spesifikk avtale, få den rett som er rimelig og nødvendig for at oppdragsavtalen skal nå sitt formål, jfr. analogi til situasjonen for arbeidstaker.¹²⁷ Hvis konsulenten har utviklet et program eller del av et program vil det typisk innebære at

124 Se Knoph, 1936, s. 84

Se Vigen, 2001 for en generell behandling av emnet.

125 Se f.eks. Vigen, 2001, s. 25

126 Åvl. § 39g er et resultat av Rdir. 91/205 Art. 2 3. ledd

Se også Vigen, 2001, s. 59, 60-61 og 83-85 og Torvund, 1999, s. 124-125

127 Tilsvarende Torvund, 1999, s. 128

oppdragsgiver får rett til uinnskrenket og tidsubegrenset bruk av programmet innenfor oppdragsgivers egen organisasjon¹²⁸, men hvis oppdragsgiver f.eks. vil gjøre programmet tilgjengelig som åpen kildekode er det ikke klart at han har rett til det. Det vil bero på en konkret vurdering av situasjonen om det kan sies å være innenfor formålet til oppdragsavtalen at kunden skal ha denne retten. For å unngå slike problemer, bør rettighetene avtalereguleres på en klar måte.

3.10 Fellesverk¹²⁹

I mange tilfeller vil opphavsmennene samarbeide om et verk, f.eks. et helt program eller en modul i programmet. Skjer samarbeidet på en måte som gjør at «de enkeltes ytelser [ikke] kan skilles ut som særskilte verk», jfr. åvl. § 6, oppstår det et fellesverk. Ved fellesverk blir opphavsretten til hele fellesverket et sameie¹³⁰ mellom de samarbeidende programmererne.

At mange utviklere samarbeider på samme modul er typisk for utvikling av dataprogrammer. Mange åpen kildekodeprosjekter er under kontinuerlig utvikling. Det vil nok her ofte være vanskelig å trekke grensen mellom hva som er et fellesverk og hva som er en bearbeidelse, men, som vi skal se, er dette mer et teoretisk problem i forhold til kategorisering enn et praktisk problem.

3.10.1 (U)selvstendige verk

For at et verk skal kunne betegnes som et fellesverk, må det ikke kunne skilles ut «særskilte verk», jfr. § 6. Ordet «særskilt» må bety det samme som selvstendig.¹³¹

Et verk er selvstendig når det kan atskilles og det er uavhengig (dvs. at det kan «stå på egne bein»)¹³².

128 Se Andersen, 2005, s. 349

Alcatel STK-dommen fra Oslo Byrett i 1992 ser ut til å ha en noe strengere tilnærming. Retten skriver «Det er ikke bevist at det er noen avtale om at programmet var avtalt å brukes for andre anlegg i Alcatel STK og retten finner derfor at Alcatel STK bare kan bruke ett eksemplar av programmet». Se Torvund, 1999, s. 133

129 Fellesverk kalles også ekte samarbeidsverk og udelelige samarbeidsverk.

130 Om sameielovens anvendelse se Lassen, 1983, særlig fra side 383. Det er sterke begrensninger i lovens anvendelse på fellesverk.

131 Ordet selvstendig er brukt i de andre nordiske lovene og her er det antakelig rettsenhet. Se Lassen, 1983, s. 327

132 Se Rognstad, 2004 s 56 og Lassen, 1983, s. 327-328

For at et verk skal kunne atskilles må en kunne peke ut verket og dermed separere det fra andre.¹³³ Hvis en ikke kan peke ut hvilken del av verket som stammer fra en enkelt programmerer eller en enkelt gruppe, vil en ikke kunne skille ut et selvstendig verk. Forutsatt samarbeid vil det i disse tilfellene være et fellesverk.

Selv om en kan peke ut hvem som har gjort hva, er det alene ikke nok til at et verk er selvstendig.¹³⁴ Den andre siden av at et verk er selvstendig er som sagt uavhengighet. Dvs. verket må kunne «stå på egne ben».¹³⁵ Helt hvor grensene går for programmer er ikke lett å si. Dette spørsmålet har, så vidt jeg kan se, ikke blitt særlig utforsket i norsk juridisk litteratur eller hos domstolene. En må her foreta en konkret vurdering, hvor en legger vekt på om verkene i tilstrekkelig grad fremtrer enhetlige eller ikke.¹³⁶ En kan nok ikke her basere seg på teknisk avhengighet. Dette er et annet begrep enn det opphavsrettslige. Å likestille disse begrepene ville betydd at de fleste programmer ikke ville være selvstendige i forhold til operativsystemet, noe som er en klart uholdbar løsning.

Opphavsretten var opprinnelig laget for å beskytte bøker,¹³⁷ som ikke har slike funksjonelle egenskaper som programmer. Opphavsretten beskytter ikke funksjonalitet som sådan, bare uttrykket av funksjonaliteten når dette uttrykket ikke er diktet av funksjonaliteten. Siden programmer er beskyttet som litterære verk, jfr. EDB-direktivet artikkel 1, må en være forsiktig med å argumentere med funksjonelle egenskaper for å utvide hva som utgjør et selvstendig verk (eller et avhengighetsverk).

En parallell til bøker kan være illustrerende. Selv om en er avhengig av å ha lest en grunnbok for å kunne bruke den viderekommende boken i en lærebokserie, gjerne fordi den viderekommende boken refererer til sider og tabeller i grunnboken, vil det som regel være naturlig opphavsrettslig å se på bøkene som to forskjellige verk, selv om leseren er avhengig av begge bøkene.¹³⁸ Se også argumentasjonen under i kapittelet om grensesnitt.

133 Se Bing, 1985, s. 52, Rognstad, 2004, s. 56 og Knoph, 1936, s. 80

134 Se Lassen, 1983, s 327

135 Se Rognstad, 2004, s. 56 og Lassen, 1983, s. 327-328

136 Se Lassen 1983, s 331 som behandler eksempelet der en tverrfaglig gruppe går sammen om en bok som skal belyse samfunnsproblemer fra ulike synsvinkler. Han skriver «Selv om deltagerne her i mange henseender underordner seg et avtalt felles opplegg, for eksempel når det gjelder kapitlenes omfang og inndelingsmåte, graden av popularisering, språkform etc., og gir interne henvisninger til hverandres kapitler, er det ikke sikkert at boken bør sees som et fellesverk. Her må man trolig legge avgjørende vekt på i hvilken grad den fremtrer som en enhetlig fremstilling, men hvor grensen mot sammensatt verk bør trekkes er likevel nokså usikkert.»

137 Se Samuels, 2000, s 11, 131

138 Se Lassen, 1983, s 331 som gir et nokså tilsvarende eksempel.

Når et program er sammensatt av flere selvstendige moduler vil hver selvstendig modul kunne være et fellesverk, mens programmet som helhet vil være et sammensatt verk.

3.10.2 Samarbeid

Selv om programdelen ikke kan deles opp i selvstendige verk, vil den bare være et fellesverk hvis delen har blitt til i samarbeid mellom opphavsmennene. Kravet til samarbeid kommer ikke klart frem av loven, men er sikker rett.¹³⁹ Spørsmålet blir da hvilket samarbeid som er tilstrekkelig.

Samarbeid innen én enkelt bedrift er lite problematisk når § 39g får anvendelse, da er det jo bare én som ender opp med opphavsretten, bedriften. (Selv om det er to eller flere originære opphavsmenn).

Den som bidrar med ideer og kritikk, men som ikke har vært med på å prege den konkrete utformingen av programmet med skapende innsats blir ikke medopphavsmann.¹⁴⁰ Derimot preger en den konkrete utformingen med skapende innsats blir en medopphavsmann.¹⁴¹

Utarbeider f.eks. en person spesifikasjoner og en annen implementerer disse kan det oppstå et fellesverk så langt det skapende og originale skjer hos begge. Anvisningen i spesifikasjonene må altså være så spesifiserte at de bidrar til den konkrete utformingen av programmet for at det skal oppstå et fellesverk,¹⁴² f.eks. ved bruk av pseudokode¹⁴³ eller klassediagrammer etc. En som lager mer abstrakte spesifikasjoner, som ikke bidrar til den konkrete utformingen av programmet, men kun til ideen om programmet, blir ikke medopphavsmann.

En av fordelene som ofte blir fremhevet ved bruk av åpen kildekode er at tilgangen til kildekode også gjør at andre likemenn kan undersøke kvaliteten av den («peer review»). Dette kan skje ved at bidragsyteren sender en diff¹⁴⁴ med de endringene han har gjort til mailing-listen for programmet. Hvis forslagene bidragsyterne får er så konkrete og originale at de har verkhøyde vil det fort oppstå et fellesverk når bidragsyteren benytter seg av endringene. Hvis ingenting er sagt om lisens, vil det nok ofte være naturlig i disse situasjonene, der kommentaren er gitt for at den skal kunne brukes i kode som inngår i et åpen kildekodeprosjekt, å si at kommentatoren har gitt en implisitt lisens til programmereren av bidraget for å bruke kommentarene i prosjektet.

139 Se Rognstad, 2004, s. 57 og Lassen, 1983, 327

140 Se Lassen, 1983, s 332

141 Se Lassen, 1983, s 332

142 Se Bing, 1985, s. 53 og 36 og mer generelt Knoph, 1936, s. 78

143 Kode som ikke er ment å kjøre på maskinen og som gir en kompakt beskrivelse på et høyt nivå over algoritmen eller metoden.

144 Ordet diff kommer fra difference. Diff-er er en spesiell måte å uttrykke endringer på.

Fellesverk skiller seg fra sammensatte og samleverk fordi ved fellesverk kan den enkeltes bidrag ikke skilles ut som særskilt verk. Fellesverkene skiller seg fra bearbeidelsene ved at bearbeidelser skjer på et allerede eksisterende verk. Selve bearbeidelsen kan forøvrig være et fellesverk mellom flere medopphavsmenn.

3.11 Sammensatte verk

Et sammensatt verk er et verk som er satt sammen av flere deler. Disse delene må være originale (altså verk i seg selv). I tillegg må de være selvstendige¹⁴⁵. Er delene ikke selvstendige verk kan det være en står ovenfor et fellesverk.

Et program som er sammensatt av selvstendige moduler (med verkshøyde) vil være et sammensatt verk¹⁴⁶. Når en benytter seg av ferdige biblioteker i programmet en lager vil det ofte oppstå et sammensatt verk. Dette kommer jeg tilbake til nedenfor. Mange åpen kildekodeprosjekter går nettopp ut på å lage slike biblioteker.

Ved sammensatte verk vil de forskjellige opphavsmennene være selvstendige, dvs. den enkelte opphavsmann har opphavsrett til sin del av det sammensatte verket. De må derfor sammen være enige om utnyttelsen for at hele det sammensatte verket kan utnyttes.

3.12 Samleverk

Samleverk betyr det samme som et sammensatt verk, men selve sammensetningen har her verkshøyde. Altså resultatet av sammensetningen må uttrykke original åndsinnsats fra sammensetterens side for at det skal være et samleverk. Opphavsmannen til samleverket får opphavsrett til selve sammenstillingen, men selvfølgelig ikke til de enkelte verkene han har sammenstilt, jfr. åvl. § 5. Det innebærer bl.a. at sammenstilleren må ha samtykke fra opphavsmennene til de enkelte verkene som er sammenstilt hvis han vil fremstille eksemplarer av samleverket.

For dataprogrammer kan samleverk være aktuelt f.eks. ved bruk av noen typer CASE-programmer¹⁴⁷ som lar en «klikke-og-dra» ferdige moduler

145 For mer om hva som er et selvstendig verk viser jeg til det jeg har sagt over.

146 Se Bing, 1985, s. 52

147 Computer-aided software engineering

sammen til et program. «Programmereren» kan da få opphavsrett til selve sammenstillingen av modulene.

3.13 Avledede verk (endringer og bearbeidelser) og nye selvstendige verk

Dersom noen lager et verk basert på et annet verk kan det oppstå et bearbeidet verk. En sier gjerne at bearbeideren lager verket sitt «oppå» eller «inni» en annens verk¹⁴⁸. For bearbeidelser av et program, som f.eks. oppdateringer og oppgraderinger kan være, vil bearbeideren ha opphavsrett til selve bearbeidelsen, jfr. åvl. §§ 1 nr. 13 og 4 2. ledd.

For at bearbeidelsen skal være vernet opphavsrettslig må selve bearbeidelsen oppfylle kravet til verk som er diskutert over. For eksempel vil en feilretting som innebærer små endringer for å rette opp en «skrivefeil» e.l. ikke oppfylle kravet til verkshøyde. Slike endringer vil derfor ikke være beskyttet. Det er altså et skille mellom begrepet endring og bearbeidelse i opphavsrettslig forstand. Bearbeidelser er endringen som har verkshøyde og som derfor blir beskyttet, mens endringer som sådan ikke blir beskyttet opphavsrettslig. Som en fellesbetegnelse kalles endrede og bearbeidede verk avledede verk.

Rettighetshaveren til det opprinnelige verket har selvfølgelig, på tross av endringen, fremdeles vern for det opprinnelige verket, jfr. åvl. §§ 1 og 5 2. ledd. Det gjør at den opprinnelige opphavsmannen må samtykke til eksemplarframstilling og offentliggjøring av verket i den endrede skikkelsen, jfr. åvl. §§ 2 og 5 2. ledd. Derfor kalles avledede verk av og til avhengighetsverk.

I mange åpen kildekode-lisenser, særlig de med copyleft, står begrepet «derivative work» sentralt. Et «derivative work» etter amerikansk rett faller i utgangspunktet inn under kategorien vi i norsk rett kaller avledede verk (altså inkluderes både endrede og bearbeidede verk i begrepet)¹⁴⁹.

Se også argumentasjonen i underkapittelet «(U)selvstendige verk» i kapitlet over om fellesverk.

148 Se RG 1999 s. 330 (MOCS) og Lassen, 1983, s 329

149 Se Sterling, 2003, s. 242 til 244

Se 17 USC § 101: «A 'derivative work' is a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or *any other form in which a work may be recast, transformed, or adapted.* ...»

Se også US Copyright Office Circular 14

3.13.1 Skillet mellom avledede verk og nye selvstendige verk

De frie elementene i det opprinnelige verket (som ideer etc.) kan fritt benyttes slik at nye og selvstendige verk oppstår, jfr. åvl. § 4.1. Da vil den eller de som har laget dette nye verket alene ha opphavsretten. En kan derimot ikke benytte seg av de individuelle trekk beskyttelsen er knyttet til i det opprinnelige verket hvis en vil laget et nytt ikke-avledet verk. Benytter en seg av disse vernede elementene oppstår det et avledet verk.

Altså kan et verk (A), som i all hovedsak er laget uten å basere seg på et annet verk (O), bli sett på som et avledet verk hvis det inneholder et lite, men beskyttet element fra det andre verket (O). For programmer betyr dette at å kopiere noen få linjer kode fra et program O til et program A kan være nok til at A blir sett på som et avledet verk av O *hvis* linjene til sammen har verkshøyde¹⁵⁰.

Mange åpne programvareprosjekter er under konstant utvikling. En kan derfor spørre om en gjennom bearbeidinger og endringer kan «arbeide ut» tidligere opphavsmenn, altså at det oppstår et nytt og selvstendig verk gjennom endringer og bearbeidinger. Dette ble anført av Philips i «MOCS»-dommen¹⁵¹. Retten avviste ikke muligheten, imidlertid fant den at det i denne saken ikke var foretatt tilstrekkelige omfattende endringer og bearbeidelser til at et nytt verk hadde oppstått. Så lenge vernede elementer fra en opphavsmanns verk er i behold i det endrede programmet, vil han ha opphavsrett til disse.

3.13.2 Avledede verk ved relisensiering av programmer

Grensen mellom bearbeidelse og nye selvstendige verk er aktuell ved relisensiering; Dvs. når en ønsker å skifte lisens for et program, f.eks. fra proprietær lisensiering til GPL-lisensen. Da vil situasjonen kunne være den at det er deler av programmet hvor en selv ikke har opphavsrett til delen, og at opphavsmannen til disse delene ikke vil eller kan gi tillatelse til å benytte den nye lisensen på delen. At han ikke vil eller kan gi denne tillatelsen kan ha mange årsaker: han kan være uenig i den nye lisensen, han kan være død, han kan være flyttet til en ukjent adresse etc. Når et prosjekt har tusenvis av bidragsytere og det pågår over et langt tidsrom blir disse problemstillingene fort aktuelle.

150 Ofte vil noen få linje ikke være tilstrekkelig, jfr. partisjonering som er omtalt over.

151 Se RG 1999 s 300:

«Philips har påberopt seg at Philips etterhvert foretok så mange bearbeidelser og endringer av MOCS at det oppsto et nytt verk som Philips har opphavsrett til. ... Lagmannsretten kan etter bevisførselen ikke se at Philips kan høres med at man har bearbeidet MOCS på en så omfattende måte at det er oppstått et nytt, uavhengig åndsverk som Philips kan nyte sitt eget opphavsrettslige vern for.»

Da blir det aktuelt å kode disse delene på nytt. I denne situasjonen er det lett for programmereren av den nye delen å bygge på det tidligere verkets vernede elementer. Gjør programmereren det vil det oppstå en bearbeidelse som opphavsmannen til den gamle delen har rettigheter til (et plagiat) og bruk av dette uten hans tillatelse vil innebære krenkelse av opphavsretten (vanhjemmel). Dette kommer jeg tilbake til nedenfor under plagiat.

3.14 Særlig om forholdet mellom kildekode og programmet på objektform

Som nevnt ovenfor vil en kompilere kildekoden når en ønsker den over på objektform. Da oppstår spørsmålet om det foreligger en bearbeidelse av verket. Ved bearbeidelse kan bearbeideren få opphavsrett til selve bearbeidelsen, jfr. åvl. §§ 1 1. ledd og 4 2. ledd.

Når kompilatoren omgjør kildekode til objektkode følger den faste regler, som vi ikke trenger å gå inn på her. Siden denne prosessen er fast definert, og derfor ikke representerer noen skapende åndsvirksomhet, representerer den ikke en bearbeidelse, men en endring¹⁵². Det er derfor ingen bearbeidelse noen (f.eks. kompilatorprodusenten) kan ha opphavsrett til. Kildekoden fremstår derimot i en annen skikkelse (objektkode) og er beskyttet av eneretten til de(n) som har opphavsrett til kildekoden, jfr. åvl. § 2.

Etter kompileringen vil objektkoden, som nevnt over, kunne bli linket med annen objektkode, bl.a. for å gi nødvendig «infrastruktur» for at programmet skal kunne kjøre. Ved statisk linking vil nok ikke verkene for praktiske formål kunne atskilles, det oppstår da et endret verk. Altså et avledet verk (derivate work) som faller inn under GPLv2s copyleft bestemmelse som jeg vil omtale nedenfor. Dynamisk linking vil jeg behandle nedenfor.

3.15 Særlig om kombinasjon av moduler og bruk av grensesnitt

Et særlig spørsmål oppstår når et program, eller en modul i et program, «snakker» med andre programmer og moduler.

152 Se Bing, 1985, s. 39 og mer generelt Rognstad, 2004 s. 20

Mekanismen en modul bruker for å kommunisere med andre moduler kalles et grensesnitt¹⁵³. (Dette må ikke forveksles med f.eks. et brukergrensesnitt, derfor bruker noen begrepet tekniske grensesnitt.)

Grensesnittet er en abstraksjon av modulen som modulen har for kommunikasjon med omverdenen. I og med at grensesnittet er en abstraksjon for kommunikasjon mellom moduler vil grensesnittet ha metoder for ekstern kommunikasjon, mens det skjuler metoder for intern kommunikasjon inne i moduler. Dette gjør bl.a. at modulen kan bli endret på innsiden uten at det endrer måten kommunikasjonen skjer på med omverdenen. Grensesnittet et program bruker for å kommunisere med operativsystem eller med et bibliotek kalles gjerne et API¹⁵⁴.

Grensesnitt er en ganske abstrakt kategori. Det kan variere fra det svært enkle, som at et program bruker en parameter når det starter et annet program, til komplekse protokoller.

Når en implementerer at en modul skal kommunisere med en annen modul, må en nødvendigvis benytte seg av opplysninger om måten de kommuniserer på, dvs. grensesnittet. Et aktuelt spørsmål er da i hvilken grad slike opplysninger er opphavsrettslig vernet. Hvis en benytter seg av vernede elementer vil jo det verket disse elementene inngår i kunne være en bearbeidelse.

Dette spørsmålet er interessant i forbindelse med åpen kildekode fordi mange åpen kildekodeprogrammer og biblioteker blir benyttet som «byggeklosser» for andre programmer, og, som vi skal se, omfatter copyleft bl.a. avledede verk.

Det er ingen særregel for grensesnitt i opphavsrettslovgivningen. Det må foretas en konkret vurdering av hvert enkelt grensesnitt. Normalt vil grensesnittene være uten vern¹⁵⁵.

Siden abstrakte regler og metoder i seg selv ikke er opphavsrettsbeskyttet vil en kunne benytte disse konseptene for å lage (uttrykke) et kompatibelt grensesnitt uten at det oppstår et avledet verk.

Utformingen av grensesnittet mot en modul vil normalt være diktert av spesifikasjonene til den modulen det skal interageres med. Når utformingen av

153 EDB-direktivet 19/250/EEC definerer grensesnitt slik «the parts of the program which provide for ... interconnection and interaction between elements of software and hardware» i fortalen. Fokuset mitt i denne oppgaven er bare på software.

154 Application Programming Interface

155 Se tilsvarende syn Vinje, 1992, s. 91

grensesnittet er diktet av funksjonalitet vil implementeringen av grensesnittet heller ikke være vernet¹⁵⁶.

Åvl. § 39i gir en ufravikelig rett¹⁵⁷ til på visse vilkår å foreta nødvendig eksemplarframstilling i forbindelse med omvendt utvikling («reverse engineering») for at det skal bli mulig å lage kompatible programmer. Eksemplarframstillingen er etter bestemmelsen bare tillat «for å skaffe tilveie ... opplysninger», jfr. § 39i. Bestemmelsen gir derfor ikke eksplisitt en rett til å benytte opphavsrettsbeskyttet materiale i et annet program¹⁵⁸.

Hensikten med bestemmelsen er å sikre interoperabilitet, og bestemmelsen viser at dette fra lovgiverhold er verdsatt. Det ville gitt lite mening å ha denne bestemmelsen for så å ødelegge virkningen av den med en for sterk beskyttelse av grensesnitt.

Det må derfor være rimelig å anta at i tilfeller der grensesnittet er nær den nedre grensen for hva som har verkshøyde, vil en kunne gå i retning av at grensesnittet ikke er vernet for å sikre målet om interoperabilitet¹⁵⁹.

Reelle hensyn i forhold til konkurranse kan også tale for at en ved vurderingen av om en implementering av et grensesnitt strider mot opphavsretten bør være liberal med hva en tillater. En for sterk beskyttelse av grensesnitt vil hindre konkurrenter i å produsere programmer som kan integreres med andre programmer.¹⁶⁰ Mens en svak eller fraværende beskyttelse vil øke muligheten

156 Tilsvarende Vinje, 1992, s. 90 og Riis, 2001, s. 40

Se COM(88) 816 final 17 Mars 1989 paragraf 3.13 «If similarities in the code which implements the ideas, rules or principles occur as between interoperative programs, due to the inevitability of certain forms of expression, where the constraints of the interface are such that in the circumstances no different implementation is possible, then no copyright infringement will normally occur, because in these circumstances it is generally said that idea and expression have merged»

157 Se § 39i 3. ledd

158 Dette blir understreket av 2. ledd bokstav c som forbyr at opplysningene «nyttes til utvikling, framstilling eller markedsføring av et datamaskinprogram som vesentlig svarer til det opprinnelige i sin utforming, eller til *noen annen handling som krenker opphavsretten til programmet.*»

159 Se Vinje, 1992), s. 90 for samme syn i forhold til grensesnitt og dekompilering.

160 Se Samuelson, 1999, s. 586: «Copyright protection for program interfaces and outlawing decompilation would insulate certain leading software and hardware companies, such as Microsoft and IBM, from certain kinds of competition. There is, however, a fairly strong consensus among economists who have studied intellectual property rights issues in the software industry that high protectionist rule would harm competition and follow-on innovation in the software industry out of proportion to any assuring social benefit»

Se også Riis, 2001, s 40

til å lage kompatible moduler og vil kunne føre til økt konkurranse.¹⁶¹ Hvor aktuell konkurransefordelen er må selvsagt vurderes i hvert enkelt tilfelle.

3.15.1 Dynamisk linking

Dynamisk linking er, som sagt, en av metodene som brukes for å «binde sammen» programmoduler. I forbindelse med åpen kildekode er ofte skillet mellom dynamisk og statisk linking diskutert i forhold til copyleft. Allikevel vil avgjørelsen av hva som faller inn under copyleft til sist basere seg på opphavsrett og avtalerett heller enn disse tekniske konseptene. Begrepene statisk og dynamisk linking er ikke nevnt i GPLv2.

Hvordan dynamisk linking fungerer i detalj varierer fra plattform til plattform. Ved dynamisk linking blir, som sagt, modulen ikke en del av programfilen. Derimot vil programfilen inneholde deler av grensesnittet til modulen, som navn på funksjoner etc. Dette er nødvendig for at modulene skal kunne virke sammen. Disse delene av grensesnittet vil nok typisk i seg selv ikke ha verks-høyde, jfr. over. Derfor er nok ikke dynamisk linking i seg selv nok til at hovedprogramfilen kan betegnes som et avledet verk av modulen det linkes til.

Hadde kommunikasjonen mellom to moduler skjedd med f.eks. med en kompleks protokoll, som en for å implementere må benytte vernede elementer fra, er det mulig at situasjonen blir annerledes. I hvor stor grad protokoller kan være vernet er noe usikkert, se over om opphavsrett til grensesnitt, men mye taler for at de vanligvis ikke vil være det.

Velger en å vektlegge hovedprogrammets *tekniske* avhengighet av modulen (i motsetning til opphavsrettslig) vil løsningen kunne bli at hovedprogrammet blir et avledet verk. Dette er antakelig ikke riktig, jfr. diskusjonen over.

Konklusjonen må derfor bli at som hovedregel vil dynamisk linking av moduler *i seg selv* ikke innebære at hovedprogrammet er å anse som et avledet verk av biblioteket, men en kan ikke kategorisk utelukke at det kan være situasjoner der hovedprogrammet kan sees på som et avledet verk av modulen. Avgjørelsen beror til sist på et spørsmål om opphavsrettslig kategorisering heller enn teknisk. Disse spørsmålene har imidlertid ikke blitt utforsket i norsk rettspraksis og løsningen er derfor noe usikker.

¹⁶¹ Se Samuelson, 1999, s. 586

Problemstillingen var oppe i Microsoft v Commission (T-201/04). Saken vier at en tillegger interoperabilitetshensynet og konkurransehensynet stor vekt. I saken vurderer en imidlertid ikke om grensesnittene er vernet av opphavsrett eller ikke.

3.16 Friere enn fri? «The public domain»

Verk kan være i «det fri», eller i «the public domain», som det kalles på engelsk. Når verket er i «det fri» er det ikke opphavsrett knyttet til verket. Før åpen kildekodelisenser ble populære ble, som sagt, programmer ofte lagt¹⁶² i «the public domain». Siden en del åpen kildekodeprogrammer vil kunne benytte seg av deler av programmer «i det fri» vil jeg se nærmere på dette.

At et verk¹⁶³ er i «det fri» kan skyldes hovedsakelig tre grunner:

1. Vernetiden er forbi. Dette er lite aktuelt for dataprogrammer siden verntiden er så lang. Av åvl. § 40 følger det «Opphavsretten varer i opphavsmannens levetid og 70 år etter utløpet av hans dødsår. For [fellesverk] ... regnes de 70 år fra utløpet av den lengstlevende opphavsmanns dødsår.»
2. Verket ikke kan vernes pga sin natur (f.eks. en lov)¹⁶⁴. Dette er også lite aktuelt for dataprogrammer.
3. Opphavsmannen har oppgitt sin rett (abandonering eller dereliksjon)¹⁶⁵ En opphavsmann kan antagelig oppgi retten sin til verket ved å erklære at det er fritt for alle¹⁶⁶. Dette er faktisk ikke så upraktisk for dataprogrammer som en kunne tro siden, som sagt, dedikering til «the public domain» ofte ble gjort før åpen kildekodelisenser ble så utbredt. Opphavsmannen kan selvfølgelig bare oppgi den delen av opphavsretten som ikke er overdratt til andre. Siden opphavsretten er så stekt knyttet til opphavsmannen er det ikke rom for okkupasjon, som f.eks. kan tenkes ved løsøre. De ideelle rettighetene kan antakelig ikke abandoneres i sin helhet, jfr. åvl. § 3 2. ledd som sier at opphavsmannen ikke uavgrenset kan «fraskrive seg» retten etter 1. ledd. Retten til abandonering er noe usikker i andre land¹⁶⁷ og av

162 Tidligere var det nødvendig å registrere verk i USA for at de skulle være opphavsrettsbeskyttet. (Dette var før Berne Convention Implementation Act of 1988, i kraft 1. mars 1989.) Unnlatelse av å registrere verket ville føre til at verket ble i «det fri». I dag må en aktivt legge verket i «det fri».

163 For å være et verk må det, som sagt ha, verkshøyde. Uttrykk som ikke har verkshøyde kan også bli sett på som å være i «det fri».

164 Se f.eks. åvl. § 26

165 Begrepene brukes om hverandre. Abandonering brukes her i tingsrettslig betydning, ikke konkursrettslig.

166 Se Knoph, 1936, s.147. «Svært stor praktisk betydning har det heler ikke at opphavsmannen kan gi op sin rett, og erklære at hans verker er fri for hvermann. Leo Tolstoi gjorde det ganske visst i sin ti, men noen norsk Tolstoi har såvidt jeg vet ikke forekommet. Og skulde han dukke op, sier det sig selv at det bare er de delene av retten som ikke er overdradd til andre, han har anledning til å stille til disposisjon for almenheten gjennom «dereliksjon»

167 Se Greenleaf, 2007

noen antatt å ikke eksistere¹⁶⁸. I så fall er det nok naturlig å se på en dedikering av et verk til «det fri» som en i utgangspunktet simpel, ubetinget og ubegrenset lisens til verket for alle. Dette fordi det er liten grunn til å hindre opphavsmannen å fritt råde over sitt verk.

Beskyttelse etter opphavsretten faller antakelig ikke bort selv om opphavsmannen med vilje lar være å bruke retten¹⁶⁹. Loven oppstiller ikke noe krav om at opphavsmannen må benytte retten. Derfor kan ikke såkalt «abandonware» utnyttes lovlig i åpen programvare med mindre ett av vilkårene over er oppfylt eller en får en tillatelse av den berettigede.

Opphavsmannens passivitet i forhold til en (faktisk eller juridisk) person i et konkret tilfelle vil noen ganger kunne anses som en tildeling av lisens. I tillegg kan passivitet i konkrete tilfeller tale for at opphavsmannen taper sin rett til vindikasjon. Jeg kommer tilbake til det siste under temaet ekstinksjon.

168 Se Rosen, 2002. Synspunktet er antakelig feil, se f.eks. Ninth Circuit Model Civil Jury Instructions 17.19

169 Se Knoph, 1936, s. 147. «Og heller ikke om ophavsmannen lar være å bruke retten, d.v.s ikke holder verket offentlig tilgjengelig, faller beskyttelsen bort»

4 ÅPEN KILDEKODELISENSER VS. PROPRIETÆRE LISENSER

4.1 Kjennetegn for proprietære lisenser

Den lisensformen flest er kjent med når det gjelder lisensiering av programmer er nok den proprietære, siden den lenge har vært den dominerende. Ved proprietære lisenser får kunden vanligvis en rett til å bruke programmet mot betaling¹⁷⁰. Betalingen kan f.eks. skje i form av en enkeltsum eller i form av abonnement, og den kan være avhengig av bruksområde og intensitet. Den proprietære modellen er også kjennetegnet ved at kunden vanligvis kun får tilgang til programmet på objektform. Ved proprietær lisensiering får kunden ofte ingen rett til å tilgjengeliggjøre programmet for allmennheten eller bearbeide programmet. Det kan også være begrensninger i hvilke sammenhenger kunden kan bruke programmet, f.eks. bare til undervisningsformål.

4.2 Mylderet av lisenser

Akkurat som det er et utall proprietære lisenser, er det mange forskjellige åpen kildekode-lisenser. Disse har ganske forskjellig innhold.

Å avgjøre hvilken åpen-kildekode-lisens som er mest brukt er ingen enkel oppgave. Ofte blir statistikk oppgitt, men uten kilde. Går en inn og ser på statistikken fra de store samarbeidsportalene for åpen kildekode, ser en at GPL-lisensen versjon 2 antakelig for tiden er den største¹⁷¹. Jeg har derfor, som sagt, i denne fremstillingen valgt å se særlig på GPL, men jeg vil også se på

¹⁷⁰ Unntak er f.eks. freeware

¹⁷¹ Statistikk hentet fra <http://freshmeat.net/stats/> den 27. september 2007. GPL-lisensen ble brukt av 64.87% av de 43,575 prosjektene som er på Freshmeat. Dette er ikke en helt god måte å måle på. Problemet med metoden er at prosjekter med mange brukere teller like mye som de med få. I tillegg er det mange som benytter andre tjeneste (som <http://sourceforge.net/>) eller har sine egne løsninger. Det siste gjelder ofte prosjekter med mange brukere.

andre lisenser når de er illustrerende. Jeg vil da særlig se på (den endrede)¹⁷² BSD-lisensen. BSD-lisensen er på mange måter en direkte motsetning til GPL, selv om den også er en åpen kildekode-lisens, siden BSD-lisensen inneholder få forpliktelser for lisensstakeren i forhold til GPL. BSD lisensen blir derfor ofte kategorisert som «tillatende» («permissive»).

GPL lisensen er kommet i en ny versjon, versjon 3. Siden den foreløpig (i 2007) er mindre brukt, vil jeg ha hovedfokus på versjon 2, men jeg vil trekke inn versjon 3 for illustrasjon. For å holde de forskjellige variantene av GPL lisensen fra hverandre vil jeg omtale dem som GPLv2 og GPLv3.

4.3 Kjennetegn for åpen kildekode-lisenser

Åpen kildekode-lisenser skiller seg fra de proprietære lisensene på flere punkter. De viktigste egenskapene som er annerledes ved åpen kildekode-lisenser vil jeg gå gjennom nedenfor.

4.3.1 Tilgang til kildekoden

Åpen kildekode-lisenser gir brukeren tilgang til kildekoden. Avhengig av lisensen vil de som tilgjengeliggjør et program i objektform for allmennheten enten ha mulighet til, eller ha plikt til, å gjøre kildekoden til det åpen-kildekodeverket de benytter tilgjengelig for andre. Lisensstaker kan også ha plikt til å gjøre tilgjengelig kildekoden til endringer han selv har gjort på det åpne programmet eller kildekoden til programdeler som blir satt sammen med det åpne programmet. Dette kommer jeg tilbake til under overskriften copyleft.

GPLv2 regulerer tilgang til kildekode i artikkel 3 og oppstiller en plikt til å gjøre kildekoden tilgjengelig når en tilgjengeliggjør verket i objektform for allmennheten¹⁷³. BSD-lisensen har ikke regulert forholdet, og det foreligger derfor ingen slik plikt.

Objektformen av programmer er ofte vanskelig og tidkrevende å skjønne. Som en analogi kan en si at å forandre et program uten kildekoden er som å prøve å forandre en kake når en bare har kaken og ikke kakeoppskriften. Tilgang til kildekoden gjør at en kompetent bruker bl.a. får mulighet til bedre å forstå programmet, får mulighet til å endre og forbedre programmet,

¹⁷² Den opprinnelige lisensen BSD inneholdt en reklameklausul som i den nye versjonen av lisensen er blitt fjernet. Den gamle brukes sjeldent i dag. Når jeg refererer til BSD-lisensen vil jeg mene den nye hvor reklameklausulen er fjernet.

¹⁷³ Et skriftlig tilbud kan være nok, jfr. GPLv2 artikkel 3 bokstav b og c

og får mulighet til å kontrollere programmet for mangler (f.eks. sikkerhetsmangler). Tilgang til kildekode gjør også at en kan lære av tidligere arbeid. Komplexiteten til programmet og tiden brukeren har til rådighet vil selvsagt sette grenser for dette.

Tilgang til kildekode gjør det også enklere å lage andre programmer som er kompatible med programmet. Dvs. det gjør det enklere å lage programmer som kan utveksle informasjon, f.eks. ved at programmene «snakker» direkte med hverandre, ved at de bruker samme filformat etc. En kan på denne måte unngå fastlåsing («lock-in») til en bestemt leverandørs formater. Tilgang til kildekode vil derfor kunne fremme interoperabilitet.

4.3.2 Mulighet for endring

At lisensen tillater endringer (avledede verk) gjør at en kan videreutvikle programmet, foreta feilrettinger etc. Muligheten til å foreta endring henger tett sammen med tilgangen til kildekode. Uten tilgang til kildekode vil det, som sagt, være svært upraktisk å foreta mange former for endringer av programmet.

Ulempen med en liberal rett til endring er at det kan oppstå inkompatible versjoner av programmet når noen lager egne versjoner. (Se også «forking» nedenfor). Copyleft vil imidlertid kunne «bøte» på dette, siden de som ønsker å lage kompatible versjoner kan få rett til å benytte endringene som førte til inkompatibiliteten.

Strengt tatt (og høyst teoretisk) faller det å endre et program utenfor opphavsmannens enerett. Så lenge det ikke skjer en eksemplarframstilling eller en tilgjengeliggjøring for allmennheten av vernede elementer skjer det ikke ett opphavsrettsbrudd. Siden det å kjøre programmet innebærer en eksemplarframstilling, og opphavsmannen har enerett til å fremstille eksemplarer i endret skikkelse,¹⁷⁴ må en i praksis ha tillatelse for å kunne endre programmet. Åvl. § 39h gir imidlertid rettmessig bruker mulighet til å foreta feilrettinger innenfor visse grenser uavhengig av hva lisensen skulle tilsi.

GPLv2 tillater avledede verk i artikkel 2 på nærmere angitte vilkår (bl.a. copyleft). BSD lisensen gir også tillatelse til avledede verk når den sier at «Redistribution and use ... with or without modification .. are permitted...».

4.3.3 Liberal rett til eksemplarframstilling og tilgjengeliggjøring

Å distribuere et program over Internett vil innebære at lisensgiver gjør verket tilgjengelig på Internett (fremfører verket for allmennheten) og at lisenstaker

174 Jfr. åvl. § 2 1. ledd

laster ned programmet (eksemplarframstilling). Begge deler krever i utgangspunktet opphavsmannens tillatelse, jfr. åvl. § 2.

Åpen kildekode-lisenser gir derfor bl.a. distributøren rett til å gjøre programmet tilgjengelig for allmennheten og nedlasteren (den nye lisenstakeren) rett til eksemplarframstilling. For GPLv2-lisensen følger det av artikkel 1 til 3, og for BSD-lisensen følger det av « Redistribution ... with or without modification ... [is] permitted...»

4.3.4 Liberal rett til bruk

Siden bruk innebærer en eksemplarframstilling, som nevnt over, må en ha tillatelse til å kunne bruke programmet, hvis ikke strider det mot opphavsmannens enerett til eksemplarframstilling, jfr. åvl. § 2

Åpne kildekode-lisenser gir vanligvis lisenstaker rett til å bruke programmet uten å begrense bruken til et bestemt formål eller en bestemt persongruppe.

I GPLv2 artikkel 0 står det: «Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted ...». Dette må nok forstås slik at lisenstaker får en tillatelse til å bruke programmet, men at lisenstaker ikke blir forpliktet i forhold til de andre delene av GPLv2-lisensen når han kun kjører programmet.

GPLv3 artikkel 2 regulerer dette klarere «You may .. run ... covered works that you do not convey, without conditions so long as your license otherwise remains in force.»

Av BSD lisensen følger den liberale bruksretten av «use ... [is] permitted...»

4.3.5 Gratis?

Åpen programvare er vanligvis gratis å lisensiere. Dette står i kontrast til mange former for proprietær programvare¹⁷⁵. Dette er av stor betydning når en trenger mange lisenser til et program eller når alternative programmer er dyre. At lisensen vanligvis er gratis er selvfølgelig et stort incentiv til å bruke åpen programvare i forhold til dyrere alternativer. For å få det fulle bildet må en selvsagt se på total eierskapskostnad (TCO)¹⁷⁶.

Av GPLv2 artikkel 1 2. ledd følger det «You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.» En kan altså ta seg betalt for å tilgjengeliggjøre programmet. Har en først tilgjengeliggjort objekt-koden, kan en imidlertid ikke ta seg betalt mer enn kostpris for å tilgjengeliggjøre kilde-koden. Dette

¹⁷⁵ Gratis proprietær programvare kalles gjerne «freeware»

¹⁷⁶ Total cost of ownership

er nærmere regulert i GPLv2 artikkel 3 bokstav b. Av BSD-lisensen følger det ingen begrensninger i prisen som kan tas. P.g.a. de liberale rettighetene til eksemplarframstilling og tilgjengeliggjøring som lisenstaker får til åpen kildekodeprogrammer, vil det normalt på sikt ikke fungere å ta betalt for programvaren. Lisenstaker kan imidlertid ha behov for å utnytte programmet på andre vilkår enn de åpen kildekode-lisensen tilbyr, og da blir vederlag fort aktuelt. Dette kommer jeg tilbake til under kapittelet om dobbelt lisensiering.

Det blir ofte understreket at fri programvare er «free as in free speech, not as in free beer». Dette gjelder filosofien bak fri programvare. Det refererer til de liberale rettighetene fri programvare gir lisenstaker. At f.eks. GPL-programvare vanligvis er gratis er derfor ikke det som det blir siktet til med betegnelsen fri programvare.

4.3.6 Forgreninger - «forking»

Kombinasjonen av å få tilgang til kildekode, rett til lage avledede verk og å kunne tilgjengeliggjøre disse for allmennheten innebærer at enhver i teorien kan «overta» utviklingen av et åpen kildekodeprosjekt. Å kopiere kode fra et prosjekt for så å starte et nytt prosjekt rundt denne kalles gjerne «forking».¹⁷⁷

Dette har flere praktiske effekter. Er ledelsen i prosjekt ikke god, kan andre starte et alternativt prosjekt. Dette kan ha en demokratiserende effekt innen prosjektet.

«Forking» innebærer en mulighet for å komme seg ut av en fastlåst situasjon i prosjektet. F.eks. dersom de som tar i mot ny kildekode, dvs. de som har «commit access»¹⁷⁸, slutter å arbeide med prosjektet uten at noen andre får rettigheter til å ta imot ny kildekode. I slike situasjoner kan andre ta over prosjektet gjennom «forking».

En annen effekt av «forking» er at en er tryggere hvis utvikleren slutter å utvikle programmet (f.eks. de går konkurs). Når en har tilgang til kildekode og rett til endring av kildekode vil brukeren kunne betale andre for å videreutvikle produktet, evt. gjøre det selv. Er det et levende samfunn rundt programmet, kan det også være at samfunnet vil videreutvikle det. Escrow-avtaler¹⁷⁹ blir her fort overflødige. Selv om en har en mulighet for «forking»

177 En kan også ha flere forker innenfor et prosjekt.

178 Som sagt, bruker en gjerne versjonkontrollsystemer for å hold styr på kildekode i et prosjekt. Mange, ofte alle og enhver, kan ha tilgang til å laste ned kildekode fra versjonkontrollsystemet. Derimot er det vanligvis begrenset hvem som kan laste opp ny kode til depoet («the repository»). De som har slik tilgang har «commit access».

179 For programvare innebærer gjerne Escrow-avtaler at en tredjepart får kildekode til et program og oppbevarer den. Hvis utviklingen av programmet skulle opphøre vil de som har rett etter Escrow-avtalen kunne kreve koden utlevert f.eks. for å kunne vedlikeholde programmet.

kan begrensninger i kompetanse, tid og økonomiske resurser selvsagt gjøre denne muligheten mindre verdt for enkeltbrukere.

For prosjekter kan «forking» være uheldig siden det deler opp det potensielle utviklingsmiljøet¹⁸⁰.

180 Se Raymond, 2000, «Homesteading the Noosphere», s. 5

5 LOVVALG OG JURISDIKSJON

Åpen kildekodeprosjekter har ofte en global natur med både utviklere og brukere fra hele verden. En er derfor nødt til å se på problemer i forhold til lovvalg og jurisdiksjon. Dette er store spørsmål som det vil falle for langt å gå grundig inn på her. Jeg vil derfor kun gi en kort oversikt.

Jurisdiksjon er spørsmålet om en domstol i et land vil akseptere saken eller avvise den. Velger den å akseptere saken må den avgjøre hvilket lands lov den skal følge for å avgjøre jussen i saken (*lex causae*). Det kan være domstol-landets rett (*lex forum*) eller et annet lands rett. Prosessspørsmål, inkludert bevisrett, følger *lex forum*.¹⁸¹ Reglene for jurisdiksjon og lovvalg er nasjonale, selv om det finnes noen folkerettslige avtaler.

5.1 Jurisdiksjon

Spørsmålet om jurisdiksjon blir avgjort av nasjonal rett, men resultatet kan følge av både norske og folkerettslige rettsregler. Tvisteloven viker for folkerettslig regulering (altså sektor-monisme), jfr. tvl. § 1-2.

Særlig er Lugano-konvensjonen aktuell for EØS-området¹⁸². Denne er inkorporert i norsk rett¹⁸³.

For saker om kontraktsforhold er saksøktes domisil (bopel), artikkel 2, eller oppfylleelsesstedet for kontrakten, artikkel 5 (1) aktuelle¹⁸⁴ som steder med jurisdiksjon. I tillegg er det mulig, på nærmere vilkår¹⁸⁵, å avtale vernetting etter Luganokonvensjonens art 17.

For saker om erstatning utenfor kontrakt, dvs. etter opphavsretten, er artikkel 5 (3) aktuell. Etter bestemmelsen kan det saksøkes ved skadestedet, dvs. «det sted der skaden ble voldt eller oppsto»¹⁸⁶. Dette kan være flere steder og valget av sted er ikke åpenbart når noe fysisk ikke er skadet. Jeg lar disse

181 Se Bing, 2001, s.1

182 Liechtenstein er ikke med.

Konvensjonen bygger sterkt på Brussel-konvensjonen fra 1968 i EU. I dag har Brussel-forordningen (Council Regulation (EC) No. 44/2001) i stor grad avløst denne.

183 Lov av 8. januar 1993 nr. 21

184 Av artikkel 13 til 15 er det særregler for forbrukeravtaler.

185 Bestemmelsen får kun virkning når «minst en av [partene] har bosted i en Konvensjonstat».

I tillegg er det visse formkrav. Se også SACIS-kjennelsen, Rt. 1993 s. 1583.

186 Se artikkel 5 (3) i Lugano-konvensjonen

spørsmålene ligge.¹⁸⁷ I tillegg kan saksøktes domisil (bosted) benyttes¹⁸⁸ og en kan avtale (i etterkant) et vernetings¹⁸⁹.

Artikkel 5 (3) gjelder bare for «saker om erstatning utenfor kontraktforhold». Berikelseskrav etter åvl. § 55 kan derfor hevdes å falle utenfor.¹⁹⁰

Distributørene, brukerne og samarbeidet rundt åpen kildekodeprosjekter strekker seg svært ofte utenfor Luganokonvensjonens grenser. Hvis folkerettslige avtaler ikke gir svar, kan vi se på tvisteloven.

Partene kan for det første i utgangspunktet selv avtale et norsk vernetings¹⁹¹, jfr. tvl. § 4-6(1). Av tvl. § 4-3(1) følger det at «[t]vister i internasjonale forhold kan bare anlegges for norske domstoler når saksforholdet har *tilstrekkelig tilknytning til Norge*». Behovet for å få avgjort tvisten i Norge og risikoen for avvising i andre land er faktorer som kan spille inn i avgjørelsen.¹⁹² Vernetings følger da av tvl. §§ 4-4, 4-5 og evt. 4-3(2).

Både etter Luganokonvensjonen og tvisteloven er det som vist mulig å avtale jurisdiksjon. Dette blir vanligvis ikke gjort i åpen kildekodeavtaler. Dette skyldes antagelig at det kunne vært motvilje hos folk fra andre nasjoner enn den hvor vernetings ble lagt til. I tillegg er det nok for en del prosjekter vanskelig å velge en representativ nasjon. EUPL¹⁹³ er et unntak der ECJ¹⁹⁴ for visse tilfeller er avtalt, jfr. artikkel 14.

5.2 Lovvalg

Åpen kildekodetisenser vil, som sagt, ofte bli inngått med lisensgiver og lisensstaker i forskjellige land. Spørsmålet blir da: Hvilket lands lov skal retten følge? Dette kalles lovvalgsspørsmålet.

For prosessspørsmål vil, som sagt, *lex forum* blir brukt.¹⁹⁵ En må skille mellom lovvalget for kontraktsforholdet og lovvalget for opphavsrettens eksistens og innhold¹⁹⁶ da disse spørsmålene kan bli regulert av forskjellige lands lover.

187 Se nærmere f.eks. Bing, 2001, s. 31-33

188 Artikkel 2 i Lugano-konvensjonen og *Fiona Shevill and Others v Press Alliance SA*

189 Artikkel 17 i Lugano-konvensjonen, jfr. «har oppstått eller måtte oppstå»

190 Se Fuglesang og Krog, 1999, s. 45-50. Særlig side 50.

191 Formkrav følger av tvl. § 4-6(2) og begrensninger i forbrukerforhold av tvl. § 4-6(3).

192 Se NOU 2001 nr. 32 B Bemerkninger til § 4-2 i kapittel 4.5

193 EUPL er, som sagt, European Union Public Licence

194 European Court of Justice (EF-domstolen)

195 Se Bing, 2001, s.1

196 Se Kottvedgaard, 2005, s. 424

5.2.1 Lovvalg for krav basert på opphavsrett (opphavsrettsstatuttet)

Spørsmål om opphavsrettens eksistens, innhold og sanksjoner avgjøres i all hovedsak¹⁹⁷ etter handlingsstedets lov.¹⁹⁸ Begrunnelsen for dette er at det er naturlig at en nasjonal lovgiver kun kan regulere sitt eget territorium.¹⁹⁹ (Selv om det ikke er ukjent at en nasjonal lovgiver av og til regulerer situasjoner i utlandet).

Den norske lovgivningen om opphavsrett etablerer derfor enerett bare i Norge og ikke i andre land²⁰⁰. Hvis norske domstoler skal dømme om opphavsrettsinngrep i utlandet må de altså anvende gjerningsstedets lov.²⁰¹

Spørsmålet som reiser seg da er hvor er «gjerningsstedet»? Når er et opphavsrettsinngrep skjedd i Norge? For nettbaserte tjenester er svaret ikke åpenbart. Er handlingen gjort i Norge hvis den er styrt fra Norge? Hvis data er sent fra Norge (avsenderprinsippet), altså opplastet eller gjort tilgjengelig for nedlasting fra en server i Norge? Hvis data er mottatt i Norge (mottakerprinsippet)? Hvis utstyr er plassert i Norge? Disse spørsmålene er omstridte og ikke endelig løst. Det vil falle for langt å gå nøye inn på dem i denne oppgaven.²⁰²

Et neste spørsmål blir da hvilke verk som beskyttes i Norge og i hvilken grad. Fulle rettigheter etter loven får kort fortalt norske statsborgere, personer bosatt i rike, verk som utgis i Norge for første gang eller verk som utgis her innen 30 dager fra førstegangsutgivelsen i et annet land, jfr. åvl. § 57. Dette kalles umiddelbar virkning.

Opphavsmenn fra andre land kan falle inn under åvl. § 59 1. ledd. Dette kalles middelbar virkning. Beskyttelsen vil her kunne avhenge av hvilke folkerettslige avtaler opphavsmannens land har inngått og vernet det aktuelle landet gir for norske verk. En kan her merke seg kapittel VI i forskrift til åndsverkloven. § 6-2 innebærer at bestemmelsene i åndsverkloven, med noen

197 Et unntak kan tenkes f.eks. for spørsmålet om hvem som er rettsinnehaver. Se Koktvedgaard, 2005, s. 425.

Se også Rognstad, 2007, s. 420

198 Se Koktvedgaard, 2005, s. 424

Se Rognstad, 2007, s. 451 «Det synes stort sett å være internasjonal enighet om at rettsvalget i opphavsrettstvister, i hvert fall når det gjelder rettighetenes eksistens, omfang, varighet og sanksjonering, bestemmes av retten på handlingsstedet».

199 Se Knoph, 1936, s 166-167

200 Se Knoph, 1936, s 167 og Rognstad, 2004, s. 154

201 Se Rognstad, 2004, s. 154 og Lassen, 1997

202 Se f.eks. Bing, 2001 om problemer generelt i Europa og Svantesson, 2007

Se også Rognstad, 2007 der mange av argumentene for og mot de forskjellige prinsippene blir behandlet. Rognstad oppsummerer behandlingen med at «Mye taler ... for at mottakerlandets rett anses som det relevante tilknytningspunktet for tilgjengeliggjøring ved onlinetransmisjoner i norsk rett, i alle fall som et utgangspunkt. Det må dog være rom for å supplere utgangspunktet med den individualiserende metode.»

unntak²⁰³, verner verk som har tilsvarende tilknytning, som i § 57, til et land som er medlem av Bernunionen eller TRIPS avtalen under WTO. (De fleste praktisk relevante land er medlem i Bernunionen når jeg skriver dette.) Etter § 6-1 er det et videre vern for EØS-tilknyttede. Det faller for langt å gå inn på de nærmere spørsmålene i denne oppgaven.

En kan jo lure på hvorfor utlendinger ikke automatisk får samme vern som nordmenn, noe de f.eks. får ved eiendomsrett til fast eiendom. Regelen kan primært begrunnes i handelspolitikk og et ønske om gjensidighet; en ønsket at andre nasjoner skulle gi norske verk en like god beskyttelse som verk som ble skapt i den aktuelle nasjonen²⁰⁴. Knoph kobler også reguleringen opp mot opphavsrettens fødsel som et privilegium staten gav til visse (privilegerte) borgere og hvor utlendinger da stod svakt²⁰⁵.

5.2.2 Lovvalg for krav basert på kontrakt (kontraktsrettsstatuttet)

Når det gjelder spørsmål om lovvalg for kontraktsforholdet har avtalepartene i utgangspunktet frihet til å avtale hvilket lands lov som skal benyttes (partsautonomi)²⁰⁶. Dette gjøres ikke vanligvis i åpen kildekode-lisenser. Verken GPLv2 eller BSD lisensen har slike bestemmelser. Dette er antakelig begrunnet, som sagt over, med at hvis en hadde innført ett lands lov i avtalen ville personer uten tilknytning til det landet fort ikke ønsket å benytte avtalen. Og at det i tillegg ofte er vanskelig å finne et representativt land for prosjektet. EUPL er igjen et unntak, jfr. artikkel 15.

Vi baserer oss på den individualiserende metoden, eller nærhetsprinsippet²⁰⁷, når partsautonomien ikke er benyttet. Prinsippet går ut på at lovvalget beror på en konkret skjønnsmessig vurdering av hvilken stat rettsforholdet har nærmeste tilknytning til²⁰⁸. Denne regelen ble første gang anvendt i Norge i Irma-Mignon-dommen²⁰⁹ og kalles derfor gjerne Irma-Mignon-formelen.²¹⁰ Den senere Tampax-dommen²¹¹ danner grunnlaget for at denne formelen gjelder generelt som utgangspunkt i kontraktsforhold. En må ved den individualiserende metoden legge vekt på både subjektive og objektive

203 Se §§ 38c, 41a, femte kapittel og § 45c

204 Se Knoph, 1936, s. 168

205 Se Knoph, 1936, s. 168

206 Se Cordes og Stenseng, 1999, s. 282

207 Se Cordes og Stenseng, 1999, s. 55

208 Se Cordes og Stenseng, 1999, s. 54

209 Se Rt. 1932 II s. 59

210 Se Cordes og Stenseng, 1999, s. 55

211 Se Rt. 1980 s. 243

tilknytningmomenter.²¹² Generelt vil aktuelle momenter være kontraktsstedet, oppfyllelsesstedet, forretningsstedet / partenes bopel, voldgiftsklausuler, vernetingsklausuler og kontraktspråk.²¹³ Det går derfor ikke an å gi noe pre-sist generelt svar for situasjonen til åpen kildekodeavtaler, en konkret vurdering er nødvendig for den enkelte situasjon.

5.2.3 Lovvalg for erstatningskrav

Erstatningskravene jeg ser på i denne oppgaven er enten basert på opphavsrett eller på kontrakt. Når det gjelder erstatning basert på kontrakt og opphavsrett har jeg omtalt lovvalget over.

Andre erstatningskrav kan være utenfor kontrakt og opphavsrett, lovvalget for disse kan bli annerledes enn omtalt ovenfor²¹⁴, men jeg går ikke inn på det siden jeg ikke omtaler slike former for erstatning.

5.3 Risiko

Konsekvensene av de liberale rettighetene som en finner i åpne kildekode-lisenser er at åpen kildekodeprogrammer kan blir brukt over hele verden. Det betyr at lisensgiver potensielt blir kan bli holdt ansvarlig etter ethvert lands lov siden det kan potensielt være en lisenstaker i ethvert land. Dette betyr selvfølgelig en høy grad av usikkerhet for den rettslig stillingen. Bestemmelser for opphavsrett, obligasjons- og avtalerett vil variere fra land til land. Selv om vi har enkelte internasjonale avtaler som Bern-konvensjonen og TRIPS-avtalen som harmoniserer opphavsretten på et (noen ganger) abstrakt nivå vil det kunne være store forskjeller. Selv om land har implementert de samme bestemmelsene kan disse tolkes forskjellig i de forskjellige landene. Lisensgiver bør derfor være veldig forsiktig med å basere seg på de finere nyanser i jussen og å basere seg utelukkende på en nasjons rett.

²¹² Se Cordes og Stenseng, 1999, s. 280

²¹³ Se Cordes og Stenseng, 1999, s. 280-281

²¹⁴ Se Cordes og Stenseng, 1999, s. 311 og Cordes og Stenseng, 1999, s. 313-330

6 TOLKNING AV ÅPEN KILDEKODEAVTALER

6.1 Generelt om tolkning av åpen kildekode lisenser

Ved tolkning av avtaler bygger vi i Norge på en tolkingsart som gjerne kalles «intersubjektiv» tolkning²¹⁵.

Det betyr at for skriftlige avtaler tar en i utgangspunkt i en naturlig språklig forståelse av avtalen når en tolker den. Hva ville en fornuftig person i samme stilling som avtaleparten ha lagt til grunn?²¹⁶ Allikevel er det en viktig reservasjon: kan en etablere en felles oppfatning mellom partene som avviker fra den naturlige språklige forståelsen av avtalen vil denne oppfatningen gå foran²¹⁷.

Lisensiering av åpen programvare skjer ofte uten at lisensgiver og lisenstaker snakker sammen eller kommenterer lisensen nærmere enn ved å henvise til den. Det vil i disse tilfeller derfor ikke foreligge en annen felles oppfatning om lisensen mellom partene enn lisensen. En del lisensgivere til åpen kildekode gir nærmere detaljer på hjemmesidene sine om forståelsen av lisensen de bruker. Dette vil, avhengig av situasjonen, kunne spille inn på hva som er avtalt.

Nedenfor vil jeg skrive om noen aktuelle «tolkningsregler» eller «tolkningsprinsipper». Det er verd å merke seg at mange av disse har mer karakter av å være retningslinjer enn mange andre regler i jussen.

De fleste åpen kildekode lisensene er skrevet på engelsk og har sitt opphav i USA. Et spørsmål som reiser seg, når norsk rett skal anvendes, er om en skal benytte seg av den forståelse av begreper som er vanlig under amerikansk rett, eller om en skal legge til grunn den forståelsen som nordmenn vil ha av det engelske uttrykket? Er det en felles oppfatning mellom partene om spørsmålet, må denne legges til grunn.²¹⁸ Er det ikke noen felles oppfatning mellom partene, er praksis ikke helt klar, men antagelig må en bygge på en norsk forståelse når norsk rett anvendes²¹⁹. Den vil i alle fall for norske avtaleparter ligge nærmest

215 Se Hov, 2002, s 148

216 Se Hov, 2002, s.151 og kjl. § 89 1. ledd 2. punktum (den kan imidlertid ikke direkte anvendes på alle avtaler)

217 Se Hov, 2002, s 148 og 150

218 Se Hov, 2002, s 154

219 Se Hov, 2002, s 154-155 Hov oppsummere på side 155: «Den praksis som foreligger, trekker etter min oppfatning nokså entydig i retning av at en skal legge til grunn den tolkningen som ville vært naturlig etter språket i det land hvis rett anvendes».

en naturlig språklig forståelse. På den annen side kan en hevde at den som inngår en avtale på et fremmed språk bør bære risikoen for det.²²⁰ GPLv2 bruker begreper som stammer fra amerikansk opphavsrett. Det står f.eks. i artikkel 0 «any derivative work under copyright law»²²¹. Kan en her bruke det norske begrepet avledet verk når en anvender norsk opphavsrett? Antagelig må en på grunn av konklusjonen over kunne bruke det norske begrepet. Siden Norge og de fleste andre land, inkludert USA, er underlagt bl.a. Bern-konvensjonen og, noen færre land, TRIPS-avtalen, vil ikke alltid konsekvensene blir så store på et noe abstrakt nivå.

Åpen kildekode-lisenser blir vanligvis ikke individuelt forhandlet frem mellom lisensgiver og lisenstaker, de er altså standardavtaler. For GPL avtalene sin del er de laget av Free Software Foundation, men de fleste lisensgivere og lisenstakere som benytter lisensen har ingen tilknytning til organisasjonen.

Ved tolkning av avtaler er det et prinsipp at en tolker til ulempe for den som kan bebreides for uklarheter i kontrakter.²²² Han kunne jo sørget for at avtalen ble klar. Dette kan tale for å tolke lisensen mot lisensgiver siden han vanligvis har valgt standardavtalen.

Prinsippet om å tolke til ulempe for den som kan bebreides for uklarheten innebærer at en kan tolke imot forfatteren av avtalen (forfatterregelen)²²³. Mange leverandører av åpen kildekode benytter egne lisenser og da blir regelen aktuell, men ofte er forfatterne av lisensen, konsipistene om en vil, ikke en del av avtalen og forfatterregelen gir oss da lite.

Kyndighetsregelen tilsier at et utsagn ved uklarhet skal tolkes til ulempe for den kyndigste parten²²⁴. Han hadde best mulighet til å unngå uklarhetene. Kyndighetsregelen innebærer at hvis f.eks. et stort, profesjonelt selskap lisensierer programvare under en åpen kildekode-lisens til en vanlig hobbybruker, kan avtalen ved uklarhet tolkes til fordel for hobbybrukeren. Her er det verd å merke for avtaler med forbrukere at avtl. § 37(1) nr. 3 sier at standardavtaler ved uklarhet skal tolkes til forbrukerens fordel når avtalemotparten er en næringsdrivende. En del åpen kildekode-avtaler inngås nettopp med forbrukere.

220 Se Hov, 2002, s. 154

221 Se 17 USC

222 Se Hov, 2002, s. 158-159

223 in dubio contra stipulatorem

224 Se Hov, 2002, s. 158-159

6.2 For- og etterarbeider til avtalene

Under utviklingen av GPLv3 har det foreligget forskjellige utkast og mange kommentarer. Et spørsmål blir da hvilken vekt en skal tillegge slike «forarbeider». Forarbeider til standardavtaler²²⁵ har særlig blitt vektlagt når det har vært snakk om avtaler som er blitt til i samarbeid mellom interessentene. Slike avtaler kalles gjerne «agreed documents».

Under utviklingen av GPLv3 ble det laget et eget nettsted²²⁶ som la de forskjellige utkastene av avtalen ut på «høring», slik at interesserte på Internett kunne kommentere på utkastene. Muligheten for alle til å kommentere er allikevel ikke det same som grundige forhandlinger mellom interessegruppene. En kan derfor vanskelig si at GPLv3 fremstår som et «agreed document».

Å ha oversikt over slike kommentarer som er blitt gitt i prosessen og få tilgang til dem er ingen lett oppgave. Det er derfor gode grunner til å tillegge dem lite eller ingen vekt på det generelle planet.

Free Software Foundation (FSF) har også på sine Internettssider lagt ut FAQ²²⁷ etc. om hvordan GPL skal forstås. Spørsmålet blir da hvordan en skal vektlegge slike «etterarbeider». FSF er, med unntak for deres egne programmer, ikke en part i avtalen. Utsagn som FSF kommer med kan derfor i utgangspunktet ikke tillegges mer vekt enn juridisk litteratur ellers. Kan det derimot bevises at avtalepartene begge har lagt FSF sine uttalelser til grunn, vil avtalepartenes felles forståelse av uttalelsene måtte legges til grunn.

6.3 Bransjepraksis

Der avtalen er uklar og det har utviklet seg en oppklarende bransjepraksis er det liten grunn til ikke å se hen til den. Det vil skape forutberegnelighet for partene. En må imidlertid være forsiktig med å trekke dette for langt, da avtalens innhold er fastsatt av hva partene har inngått.

225 Om vekt av forarbeider til standardavtaler, se Hov, 2002, s 175

226 <http://gplv3.fsf.org/>

227 Frequently Asked Questions

7 COPYLEFT

7.1 Introduksjon

Copyleft er et sentralt tema ved bruk av åpen kildekode-lisenser. Copyleft bruker opphavsretten (copyright) til å oppnå et resultat som på mange måter er det motsatte av tradisjonell bruk av opphavsrett, derav navnet.

Copyleft er en type vilkår som finnes i noen åpen kildekode-avtaler. Hvis lisenstaker tilgjengeliggjør et avledet verk, der det opprinnelige verket²²⁸ er under en lisens med en copyleft-klausul, må lisenstaker (som blir lisensgiver) lisensiere det avledede verket under copyleft-lisensen, eller under tilsvarende liberale vilkår, til den som mottar det avledede verk. Dette kalles gjerne svak copyleft. Sagt med andre ord, svak copyleft betyr altså at copyleft-lisensen «sprer» seg til avledede verk.

Copyleft kan minne om tilbakelicensieringsklausuler («grant-back clauses»), men er forskjellige bl.a. fordi mottaker av lisensen til det avledede verket ikke nødvendigvis er lisensgiver av verket som det bli avledet fra. Copyleft kan derfor betegnes som en slags «grant-forward clause».

Det finnes også copyleft-klausuler som gjør at verk som blir satt sammen med et verk under en copyleft-lisens også må underlegges copyleft-lisensen (eller tilsvarende vilkår). Dette kalles gjerne sterk copyleft.

Siden mulighet for endring av programmet er så sterkt knyttet til tilgangen til kildekode, som nevnt over, krever copyleft lisensene også at lisenstaker, som mottar programmet, skal ha tilgang til kildekoden.

Siden copyleft vilkårene «sprer» seg til avledede verk, og i noen tilfeller også til verk som er satt sammen med et verk under en copyleft lisens, blir det gjerne sagt at copyleft har en «viruseffekt». Dette kan virke negativt ladet. Jeg vil her bruke begrepet «copyleft» siden jeg oppfatter det som normativt nøytralt. Copyleft-klausuler omtales av og til også som gjensidighetsklausuler.

Grovt sett kan en altså dele åpen kildekode-lisenser i tre kategorier i forhold til copyleft:

- I lisenser med sterk copyleft vil avledede verk og visse verk programmet blir satt sammen med måtte lisensieres under copyleft-lisensen eller tilsvarende liberale vilkår. Dette er som vi skal se kanskje tilfellet for GPL-lisensen.

228 Altså det verket som er blitt bearbeidet eller endret

- I lisenser med svak copyleft, vil bare avledede verk måtte lisensieres under copyleft-lisensen eller på tilsvarende liberale vilkår, som f.eks. EUPL-lisensen.
- Den siste gruppen åpen kildekode-lisenser har ikke copyleft, som f.eks. BSD-lisensen.

Jeg vil nedenfor se nærmere på lisenser med sterk og svak copyleft.

Det er verdt å merke at hvis et program, eller, mer praktisk, en liten del av et program, ikke er beskyttet av opphavsretten vil en programmerer lovlig og *uten lisens* kunne utnytte programdelen som er uten opphavsrettsbeskyttelse. (Forutsatt at den heller ikke er vernet av andre grunner som f.eks. databasevernet²²⁹.)

7.2 Svak copyleft

Svak copyleft omfatter, som sagt, kun avledede verk. Verk som bli sammensatt med verket under copyleft-lisensen blir ikke omfattet. EUPL er illustrerende. Av artikkel 5 følger det: «If the Licensee distributes and/or communicates copies of the Original Works or Derivative Works based upon the Original Work, this Distribution and/or Communication will be done under the terms of this Licence. The Licensee (becoming Licensor) cannot offer or impose any additional terms or conditions on the Work or Derivative Work that alter or restrict the terms of the Licence.»

Og av artikkel 1 i lisensen blir avledede verk i lisensens forstand definert som: «Derivative Works: the works or software that could be created by the Licensee, based upon the Original Work or modifications thereof. This Licence does not define the extent of modification or dependence on the Original Work required in order to classify a work as a Derivative Work; this extent is determined by copyright law applicable in the country mentioned in Article 15.»

Hva som blir omfattet av copyleft blir altså overlatt helt til opphavsrettens vurdering av hva som er et avledet («derivative») verk. (For EUPL sin del må en imidlertid være oppmerksom på at belgisk opphavsrett kan bli avgjørende i visse tilfeller, jfr. artikkel 15.)

229 Se åvl. § 43

7.3 Sterk copyleft etter GPL

7.3.1 Hva blir omfattet av copyleft etter GPL?

Hvor langt copyleft etter GPLv2 strekker seg er et av de store spørsmålene innen åpen kildekodejus. Uheldigvis er det ikke noe klart svar. Dette skyldes dels, som vi vil se, at språket i lisensen ikke er klart. Dels at GPL er avhengig av opphavsretten og, som vist over, er opphavsretten ikke helt klar på alle områder.

Et annet problem med avtalen er at de samme termene blir repetert, men med litt forskjellig vinkling. En må her huske på at avtalen er laget for å kunne brukes av ikke-jurister uten juridisk bistand. Derfor gjengir avtalen ofte en beskrivelse av vurderingstemaer i opphavsretten som for en jurist ville vært overflødig²³⁰. Når denne gjengivelsen er upresis gjør det en presis tolkning vanskeligere.

Resultatet har blitt at det har kommet mange sprikende tolkninger av avtalen i juridisk litteratur.²³¹

Hovedbestemmelsen for copyleft i GPLv2 finner vi i 2 b): «You must cause *any work* that you distribute or publish, that in *whole or in part contains* or is *derived from* the Program or any part thereof, to be licensed *as a whole* at no charge to all third parties under the terms of this License.»

Hovedbestemmelsen blir så nærmere presisert i artikkel 2 2. til 4. ledd. Etter bestemmelsen blir verk i tilsynelatende to tilfeller omfattet av copyleft:

1. tilgjengeliggjøring av avledede verk av et program lisensiert under GPLv2. («any work that... *is derived from* the Program»)
2. tilgjengeliggjøring av verk som inneholder et program lisensiert under GPLv2. («any work that ... *in whole or in part contains* ... the Program»)

Virkningen av at et verk blir omfattet av copyleft er at det inngår i den helheten («*a whole*») som må lisensieres under GPLv2.

7.3.2 «any work that... *is derived from* the Program»

Den første kategorien av verk som faller inn under copyleft er de som er et avledet verk (endring eller bearbeiding) av verk som er underlagt GPLv2.

²³⁰ Se f.eks. definisjonen av «derivative work» i artikkel 0.

²³¹ Tolkninger av copyleft bestemmelsen er gitt av bl.a.: Jaeger (2006) s. 30-40, Kielland (2006) særlig kapittel 6, Nimmer (2006), St. Laurent (2004) særlig s. 39 og s. 154-158, Välimäki (2005) s. 130-138

I amerikansk rett svarer «derivative work» konseptuelt til det vi på norsk kategoriserer som avledede verk²³². Når avtalen sier «is derived from» er det naturlig å anse «derived» som samme begrep som «derivative» bare i verb-form. Avtalen viser altså til opphavsretten. Om hva som er et avledet verk viser jeg til det jeg har sagt over. Jeg minner om at et verk som inneholder vernede elementer fra et annet verk vil være en avledning av dette andre verket. Og at et statisk linket bibliotek med et hovedprogram antakelig vil være å anse som en endret variant (en avledning) av biblioteket, jfr. diskusjonen over.

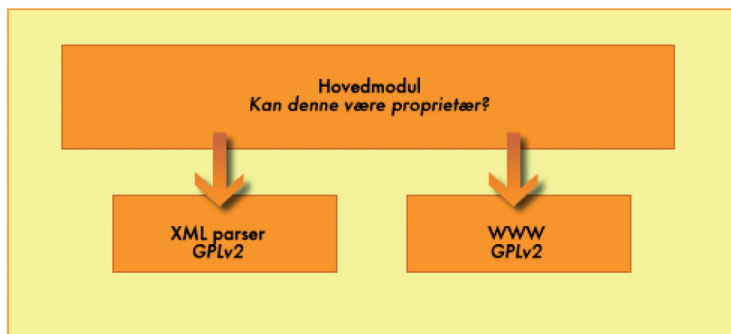
Det at en modul benytter seg av et grensesnitt mot en annen selvstendig modul, som mot et ferdiglaget dynamisk bibliotek, innebærer i utgangspunktet ingen endring av bibliotek-modulen og antakelig heller ikke at modulen som benytter biblioteket danner et avledet verk av biblioteket, jfr. tidligere diskusjon²³³.

Spørsmålet blir da om slik bruk av biblioteker allikevel kan føre til at en er innenfor copyleft bestemmelsens rekkevidde. Altså selv om et avledet verk ikke blir dannet. Dette vil jeg se på i neste kapittel.

7.3.3 «any work that ... in whole or in part contains ... the Program»

Den andre kategorien av verk som *kan* falle inn under GPLv2 sin copyleft er de som «inneholder» helt eller delvis et program som er lisensiert under GPLv2.

Ofte benytter en ferdige biblioteker når en lager programmer; f.eks. hvis en vil lage et kokebok-program, vil en f. eks. kunne ønske å benytte en modul for å lese filer i XML-format og en annen modul for å laste ned filer fra Internett for at brukeren skal kunne få tak i flere oppskrifter. Situasjonen kan illustreres på denne måten:



Illustrasjon 2: Eksempel på et program som benytter to dynamisk linkede biblioteker

²³² Se over og 17 USC § 101

²³³ Se over under overskriftene «Særlig om kombinasjon av moduler og bruk av grensesnitt» og «Dynamisk linking».

Her er hovedmodulen teknisk sett avhengig av bibliotekene, men når den bare benytter grensesnittet til bibliotekene er den antakelig vanligvis ikke et avledet verk, jfr. over. Hvis vi videre forutsetter at en ønsker at hovedprogrammet skal være proprietært, og at disse ferdige bibliotekene en benytter seg av er underlagt GPL, blir spørsmålet: Er en forpliktet etter copyleft-bestemmelsen til å underlegge hovedprogrammet GPL?

Free Software Foundation skriver: «using the ordinary GPL for a library makes it available only for free programs.»²³⁴ og videre «Some libraries are released under the GNU GPL alone; you must use a GPL-compatible license to use those libraries.»²³⁵ Det ser her ut som en har lagt til grunn en tolkning hvor hovedprogrammet blir omfattet av copyleft når en benytter seg av disse bibliotekene.

Av artikkel 2 4. ledd i GPLv2 følger det «... the intent is to exercise the right to control the distribution of *derivative or collective works based on the Program*.» Altså i tillegg til at copyleft skal omfatte avledede verk, som vist over, virker det som om hensikten også har vært, i alle fall i visse tilfeller, at sammensatte verk (collective works) skal bli omfattet. Et sammensatt verk utgjør nettopp en helhet («a whole»)²³⁶ som artikkel 2 bokstav b nevner.

Innledningsvis i artikkel 2 ser en ut til å ha begrenset copylefts virkeområde til verk som er «based on the Program». Av definisjonen i artikkel 0 følger det at «work based on the Program» betyr «either the Program or any *derivative work under copyright law*: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language.» Benytter en definisjonen i artikkel 0, i betydningen at kun direkte kopier og avledede verk omfattes, blir det vanskelig å se at det er rom for å omfatte sammensatte verk, slik som artikkel 2 4. ledd gir uttrykk for. Avtalen kan derfor virke inkonsistent.

Som et annet motargument mot at sammensatte verk skal omfattes kan anføres begynnelsen av artikkel 2 som sier «You may *modify* your copy or copies of the Program ... provided that you also meet all of these conditions» og videre at artikkel 2 2. ledd sier «These requirements apply to *the modified work* as a whole.» Hvis en tar disse bestemmelsene på ordet, virker det som om verk som ikke er en endring («modification») av et GPL-verk ikke skal omfattes av copyleft. (Et sammensatt verk utgjør jo bare en sammensetting, ikke en endring). Dette gjør igjen at avtalen kan virke inkonsistent.

234 «Why you shouldn't use the Lesser GPL for your next library»

235 «Frequently Asked Questions about version 2 of the GNU GPL»

236 Se 17 USC § 101 «A 'collective work' is a work... in which a number of contributions, constituting separate and independent works in themselves, are assembled into a *collective whole*.» (min utheving)

Et annet motargument mot at slike sammensatte verk skal omfattes, som har vært brukt, kan en finne i ordlyden i artikkel 2 b). Bestemmelsen sier at copyleft gjelder «work ... that *in whole or in part contains* or is derived from the Program». Et avledet verk er nettopp et verk som inneholder vernede elementer fra et annet verk. Det er derfor mulig å se på «*in whole or in part contains*» som en eksemplifisering av hva et «derived work» er²³⁷. Denne forståelsen er nok ikke helt naturlig siden ordet «or» («eller») er brukt²³⁸. Derimot blir det naturlig at også andre verk enn de avledede verkene skal omfattes, jfr. «or».

Et mulig argument for at sammensatte verk skal omfattes følger også av siste setning i lisensen, altså etter instruksjonene om hvordan en skal bruke lisensen. «This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License». Problemet er imidlertid at lisensen ikke sier noe om formen for linking.

Det er mulig at forfatterne av avtalen så på dynamisk linking som tilstrekkelig til at det ble dannet et avledet verk. Altså at de så på funksjonell eller teknisk avhengighet som avgjørende for hva som danner et avledet verk.²³⁹ Dette kan passe med trender i den amerikansk opphavsretten på den tiden GPLv2 ble laget.²⁴⁰

Hvor langt GPLv2 sin copyleft strekker seg må sies å være noe uklart. Intensjonen taler klart for at visse sammensatte verk bør omfattes, det samme gjør ordlyden noen steder og avgrensingen mot «aggregation», som jeg vil komme tilbake til. Samtidig taler presiseringen av artikkel 2 imot denne løsningen. Løsningen er derfor ikke helt klar. Siden det finnes gode grunner for at copyleft også skal omfatte visse sammensatte verk vil jeg se nærmere på dette.

Da blir spørsmålet: Hvor skal yttergrensene gå for dette sammensatte verket? Dette beror på en vurdering av hva som utgjør en helhet, jfr. «*as a whole*» i artikkel 2 b). For dataprogrammer har vi dessverre ikke en perm som for bøker som vanligvis gir en avgrensning for hva som inngår i verket. Derimot har en det noe upresise tekniske begrepet «program» som en kan se hen til, og som nok er den beste måten å avgrense denne «helheten» det snakkes om.

237 Se Välimäki, 2005, s. 131

238 Jeg sikter altså til: «...in whole or in part contains or is derived from the Program...». Siden en har valgt å bruke «or» kan det virke som «in whole or in part contains» er noe annet enn en eksemplifisering av hva et «derived work» er.

239 Se diskusjonen over om «Særlig om kombinasjon av moduler og bruk av grensesnitt» Se også Jaeger, 2006, s. 41

240 Se Stoltz, 2006

Noen begrensninger av «helheten» følger av avtalen. For det første kan «helheten» ikke strekke seg lengre enn til det en tilgjengeliggjør sammen, slik som 2 b) er utformet: «...work that *you distribute or publish* ... as a whole...» «Helheten» vil derfor bare omfatte verk som er tilgjengeliggjort med en viss tilknytning til et GPLv2 verk. Dette kommer jeg tilbake til.

Selv om flere verk blir gjort tilgjengelige sammen, er det ikke sikkert at de er en del av «helheten». Siste ledd av artikkel 2 sier «In addition, *mere aggregation* of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.» En må altså trekke et skille mellom tilgjengeliggjøring av programmer som en del av «helheten» og tilgjengeliggjøring av programmer sammen med andre programmer der tilknytningen er så svak at programmene ikke er en del av det *sammensatte* verket, men bare to verk som er *sammenstilt* («aggregated»).

7.3.3.1 Nærmere presisering av «helheten»

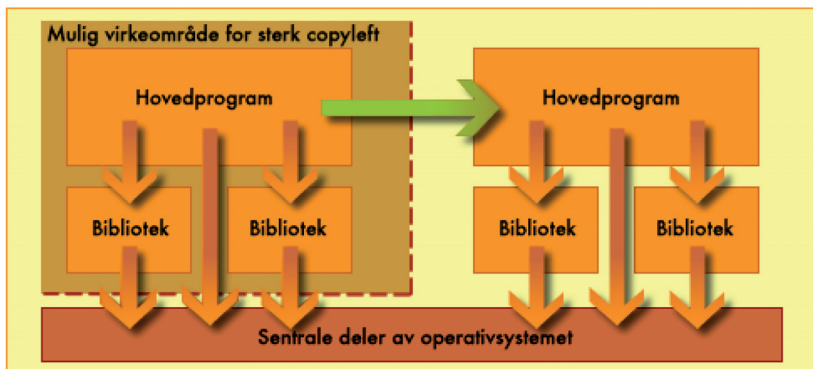
Et problem med å tolke GPL er at den ikke regulerer nøyaktig hvor grensene for «helheten», eller det sammensatte verket, skal gå. Som sagt, kan det være nyttig å se hen til det tekniske begrepet «program», altså noe tilsvarende begrepet «bok» for litterære verk. Allikevel kommer en ikke unna at begrepet «helheten» er svært vagt.

Siden begrepet «helheten» er så vagt bør en vise tilbakeholdenhet med å tolke det for omfattende. Der bransjepraksis har gitt praktiske avgrensninger er det liten grunn til å gå imot praksisen med mindre det er gode holdepunkt for noe annet.

Nedenfor vil jeg prøve å trekke opp grensen for hva som utgjør denne «helheten».²⁴¹ Jeg vil da se nokså teknisk på hva som naturlig utgjør et program. Begrepet «program» har imidlertid ingen eksakt definisjon. En må også være forsiktig med å sette et likhetstegn mellom juridiske vurderinger og tekniske begrep da de sjeldent er helt overlappende. Å sette et likhetstegn vil øke faren for omgørelser. En må basere avgjørelsen på en helhetsvurdering.

For vurderingen av hva som inngår i denne «helheten» blir det da naturlig å ta utgangspunkt i at programmet omfatter alle moduler som hovedmodulen er teknisk avhengig av, for så å foreta avgrensninger. Teknisk avhengighet må ikke likestilles med opphavsrettslig avhengighet (altså spørsmålet om noe er et selvstendig verk eller ikke), jfr. over.

241 Fremstillingen støtter seg på «Frequently Asked Questions about version 2 of the GNU GPL»



Illustrasjon 3: Mulig forståelse av virkeområdet for sterk copyleft

Utgangspunktet må bli at moduler som hovedmodulen er funksjonelt avhengig av (og de modulene disse igjen er avhengig av) blir omfattet. Men selv om en programmodul er funksjonelt avhengig av et annen programmodul vil en nok allikevel kunne si at den ikke er en del av helheten. De fleste programmer er f.eks. avhengige av operativsystemet uten at det er naturlig å se på operativsystemet som en del av programmet. Dette kan en betegne som en horisontal avgrensning. For programmer som selv er naturlige å se på som en del av operativsystemet kan kanskje situasjonen være en annen.

Dette blir støttet av GPLv2 artikkel 3 2.ledd siste setning: «However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the *major components* (*compiler, kernel, and so on*) of the operating system on which the executable runs». At bestemmelsen er kalt et unntak kan nok ikke tale for at det menes at operativsystemet skal omfattes av «helheten». Den er nok mer en sikkerhetsventil siden en er usikker på hvordan avtalen vil bli tolket. Bestemmelsen fortsetter med «unless that component itself *accompanies* the executable.» Hva en skal legge i ordet «accompanies» er ikke klart. En rimelig forståelse er nok at sammenstilling (aggregation) ikke er tilstrekkelig jfr. siste ledd i artikkel 3. (Siste ledd med «aggregation»-bestemmelsen kommer jo som et unntak til det ovenforstående 2. ledd).

Det kan også være aktuelt å foreta en avgrensning mot moduler som følger med pga. valget av programmeringsspråk. Typisk er dette moduler som gir grunnleggende funksjonalitet til programmet (som run-time biblioteker). Free Software Foundation skriver²⁴² «Is dynamically linking my program with the

242 Se «Frequently Asked Questions about version 2 of the GNU GPL»

Visual C++ (or Visual Basic) run-time library permitted under the GPL? The GPL permits this because that run-time library normally accompanies the compiler or interpreter you are using. So it falls under the exception in GPL section 3.» Henvisningen til artikkel 3 er ikke særlig presis. Det er flere unntak i artikkel 3, men ut fra ordlyden er det kanskje nest siste paragraf siste setning de refererer til. Det er allikevel vanskelig å se begrunnelsen siden unntaket der gjelder «major components» i «the operating system». Det er nok ikke alltid naturlig å se på run-time bibliotek som dette. Begrunnelsen virker derfor tynn. Løsningen har allikevel svært gode grunner for seg siden en annen løsning ville gjort det vanskelig å bruke lisensen i forbindelse med mange programmeringsspråk. Slike biblioteker gir grunnleggende «infrastruktur», noe også operativsystemet gjør. Praksis og rimelighet kan tale for at slike moduler ikke skal omfattes av helheten.

En må nok i vurderingen av hvorvidt en modul i forhold til en annen modul er en del av «helheten» eller ikke også avgrense i forhold til andre selvstendige programmer. Også når modulen er funksjonelt avhengig av disse. Dette kan betegnes som en vertikal avgrensning. For å trekke denne grensen kan en både se hen til kommunikasjonsformen og til innholdet i kommunikasjonen mellom modulene.²⁴³ Selvstendige programmer vil ofte bruke pipes²⁴⁴, sockets²⁴⁵ og kommandolinjeargumenter som kommunikasjonsformer mellom hverandre. Dette kan være en indikasjon på at de er separate. Avgjørende kan det imidlertid ikke være fordi det blir for lett å omgå. Derfor kan det også være aktuelt å se hen til innholdet i kommunikasjonen.²⁴⁶

At to moduler deler et adresseområde i hurtigminnet når programmet kjører, kan være en indikasjon på at de har tett forbindelse, og dermed på at de er en del av samme program.

Også i situasjoner når en modul ikke kommuniserer med en modul i hurtigminnet til datamaskinen, kan teknikker som RPC (Remote procedure call) gjøre at en befinner seg i en tilsvarende situasjon. Ved RPC vil modulen det interageres med gjerne befinne seg på en annen maskin i et nettverk, altså distribuert databehandling. Forenklet sagt gjør RPC at programmereren kan gjøre funksjonsskall til en annen ekstern modul som om modulen var lokal. Situasjonen vil også kunne oppstå i en del web-applikasjoner der «web-siden» (klienten) kommuniserer med programmet på web-serveren (f.eks. AJAX²⁴⁷). (Her vil GPL sitt

243 Se GPL FAQ «What is the difference between «mere aggregation» and «combining two modules into one program»?»

244 En «pipe» er kort fortalt en kanal for kommunikasjon mellom to prosesser innen en maskin.

245 En «socket» er kort fortalt en kanal for kommunikasjon mellom to prosesser vanligvis over et nettverk.

246 Se GPL FAQ «What is the difference between «mere aggregation» and «combining two modules into one program»?»

247 Asynchronous JavaScript and XML

krav om tilgjengeliggjøring være særlig aktuelt for å begrense helheten. Dette fordi en ofte kan ha programvare på serveren («serverside») som ikke blir tilgjengeliggjort, og som derfor ikke omfattes av copyleft. Hvis vernede elementer under GPL ikke blir overført til klienten vil disse heller ikke omfattes.)

Rettskildegrunnlaget for å trekke disse linjene er langt fra bunnsolid. De endelige grensene må trekkes opp av rettspraksis. Det som er sagt over blir derfor kun et forslag til en mulig løsning. Siden det i stor grad er opp til programmererne av modulene å utforme integreringen mellom modulene er det mange forskjellige løsninger, og en må foreta en konkret vurdering av den enkelte situasjonen.

7.3.4 Avgrensning av copyleft mot tilgjengeliggjorte programmer

Det er særlig to former for tilgjengeliggjøring som er interessante i forbindelse med åpen programvare. Den ene formen er distribusjon av programmer på fysiske media som f.eks. en DVD e.l. og den andre er distribusjon over Internett.

Opphavsrettslig kan en skille mellom tre typer tilgjengeliggjøring: Spredning, fremføring og visning.

Spredning omfatter besittelsesoverføring av fysiske eksemplarer av et åndsverk, jfr. åvl. § 2 3. ledd bokstav a. Et typisk eksempel er et program på en DVD e.l. Enhver form for besittelsesoverføring, som gratisutdeling og salg, av fysiske eksemplarer faller inn under begrepet spredning²⁴⁸.

Overføring av programmer over Internett eller andre digitale nett blir ikke sett på som spredning siden spredning kun gjelder fysiske eksemplarer²⁴⁹. Fremføring vil da stå igjen som en kategori å plassere digital overføring i²⁵⁰, jfr. åvl. § 2 4. ledd.

Hovedbestemmelsen for copyleft i GPLv2 artikkel 2 b) stiller som betingelse for at et verk skal bli omfattet av «helheten», altså virkeområdet for copyleft, at verket er blitt distribuert eller publisert som en del av denne helheten: «You must cause any work *that you distribute or publish*, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed *as a whole* at no charge to all third parties under the terms of this License».

248 Se Rognstad, 2004 s. 79

249 Se Rognstad, 2004, s. 80 og opphavsrettsdirektivet (2001/29/EC) art. 4 jfr. fortalen pkt. 28.

250 Tilsvarende konklusjon Rognstad, 2001, s 11

Spørsmålet blir, for det første, hva «distribute» innebærer? Av definisjonene i § 101²⁵¹ i 17 USC og av hovedbestemmelsen i § 106 (3)²⁵² ser vi at spredning (av fysiske eksemplarer) må falle inn under «distribution», men også at overføring over digitale nett også må falle inn under «distribution» siden de andre kategoriene passer dårlig²⁵³. Når det gjelder «publish» er det antakelig her synonymt med «distribute», men det er noe omstridt²⁵⁴. Ordlyden er noe annerledes når det gjelder bestemmelsen som krever at det gis tilgang til kildekode, der brukes betegnelsen «copy and distribute», jfr. artikkel 3. Antakelig er det ingen praktisk forskjell.

Opphavsmannens enerett til å tilgjengeliggjøre verket gjelder bare tilgjengeliggjøring som skjer utenfor det private området, altså til «almenheten», jfr. åvl. § 2.²⁵⁵ Det private området kan defineres som «den forholdsvis snevre krets som familie-, vennskaps- eller omgangsbånd skaper»²⁵⁶. For eksempel ble det i Bedriftsmusikkdommen²⁵⁷ bl.a. slått fast at musikk (et åndsverk) som ble spilt

251 «Distribute» er ikke direkte definert.

252 Fra § 101:

«Publication' is *the distribution* of copies or phonorecords of a work to the public by sale or other transfer of ownership, or by rental, lease, or lending. The *offering to distribute copies or phonorecords to a group of persons for purposes of further distribution*, public performance, or public display, *constitutes publication*. A public performance or display of a work does not of itself constitute publication.
[...]

Fra § 106:

«Subject to sections 107 through 122, the owner of copyright under this title has the exclusive rights to do and to authorize any of the following: [...]

(3) *to distribute copies* or phonorecords of the copyrighted work *to the public* by sale or other transfer of ownership, or by rental, lease, or lending;
[...].»

253 Se A&M Records, Inc. v. Napster, Inc., 239 F.3d 1004 (9th Cir. 2001) for et kjent eksempel. (Saken gikk videre i UMG Recordings Inc. et. al. v. Bertelsmann AG Inc. et. al.)

254 Dette spørsmålet dreier seg bla. om den såkalte «making available»-teorien i amerikansk opphavsrett. Den er, så vidt jeg kan se, omstridt. Se f.eks. Elektra v. Barker for en drøftelse av denne.

Se også Ford Motor Co. v. Summit Motor Prod., Inc. 930 F.2d 277 (3d Cir. 1991) «'Publication' and the exclusive right protected by section 106(3) . . . are for all practical purposes, synonymous.»

Se også f.eks. Atlantic v. Howell om ulisensiert fildeling: «Unless a copy of the work changes hands in one of the designated ways, a «distribution» under [sec.] 106(3) has not taken place. Merely making an unauthorized copy of a copyrighted work available to the public does not violate a copyright holder's exclusive right of distribution.»

255 Merk at også 17 USC § 106(3) sier «to the public»

256 Knoph, 1936, s. 89. Definisjonen fikk prejudikatstatus gjennom Rt. 1953 s. 633 Bedriftsmusikk.

257 Rt 1953 s 633

på høyttaleranlegget (fremført) i en bedrift der over 180 personer kunne høre musikken, var utenfor det private område. Det er ikke noe fast antall som utgjør det private området, det er en situasjonsbetinget avgjørelse. En vil kunne være utenfor det private området i lang mindre bedrifter.²⁵⁸

Spørsmålet blir da: Er det at et GPLv2 lisensiert program blir tilgjengelig gjort utenfor det private området tilstrekkelig til at andre utenfor dette private området kan kreve tilgang til programmet (og kildekoden) under GPLv2?

Det er verdt å merke at copyleft bestemmelsen kun får anvendelse når lisenstaker selv velger å tilgjengeliggjøre programmet, jfr. "*work that you distribute or publish*", jfr. artikkel 2. En har ingen plikt til å tilgjengeliggjøre programmet til andre. Av artikkel 2 og 3 i GPLv2 følger det "*You may ... copy and distribute*".

Det betyr at selv om en innenfor en bedrift bruker et program slik at en er utenfor det private området, trenger bedriften ikke å gi programmet (og kildekoden) til andre utenfor bedriften. Retten til å få tilgang til kildekode er begrenset til programvare som lisenstaker selv velger å tilgjengeliggjøre. Lisenstaker står følgelig fritt til å velge å ikke gjøre dette.

7.3.5 Programvare som tjeneste (SaaS) - «The ASP loophole»

Programvare som tjeneste («software as a service» (SaaS)) innebærer at programvareleverandøren lager et grensesnitt til programmet sitt som er tilgjengelig over Internett e.l. Leverandøren selv, eller en tredjepart som leverandøren engasjerer, drifter serveren hvor programmet ligger. Kunden får verken en lisens til selve programmet eller selve programmet²⁵⁹ tilgjengeliggjort for seg, men kunden betaler for å bruke programmet. Brukeren, med sitt eget program, sender altså data til leverandørens program for så å motta resultatet fra leverandørens program. Leverandører av slike tjenester kan også betegnes som Application Service Providers (ASP). («Software streaming» der deler av det eksekverbare programmet blir overført til brukeren faller altså utenfor SaaS slik jeg bruker begrepet her.)

Spørsmålet som reiser seg er: Hvis en leverer en SaaS-tjeneste, må leverandøren da gi fra seg kildekoden til sitt program hvis det er basert på GPLv2-programmer? Som sagt trenger ikke leverandøren å gi fra seg kildekode til

²⁵⁸ Se også Rt. 1970 s. 1172 Diskotekdommen

Se også vedtak 20. mars 1987 av Nemnda til behandling av tvister vedrørende videresending av kringkastingsprogram (med Lassen som formann) i NIR 1989 s. 369-373. Vedtaket gjaldt tidligere åvl. § 20 a (nå åvl. § 34) som bruker begrepet «tilgjengelig for almenheten» noe også åvl. § 2 gjør. For små borettslag på 15 boliger eller færre med et antatt beboerantall på 50-60 og hvor beboerne ikke har spesielt nære vennskapsbånd antok nemnda at det sannsynligvis ville utvikles et slikt tett fellesskap at det var naturlig å oppfatte dette som en privat krets.

²⁵⁹ Altså koden til programmet enten i kildekode eller objektform.

noen for annet enn verk under GPLv2 som han «*distribute or publish*», jfr. artikkel 2 bokstav b. Siden han ikke tilgjengeliggjør selve programmet for kunden, bare utdata («output»), trenger han ikke å gjøre kildekode tilgjengelig for kunden. (Unntak kan selvsagt tenkes hvis leverandørens «output» inneholder en «programsnutt» eller andre vernede elementer fra et GPL-program²⁶⁰).

Dette «smutthullet» mener noen strider mot filosofien bak fri programvare. Derfor laget Affero Inc. en lisens for å dekke nettopp dette. Free Software Foundation har, basert på denne lisensen, laget en ny lisens «GNU Affero General Public License». Lisensen bygger også på GPLv3. I tillegg til copyleft som følger av GPL v3 må leverandøren i, situasjonen beskrevet over, gi fra seg kildekode til programmer som omfattes av GNU Affero til de som interagerer med programmet. Det er artikkel 13 som er av særlig interesse for dette. Den starter med: «Notwithstanding any other provision of this License, if you modify the Program, your modified version must prominently *offer all users interacting with it remotely through a computer network* (if your version supports such interaction) *an opportunity to receive the Corresponding Source of your version* by providing access to the Corresponding Source from a network server at no charge, through some standard or customary means of facilitating copying of software.»

Det sentrale spørsmålet her blir: Hva er interaksjon («interacting»)? Ut fra en naturlig språklig forståelse blir den ovenfornevnte situasjonen, hvor brukers program kobler seg opp til leverandørens program over, f.eks. Internett, for så å benytte seg av tilgjengelige funksjoner, omfattet. Ordet interaksjon er såpass vidt at andre situasjoner også kan falle innenfor, som f.eks. at en bruker benytter et brukergrensesnitt (som en web-side) til et program under GNU Affero-lisensen.

7.3.6 Diff-er og patch-er

Når en bidragsyter til et åpen kildekodeprosjekt skal overlevere bidraget sitt til prosjektet skjer det ofte i form av en diff²⁶¹. (Hvis bidragsyteren har «commit»-tilgang kan bidragsyteren legge bidraget direkte inn i versjonskontrollsystemet). Diff-en viser endringene programmereren har gjort, altså forskjellen mellom den uendrede koden og den endrede koden. Diff-er kalles også patch-er.

Et interessant spørsmål er om copyleft også vil omfatte diff-er. Hvis copyleft ikke omfatter diff-er av et GPLv2-program, vil diff-en kunne være proprietære.

²⁶⁰ Se diskusjonen over om grensesnitt

²⁶¹ difference

Diff-er finnes i flere typer. Jeg vil nedenfor se nærmere på den mest populære typen, «unified diff». For illustrasjonens skyld følger et eksempel nedenfor. Eksempelen viser et tilfeldig program som skriver ut Fibonacci-tall²⁶². Det opprinnelige programmet skriver ut kun det nte Fibonacci-tallet, mens det endrede programmet skriver ut rekken frem til og med det nte tallet. Fibonacci-tall er ikke poenget her, men innholdet i diff-en kan ha opphavsrettslig betydning.

Den opprinnelige filen:

```
public class DiffExample
{
    public static void main( String[] args )
    {
        int n = Integer.parseInt( args[0] );

        int previous = -1;
        int current = 1;
        int sum;

        for ( int i = 0; i <= n; i++ )
        {
            sum = previous + current;
            previous = current;
            current = sum;
        }

        System.out.println( current );
    }
}
```

Diff-en (i «unified format»):

```
--- DiffExample.java 2008-04-13 18:55:13.000000000 +0200
+++ DiffExample2.java 2008-04-13 18:55:16.000000000 +0200
@@ -13,8 +13,10 @@
         sum = previous + current;
         previous = current;
         current = sum;
-    }
-
-    System.out.println( current );
+    System.out.print( current + " " );
+}
+
+    System.out.println();
+}
}
```

262 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ... Altså $F_n = F_{n-1} + F_{n-2}$ og $F_0=0$ og $F_1=1$

Den endrede filen:

```

public class DiffExample
{
    public static void main( String[] args )
    {
        int n = Integer.parseInt( args[0] );

        int previous = -1;
        int current = 1;
        int sum;

        for ( int i = 0; i <= n; i++ )
        {
            sum = previous + current;
            previous = current;
            current = sum;

            System.out.print( current + " " );
        }

        System.out.println();
    }
}

```

Inneholder diff-en vernede elementer, og dermed utgjør et avledet verk, må en ha opphavsmannens tillatelse for å tilgjengeliggjøre den. Ingen av «filene» over er nok beskyttet av opphavsretten siden utformingen er diktert av algoritmen (funksjonaliteten). Derfor må vi, for eksempelets, skyld legge til grunn at vi har et originalt program som har tilstrekkelig verkshøyde, og dermed er beskyttet av opphavsretten. Har opphavsmannen av det opprinnelige programmet bare gitt tillatelse etter GPL vil diff-en måtte underlegges GPL etter reglene om copyleft når diff-en utgjør et avledet verk, og den tilgjengeliggjøres for allmennheten, som omtalt over.

For å avgjøre om diff-en opphavsrettslig skal kategoriseres som et avledet verk av originalprogrammet, må en finne ut om den inneholder vernede elementer fra originalprogrammet.

Opphavsrettslig er det interessant å legge merke til at diff-en inneholder deler av det originale programmet både for at en skal kunne orientere seg (linjene uten + eller - under @@) og at den inneholder linjer som skal bli fjernet fra det opprinnelige programmet (linjene som begynner med minus). Noen få linjer kode vil sjeldent være opphavsrettslig beskyttet, jfr. over om partisjonering. Derimot kan det være at linjene som blir slettet er mange. Er linjene til sammen nok til at de inneholder vernede elementer, noe som fort kan skje når det er mange linjer, vil diff-en utgjøre et avledet verk av programmet.

Videre inneholder filen noen linjer som skal legges til det opprinnelige programmet (linjene som begynne med +). Hvis disse utgjør en «modul» som

bare benytter seg av grensesnittinformasjon (og ikke vernede elementer) vil de antakelig ikke utgjøre et avleitet verk, men se diskusjonen over om grensesnitt. Ofte vil linjene endre eksisterende «moduler». Linjene vil da typisk bare fungere og gi full mening når de sees sammen med det opprinnelige programmet. Videre vil de benytte seg av variabler etc. fra «modulen» de endrer. I disse tilfellene blir nok avhengigheten fort så sterk at de nye linjene kan bli sett på som et avleitet verk.

Selv om diff-en ikke utgjør et avleitet verk, kan det være at en må lisensiere diff-en under GPL hvis diff-en tilgjengeliggjøres sammen med et GPL-lisensierte programmet slik at diff-en utgjør en del av «helheten». Selv om lisenstaker selv må kjøre et program som gjennomfører endringene diff-en gir oppskriften på på sin egen maskin, vil det nok være naturlig å se på diff-filen som en del av «helheten» som utgjør programmet når det originale programmet og diff-filen tilgjengeliggjøres sammen.²⁶³

Patch-er eller diff-er kan også være laget for programmer på objektform;²⁶⁴ der deler av objekt-koden i det gamle programmet blir oppdatert med objekt-koden fra det nye. Vurderingen blir i prinsippet den samme som over.

Konklusjonen blir altså at det må foretas en konkret vurdering av den enkelte diff-en for å avgjøre om den er et avleitet verk, men at en absolutt ikke kan utelukke at diff-er kan utgjøre et avleitet verk.²⁶⁵

7.3.7 GPLv3

GPLv3, som er oppfølgeren til GPLv2, har fått en ganske annerledes utforming. En har ved den nye avtalen forsøkt å unngå en del uklarheter som følger av versjon 2 av avtalen. Dette har en langt på vei lyktes med, men, som vi skal se nedenfor, er det fremdeles uklarheter. GPLv3 er ny (når jeg skriver dette), og det er fremdeles usikkert hvor populær den vil bli.

GPLv3 har også fått en rekke nye bestemmelser, bl.a. i forhold til patenter, DRM²⁶⁶ og såkalt Tivoization. Med Tivoization²⁶⁷ mener en situasjonen når hardware blir solgt med GPL-programvare installert, og hvor produsenten bruker hardware-en til å hindre brukeren i å kjøre endrede utgaver av programvaren på hardware-en. Dette kan sees på som å stride mot filosofien bark fri programvare; der filosofien jo er at brukeren skal kunne tilpasse

263 Se tilsvarende Kielland, 2004, s 140-141

264 Se f.eks. Percival, 2003 for en nærmere beskrivelse av hvordan slike programmer kan fungere

265 Tilsynelatende annerledes hos Kielland, 2005, s 140-141 som støtter seg på Metzger og Jaeger. De ser ut til å ha en annen forståelse av faktum.

266 Digital rights management.

267 Ordet stammer fra TiVo som er et merkenavn for en digital videooptaker produsert av TiVo Inc.

programvaren til sine behov. Disse nye bestemmelsene lar jeg av plasshensyn ligge, men jeg skal nedenfor se litt nærmere på copyleft bestemmelsene. Hovedbestemmelsen om copyleft følger av artikkel 5 bokstav c: «You must license *the entire work, as a whole*, under this License to anyone *who comes into possession of a copy*. This License will therefore apply, along with any applicable section 7 additional terms, to the *whole of the work*, and all its parts, regardless of how they are packaged...».

Hva som er verket i denne setningen her følger av artikkel 5 første setning. Det er et «work based on the Program». Av definisjonen i artikkel 0 følger det hva som er et verk («a work») basert på programmet: «To ‘modify’ a work means to *copy from or adapt all or part of the work in a fashion requiring copyright permission*, other than the making of an exact copy. The resulting work is called a ‘modified version’ of the earlier work or *a work ‘based on’ the earlier work*.»

Avledede verk faller innenfor definisjonen «work based on the Program».²⁶⁸ Derimot er det vanskelige å se at sammensatte verk (som ikke utgjør et avleddet verk av programmet) faller innenfor. Hvis vi antar at dynamisk linkede bibliotek ikke danner avledede verk, jfr. diskusjonen over: Hvordan blir situasjonen da hvis en dynamisk linker til et GPLv3 bibliotek fra et hovedprogram som en ønsker at skal være proprietært?

Artikkel 5 c), som omhandler copyleft, sier «You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. ...» Men artikkel 5 gjelder jo, i henhold til innledningen, bare «work based on the Program», altså antakelig bare avledede verk.

Slik avgrensningen mot sammenstilte («aggregate») verk er utformet, i siste ledd i artikkel 5, kan det allikevel virke som om også visse sammensatte verk skal være omfattet: «A compilation of a covered work with *other separate and independent works*, which are *not by their nature extensions of the covered work*, and *which are not combined with it such as to form a larger program*, in or on a volume of a storage or distribution medium, is called an ‘aggregate’ if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.»

Tolker en bestemmelsen antitetisk virker det som om verk som ikke er avledede (altså «other separate and independent works»), men som er «combined with it such as to form a larger program», som tilfellet i kokebok eksempelet

268 Sml. f.eks. definisjonen av avledede verk i 17 USC § 101 «A ‘derivative work’ is a work based upon one or more preexisting works [...] or any other form in which a work may be recast, transformed, or adapted.»

mitt over skal, omfattes av copyleft etter GPLv3. I tillegg virker det som verk som «by their nature [are] extensions of the covered work» skal omfattes. Dette er imidlertid også avhengig av hva en legger i ordet «combined»; er dynamisk linking nok til at det er «combined»? Dette skal vi se nærmere på nedenfor.

De som ønsker å unngå GPLv3 på «hovedprogrammet» ønsker antakelig også å tilgjengeliggjøre programmet i objektform. I så fall må en se hen til artikkel 6.

Av artikkel 6 a følger det at: «You may convey *a covered work* in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License...»

Et «covered work» er i artikkel 0 definert som «either the unmodified Program or a work based on the Program.» Altså antakelig kun direkte kopier eller avledede verk.

Definisjonen av den korresponderende kildekoden finner vi i artikkel 1: «The ‘Corresponding Source’ for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and *the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require*, such as by intimate data communication or control flow between those subprograms and other parts of the work.»

En ser at bestemmelsen om «corresponding source» kan innebære en forpliktelse til å formidle flere verk enn kun de avledede, men den svarer likevel ikke helt på spørsmålet om dynamisk linking. Hvis et GPLv3-program gjøres avhengig av et dynamisk linket bibliotek, må kildekoden til dette biblioteket som hovedregel²⁶⁹ også tilgjengeliggjøres fordi biblioteket er et «*dynamically linked subprogram[] that the work is specifically designed to require*». Endrer vi derimot litt på situasjonen blir ikke situasjonen like klar. Når det er «hovedprogrammet» som er avhengig av et bibliotek under GPLv3 er det ikke klart at «hovedprogrammet» skal falle innenfor. («The work» i sitatet over er jo da biblioteket under GPLv3 og ikke «hovedprogrammet»). «Hovedprogrammet» kan allikevel bli omfattet av artikkel 5, jfr. over.

²⁶⁹ Merk unntak for «System Libraries», som også er definert i 1 og unntak for «general-purpose tools or generally available free programs» etc.

Oppsummert kan en si at GPLv3 nok har forsøkt å løse en del av uklarhetene i GPLv2, men at det fremdeles er uklare situasjoner. Disse kunne nok vært unngått ved å forenkle avtalen.

7.4 Copyleft's avhengighet av opphavsretten

Som vist over er vurderingene i copyleft bestemmelsene sterkt knyttet til opphavsretten. Begrepene avledet og sammensatte verk står sentralt. Det har den konsekvens at innholdet i lisensen vil utvikles i samsvar med opphavsretten. Dette kan både være en fordel og en ulempe. Selv om hovedpunktene i de fleste lands opphavsrett er nokså like, pga. bl.a. Bern-konvensjonen og WTOs TRIPS-avtale, er det fremdeles rom for store forskjeller på detaljnivå. Dette kan skape usikkerhet om hvor grensene går. Hva begrepet avledet verk innebærer vil kunne endre seg over tid²⁷⁰, og kan føre til at f.eks. GPL ikke får den rekkevidden som opprinnelig var tenkt.

7.5 Hybridlisensiering

Begrepet «hybridlisensiering» blir gjerne brukt om å kombinere åpen kilde-kodelisenser med proprietære lisenser. Jeg vil nedenfor se nærmere på en del spørsmål og begreper knyttet til denne problemstillingen.

7.5.1 Dobbelt- og multi-lisensiering

Som viste over innebærer copyleft at programmer som baserer seg på et program under en copyleft-lisens kan måtte bli lisensiert under copyleft-lisensen. Hvis det åpne programmet med sterk copyleft f.eks. er et bibliotek til et hovedprogram en ønsker at skal være proprietært kan det by på problemer. Derfor er det en del leverandører som tilbyr biblioteker etc. under flere lisenser som f.eks. GPL og en proprietær lisens. Dette kalles dobbelt-lisensiering. Benyttes en svak copyleft lisens, eller en lisens uten copyleft, vil incentivet til å kjøpe den proprietære lisensen fort falle bort siden virkningene av lisensene kan begrenses. Å selge biblioteker under en lisens som gjør at en kan benytte biblioteket i proprietær

²⁷⁰ Se f.eks. Stoltz (2006) s 1453 som påviser endringer i hva som legges i begrepet avledet verk i USA

programvare, vil kunne være en inntektskilde for produsenter av åpne programvaren. I Norge er²⁷¹ Trolltech kanskje mest kjent for denne lisensformen²⁷².

Det er selvfølgelig også mulig at lisensgiver tilbyr enda flere forskjellige lisensavtaler enn kun to. Dette kalles multi-lisensiering. Dette kan bl.a. være aktuelt for at lisenstaker skal få en lisens han føler seg trygg på.

7.5.2 «Proff»-versjoner og proprietær deler

En annen variant av hybridlisensiering er å lisensiere en standardutgave av programmet under en åpen kildekode-lisens, og å lisensiere tilleggsfunksjonalitet (en «proff»-utgave) under en alminnelig proprietær lisens. På den måten kan selskapet nyte godt av oppmerksomheten den åpne programmet kan få gjennom å bli mye brukt og få inntekter fra tilleggsfunksjonaliteten.

7.5.3 Forholdet til bidragsyttere ved dobbel lisensiering

Hvis et program en ønsker å lisensiere under en proprietær lisens inneholder bidrag (beskyttede elementer), som er gitt under lisenser som ikke tillater en proprietær lisensiering, vil det kunne oppstå vanhjemmel. Har en f.eks. fått et bidrag til et prosjekt lisensiert kun under GPLv2, og en ønsker å også tilby bidraget under en proprietær lisens, vil det oppstå vanhjemmel, siden tilgjengeliggjøringen vil stride mot vilkårene i GPLv2. Den som benytter eller ønsker å benytte seg av «dual lisencing» med en proprietær lisens kan derfor ikke akseptere bidrag kun under GPLv2.

I slike situasjoner er det aktuelt for prosjektet enten å ikke tillate bidrag fra utenforstående, eller kun å akseptere bidrag under en lisens som gir tilstrekkelig fleksibilitet.

F. eks. MySQL, som benytter dobbel lisensiering, har i MySQL Contributor License Agreement gått langt i å sikre selskapet muligheten til å benytte verket proprietært: «You hereby irrevocably assign, transfer, and convey to MySQL all right, title and interest in and to the Contribution. Such assignment includes all copyrights ...»²⁷³

Her er det ikke snakk om en lisens, men en fullstendig overføring²⁷⁴ av opphavsrett. Generelt er en fullstendig overføring av opphavsretten selvsagt ikke nødvendig for å sikre at prosjektet kan benytte bidraget, men det stiller

271 Trolltech er nå kjøpt av Nokia.

272 Se «Business Model - Trolltech»

273 Se «MySQL Contributor License Agreement»

274 En reservasjon må taes i forhold til ideelle rettigheter

.....

prosjektet i en posisjon til å håndheve opphavsretten rettslig²⁷⁵ og gir fleksibilitet hvis en ønsker å endre lisensene.

275 Se kapittelet over om «Lisenser og overføring av opphavsrett»

8 NÆRMERE OM ULIKE FORMER FOR LISENSIERING

Siden opphavsmannen har enerett til eksemplarframstilling, både i opprinnelig og endret skikkelse, og til tilgjengeliggjøring for allmennheten, jfr. åvl. § 2, må brukere, videreutvikler og distributører etc. ha lisens dersom de ønsker å gjøre disse handlingene.

For åpen kildekode-lisenser kan lisensieringen skje etter to forskjellige modeller. For det første kan opphavsmannen gi en lisens til sitt verk som også gir lisensmottaker rett til å tildele lisenser på samme vilkår, altså å sublisensiere.²⁷⁶ Se f.eks. MIT-lisensen.

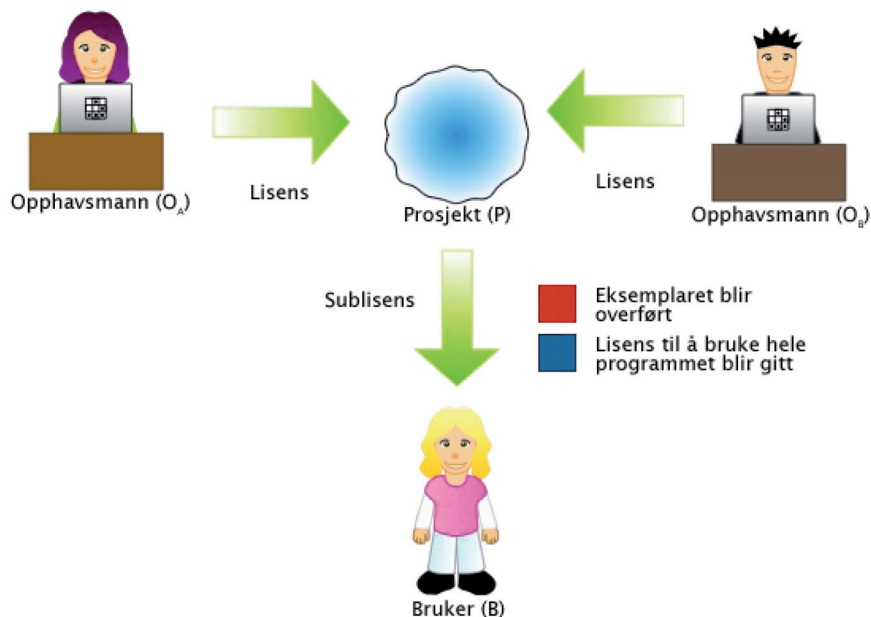
En annen form for lisensiering, som nok benyttes mest, er direkte nedstrøms-lisenser. Se f.eks. GPLv2, GPLv3 og BSD. Her gir den opprinnelige lisensgiver (bidragsyter) en lisens direkte til lisensmottakeren, selv om lisensmottakeren ikke mottok programmet fra den opprinnelige lisensgiver. Distributøren av programmet (eksemplaret) gir bare lisens til det han har bidratt med. Brukeren mottar og er avhengig av et knippe lisenser fra alle bidragsyterne for å kunne benytte programmet.

Nedenfor vil jeg se nærmere på disse ulike formene for lisensiering og på konsekvensene av dem. For illustrasjon kommer jeg til å bruke noen figurer. Disse gir en forenklet fremstilling av situasjonen i mange åpen kildekodeprosjekter. I åpen kildekodeprosjekter vil det, noen ganger, kunne være mange tusen bidragsytere. Dette vil gjøre modellene atskillig mer komplekse, men grunnprinsippene forblir de samme.

8.1 Sublisensiering

Figuren nedenfor illustrerer sublisensiering (eller underlisensiering om en vil). Dette er kanskje den enkleste modellen, så jeg starter med den. Modellen er nok mindre brukt enn direkte nedstrømslisens-modellen.

276 På norsk blir sublisensiering noen ganger kalt underlisensiering.



Illustrasjon 4: Sublisensiering

I en ren sublisensieringsmodell, som over, mottar brukene (B) retten til å bruke hele programmet fra det åpen kildekodeprosjektet (P). B mottar også vanligvis eksemplaret fra P.²⁷⁷ P vil kunne være et selskap, en forening etc.

I eksempelet over får P en lisens fra O_A og en lisens fra O_B. I kraft av at P har disse lisensene kan P lisensiere bidragene fra O_A og O_B i tillegg til evt. egne bidrag til B. Siden P lisensierer O_As og O_Bs verk i kraft av lisensen han har fått fra dem, kalles det sublisensiering.

Hvis P ikke skulle ha rettigheter til hele eller deler av programmet han lisensierer til B vil det oppstå en vanhjemmelssituasjon. B vil da, etter alminnelig obligasjonsrett, ha kontraktsrettslige krav mot P siden P ikke kan gi B de rettighetene B ble lovet. Dette kommer jeg tilbake til nedenfor.

Av avl. § 39b 2. ledd følger det at opphavsretten «heller ikke overdras videre uten samtykke». Dette innebærer altså at det er en presumpsjon for at en opphavsrett ikke kan videreoverdras (eller viderelicensieres) av lisensmottaker.²⁷⁸ For at P skal kunne sublisensiere må retten til det altså være klart avtalt.

²⁷⁷ En distributør, som ikke er lisensgiver, kan selvsagt også brukes.

²⁷⁸ Se Stray, 1989, s 178

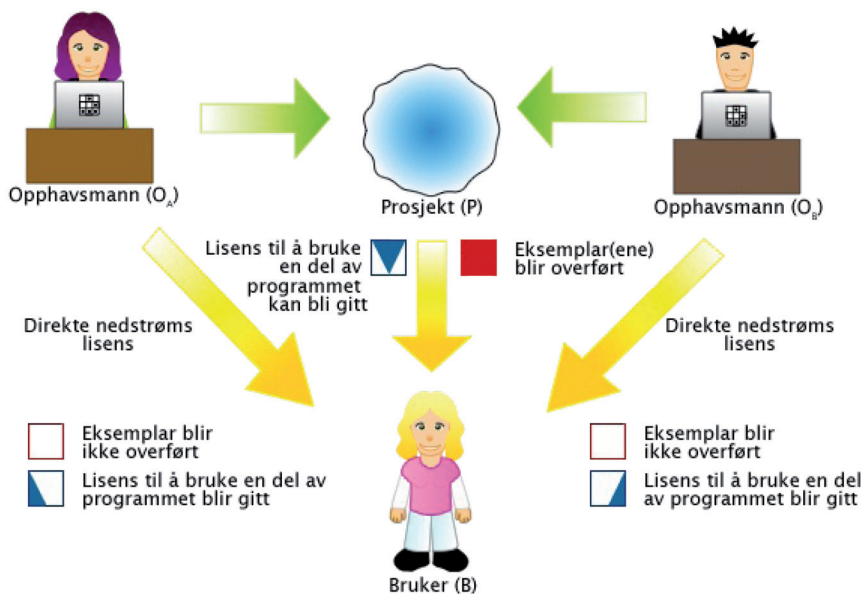
8.1.1 Sublisensiering i åpen kildekodeavtaler (MIT-lisensen)

MIT-lisensen gir et enkelt eksempel på sublisensiering. I lisensen står det: «Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the «Software»), to *deal in the Software without restriction*, including without limitation the rights to use, copy, modify, merge, publish, distribute, *sublicense*, and/or sell copies of the Software, and to *permit persons to whom the Software is furnished to do so,...*»

Lisensmottaker kan her sublisensiere og gi sine lisensmottakere samme rett til å sublisensiere.

8.2 Direkte nedstrømslisenser

Direkte nedstrømslisenser er den andre metoden som blir brukt for lisensiering i åpen kildekodeavtaler, GPLv2 og v3 er eksempler på dette. Den er antagelig den mest brukte formen for lisensiering. Figuren nedenfor illustrerer situasjonen.



Illustrasjon 5: Direkte nedstrømslisenser

Ved direkte nedstrømslisenser får brukeren (B), i likhet med sublisensiering, eksemplaret normalt overført fra prosjektet (P)²⁷⁹. Derimot gir P ikke B rett til å bruke hele programmet. P gir bare rett til de delene P har bidratt med. B får rettighetene til å bruke de andre delene av programmet fra bidragsyterne til de respektive delene, altså O_A og O_B i eksempelet over. Bs rett til å bruke programmet er altså avhengig av det knippe rettigheter han får fra bidragsyterne.

Ved sublisensiering inngikk, som sagt, B bare én avtale; avtalen med P. Ved direkte nedstrømslisenser inngår B tre avtaler; én med O_A , én med O_B og én med P. Disse avtalene gir til sammen B rett til å bruke programmet. Etter denne avtalestrukturen kan B, i utgangspunktet, ikke rette et kontraktsrettslig krav mot P for at det hefter vanhjemmel til O_A s og O_B s bidrag. Dette fordi P ikke har gitt B rett til å bruke disse delene.

Nedenfor vil jeg se nærmere på to eksempler på direkte nedstrømslisenser, GPL-lisensen og BSD-lisensen.

8.2.1 GPLs nedstrømslisensbestemmelse

Det følger av GPLv2 artikkel 6 at: «Each time you redistribute the Program (or any work based on the Program), *the recipient automatically receives a license from the original licensor* to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.»

Anvender vi avtalebestemmelsen på eksempelet over innebærer det at når P distribuerer programmet, mottar B en lisens direkte fra O_A , O_B og P som alle er originale lisensgivere (bidragsytere). Lisensene omfatter hvert enkelt bidragsyter sitt respektive bidrag. Tilsvarende nedstrømslisensbestemmelse finner vi også i GPLv3 artikkel 10.

Sublisenser er ikke tillatt etter GPLv2 og v3. Av GPLv3 følger det eksplisitt: «Sublicensing is not allowed; section 10 makes it unnecessary.», jfr. artikkel 2 3. ledd.

At sublisensiering ikke er tillatt etter GPLv2 er ikke like klart. Av artikkel 4 følger det «You may not copy, modify, *sublicense*, or distribute the Program *except as expressly provided under this License*.» En antitetisk tolkning taler for at avtalen er basert på å gi sublisenser. Antiteser må en alltid være forsiktig med. Siden det gir lite mening at lisenstaker skal få dobbelt opp med lisenser²⁸⁰, siden det ikke står at sublisenser kan bli gitt og siden det er en

²⁷⁹ En distributør kan også brukes.

²⁸⁰ Det gir altså lite mening at lisenstaker skal få en rett til å bruke en del av verket gjennom en sublisens og en rett til å bruke samme del av verket gjennom en direkte nedstrømslisens.

presumsjon mot at sublensiering er tillatt, jfr. åvl. § 39b 2. ledd, er sublensiering nok ikke mulig.

8.2.1.1 Avgrensningen av den direkte nedstrømlisensen

Et problem i forbindelse med de direkte nedstrømlisensene i GPL er: Hva er «the Program» som lisensmottaker (B) får lisens til fra bidragsyteren (O_A) i artikkel 6 i GPLv2? Problemet er ikke åpenbart i avtalen mellom O_A og P, men i senere ledd ser en problemet. Skal en ta artikkel 6 i GPLv2 på ordet, jfr. definisjonen i artikkel 0, vil «the Program» i avtalen mellom P og B gi inntrykk av at O_A skal lisensiere Ps endringer til B. Dette er imidlertid ikke en rimelig forståelse. O_A har i utgangspunktet ingen rett til Ps endringer. Avtalen gir ikke bidragsyter (O_A) automatisk en lisens på Ps endringer; GPL har ingen «grant-back»-klausul. Avtalen bør derfor forstås slik at bidragsyter (O_A) kun gir nedstrømlisenstaker (B) lisens til den delen av programmet som bidragsyteren (O_A) har bidratt med.

8.2.2 Kontraktingåelsen ved nedstrømlisenser

Her vil jeg se nærmere på hvordan lisenstaker (B) inngår avtalene med lisensgiverne (O_A , O_B og P).

Situasjonen kan forklares ved at bidragsyteren (lisensgiveren), ved å gjøre verket tilgjengelig under GPL lisensen, gir et tilbud til allmennheten (verden) som kommer i kontakt med verket og lisensen om at disse kan lisensiere verket.²⁸¹

Lisenstaker må så akseptere tilbudet. Dette vil kunne skje vet at lisenstaker gjøre en realhandling som forutsetter at han skal tiltre lisensen.²⁸² Altså at lisenstaker gjør en handling som krever opphavsmannens tillatelse slik den blir gitt i GPL-lisensen for at handlingen skal være lovlig. Lisenstakers handling kan nok også betegnes som konkludert adferd.²⁸³ En slik forståelse ser ut til å ligge til grunn i GPL, jfr. artikkel 5 siste setning²⁸⁴. Videre må nok lisenstakers

281 «Et depositivt utsagn kan være rettet til flere på samme tid – eller til en ubestemt krets av personer», jfr. Krüger, 1989, s 40.

282 Se Hov, 2002, s. 85. Realhandling blir gjerne betegnet som «uegentlig dispositive utsagn». Se også RG 1994 s 1203 der en person parkerte en personbil i strid med skilting. Lagmansretten uttalte: «Opplysningene om konsekvensene av ikke å følge grunneierens parkeringsnektelse sammenholdt med parkeringshandlingen må bli å anse som - eller likestilles med - en realavtale, med mindre parkeringsnektelsen fremtrer som urimelig eller sjikanøs.»

283 Se Hov, 2002, s. 86

284 «[...] by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.»

aksept være avhengig av at han kjente eller burde ha kjent til vilkårene.²⁸⁵ Slike avtaler, som blir inngått her, må nok kunne beskrives som realavtaler.²⁸⁶

I tysk rettspraksis er det to dommer av interesse. Den ene «netfilter/iptables»-saken, som gikk for Landgericht München I i 2004, omhandlet GPLv2. I dommen legges det til grunn en tilsynelatende tilsvarende betraktningmåte. I dommen legges det til grunn at en avtale «in rem» er sluttet, altså en tinglig avtale, jfr. del A av dommen. Dette må nok være en parallell til realavtalene våre. I tysk rett har en også «D-Link»-saken, som gikk for Landgericht Frankfurt am Main i 2006. Retten har lagt til grunn en tilsvarende forståelse som over.²⁸⁷

Avtaler komme i mange former. Partenes forutsetninger,²⁸⁸ bransjepraksis²⁸⁹ og reelle hensyn²⁹⁰ er relevante disposisjonskriterier. Når millioner benytter seg av disse avtalene, og disse ser ut til å forutsette en slik form for avtaleinngåelse, er det liten grunn til å ikke tillate dette.

Spørsmål i forhold til mindreårig lisensgiver, lisensgivers død og lisensgivers konkurs, vil jeg ikke gå nærmere inn på her. Det er nok mulig å se på P som en mellommann for Bs avtaler med O_A og O_B . Men siden P ikke er en del av avtaleforholdet B- O_A og B- O_B er nok det vanligvis av begrenset betydning.²⁹¹ Jeg vil heller ikke gå nærmere inn på denne problemstillingen.

8.2.3 Direkte nedstrømslisenser i BSD-lisensen

BSD-lisensen, som nevnt over, er et annet eksempel på en mye brukt lisens som også må innebære en direkte nedstrømslisens, men på en litt mer subtil måte. I lisensen heter det:

«Copyright (c) <year>, <copyright holder>

...

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

285 Se f.eks. RG 1994 s 1203, omtalt over, der retten legger vekt på at skiltingen var «klart i sitt innhold».

286 Sml. RG 1994 s. 1203 omtalt over.

287 Fra side 10 av dommen følger det: «Die GPL ist auf das Rechtsverhältnis zwischen den Urhebern und der Beklagten anwendbar. Die drei Softwareprogramme werden unstreitig ausschließlich unter den Bedingungen der GPL lizenziert. Im Fall der freien Software ist anzunehmen, dass der Rechteinhaber durch die Unterstellung des Programms unter die GPL ein Angebot an einen bestimmbar Personenkreis abgibt, das von den Nutzern durch Vornahme der zustimmungsbedürftigen Handlung angenommen wird; insoweit wird man von einem Verzicht auf den Zugang der Annahmeerklärung beim Anbietenden (§ 151 BGB) ausgehen können.»

288 Se Woxholth, 2005, s 99

289 Se Woxholth, 2005, s 99

290 Se Woxholth, 2005, s 103

291 Nærmere om forholdet fullmektig – tredjemann, se Woxholth, 2005, s. 279-281

* Redistributions of source code must *retain the above copyright notice, this list of conditions* and the following disclaimer.

* Redistributions in binary form must *reproduce the above copyright notice, this list of conditions* and the following disclaimer in the documentation and/or other materials provided with the distribution.»

Slik avtalen er utformet er det naturlig at den som står oppført som opphavsmann («copyright holder») er løftegiver (lisensgiver) i avtalen. Lisenstaker får nok ikke rett til å sublensiere siden dette ikke er nevnt, og, som sagt, er det en presumsjon mot sublensiering, jfr. åvl. § 39b 2.ledd. Lisenstaker i senere ledd (B) må derfor motta lisensen fra de opprinnelige lisensgiverne (O_A og O_B).

8.2.4 Konsekvensene av direkte nedstrømslisenser

Som sagt, får brukeren (B) rettighetene til programmet gjennom et knippe avtaler med de forskjellige bidragsyterne. Dette innebærer at den brukeren mottar programmet fra (P), ikke har gitt lisens til bidragene fra andre (O_A og O_B) gjennom avtalen han har med brukeren (B). B kan derfor i utgangspunktet ikke holde den han mottar programmet fra (P) ansvarlig for at det er vanhjemmel knyttet til andre bidragsyteres bidrag. P har etter avtalen ikke lovet rettigheter til disse bidragene. (I en konkret situasjon må en selvsagt se om det finnes holdepunkter for at Ps løfte er mer vidtrekkende.)

8.3 Lisensmodellens betydning for opphavsrettslige krav

Selvsagt vil en krenket opphavsmann kunne rette sine opphavsrettslige krav uavhengig av hvilken av disse to lisensmodellene som er valgt. Hvis O_A har plagiert T, vil T kunne rette krav grunnet bruddet på opphavsrett både mot O_A , P og B. Dette kommer jeg tilbake til nedenfor.

9 NÆRMERE OM SKILLET MELLOM OPPHAVSRETTLIG OG KONTRAKTSRETTLIG VIRKENDE VILKÅR I LISENSENE

Et spørsmål som reiser seg når lisenstaker eller andre bryter med vilkårene i lisensen er: Kan lisensgiveren eller opphavsmannen reise krav etter opphavsretten eller kontraktretten? Dette skille har betydning bl.a. for den internasjonale privatretten, som vist over, og for erstatningskrav, som jeg vil behandle nedenfor.

Krav basert på opphavsrett kan opphavsmannen gjøre gjeldende mot rettskrenkeren, mens rettigheter basert på kontrakt som hovedregel bare kan gjøres gjeldende mellom avtalepartene.²⁹² Rettigheter basert på opphavsrett er altså ikke avhengige av et inngått avtaleforhold, de følger av loven. Det er også forskjell på hvilke sanksjoner som kan rettes mot en som bryter en opphavsrettslig forpliktelse, og mot en som bryter en kontraktsrettslig forpliktelse. Det er særlig verdt å merke at naturaloppfyllelse, som f.eks. det å kreve at et verk skal underlegges GPL, ikke er en sanksjon opphavsretten hjemler. Kontraktsretten hjemler derimot naturaloppfyllelse på nærmere vilkår.²⁹³

Spørsmålet blir da: Hvilke bestemmelser i en lisens har betydning for opphavsrettslige krav og hvilke bestemmelser har kun kontraktsrettslig virkning? Avtalebestemmelser om de kvalifiseringer som åndsverkloven selv gjør er, som hovedregel, opphavsrettslig relevante.²⁹⁴ Vanligvis er hvor lenge lisensen varer (tid), hvor mange eksemplarer som kan fremstilles (antall) og geografisk utnyttelseområde opphavsrettslig relevant. Dette følger av et opphavsmannen har rett til å delvis overdra opphavsretten, jfr. åvl. § 39 (1). Grenser i forhold til «å bruke verket på en bestemt måte eller ved bestemte midler» kan også være relevante, jfr. åvl. § 39a²⁹⁵. Grensen mellom kontraktsrett og opphavsrett er imidlertid relativt lite utforsket i norsk rett og er derfor ikke helt klar²⁹⁶.

292 Se også Kinander, 2003 om skillet mellom tinglige og obligatoriske rettigheter og Hagstrøm, 2003, s. 27-28

293 Se Krüger, 1989, s. 13

294 Se Michaelsen, 2006, s 44. F.eks. A kan få rett til å eksemplarframstille uten at han får rett til å tilgjengeliggjøre verket for almenheten. Tilgjengeliggjør A verket for almenheten, på tross av det, bryter A i utgangspunktet med opphavsretten.

295 Se kommentarer til bestemmelsen i Ot.prp.nr.15 (1994-1995). Se også kommentarer til § 25 som § 39a viderefører i Ot.prp.nr. 26 (1959-1960)

296 Men se Michaelsen, 2006 og Kielland, 2005

Nedenfor vil jeg se på to eksempler som illustrerer skillet. Jeg vil først se på i hvor stor grad bestemmelsen om copyleft i GPLv2 er opphavsrettslig relevant. Så vil jeg se på virkningen av opphørklausulen i GPLv2.

9.1 Copyleft

Eksempelvis kan vi se på en situasjonen der A lisensierer et verk under en copyleft lisens, og hvor B lager en bearbeidelse av dette verket. Og videre at B stiller seg slik at copyleft etter lisensen skal bli utløst, altså at B tilgjengeliggjør bearbeidelsen for allmennheten (C). Og at situasjonen videre er slik at B nekter å underlegge sin bearbeidelse copyleft-lisensen og benytter avvikende vilkår. Er Bs opptreden en krenkelse av opphavsretten eller kontraktretten?

GPLv2 gir ikke tillatelse til en slik tilgjengeliggjøring som B gjør. For å tilgjengeliggjøre bearbeidelser er B avhengig av opphavsmannens tillatelse, jfr. åvl. § 2 1. ledd. A kan derfor rette sitt krav mot B etter opphavsretten²⁹⁷ og kreve opphavsrettslige sanksjoner mot B. Siden A, i eksempelet her, også i et kontraktsforhold til B, så vil A kunne rette kontraktsrettslige krav mot B, som krav om naturaloppfyllelse.

Når det gjelder copyleft i forhold til andre selvstendige verk, som altså ikke er avledede verk, strekker ikke argumentet over til. Åvl. gir ikke A relevante rettigheter i forhold til andre selvstendige verk. Slike brudd vil derfor bare ha kontraktsrettslig virkning²⁹⁸.

9.2 Opphørsklausuler

Av GPLv2 artikkel 4 følger det at brudd på lisensens vilkår (både kontraktsrettslige og opphavsrettslige) skal føre til at lisensen opphører: «You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. *Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License.* However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.»

²⁹⁷ Se tilsvarende, Kielland, 2005, s 86-87

²⁹⁸ Se tilsvarende, Kielland, 2005 s. 125

GPLv3 artikkel 8 viderefører denne bestemmelsen, men oppstiller et mer komplekst system med melding, tidsfrister og gjenopprettelse av lisenser, og kom nok som en reaksjon på noen av de problemene vi skal se på nedenfor.

Bestemmelsene i GPLv2 artikkel 4 kan forstås slik at den, etter sin ordlyd, innebærer at brudd på lisensen vil føre til at den opphavsrettslige tillatelsen tilbakekalles, enten plikten lisensen oppstiller er opphavsrettslig eller kontraktsrettslig virkende. Artikkel 4 er altså resolutiv. Etter ordlyden er et forsøk på brudd nok til rettighetstap, jfr. «atempt».

I «netfilter/iptables»-saken, som gikk for Landgericht München I i 2004, fikk Harald Welte en midlertidig forføyning mot Sitecom Germany. Sitecom hadde tilgjengeliggjort programvare i objektform som benyttet «netfilter/iptables» som er underlagt GPLv2. Verken lisensvilkårene eller kildekoden var gjort tilgjengelig slik GPLv2-krever. Welte, som var en av opphavsmennene, gikk derfor til sak. Artikkel 4 ble drøftet i saken. Retten anerkjente at artikkel 4 virker resolutivt og at brudd på lisensen vil føre til tilbakekallelse av lisensen.²⁹⁹

Tar vi bestemmelsen på ordet, innebærer i praksis ethvert brudd på lisensen at lisenstaker taper rettighetene sine med en gang. Dette kan, i en del situasjoner, åpenbart virke urimelig, og mye kan tale for å tolke bestemmelsen mer lempelig. I «netfilter/iptables»-dommen er det et interessant poeng: «Weiter ist zu berücksichtigen, dass das dingliche Angebot bei Verstößen nicht erlischt, sondern der Verletzer die Rechte durch Annahme und Befolgung der Bedingungen jederzeit wieder erwerben kann. Der automatische Verlust ist daher für Verletzer nicht besonders gravierend.»

Retten sier altså at et tinglig tilbud ikke opphører ved brudd på lisensen, og at rettskrenkeren ved å akseptere og å følge betingelsene kan gjennerverve rettighetene til en hver tid. Retten mener derfor at det automatiske tapet av rettigheter ikke er spesielt graverende for rettskrenker.

Bygger vi videre på parkeringsplassstilfellet (realavtalen)³⁰⁰, som omtalt over under avtaleinngåelsen, er tankegangen kanskje lettere å se. Selv om en én gang ikke har betalt for parkeringen, og dermed tapt retten til å stå parkert på parkeringsplassen, betyr ikke det at en aldri kan parkere på parkeringsplassen igjen.

299 Se tilsvarende Höppner, 2004, s. 633

300 Se RG 1997 s. 338

I GPLv3 har en, som sagt, kommet til en noe tilsvarende løsning i artikkel 8 2. ledd og 3. ledd³⁰¹ som den i dommen over. En slik forståelse av GPLv2, som vist over, strider ikke med lisensgivers interesser; forståelsen sikrer at avtalen blir fulgt, noe som må være i samsvar med lisensgivers ønske.

Når det gjelder opphavsrettslig relevante handlinger, byr bestemmelsen i GPLv2 artikkel 8 ikke nødvendigvis på problemer. Gjør lisenstaker en opphavsrettslig relevant handling, som han ikke har tillatelse til, er jo konsekvensen tilsvarende bestemmelsen, dvs. handlingen er, som GPLv2 sier, ugyldig («void») og lisenstaker er i en vanhjemmelssituasjon. (Derimot vil en lisenstaker vanligvis ikke tape alle rettigheter etter avtalen. Dette kan som sagt lett bli urimelig). Bestemmelsens forhold til konsumpsjonsretten, jfr. åvl. § 19, vil jeg avgrense mot her.³⁰²

Når det gjelder handlinger som kun er kontraktbrudd, kan bestemmelsen hevdes å være problematisk. Dette fordi den kan innebære at handlinger som ellers utgjør et kontraktbrudd vil føre til en opphavsrettslig krenkelse. Et eksempel kan være dersom en lisensgiver tar seg betalt for å gjøre kildekoden tilgjengelig for et avledet program han tilgjengeliggjør i strid med GPLv2 artikkel 3 bokstav b. Dette er antakelig bare et kontraktbrudd siden prisfastsetting kan sees på som en konsekvens av eneretten som åvl. § 2 gir og ikke som en del av den³⁰³.

På den ene siden kan en hevde at det er opp til den enkelte å sikre sin rettsstilling, og at avtalefriheten må tillate slike løsninger der lisenstaker risikerer at hans kontraktsbrudd får en opphavsrettslig konsekvens. På den annen side kan en hevde at en slik løsning passer dårlig med de sanksjonene, inkludert straff, som ligger bak opphavsretten, og at det er liten grunn til å gi lisensgiver slike virkemidler ved kontraktbrudd.³⁰⁴ Det er derfor mulig at bestemmelsen ikke bør gis virkning i disse tilfellene, og at bruddet heller bør behandles etter alminnelige regler om kontraktbrudd³⁰⁵.

301 Av GPLv3 artikkel 8 2. ledd og 3. ledd følger det:

«However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.»

302 Se Hoeren, 2004, s. 2-3 Han peker særlig på at slike bestemmelser kan undergrave konsumpsjonsretten. Jeg kommenterer ikke konsumpsjonsretten siden den bare gjelder spredning (overdragelse av fysiske eksemplar), fremføring (digital overføring) er mer aktuelt her. Se også Nimmer, 1998 om problemstillingen mer generelt. Se også Wiebe, 2004, s. 158

303 Se Rognstad, 199, s. 334-335

304 Se Michaelsen, 2006, s. 70

305 Se Hoeren, 2004, s. 2 og Michaelsen, 2006, s. 72

10 VANHJEMMEL OG SANKSJONER

10.1 Introduksjon

Det foreligger vanhjemmel³⁰⁶ i forhold til opphavsmannen når en utnytter et åndsverk (en opphavsrett) uten opphavsmannens tillatelse eller utover den opphavsrettslig relevante utnyttelsen opphavsmannen har tillatt. For lisensavtalene foreligger det vanhjemmel når lisensgiver inngår en avtale om å lisensiere et program som tredjemann har privatrettslige rettigheter til og tredjemanns rett går foran lisenstakers rett.^{307 308}

Den som disponerer på en slik ulovlig måte som beskrevet over kalles gjerne rettskrenkeren (eller litt mer upresist delinkventen). Vi ser altså at det kan oppstå krav som følge av vanhjemmel både i kontrakt og etter åndsverkslovens regler.

Overdragelse av ikke-eksisterende rettigheter, slik som f.eks. «opphavsrett» til et dataprogram som ikke har verkshøyde, ligner på vanhjemmel, men er prinsipielt forskjellig. Jeg vil ikke behandle disse situasjonene nærmere.

Vanhjemmel kan oppstå på flere måter enn bare i tilknytning til opphavsrettsbrudd. Avtalerettslig ugyldighet som habilitetsmangler (umyndige), tilbblivelsesmangler eller innholdsmangler³⁰⁹ er en annen mulighet. Nedenfor vil jeg kun fokusere på vanhjemmel knyttet til opphavsrettsbrudd.

10.1.1 Forskjellige typer vanhjemmel i kontrakt

Når lisensgiver disponerer rettslig over en eksisterende opphavsrett uten at den tilhører ham, og uten at han har noen kompetanse til å råde over den, foreligger det total vanhjemmel.

Hvis lisensgiver disponerer over en eksisterende opphavsrett i større grad enn det han har rett til, foreligger det en partiell (delvis) vanhjemmel. F.eks.

306 Rettsmangler brukes ofte også om vanhjemmel, men begrepet rettsmangel kan også innebære offentligrettslige mangler i tillegg til privatrettslige.

307 Det foreligger ikke vanhjemmel (i avtale) når vanhjemmelen ikke er i strid med avtalen. F.eks. en inngår avtale om et gode der rettighetsforholdet er usikkert og rettighetsmottaker har akseptert den risikoen. Dette er nok ikke praktisk for åpen kildekode-lisenser.

308 Se f. eks. Hagstrøm, 2004, s. 510 og Krüger, 1999, s. 222 for definisjoner på vanhjemmel

309 Se nærmere Vyrje, 1985, s. 25

hvis A har lisens til bruk i ett år og rett til å overføre lisensen, og A overfører lisensen til B for bruk i 2 år, vil det foreligge en partiell vanhjemmel. Ved partiell vanhjemmel vil lisenstaker få rett etter avtalen i så stor grad som lisensgiver kan gi den, men altså ikke alt avtalen gir ham krav på. I eksemplet over vil Bs bruk utover det første året være et opphavsrettsbrudd.

En skiller også mellom opprinnelig og etterfølgende vanhjemmel, men dette har begrenset praktisk betydning for åpen kildekodeavtaler siden vanhjemmelen her i all hovedsak er opprinnelig. Opprinnelig vanhjemmel foreligger allerede ved tidspunktet for avtaleinngåelsen, altså in contrahendo. Etterfølgende vanhjemmel foreligger hvis vanhjemmelen først oppstår etter avtaleinngåelsen, men før risikoovergangen til lisenstaker (typisk ved oppfyllelse)³¹⁰, altså in contractu. Siden oppfyllestidspunktet for åpen kildekode lisenser vanligvis faller sammen med tidspunktet for avtaleinngåelsen er det opprinnelig vanhjemmel som er mest aktuelt. Lisensavtalen blir ofte inngått når en laster ned programmet, men i noen tilfeller først etter nedlastingen.³¹¹

10.1.2 Vindikasjon og ekstinksjon ved overføring av opphavsrett

Det grunnleggende utgangspunktet ved overføring av rettigheter er at ingen kan overføre en bedre rett enn han selv har (*nemo dat quod non habet*). Det betyr at suksessors (lisenstakers) rett ikke kan gå lengre enn den rett suksessor avleder fra sin hjemmelsmann (lisensgiver). Dette gjelder for alle ledd bakover til opphavsmannen. Fra denne regelen er det et unntak, ekstinksjon. Som vi skal se, får ekstinksjon i all vesentlighet ikke anvendelse på opphavsrett.

Her er det også verd å peke på at opphavsretten, som sagt, ikke verner mot dobbeltfrembringelse. Den gir derfor ikke et prioritetsvern, kun et etterligningsvern.³¹²

Ved ekstinksjon får suksessor (B) en bedre rett enn den han utleder fra sin hjemmelsmann (A). Ekstinksjon betyr at den rette rettighetshavers (Ts) rett blir utslettet til fordel for suksessoren (B). Dette vil i mange tilfeller kunne være rimelig når T får et erstatningskrav mot A, A fremstår for B som rette eier, B er i god tro og har foretatt sikringsakt.

Ved ekstinksjon vil, som i eksemplet over, Ts rettigheter måtte stå tilbake for Bs utnyttelse. Ved ekstinksjon vil opphavsmannen altså tape deler av rettighetene. På opphavsrettens område er det ikke som hovedregel ekstinksjon,

310 Se Krüger, 1999, s. 222

311 Som omtalt ovenfor har GPL særskilt regulering for hvilke deler av avtalen som er aktive.

312 Se Gaarder, 1964, s. 248

siden en mangler hjemmel for det.³¹³ En sentral grunn til dette er at det er vanskelig utad å se hvem som er den rette eier av en opphavsrett. Hvis en kjøper et løsøre kan kjøperen fjerne selgerens legitimasjon ved å ta fra ham den fysiske gjenstanden. Dette kan han selvfølgelig ikke gjøre med noe immaterielt. Dette gjør ekstinksjon vil bli vanskelig å gjennomføre for immaterielle rettigheter. Løsninger som registre kan tenkes, men er ikke særlig praktisk for åndsverk.³¹⁴

Dette betyr altså at lisenstaker (B) i praksis kun kan vinne rett fra opphavsmannen eller en som utleder en tilstrekkelig rett fra opphavsmannen. En annen sak er at en kan ved ekstinksjon erverve rett til det fysiske mediet et program ligger på gjennom ekstil. § 1, men det faller utenfor denne drøftelsen.

Hvis rette eier (T) har opptrådt uaktsomt, slik at han har bidratt til at godtroende B har fått den mangelfulle rettigheten fra A, kan det virke lite rimelig at B ikke skal kunne vinne retten. Vyrje, basert på rimelighetsbetraktninger og tingsrettslig teori, har derfor tatt til ordet for at ekstinksjon skal være mulig i slike situasjoner også for åndsverk,³¹⁵ men at det må vises forsiktighet³¹⁶. Løsningen virker rimelig i alle fall når ekstinksjonen ikke kolliderer med eksklusive rettigheter som opphavsmannen (T) har gitt til andre.

Et annet spørsmål er om opphavsmannens passivitet kan hindre hans vindikasjon. Opphavsmannen bør jo si ifra når noen krenker retten hans slik at krenkeren kan bli klar over krenkelsen og holde opp så ikke krenkelsen blir verre. Godtroervervloven (som ikke gjelder for immaterielle rettigheter³¹⁷) § 3 1. ledd sier: «Krav... mot godtroende mottaker må settes fram uten ugrunnet opphold etter at eieren har fått slik viten at han kunne gjøre kravet gjeldende eller burde ha skaffet seg slik viten.» Basert på de reelle hensynene over og at det kan hevdes at § 3.1 oppstiller et generelt prinsipp, har det i juridisk

313 Se Knoph, 1936, s 139.

Se Koktvedgaard, 2005, s. 420

Se Gulating lagmannsretts dom av 14. november 1988 Studio 46 Reklamebyrå v/Egil Korsnes mot A/S Papir (særlig side 8).

314 Minner her om at registrering som *vilkår* for opphavsrett er forbudt etter Bern-konvensjonen artikkel 5 (2)

315 Se Vyrje s. 32 og 100

316 Se Vyrje, 1985, s. 100-101, «Avslutningsvis bør det imidlertid presiseres at synspunktet om at rette eiers uaktsomhet avskjærer ham fra å vindisere må anvendes med stor grad av forsiktighet innen immaterialkontraktsretten, da man ellers lett kan komme til å «uthule» vindiksjonsretten.»

317 Jfr. bl.a. tittel «Lov om godtroerverv av løsøre.»

teori, hos Vyrje,³¹⁸ blitt antatt at passivitet kan føre til at opphavsmannen taper sin vindikasjonsrett.

Uavhengig av dette vil foreldelsesloven § 9 få anvendelse for erstatningskrav utenfor kontrakt og foreldelsesloven § 3 jfr. § 2 få anvendelse ved kontraktbrudd.

10.2 Årsaker til vanhjemmel

Det kan være mange grunner til at vanhjemmel kan oppstå ved lisensiering av åpen kildekodeprogrammer. Mange av grunnene vil sammenfalle med årsaker som er vanlig for proprietær programvare, men arbeidsformen som mange åpen kildekodeprosjekter benytter skaper noen forskjeller. Kategoriene nedenfor er ikke uttømmende og kan også overlappe. Hensikten er kun å vise noen av de typiske tilfellene hvor vanhjemmelsproblemer vil oppstå.

10.2.1 Privat reguleringsvillfarelse

Hvis vanhjemmelen skyldes villfarelse med hensyn til hvordan en avtale skal forstås, kan den kategoriseres som privat reguleringsvillfarelse. I Norge er dette er nok særlig aktuelt for en del åpen kildekodeavtaler på grunn av det engelske språket og at avtalene ofte er basert på amerikansk (US) rett³¹⁹, som nok ikke er lett tilgjengelig for alle.

Hvis en lisenstaker av et program under GPLv2-lisensen misforstår den, og han lisensierer programmet under en proprietær lisens, er det snakk om privat reguleringsvillfarelse. Privat reguleringsvillfarelse kan både være forsettlig og uaktsom.

For orden skyld minner jeg om at hvis lisensgiver og lisenstaker er enige om forståelsen av avtalen er det deres forståelse som er avtalt, uavhengig av om den forståelsen er eller ikke er i samsvar med forståelsen til de(n) som har skrevet eller «forvalter» lisensen.

318 Se Vyrje, 1985, s. 97-99 «Skal jeg derfor våge meg på en konklusjon, må den bli at den rette eiers passivitet medfører at hans vindikasjonsadgang faller bort – og dette gjelder uansett om han «... kommer under vær.» med vanhjemmelsmisligholdet før eller etter at opphavsrettservervren har begynt sin «utnyttelse».

319 «Amerikansk rett» er noe upresist siden lovgivning også kan variere mellom delstatene i USA.

10.2.2 Rettsvillfarelse og bevisste lovbrudd

Rettsvillfarelse (*error juris*) er villfarelse med hensyn til eksistensen av eller innholdet i en rettsregel. Rettsvillfarelse er som hovedregel ikke unnskyldeelig (*error juris semper nocet*). Har f.eks. oppdragsgiver, uten nærmere avtale, benyttet en ekstern konsulent til å programmere for seg, og oppdragsgiver tror at han har fått hele opphavsretten overdratt til seg siden han jo har betalt for programmeringen, vil han være i rettsvillfarelse. Som sagt, har oppdragsgiver i slike situasjoner kun fått en lisens til det som er nødvendig og rimelig ut fra oppdraget, ikke mer. Gir oppdragsgiver lisenser utover over dette foreligger det en partiell vanhjemmel. Denne problemstillingen kan fort bli aktuell der en konsulent har arbeidet på et program som oppdragsgiver senere ønsker å gjøre om til åpen kildekode.

Det forekommer selvsagt også at rettskrenker bryter bevisst med opphavsretten.

10.2.3 Direkte kopi og plagiat

I «direkte kopi»-situasjonen lisensierer bidragsyteren til prosjektet uendret kode han ikke har rettigheter til. Lisensgiver bidrar f.eks. med en del av et program noen andre har rettigheter til (f.eks. arbeidsgiver) eller han «klipper-og-limer» inn et beskyttet verk i sin egen kode.

I plagiat-situasjonen har programmereren plagiert (ettergjort) en annens program uten tillatelse. Det er, som nevnt over, en grense mot nye selvstendige verk. Programmereren kan lovlig lage et selvstendig verk basert på ideene og prinsippene i et tidligere verk. Det er bare de individuelle trekkene som er beskyttet av opphavsretten, og som derfor ikke kan benyttes uten tillatelse.

En skiller mellom bevisste og ubevisste plagiat. Begge deler er ulovlig. Ved ubevisste plagiat mener plagiator («opphavsmannen» til plagiatet) han har skapt et originalt program, selv om det er basert på de individuelle trekkene til et tidligere verk. Ved bevisste plagiat er plagiator klar over at han bygger på de individuelle trekkene til et tidligere verk.

10.2.3.1 Relisensiering

Når en ønsker å skifte fra en proprietær lisens til en åpen kildekode-lisens, eller fra en åpen kildekode-lisens til en annen, kan det oppstå problemer for prosjektet når prosjektet ikke får innhentet tillatelse fra en av lisensgiverne. Prosjektet kan, som nevnt, oppleve en innlåsing («lock-in») i forhold til den gamle lisensen. Dette kan, som sagt, skyldes at lisensgiver nekter å gi tillatelse til relisensieringen, men også en ikke får kontaktet lisensgiver (altså

«foreldreløse verk»³²⁰ For prosjekter med mange bidragsytere er dette klart aktuelt. I disse tilfellene kan det bli aktuelt å skrive de delene av programmet som en ikke får lisens til på nytt.

I denne situasjonen er nok risikoen for ubevisst plagiering særlig stor siden det er lett å bygge på individuelle trekk til den delen som skrives på ny.

10.2.4 Dobbeltsuksesjon

Dobbeltsuksesjonskonflikter oppstår når to suksessorer utleder en rettighet fra samme hjemmelsmann og rettighetene til de to ikke kan forenes. F.eks. dersom jeg overfører samme eksklusive rett til to andre vil det gå galt siden begge ikke kan få den. I dobbeltsuksesjonssituasjonene vil den disposisjonen (overføringen eller lisensieringen) som er først i tid vinne rett.³²¹

Hvis en utvikler har laget et program eller en del av det, og så overdrar opphavsretten til en annen, vil det oppstå en total vanhjemmel hvis utvikleren så lisensierer programmet under en åpen kildekode-lisens. Utvikleren disponerer jo da over noe han ikke lengre har. En tilsvarende situasjon vil oppstå hvis utviklerne selger en eksklusiv lisens til et program, for så å lansere programmet under en åpen kildekode-lisens eller en annen lisens.

10.2.4.1 Arbeidstakers bidrag til prosjekter

Dobbeltsuksesjon kan være resultatet når arbeidstaker i arbeidsforholdet har bidratt til et åpent kildekodeprosjekt. Hvis arbeidet han gjør er innenfor oppgavene som omfattes av arbeidsforholdet vil jo opphavsretten automatisk bli overført til arbeidsgiver, jfr. åvl. § 39g som omtalt over.

Her kan det oppstå problemer hvis programmereren sender inn arbeidet han har gjort som et bidrag til et prosjekt uten at arbeidsgiver har godkjent dette. Programmereren råder jo da over noe han ikke har rettigheter til. I slike situasjoner er arbeidstaker avhengig av godkjennelse fra arbeidsgiver, siden arbeidsgiver har fått overført opphavsretten.

Godkjennelsen fra arbeidsgiver kan være eksplisitt, dvs. at de avtaler at programmereren kan bidra, eller den kan være implisitt, f.eks. at arbeidsgiver forholder seg passiv etter at han vet om at programmereren har planer om å bidra.

³²⁰ «orphan works»

³²¹ jfr. *nemo dat quod non habet* regelen omtale over.

10.2.5 Selvplagiat

Selvplagiat³²² minner litt om dobbeltsuksesjon. Denne situasjonen oppstår hvis programmereren har overført hele opphavsretten til et tidligere verk til en annen eller gitt en annen en eksklusiv lisens³²³, og den samme programmereren så lager en gjentakelse eller en bearbeidelse av det tidligere verket for hans egen eller andres utnyttelse. Ved selvplagiat vil den som er først i tid vinne rett ovenfor den som er senere i tid.

Selv om en programmerer har laget et program og overdratt opphavsretten til dette eller overdratt en eksklusiv lisens, betyr ikke det at han er avskåret fra å lage et tilsvarende program igjen. Han kan bare ikke benytte seg av de beskyttede elementene i verket han har overdradd eller gitt en eksklusiv lisens til. Derimot kan han (som alle andre³²⁴) bygge på de frie elementene i verket og skape et nytt og selvstendig verk som ikke vil være et plagiat.

10.2.5.1 Selvplagiat ved fritidsarbeid

Hvis en programmerer i et arbeidsforhold arbeider med en problemstilling, samtidig som han også er bidragsyter i et relatert åpen kildekodeprosjekt på fritiden kan det oppstå problemer: Arbeidet programmereren gjør i arbeidsforholdet vil som sagt arbeidsgiver få opphavsrett til, jfr. åvl. § 39g, forutsatt at ikke annet er avtalt.

Hvis det programmereren på fritiden bidrar med til prosjektet ligger så nært opptil det han har laget i arbeidstiden at det er å anse som et plagiat, vil det foreligge så godt som total vanhjemmel³²⁵ i forhold til det programmereren lisensierer til prosjektet. I situasjonen her vil åpen kildekodeprosjektet tape rettigheten siden arbeidsgiver er først i tid.

Situasjonen blir ikke helt den samme hvis programmereren først skaper verket på fritiden for så å lage det samme på jobben. Mens arbeidstaker får (hele) opphavsretten overdratt, får prosjektet vanligvis bare en ikke-eksklusiv lisens. Det betyr at det i forhold til det arbeidstaker skal yte til arbeidsgiver bare foreligger partiell vanhjemmel. Arbeidsgiver som forventet å få hele opphavsretten overført må stå tilbake for den lisensen prosjektet har fått overført fra arbeidstaker.

322 Se f.eks. Bing, 1975, s 48

323 som gir berettigede forventninger om eksklusiv utnyttelse

324 Praktisk og bevismessing er det selvsagt en stor forskjell: Programmereren som har laget det opprinnelige programmet har jo selvsagt bedre kjennskap til de vernede elementene i det opprinnelige programmet enn andre.

325 Der selvplagiatet innebærer en bearbeidelse har jo programmereren opphavsrett til selve bearbeidelsen og denne kan han råde over. Problemet er bare at bearbeidelsen er avhengig av det ubearbeidede verket som arbeidsgiver har opphavsrett til, noe som gjør at arbeidsgivers samtykke er påkrevd.

I praksis vil det nok ofte være vanskelig å skille ut hva som er gjort når, selv om koden kan være tidsstemplet i versjonskontrollsystemer, og dette kan gi en indikasjon.

10.3 Sanksjoner ved vanhjemmel

Sanksjoner som følger av opphavsrettsbrudd kan være straff (åvl. § 54), erstatning m.m. (åvl. § 55) og inndraging etter (åvl. § 56). I tillegg kan det reises sivilt søksmål mot rettskrenkeren for å få kjent en bestemt utnyttelse ulovlig og midlertidig forføyning kan være aktuelt.

Når det gjelder sanksjoner som følge av vanhjemmel ved kontrakter vil erstatning være særlig aktuelt, selv om andre krav også er mulig.

Jeg vil nedenfor se på noen av de meste praktiske erstatningskonfliktene som kan oppstå ved vanhjemmel til åpen kildekodeprogrammer. Som vi skal se, er det nærliggende krav som ikke godt kan begrunnes av erstatningsretten, disse vil jeg også behandle. Som sagt er også andre sanksjoner aktuelle ved vanhjemmel, men jeg vil ikke gå inn på disse.

10.4 Erstatning og restitusjon av berikelse

Erstatning er en økonomisk kompensasjon for tap eller skade. Det vil her si en kompensasjon for tapet som er et resultat av rettsbruddet (vanhjemmelen). For å få erstatning må det foreligge et ansvarsgrunnlag og en tilstrekkelig (adekvat) årsakssammenheng (kausalitet) mellom den skadevoldende handlingen og tapet. Det er i utgangpunktet kun faktisk økonomiske tap som dekkes.

Nedenfor vil jeg se på de forskjellige ansvarsgrunnlagene som er aktuelle i to situasjoner, når forholdet er regulert av et kontraktsforhold (en åpen kildekodeavtale) og når kravet er basert på åndsverklovens regler.

Funksjonen til erstatning er hovedsakelig gjenoppretting og prevensjon³²⁶. Gjenoppretting innebærer at en skal stille skadelidte økonomisk som om skaden (rettsbruddet) ikke var skjedd. Immaterielle rettigheter kan selvfølgelig ikke fysisk skades, men f.eks. markedsposisjonen kan bli skadet³²⁷. Prevensjonsargumentet innebærer at et erstatningsansvar vil bidra til å forhindre potensielle skadevoldere i å volde skade. Erstatningsansvaret skal

326 Se Lødrup, 1999, s. 85

327 Se Rognstad og Stenvik, 2002, s. 513

derfor bl.a. sikre at en respekterer opphavsretten. Det kan imidlertid reises spørsmål om erstatningsnivået for åndsverk er høyt nok til å virke særlig preventivt.³²⁸ Der hvor rettskrenkeren er i god tro vil erstatning ikke ha særlig preventiv virkning.³²⁹

Berikelseskraft kan også bli aktuelt. Berikelseskraft innebærer her at rettskenker (debitor) må avstå økonomiske fordeler som han har mottatt ved rettsstridig atferd ovenfor opphavsmannen (kreditor). Jeg vil behandle dette nærmere nedenfor.

10.5 Krav basert på krenkelse av opphavsretten

Her vil jeg se på erstatnings- og berikelseskraft basert på krenkelse av opphavsrett. F.eks. hvis det i et åpen kildekodeprosjekt blir benyttet opphavsrettslig vernede elementer uten tillatelse, kan opphavsmannen både ha krav mot lisensgiver og mot lisenstaker av et åpent kildekodeprogram. Videre kan bidragsyteren (opphavsmannen) til et åpent kildekodeprosjekt ha krav mot (fysiske eller juridiske) personer som benytter programmet uten lisens eller i strid med opphavsrettslig relevante bestemmelser i lisensen.

10.5.1 Erstatningskrav

Av åvl. § 55 1. ledd følger det: «Skade som volder ved overtredelse som nevnt i § 54 ... kan kreves erstattet etter *alminnelige erstatningsregler*.» § 54 nevner i bokstav a «bestemmelser gitt til vern for opphavsretten i eller i medhold av 1. og 2. kapittel». I kapittel 1 § 2 finner vi opphavsmannens enerett til eksemplarframstilling og tilgjengeliggjøring for allmennheten. Dette betyr at opphavsmannen i utgangspunktet har et erstatningskrav mot de som bryter hans enerett og de som medvirker til dette.³³⁰

Culpa er den alminnelige erstatningsregelen i norsk rett og den regelen § 55 1. ledd sikter til.³³¹ For å bli ansvarlig etter culpa-regelen må skadevolder ha utvist culpøs adferd i forhold til krenkelsen, dvs. han må ha oppradd uaktsomt eller forsettlig. Sempel uaktsomhet er nok, dvs. rettskrenker burde handlet

328 Se Rognstad og Stenvik, 2002, s. 512

329 Se Rognstad og Stenvik, 2002, s. 519

330 Medvirkning rammes av § 54 2.ledd. Se også Rt. 2005 s. 42 (Napster) avsnitt 59

331 Tilsvarende Garder, 2002, s 248

annerledes. Det er selvfølgelig ikke krav om straffeskyld for at erstatning skal bli aktuelt³³².

De typiske postene for erstatningskrav ved krenkelse av opphavsrett er tappt vederlag, tappt avsetning (markedsforstyrrelser) og oppsporingskostnader.

Vederlagstapet³³³ er det tapet i lisensinntekter opphavsmannen har hatt fordi skadevolder ikke har kjøpt en lisens av ham. I tilfeller der utnyttelsen ikke ville funnet sted hvis det ikke var for opphavsrettsbruddet, er det ikke sikkert kravet på vederlag kan begrunnes i erstatningslæren. Dette kommer jeg tilbake til nedenfor under restitusjon av berikelse.

Hvis vederlagskravet vinner frem har opphavsmannen krav på «rimelig vederlag»³³⁴. Retten har et relativt fritt skjønn³³⁵. Har opphavsmannen faste vilkår for å gi lisens, danner disse vanligvis utgangspunktet.³³⁶ For at erstatningen skal virke preventivt kan er det være grunn til å sette den høyere enn normalt vederlag, ellers risikerer rettskrenker ikke annet ved brudd enn å betale det han ellers hadde måtte³³⁷. Faste vederlagvilkår kan en finne både ved proprietær programvare og ved dual licensing. Hvis ikke opphavsmannen har faste vilkår, må vederlagskravet fastsettes skjønnsmessig.

Ved krenkelser av opphavsretten til åpent kildekodeprosjekt kan en stille spørsmålet om opphavsmannen kan ha et vederlagstap siden programvaren jo ofte er tilgjengelig gratis for alle. Hensynet til at opphavsrettsbrudd skal ha en konsekvens taler for at opphavsmannen kan ha et slikt tap.³³⁸ I tilfeller der den åpne kildekodelisensen er basert på copyleft, eller oppstiller andre forpliktelser av betydning, vil rettskrenkerne kunne ha utnyttet rettigheten på gunstigere «betingelser» i krenkelsesperioden enn om han hadde lisensiert verket lovlig. Det er da altså snakk om en utnyttelse som ikke kan sammenlignes med den gratis³³⁹ utnyttelsen som tilbys gjennom den åpne kildekodelisensen med copyleft. Det kan derfor være rimelig å kreve vederlag basert på hva en tilsvarende lisens ville ha kostet, altså å se hen til vederlagsnivået for proprietære alternativer der det eksisterer. (Se også krav basert på uberettiget berikelse nedenfor).

332 Erstatning kan kreves selv om det ikke er grunnlag for straffeansvar, jfr. ordlyden, Ot. prp. nr. 26 (1959-60) s. 110 og Rt. 2005 s. 42 (Naspster) avsnitt 59

333 Se Stray, 1989, s. 314-315

Se Koktvedgaard, 2005, s. 463-466

334 Se f.eks. RG 1999 s. 330 «MOCS» og Stray, 1989, s. 315

335 Se Garder, 1986, s. 251

336 Se Stray, 1989, s. 315

337 Erstatning er imidlertid tradisjonelt begrenset til faktisk økonomisk tap så hvor langt dette argumentet holder er noe usikkert.

338 Se forrige fotnote.

339 Se nærmere om dette i kapittelet over som er kalt «Gratis?»

Avsetningstapet³⁴⁰ (eller omsetningstapet om en vil) er tapet av lisensinntekter fra andre enn rettskrenker som skyldes skadevolders urettmessige bruk av åndsverket. Dersom den ulovlige benyttelsen av åndsverket fører til at rettighetshaver ikke kan avsette åndsverket i så stor grad som planlagt, eller så hurtig som planlagt, på grunn av den ulovlige benyttelsen oppstår det et avsetningstap. Dette tilsvarer vanligvis tappt omsetningsvolum multiplisert med nettoinntekt per enhet³⁴¹.

Hvis proprietær programvare er brukt i et åpen kildekodeprosjekt, kan det fort bli snakk om avsetningstap. For opphavsmennene til et rent åpen kildekodeprosjekt blir det nok vanskelig å vinne frem med et krav om avsetningstap siden programvaren jo vanligvis er gratis tilgjengelig for alle, men ved dual licensing kan situasjonen bli annerledes da de som driver med dual licensing kan ha tappt lisensinntekter fra den alternative lisensen.

Markedsforstyrrelsestap³⁴² er tap som kommer av at markedssituasjonen for et åndsverk, inkludert goodwill og rénomeet, er forstyrret. Dette er ofte vanskelig å bevise og er nok mest aktuelt dersom det er piratvirksomhet av et fullstendig produkt, siden markedet jo må kunne identifisere verket.

Oppsporingskostnadene³⁴³ er kostnadene ved å påvise immaterialrettsbruddet. Det kan f.eks. være utgifter å kjøpe produktet, som program eller hardware, der den koden er ulovlig brukt³⁴⁴. I tillegg kan kostnader med å reverse engineer-e produktet etc. bli aktuelle³⁴⁵. Saksomkostninger for å få tilgang til IP-adresser for å spore krenkeren må antakelig også kunne kreves. Kravet til påregnelighet vil selvfølgelig sette en grense.

Etter åvl. § 55 1. ledd 2. setning er det også mulig å få oppreisning for ikke-økonomiske (eller ideelle) skader: «Er opphavsmanns ... rett ... krenket *forsettlig eller grovt uaktsomt*, kan retten dessuten tilkjenne ham et pengebeløp som oppreisning for *skade av ikke-økonomisk art*.» En ser av bestemmelsen at simpel uaktsomhet ikke er nok. En sier gjerne at det må foreligge «en kvalifisert klanderverdig opptreden som foranlediger sterke bebreidelser for

340 Se Koktvedgaard, 2005, s. 462

Se Stray, 1989, s. 315

Se Gaarder, 1968, s. 251

341 Se Rognstad og Stenvik (2002) s 534

342 Se Koktvedgaard, 2005, s. 462

Se Stray, 1989, s. 316

Se Gaarder, 1968, s. 251

343 Se Stray, 1989, s. 316

Se Koktvedgaard, 2005, s. 462

344 Se f.eks. «D-link»-dommen av Landgericht Frankfurt am Main i 2006

345 Se f.eks. «D-link»-dommen av Landgericht Frankfurt am Main i 2006

mangel på aktsomhet»³⁴⁶ for at en handling skal være grovt uaktsom³⁴⁷. Videre er det opp til rettens skjønn å innvilge oppreisning selv om skyldkravet foreligger, jfr. «kan»³⁴⁸. Bestemmelsen er mest aktuell ved uverdigg o.l. fremstilling av åndsverk. Alt i alt er det nok lite aktuelt å anvende bestemmelsen i forhold til opphavsrettsbrudd på programmer, enten de er proprietær eller åpne.

10.5.2 Krav basert på urettmessig berikelse

Når det gjelder vederlagskravet i tilfeller der utnyttelsen ikke ville funnet sted hvis det ikke var for opphavsrettsbruddet, som f.eks. tilfeller der rettighetshaver ikke ville ha tillatt utnyttelsen, kan det argumenteres for at utnyttelsen ikke er et tap for opphavsmannen. I det hypotetiske hendelsesforløpet, der den skadegjørende handling tenkes bort, ville det jo ikke blitt betalt et vederlag. Det kan derfor hevdes at det ikke foreligger et erstatningsberettiget tap.

Denne situasjonen er ikke så fjerntliggende hvis det skulle vise seg at de som har laget den åpne kildekoden har sterke filosofiske overbevisninger mot proprietær programvare. En kan da hevde at det ikke er et tap for dem, siden programmet ikke ville ha blitt solgt. En tilsvarende situasjon kan oppstå dersom opphavsmannen faktisk har tjent på opphavsrettskrenkelsen gjennom f.eks. publisitet.

Da «Den lille havfrue»-statuen i København ble plagiert som suvenirfigurer fikk opphavsmannen ikke erstatningen ut fra betraktningen at opphavsmannen ikke ville gitt tillatelse til utnyttelsen hvis han hadde blitt forespurt.³⁴⁹ Et slikt resultat, men ikke nødvendigvis begrunnelse, vil fort svekke opphavsretten.

I «Alle Menn»-dommen³⁵⁰ hadde ukebladet «Alle Menn» overskredet sitatretten og ble holdt erstatningsansvarlig. Ukebladet fremstilte en medisinsk rapport på en pornografisk måte og retten mente derfor at «Samtykke mot vederlag for slik bruk er det vanskelig å tenke seg...». Retten utmålte likevel «erstatning, jfr. Åndsverklovens § 55, 1. setning, jfr. 1. ledd pkt. a».

346 Se Rt. 1970 s. 1235

347 Selv om jeg ikke kan finne denne setningen anvendt på åvl. § 55, er definisjonen ofte brukt om grov uaktsomhet og gir nok god, men abstrakt, veiledning. Formuleringen er hentet fra strafferetten, men er vanlig å bruk også i obligasjonsretten. Se f.eks. Hagstrøm, 1996, s. 497

348 Se også Gaarder, 1986, s. 254

349 Dom av Vestre Landsret (Danmark) den 27. juni 1978 i NIR 1982 s 105-111

«Det må efter det fremkomne lægges til grund, at sagsøgeren ikke ville have meddelt samtykke til fremstilling mod lisensbetaling af en havfruemodel som den under sagen omhandlede. På baggrund heraf ... findes det ikke godtgjort, at sagsøgerne har lidt noget til erstatning berettigende tab i form af mistede licensindtægter...»

Se Stray, 1989, s. 314-315

350 Se Oslo byrett 28. september 1982

Erstatningen ble basert på det retten kalte et «kvasi-kontraktsmessig vederlagssynspunkt». Dette kommer jeg tilbake til.

I disse situasjonene er nok kompensasjon for vederlagskravet bedre forklart med berikelseslære enn ren erstatningsrett.³⁵¹ Argumentet er at den som på rettsstridig måte tilegner seg andres verdier, altså utnyttet den immaterielle rettigheten, bør betale for det.³⁵²

Støtte for dette synspunktet finner vi også i en gammel dom fra Høyesterett.³⁵³ En arkitekt hadde laget tegninger til et byggeprosjekt under forutsetningen at han skulle bli utførende arkitekt hvis tegningene ble brukt. Tomten ble solgt og kjøperen fikk overlevert tegningene. Selv om han kjente til avtalen, avsto han at arkitekten skulle være utførende arkitekt og benyttet deler av tegningene. Høyesterett mente kjøperen hadde fått en «vesentlig økonomisk fordel» og at «Når en på denne måten *ved en uberettiget forføyning* utnytter en annens arbeid til egen fordel, tilsier *rett og rimelighet* at det blir *ytet vederlag* for utnyttningen».

Avgjørelsen blir altså ved basert på «*rett og rimelighet*». Gaarder³⁵⁴ forklarer at byretten i denne saken hadde uttrykkelig brukt Stangs lære om rettsbrudd som rettshandel, altså det kvasi-kontraktsmessige grunnlaget³⁵⁵ som vi også så byretten bruke i «Alle Menn»-dommen. Kvasi-kontraktene stammer egentlig fra romerretten, og er ett av de tre klassiske kravgrunnlagene i romerretten.³⁵⁶ Kvasi-disposisjonen defineres av Udsen som «tilegnelse af en ydelse uden at foretage en handling, der er egnet til over for omverdenen at tilkendegive vilje til at præstere den forudsatte modydelse».³⁵⁷ Rimelighet tilsier at «disponenten» (rettskrenkeren) må ha utvist uaktsomhet for at han skal ha foretatt en kvasi-disposisjon.³⁵⁸ Selv om berikelseslæren ikke blir direkte brukt i dommene over, er det i stor grad den samme bakenforliggende tanke bak berikelseslæren og kvasikontraktene,³⁵⁹ selv om de i det ytre er forskjellige.

Det er verdt å merke at det i noen tilfeller vil være mulig å bygge vederlagskravet på den objektive bestemmelsen i § 55 2. ledd³⁶⁰. Av åvl. § 55 2. ledd følger det at «*Selv om gjerningsmannen har handlet i god tro*, kan den

351 Se også Gaarder, 1989, s. 254

352 Se Rognstad og Stenvik, 2002, s. 518 og Gaarder, 1968, s. 252

353 Se Rt. 1940 s. 48

Se omtale i Hagstrøm, 2003, s. 664-665 og Rognstad og Stenvik, 2002, s. 525

354 Se Gaarder, 1968, s. 253

355 Se nærmere om kvasikontrakter Udsen, 2006, s. 119-122, Hov, 2002, s. 84-86 og Hagstrøm, 2003, kapittel 26

356 En skilte mellom kontrakt, delikt og kvasikontrakt. Se Hagstrøm, 2003, s. 660

357 Se Uden, 2006, s. 119

358 Se Udsen, 2006, s. 120

359 Se f.eks. Hagstrøm, 2003, s 660

360 Se Rognstad og Stenvik (2002) s 524 og Vinje, 1986, s. 104

fornærmede *uansett skadens størrelse* kreve utbetalt *nettofortjenesten ved* den ulovlige handling.»

Krav etter denne bestemmelsen kan altså gjøres gjeldene uavhengig av rettskrenkers subjektive skyld og uavhengig av om opphavsmannen har lidd et tap³⁶¹. Videre er det bare nettofortjeneste som kan kreves,³⁶² utgifter (som lisensavgifter, fremstillingsomkostninger, reklame etc.) må trekkes fra. Det er videre bare merinntektene som følger av den ulovlige bruken som kan kreves, jfr. «ved»³⁶³. Det betyr altså at dersom kun et kodefragment ulovlig er brukt i et datamaskinprogram må retten skjønnsmessig fastslå hvor mye mer nettofortjenesten til rettskrenkeren ble som følge av at dette fragmentet ble brukt. Det vil åpenbart kunne være svært vanskelig i en del tilfeller.

10.5.3 Bruk av programmer etter oppdaget vanhjemmel

Hvis brukeren fortsetter å bruke et program etter oppdaget vanhjemmel kan det oppstå problemer: Vi kan ta utgangspunkt i en situasjon der en bruker i aktsom god tro har mottatt et program med vanhjemmel. Når brukeren så benytter programmet skjer det, som sagt, en eksemplarfremstilling når programmet blir laset opp i minne. Dette er et brudd på åndsverkloven, men siden brukeren er i aktsom god tro oppstår det ikke et erstatningsansvar (eller en kvasi-kontrakt etc.).³⁶⁴ Så snart brukeren kommer i ond tro om at det foreligger vanhjemmel utsetter han seg for potensielle krav ved å fortsette å bruke (eksemplarfremstille) programmet. En kan merke at åvl. § 12 2 b) gjør at unn-taket for eksemplarfremstilling til private bruk ikke får anvendelse, og at åvl. § 11a som tillater visse midlertidige eksemplarer heller ikke får anvendelse på grunn av at 2. ledd utelukker «datamaskinprogrammer».

Her ser en altså et skille mellom f.eks. litteratur og dataprogrammer. Hvis en i god tro kjøper en piratkopiert bok kan en fortsette å nyttegjøre seg den (lese den) uten at en utsetter seg for et erstatningskrav, siden det ikke skjer en eksemplarfremstilling. For dataprogrammer ser situasjonen ut til å være den motsatte.

En kan lure på om denne situasjonen alltid er heldig. Fortsatt bruk av programmet kan være nødvendig for å sikre betydelige verdier. Bruk etter et

361 Se Gaarder, 1986, s. 254 og Vinje, 1985, s. 104

362 I Stavanger byretts dom i RG 1967 s. 158 ble det uttalt når det gjaldt vurderingen av § 55 2. ledd «... at det også må legges vekt på *hva det ville kostet* for Consul gjennom forhandling enten med Røe eller med en annen konsulent (arkitekt), *å få benytte en planløsning lik eller lignende* den som Røe fremkom med. På dette grunnlag ansettes nettofortjenesten for Consul, og dermed erstatningen til Røe, til 3000 kroner.»

363 Se Gaarder, 1986, s. 254

364 §55 2. ledd er, som nevnt, objektiv.

betalt erstatningskrav er imidlertid også problematisk. Å betale erstatning gir ingen tvangslisens.

En kan reise spørsmål om åvl. §56 2. ledd kan få anvendelse, hvis «betydelige økonomiske ... verdier ville gå tapt, kan likevel retten etter omstendighetene tillate at *fremstilte eksemplarer* gjøres tilgjengelig for almenheten mot erstatning eller oppreisning til fornærmede.» Ut fra ordlyden passer bestemmelsen dårlig; fortsatt bruk av et program innebærer fremstilling av *nye* eksemplarer. Hvis en derimot ser på §56 2. ledd som et uttrykk for et mer generelt prinsipp for å unngå urimelige situasjoner som en absolutt rett til å hindre ulisensiert utnyttelse kan medføre, vil en kunne løse problemet beskrevet over.

En annen løsningsmodell finnes i lov om vern av kretsmønstre for integrerte kretser. I § 7 oppstilles det en rett til utnyttning av kretsmønstre som er fremstilt, utnyttet ervervsmessig eller importert i strid med loven på nærmere bestemte vilkår.

10.6 Krav basert på kontrakt

Her skal vi se på lisenstakers erstatningskrav mot lisensgiver når lisenstaker har fått lisens på et program hvor det er vanhjemmel. Tapet kan være et resultat av at lisenstaker må opphøre å bruke programmet som følge av vanhjemmelen. Det kan også være et regresskrav som følge av at lisenstaker har måtte betale opphavsmannen erstatning.

Nedenfor vil jeg se nærmere på hvordan ansvaret ut fra reelle hensyn bør fordeles, for så å undersøke bakgrunnsretten og avtalene. Noen av avtalene fraskriver lisensgivers ansvar i stor grad, jeg vil derfor også se nærmere på sensur av disse avtalene.

10.6.1 De lege ferenda – Hvordan bør ansvaret mellom lisensgiver og lisenstaker bli fordelt?

Åpen kildekodeprosjekter varierer mye. Noen har få bidragsyttere, mens andre har svært mange. Flere bidragsyttere må nok kunne hevdes å øke sjansen for at det oppstår vanhjemmel.

For en lisenstaker vil det nok fort bli praktisk umulig å undersøke hvem som har bidratt med hva til prosjektet, og hvilke rettigheter disse bidragsyterne har. Lisenstaker vil både kunne mangle kompetanse til å gjøre det og tilgang til informasjon. I tillegg vil det være en veldig omfattende oppgave.

Når lisensgiver er den som har bidratt med koden det foreligger vanhjemmel til er det åpenbart at han normalt vil være nærmere enn lisenstaker til å kjenne til eventuell vanhjemmel.

Når lisensgiver er den som leder prosjektet og er den som tar imot bidragene og velger hvilke bidrag som får slippe inn, vil kontrollen være atskillig enklere for ham enn for lisenstaker. (Selv om det ikke dermed er sagt at kontrollen er enkel). Prosjektleder har her mye bedre oversikt over prosjektet og muligheter til å føre kontroll med bidragsytere enn lisenstaker. Han har, i motsetning til lisenstaker, mulighet til å hindre at anonyme bidrag kommer med, og han har også mulighet til å kreve at bidragsyterne påtar seg ansvar for at deres bidrag er deres eget gjennom en spesiell avtale (ofte kalt «Contributor Agreement»). (Hvis bidragene blir ytt under en lisensavtale med sterke ansvarsfraskrivelser er det ikke i første omgang klart at bidragsyterne har et ansvar for vanhjemmel knyttet til bidragene. Dette kommer jeg tilbake til nedenfor.)

Det foreligger derfor mange gode grunner som taler for at lisensgiver ofte er nærmere til å bære risikoen for vanhjemmel enn lisenstaker.

Motargumentet mot at lisensgiver bør bære risikoen er at lisensgiver ofte ikke får noe vederlag for lisensen, og at han derfor har et klart behov for å redusere sin risiko. Med store prosjekter med mange bidragsytere og mange lisenstakere kan den økonomiske risikoen ved vanhjemmel være stort. Denne risikoen vil lisensgiver, som sagt, i noen grad kunne fordele ved å kreve at bidragsytere skal bære ansvar for vanhjemmel ved sine bidrag gjennom en bidragsyter-avtale («Contributor Agreement») eller tilsvarende. Det er allikevel langt fra sikkert at det er noe økonomisk å hente fra bidragsytere som har kommet med ulovlige bidrag.

10.6.2 Hva tilsier bakgrunnsretten?

Det første spørsmålet en kan stille seg er om kjøpsloven kan komme til anvendelse. Når det gjelder lisensiering, som den i åpen kildekodeavtaler, kommer ikke kjøpsloven til anvendelse siden den er lite egnet for å regulere stiftelser av bruksrett o.l.³⁶⁵ I tillegg kommer det at kjøp innebærer en betaling i penger

365 Se Lassen (1981): «Uansett om det her gis enlisens eller annen enerett, eller bare en ikke eksklusiv utnyttelsesrett, vil nok de fleste mene at man her står overfor overdragelser som kjøpslovgivningen overhodet ikke kommer til anvendelse på - det dreier seg ikke om kjøp, men om stiftelse av bruksrettslignende rettigheter (« upplåtelse »). Kjøpslovgivningen vil også ofte passe dårlig, og det er allerede derfor naturlig å søke andre steder etter egnede deklarasjoniske regelsett.»

Hadde det derimot vært snakk om en total overføring av opphavsretten ville nok løsningen kunne blitt en annen. Se Lassen (1981) og Stray, 1989, s. 173. Også i disse tilfellene må det selvsagt taes hensyn til opphavsrettsoverføringens spesielle karakter.

Se også Krüger, 1999, s 6

eller naturalia. Ved åpen kildekodelisenser får lisensgiver ofte ingen klar motytelse. (Selv om lisensgiver ved copyleft-avtaler vil kunne få lisens til videreutviklede deler av programmet dersom lisenstaker velger å tilgjengeliggjøre disse for ham). Situasjonen svarer altså ikke til den ved tradisjonelle kjøp.

Siden heller ingen andre obligasjonsrettslige lover passer, må vi benytte alminnelig obligasjonsrett. Når en skal finne frem til den alminnelige obligasjonsretten kan det allikevel være aktuelt å se på kjøpsloven, selv om den ikke kommer direkte til anvendelse.

Det blir på flere områder oppstilt et objektivt ansvar for vanhjemmel. Begrunnelsen som blir gitt for dette er at debitor (lisensgiver) skal være nærmere til å kjenne til rettsmangelen enn kreditor (lisenstaker). Kjøpsloven § 41(2) oppstiller objektivt ansvar for opprinnelig vanhjemmel, både total og partiell, men ikke etterfølgende³⁶⁶. Av § 41(4) ledd følger det «For tredjemannskrav som bygger på immaterialrett (jf § 96) gjelder reglene i første ledd tilsvarende»; Altså hvis det i tingen som blir kjøp er tredjemannskrav basert på immaterialrett har selger kun et kontrollansvar.³⁶⁷

Husleieloven (som jo til en viss grad ligner våre lisenser siden den også gjelder en bruksrett) har også objektivt ansvar for alle opprinnelige vanhjemler, jfr. husll. § 2-16 2. ledd. I motsetning til nåtidens kjøpslov gjaldt kjøpsloven fra 1907 sin bestemmelse i § 59 om objektivt ansvar bare opprinnelig og total vanhjemmel³⁶⁸. Det samme er situasjonen for avhendingsloven § 4-17 hvor en har basert dette på at kjøper er like nær som selger til å kjenne til partielle vanhjemler. I andre lover som håndverkertjenesteloven og tidspartloven er det ikke hjemlet objektivt erstatningsansvar for vanhjemmel.

I Rt. 1992 s. 620 uttalte Høyesterett i at «Jeg bemerker likevel at det etter mitt syn er mye som taler for at den som overdrar en rett uten selv å ha rettslig adgang til å overføre den, regelmessig vil måtte svare for medkontrahentens tap på objektivt grunnlag for slik rettsmangel.»³⁶⁹ Uttalelsen er allikevel bare et obiter dictum da det i saken forelå cupla, men signalet er klart.

366 Se Hagstrøm, 2003, s. 510

367 Ot .prp. nr. 44 (2001-2002) om forbrukerkjøpsloven punkt 3.18.1 legger en annen forståelse til grunn: «Kjøpsloven § 41 annet ledd pålegger selgeren et rent objektivt erstatningsansvar for tap som følge av en rettsmangel som forelå ved kjøpet, og som kjøperen verken kjente til eller burde ha kjent til. ... Dette særlige ansvarsgrunnlaget gjelder tilsvarende ved tredjepersonskrav som bygger på immaterialrett, jf. kjøpsloven § 41 fjerde ledd første punktum». Dette svarer imidlertid ikke til forarbeidene til kjøpsloven Ot. prp. nr. 80 (1986-1987) s. 42 hvor det står «Etter proposisjonsutkastet § 41 (4) skal det ... ikke gjelde et tilsvarende strengt ansvar som for opprinnelige rettsmangler, jf § 41 (2).» Denne siste forståelsen er nok den korrekte, jfr. ordlyden i loven.

368 Se Hagstrøm, 2003, s. 510

369 Dommen gjaldt et norsk firma som inngikk avtale med et svensk firma om å skulle være eneforhandler i Norge for lyskasser for visning av reklame.

Selv om regelen fra kjøpsloven av 1907 ble sett på som et utslag for alminnelig obligasjonsrett og uttalelsen fra Høyesterett veier tungt, gjør nyere lover, som ikke har et objektivt ansvar, at det kan hevdes at situasjonen er mer uklar³⁷⁰. I tillegg kommer det at det bare var ansvaret for total vanhjemmel som ble ansett som alminnelig obligasjonsrett. Dette bidrar til å gjøre situasjonen noe uklar.

For overdragelse av immaterielle rettigheter er det i teorien i stor grad lagt til grunn et objektivt ansvar.³⁷¹ Konklusjonen må da bli at mye taler for at det etter bakgrunnsretten i utgangspunktet foreligger et objektivt ansvar for vanhjemmel ved lisenser.

Der hvor kreditor (lisenstaker) er i ond tro om vanhjemmelen vil det være urimelig å oppstille et objektivt ansvar for debitor (lisensgiver). Dette gir seg utslag i kjl. § 41(2) som oppstiller et vilkår om at kjøper ikke må ha «kjent[] eller burde ha kjent til» vanhjemmelen for at selger skal være objektivt ansvarlig. Reglen har så gode grunner for seg at den nok må ansees som alminnelig obligasjonsrett.

Et spørsmål som kan se ut til å ha sneket seg ut av mye moderne litteratur ser ut til å være erstatningsansvaret for vanhjemler med gaver. Augdahl skriver at «Den alminnelige lære helt fra Hallager har imidlertid vært at vanhjemmelsansvaret – bortsett fra gaver – er ganske uavhengig av hvorvidt der var noget å legge avhenderen til last...»³⁷² Han antar altså at det objektivet ansvaret ikke gjelder for gavedisposisjoner. Selv om ikke noen grunn oppgis er løsningen godt i samsvar med det generelle synet om at en ved gaveløfter «som oftest er mer lemfeldig mot giverne enn ellers»³⁷³. Mye taler derfor for et culpa ansvar når det gjelder gaver. Når det gjelder gaver generelt skriver Augdahl videre at det «i *almindelighet* bør det stilles større krav til skylden her enn hvor det gjelder gjensidig bebyrdede forhold. Uaktsomheten bør være grov for å komme i betraktning.»³⁷⁴

Spørsmålet som reiser seg da er om åpen kildekodeavtaler kan betegnes som gaver. Dette spørsmålet er det nok ikke mulig å svare generelt på, en må i stedet basere seg på en konkret vurdering av det enkelte forhold. En gavedisposisjon innebærer en frivillig formueforskyvning på givers bekostning uten vederlag eller med klart redusert vederlag (gavesalg) fra gavemottaker.³⁷⁵ På den ene siden yter ofte ikke lisenstaker et vederlag eller annen klar motytelse

370 Se Hov, 2002, s. 199

371 Se, Stray, 1989, s. 173 og Torvund, 1997, s. 230

372 Se Augdahl, 1979, s. 274

373 Se Hov, 2002, s. 26

374 Se Augdal, 1978, s. 234

375 Se f.eks. Lødrup, 2001, s. 173 om gavebegrepet generelt og Hov, 2002, s. 26

når han mottar en åpen kildekode-lisens. Dette taler for at det er en gave. På en annen side påtar lisensgiver seg ofte forpliktelser (særlig ved copyleft³⁷⁶, men også andre forpliktelser er aktuelle, som f.eks. at lisenstaker skal legge ved lisensen ved tilgjengeliggjøring). Dette taler imot at det er en gave.

I tillegg oppstilles det ofte et krav om gavehensikt. Gavemottakers berikelse må ha vært givers mening. Dette er nok vanskelig å si at alltid er oppfylt. Når en bedrift lisensierer åpen kildekode i håp om å selge andre varer og tjenester har hensikten klart ikke vært den rene gave. For private bidragsytere står nok gavehensikten sterkere, men også disse kan ha alternative motiver.

Kategoriseringen «gave» passer derfor ikke alltid. Ofte er nok er likhets-trekkene med gavesituasjonen så stor at det nok er snakk om en gavelignende disposisjon og i disse tilfellene taler mye for at det kun foreligger et cupla ansvar. Et slikt syn kan en også finne støtte for når Augdahl sier om sitt syn om erstatningsansvar ved gaver at «Det anførte gjelder ikke bare gaver i snever forstand, men også andre vederlagsfrie tilsagn».

Som sagt kan det Augdahl skriver tale for et krav om grov uaktsomhet.³⁷⁷ Det er for lisensgiver klart at lisenstaker handler i tillit til at han får rett hjemmel, og at konsekvensene av vanhjemmel kan bli store for lisenstaker. Dette taler klart for at lisensgiver bør utvise aktsomhet og at terskelen for å holde lisensgiver ansvarlig ikke bør stilles svært høyt. Altså at simpel uaktsomhet bør være tilstrekkelig.

Mye taler derfor for culpa ansvar for vanhjemmel ved åpen kildekodeavtaler, men en må, som sagt, foreta en konkret vurdering i hver enkelt situasjon. At løsningen er noe usikker har allikevel ikke den store betydningen siden de fleste åpen kildekode-lisensene regulerer ansvaret eksplisitt.

10.6.3 Hva tilsier avtalene?

GPLv2 regulerer erstatning ved vanhjemmel i artikkel 11 og 12. Bestemmelsene er gjengitt nedenfor, men med normale bokstaver. (I originalteksten er bestemmelsene skrevet med store bokstaver fordi en mener det vil sikre at de ikke skal bli oversett).

³⁷⁶ Det er imidlertid opp til lisenstaker om han vil sette seg i en posisjon der det blir aktuelt.

³⁷⁷ Se Augdahl, 1978, s. 234. Selvsagt kan Augdahl ikke her ha tenkt på åpen kildekode-lisenser siden de først så lyset 10 år etter hans død, men prinsippene er de samme. I tillegg skriver han generelt om erstatning som følge av mislighold og ikke om vanhjemmel spesielt når han tar til ordet for et krav om grov uaktsomhet.

«No Warranty

11. Because the program is licensed free of charge, there is *no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated* in writing the copyright holders and/or other parties provide the program «*as is*» *without warranty of any kind, either expressed or implied*, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction.

12. In *no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages*, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.»

Artikkel 11 regulerer mangler. Kort sagt søker bestemmelsen å avskjære at noe er en mangel så langt det er rettslig tillatelig, jfr. «*the extent permitted by applicable law*». Begrepet «law» omfatter selvsagt både lovfestet og ulovfestet rett. I avtalen ser vi også at implisitte garantier («implied warranties») blir avskåret. «Implied warranties» er de garantiene som antas å foreligge ved en avtale selv om de ikke er uttalt³⁷⁸. Disse følger av retten («law»), og det er særlig «the implied warranty of non-infringement» og «the implied warranty of title» som vi er interessert i. Denne garantien skal sikre at det ikke foreligger vanhjemmel til verk som blir lisensiert. Avtalen fraskriver seg altså at det er en mangel hvis det foreligger vanhjemmel til lisensen som blir gitt. Artikkel 12 avskjærer erstatning så langt retten tillater det.

Begge bestemmelsene gjør oppmerksom på at noe annet kan avtales med lisensgiver. Dette er en tjeneste en del firmaer selger som tillegg.

Situasjonen med en «fullstendig» ansvarsfraskrivelse fra lisensgivers side er også tilfelle for de fleste andre åpne kildekode-lisenser, deriblant BSD-lisensen³⁷⁹. Som jeg påpekte ovenfor er det mange grunner som tilsier at lisensgiver normalt er mye bedre egnet til å forhindre opphavsrettsbrudd enn lisenstaker. Ut fra denne argumentasjonen har Lawrence Rosen utformet Academic Free

378 Se Beatson, 2002, s.145

379 Tilsvarende Rosen, 2002

License³⁸⁰. Det følger av avtales artikkel 7 at: «*Licensor warrants that the copyright in and to the Original Work and the patent rights granted herein by Licensor are owned by the Licensor or are sublicensed to You under the terms of this License with the permission of the contributor(s) of those copyrights ...*»

Her ser vi at lisensgiver påtar seg vanhjemmelsansvar.

I EUPL-lisensen ser vi omtrent samme løsning. Av artikkel 6 følger det: «*The original Licensor warrants that the copyright in the Original Work granted hereunder is owned by him/her or licensed to him/her and that he/she has the power and authority to grant the Licence.*

Each Contributor warrants that the copyright in the modifications he/she brings to the Work are owned by him/her or licensed to him/her and that he/she has the power and authority to grant the Licence.»

10.6.4 Kan ansvarsfraskrivelsene opprettholdes? - Sensur

Spørsmålet som reiser seg etter å ha sett på lisensavtalene over er om lisensavtaler, som GPL, som kategorisk avviser ethvert ansvar vil få gjennomslag i retten. Utgangspunktet er klart, avtaler skal respekteres av retten, jfr. pacta sunt servanda og NL-5-1-2.

Siden det er klart at å anvende denne regelen uten unntak vil føre til urimelige situasjoner, oppstiller avtl. § 36 en generell hjemmel for helt eller delvis å sette til side eller endre urimelige avtalevilkår.³⁸¹ Andre bestemmelser, som NL-5-1-2, kan også sette grenser for ansvarsfraskrivelser, men avtl. § 36 har en videre utforming og er derfor mer interessant å se på.

For at avtl. § 36 skal komme til anvendelse må urimeligheten være klar.³⁸² Det må foretas en bred vurdering.³⁸³

Avtaleloven kan selvsagt ikke brukes på standardavtalene før det foreligger et konkret avtaleforhold. Det er den enkelte avtale som skal vurderes. Dette innebærer at vurderingen av samme standardavtale kan variere fra sak til sak i forhold til profesjonaliteten, kunnskapen og andre konkrete forhold til de

380 Se Rosen, 2002

381 Se Woxholth, 2003, s 368

382 «Til sjuende og sist bør det avgjørende være om vilkåret i det konkrete tilfelle er så urimelig at dommeren ikke finner å kunne opprettholde det uforandret. ... Slik tilsidesettelse vil neppe skje uten at domstolene finner urimeligheten klart eller tydelig påvist. Det ligger også i formuleringen «urimelig» at et vilkår må være positivt urimelig for at det skal kunne lempes; det er selvfølgelig ikke nok at det kan tenkes rimeligere løsninger. «Urimelig» må oppfattes som i seg selv et ganske strengt kriterium, som det ikke vil være kurant å påberope.», jfr. Ot. prp. nr. 5 1982-83 s. 30

383 jfr. «omstendighetene forøvrig» i avtl § 36.

enkelte avtalepartene. Det er derfor ikke mulig, på generelt grunnlag, å gi noe svar på om ansvarsfraskrivelsen kan opprettholdes. Nedenfor vil jeg peke på noe typiske momenter.

Det er først når det samlede resultatet av kontrakten innebærer en ulempe for den ene parten at det vil være urimelig å gjøre avtalen gjeldende.³⁸⁴ Et gunstig vilkår kan altså veie opp for et ugunstig. At lisensgiver ikke betaler vederlag taler altså imot at ansvarsfraskrivelsene skal sensureres.

Det er ofte grunn til å sensurere ansvarsfraskrivelsen når den vil føre til et resultat som er i strid med det partene har forutsatt å oppnå ved kontrakten.³⁸⁵ Kjernen i lisensavtalen er nettopp lisensen, tillatelsen til å bruke åndsverket, en ansvarsfraskrivelse i forhold til vanhjemmel er derfor i strid med det lisenstaker har søkt å oppnå og undergraver kjernen i avtalen. Det er derfor stor grunn til å se med skepsis på slike bestemmelser med generell ansvarsfraskrivelse for vanhjemmel.

En avtalt risikofordeling mellom partene er det mindre grunn til å sensurere når risikoen er klart definert og begrenset.³⁸⁶ Dette fordi det er klarere for partene hva avtalen innebærer. GPLv2 og BSD sine ansvarsfraskrivelser for vanhjemmel er ikke spesielt klare og begrenset, og står derfor svakere.

Før den generelle lempingsregelen i avtaleloven § 36 ble innført, var det en rekke spesielle lempingsregler på formuerettens område.³⁸⁷ Deriblant en lempingsregel i åvl. § 27.³⁸⁸ Siden avtl. § 36 er laget med disse spesielle lempingsreglene som grunnlag må det fremdeles være relevant å se hen til dem. Åvl. § 27 ble ofte ansett å favorisere opphavsmannen. Dette må en imidlertid være forsiktig med å slutte fra. Grunnen til at det var mindre aktuelt å sensurere til lisenstakers fordel var at opphavsmannen ofte var den

384 Se Woxholth, 2003, s. 377

385 Se Hagstrøm, 1996, s. 458 og 459

386 Se Woxholth, 2003, s. 390

387 Se NOU 1979 nr 32, særlig kapittel 2

388 Bestemmelsen lød: «Er det for retten til å råde over et åndsverk avtalt vilkår som er i strid med god skikk på opphavsrettens område, eller viser det seg at avtalen fører til åpenbart urimelige resultater, kan avtalen kreves endret», jfr. NOU 1979 nr 32, kapittel 2.8

minst profesjonelle av partene³⁸⁹. Lisensgivers profesjonalitet i forhold til lisenstaker vil nok variere mye ved åpen kildekodeavtaler.

Det følger av avtl. § 36 at det skal tas hensyn til «partenes stilling». Det er derfor grunn til å trekke løftemottakers onde tro inn.³⁹⁰ (Lisensgiver er mottaker av løfte om at lisenstaker ikke skal gjøre erstatningskrav basert på vanhjemmel gjeldende.)

Der lisensgiver selv har opptrådt forsettlig eller grovt uaktsomt og lisenstaker er i god tro er det liten grunn til å opprettholde ansvarsfraskrivelsen. Denne læren om at forsett og grov uaktsomhet ikke kan fraskrives har lang tradisjon³⁹¹ og kan forankres i Rt. 1948 s. 370³⁹² og NL-5-1-2. Der hvor lisensgivers underordnede har handlet forsettlig eller grovt uaktsomt var situasjonen tidligere usikker, men i dag vil også slike handlinger føre til sensur³⁹³. Der hvor lisensgiver har forsettlig eller grovt uaktsomt bidratt til vanhjemmelsituasjonen bør det altså fort bli sensur av ansvarsfraskrivelsen.

Der hvor det bare foreligger simpel uaktsomhet er det også mulig at konkrete omstendigheter også kan føre til sensur.³⁹⁴

Hvis lisensgiver er klar over at han eller en annen har bidratt med ulisensiert kode foreligger det forsett og sensur av ansvarsfraskrivelsen bør bli følgen. Jeg vil også tro det fort går mot grov uaktsomhet hvis en mottar kode fra anonyme bidragsytere uten å sjekke nærmere. Anonymitet kan i seg selv selvsagt ikke være nok. I opphavsretten er det lang tradisjon for at opphavsmannen opererer under pseudonym, men mottakeren bør utvise ekstra varsomhet i forhold til disse bidragene. Jo større og jo mer komplekst bidraget er jo større grunn er det for mottaker til å undersøke opphavet.

389 Av Inst. O XI (1960-61) s. 26 følger det om åvl §27: «Resultatene må være « åpenbart urimelige » for at endring skal kunne kreves, og ved bedømmelsen av hva som er åpenbart urimelig for en part vil det selvfølgelig måtte tillegges vesentlig vekt hva denne part burde ha forutsett. Den mer forretningskyndige vil derfor vanskeligere kunne påberope seg bestemmelsen. *Dette vil - etter komitéens oppfatninger, føre til at det bare sjelden vil bli spørsmål om å endre en avtale til opphavsmannens ugunst, idet det i de tilfelle som kan oppstå normalt vil være opphavsmannens medkontrahent som er den mer forretningskyndige av partene, og som derfor ofte vil kunne sies å burde ha forutsett hva som kunne inntreffe.* Men helt å utelukke en revisjonsadgang for opphavsmannens medkontrahent ville etter komitéens mening kunne føre til støtende resultater.»

390 Se Woxholth, 2003, s. 399

391 Se Hagstrøm, 1996, s. 421

392 Dommen har i dag mest historisk interesse

393 Se Hagstrøm, 1996, s. 452: «langvarig og omfattende praksis [synes] å vise at selv ikke i typisk kommersielle kontrakter er Høyesterett tilbøyelig til å anse følgene av grov uaktsomhet fraskrevet, og det uten hensyn til om den grove uaktsomheten er utvist av ledelsen i selskapet eller underordnede»

394 Jfr. avtl. § 36 som ikke opererer med krav til grov uaktsomhet eller forsett. Se tilsvarende resultat Hagstrøm, 1996, s. 453 og 463

10.6.5 Konklusjon

Vi ser altså at selv om lisensgiver vanligvis er best egnet til å forhindre vanhjemmel knyttet til programmet, vil lisensgiver etter mange åpen kildekode-lisenser ikke bære et objektivt ansvar for slik vanhjemmel. Dette skiller seg fra situasjonen etter alminnelig obligasjonsrett når det gis en motytelse for programmet, men ikke nødvendigvis fra situasjonen som følger av lisensavtalene for slike kommersielle programmer, da svært omfattende ansvarsfraskrivelser også er vanlig for kommersielle programmer.

Ansvaret for vanhjemmel er altså, i prinsippet (de jure), i stor grad flyttet over på lisenstaker. På grunn av åpen kildekodeutviklingens ofte globale og distribuerte natur vil det for lisenstaker nok ofte være praktisk vanskelig å rette krav mot lisensgiver. Ansvaret blir nok derfor for mange tilfeller, i praksis (de facto), i enda større grad skjøvet over på lisenstaker.

11 LITTERATURLISTE

- Andenæs, Johs. (2004): *Alminnelig strafferett*. 5. utgave.
Oslo: Universitetsforlaget
- Andersen, Mads Bryde (2005): *IT Retten*. 2. utgave.
København: Gads Forlag
- Augdahl, Per (1978): *Den norske obligasjonsretts almindelige del*. 5. utgave
Oslo: H. Aschehoug & Co. (W. Nygaard)
- Beatson, J. (2002): *Anson's Law of Contract* 28. utgave.
Oxford: Oxford University Press
- Bull, Lars Jakob (1980): *Dommer og kjennelser i EDB-Rett I*
Oslo: Universitetsforlaget
- «Berkeley Software Distribution» http://en.wikipedia.org/wiki/Berkeley_Software_Distribution (11. februar 2008)
- Bing, Jon (1975): *Forlagsavtalen*. Oslo: Universitetsforlaget
- Bing, Jon (1985): *Opphavsrett og EDB*. Oslo: Universitetsforlaget
(CompLex, 2/85)
- Bing, Jon (1999): «Vurdering av opphavsrettslig krenkelse av datamaskinprogrammer - et praktisk perspektiv». I: *Nordiskt immateriellt rättsskydd*, 1991, s. 281-299
- Bing, Jon (1999) (2): «Copyright in Electronic Commerce and Private International Law». I: *Legal Aspects of Intellectual Property Rights in Electronic Commerce*. München: C.H. Beck'sche Verlagsbuchhandlung, s. 33-42
- Bing, Jon (2001): «The identification of applicable law and liability with regard to the use of protected material in the digital context». I: *E-commerce law and practice in Europe*. Cambridge: Woodhead Publishing Ltd. Section 9 Chapter 2
- Bing, Jon (2006): «De usynlige eksemplarene». I: *Lov og Data 2006*
- Bing, Jon (2008): *Ansvar for ytringer på nett*. Oslo: Universitetsforlaget

- Brodtkin, Jon (2007): «Open source impossible to avoid, Gartner says». I: Network World <http://www.networkworld.com/news/2007/092007-open-source-unavoidable.html> (8. november 2007)
- Bull, Jørgen (1973): *Rettslig beskyttelse av dataprogrammer*. Oslo: Universitetsforlaget
- Bull, Jørgen (1971): *Rettslig beskyttelse av dataprogrammer*. Særavhandling ved juridisk fakultet, Universitetet i Oslo.
- «Business Model – Trolltech» <http://trolltech.com/company/model/> (27. november 2007)
- «Computer-Implemented Inventions» <http://www.epo.org/topics/issues/computer-implemented-inventions.html> (10. desember 2007)
- «Computer programming language :: C -- Britannica Online Encyclopedia» <http://www.britannica.com/eb/article-248112/computer-programming-language> (11. februar 2008)
- «Computer programming language :: FORTRAN -- Britannica Online Encyclopedia» <http://www.britannica.com/eb/article-248109/computer-programming-language> (11. februar 2008)
- «C Programming Language History» http://www.livinginternet.com/i/iw_unix_c.htm (11. februar 2008)
- Cordes, Jørg og Laila Stenseng (1999) *Hovedlinjer i internasjonal privatrett* Oslo: Cappelen Akademiske Forlag
- DiBona, Chris, Sam Ockman og Mark Stone (1999) *Introduction*. I: «Open Sources: Voices from the Open Source Revolution» s. 19-30 Sebastopol, CA: O'Reilly & Associates, Inc.
- Fogel, Karl (2006): *Producing Open Source Software*. Sebastopol, CA: O'Reilly Media, Inc.
- Fuglesang, Andreas Frølich og Krog, Georg Philip (1999): «Internett og juridiksjon». Oslo: Institutt for rettsinformatikk (CompLex nr 3/99)
- «Frequently Asked Questions about the GNU GPL - GNU Project - Free Software Foundation (FSF)» <http://www.gnu.org/licenses/gpl-faq.html> (8. november 2007)
- Gaarder, Karsten (1968): «Erstatningsregler i norsk immaterial- og konkurranserett» I: *Nordiskt immateriellt rättsskydd*, 1968 s. 245-260

- Grad, Burton og Luanne Johnson (2002): «The Start of the Software Products Industry». I: *IEEE Annals of the History of Computing*, vol. 24, no. 1, s. 3-4, Jan-Mar, 2002
- Greenleaf, Graham (2007): «Commons-ism in One Country? – National and International Dimensions of Copyright's Public Domain» http://www2.austlii.edu.au/~graham/publications/2007/National_dimensions.html (22. november 2007)
- Hagstrøm, Viggo (1996): «Om grensene for ansvarsfraskrivelse, særlig i næringsforhold» I: *Tidsskrift for rettsvitenskap* 1996 s 421-18
- Hagstrøm, Viggo og Magnus Aarbakke (2004): *Obligasjonsrett*. Oslo: Universitetsforlaget
- Hoeren, T (2004): «The first-ever ruling on the legal validity of the GPL – A Critique of the case» http://www.oii.ox.ac.uk/resources/feedback/OIIFB_GPL3_20040903.pdf (5. januar 2008)
- Höppner, Julian P (2004): «The GPL prevails: An analysis of the first-ever Court decision on the validity and effectivity of the GPL» <http://www.law.ed.ac.uk/ahrc/script-ed/issue4/GPL-case.pdf> (5. januar 2008)
- «IBM Antitrust Suit Records» <http://www.hagley.lib.de.us/1980.htm> (31. mars 2008)
- «IBM Archives: 1969» http://www-03.ibm.com/ibm/history/history/year_1969.html (21. mars 2008)
- Jaeger, Till og Axel Metzger (2006) *Open Source Software Rechtliche Rahmenbedingungen der Freien Software*. München: Verlag C. H. Beck
- Kielland, Torger (2005): *Copyleft*. Oslo: Institutt for informatikk (Complex 7/05)
- Kinander, Morten (2003): «Sondringen mellom tinglige og obligatoriske rettigheter» I: *Tidsskrift for rettsvitenskap*, s. 673-685
- Krüger, Kai (1999): *Norsk kjøpsrett* 4. utgave Bergen: Alma Mater Forlag
- Knoph, Ragnar (1963): *Åndsretten* Oslo: Nasjonaltrykkeriet.
- Koktvedgaard, Morgens og Jens Schovsbo (2005): *Lærebog i Immaterialret*. 7. utgave København: Jurist- og Økonomforbundets Forlag.
- Lassen, Birger Stuevold (1961): «Copyrightmerket» I: *Tidsskrift for rettsvitenskap*, 1961 s. 36-62 Oslo: Universitetsforlaget

- Lassen, Birger Stuevold m. fl. (1969): «Elektronisk databehandling: Immaterialrettslige aspekter». I: *Nordiskt immateriellt rättsskydd*, 1969 s. 18-53
- Lassen, Birger Stuevold (1981): «Notat oktober 1981 av professor Birger Stuevold Lassen om tredjemanns opphavsrett som mangel i kjøpsloven». I: *Ot. prp. nr. 80 (1986-1987) om kjøpsloven*. Vedlegg 7
- Lassen, Birger Stuevold (1983): «Sameie i opphavsrett og i opphavsrettslige «naboretter»». I: *Tidsskrift for rettsvitenskap*, 1983 s. 324
Oslo: Universitetsforlaget
- Lassen, Birger Stuevold (1997): «Norske domstolars kompetanse ved opphavsrettsinngrep utenfor norsk territorium» I: *Festskrift till Stig Strömholm Del II* Uppsala: Iustus Förlag
- «Licenses by Name» <http://www.opensource.org/licenses/alphabetical>
(11. februar 2008)
- Linksvayer, Mike «The Choice of a GNU Generation An Interview With Linus Torvalds» <http://gondwanaland.com/meta/history/interview.html>
(10. desember 2008)
- Lødrup, Peter (1999): *Lærebok i erstatningsrett*. 4. utgave.
Oslo: Peter Lødrup
- McKusick(1999): «Twenty Years of Berkeley UNIX». I: *Open Sources: Voices from the Open Source Revolution* s. 31-46 Sebastopol, CA: O'Reilly & Associates, Inc.
- Merges, Robert P., Peter S. Menell, Mark A. Lemly (2006) *Intellectual Property in The New Technological Age*. 4. utgave.
New York: Aspen Publishers
- «Measuring Libre Software Using Debian 3.1 (Sarge) as A Case Study: Preliminary Results» I: *European Journal for the Informatics Professional*, Novática, 2005 (Vol. VI, nr. 3)
- Michaelsen, Terje Lundby (2006): «Opphavsrettsavtaler. Med særlig sikte på opphavsrettskrenkelser og kontraktsbrudd» <http://wo.uio.no/as/WebObjects/theses.woa/wa/these?WORKID=38402> (29. februar 2008)
- Moggridge, Bill (2007): *Designing Interactions*. Cambridge, MA: The MIT Press
- «MySQL Contributor License Agreement» <http://forge.mysql.com/contribute/cla.php> (9. november 2007)

- «Netscape announces plans to make next-generation communicator source code available free on the net» <http://wp.netscape.com/newsref/pr/newsrelease558.html> (27. nov 2007)
- Nimmer, Raymond T. (2006): «Coexisting with Free and Open Source Software». I: *Computer Law Review International*, 2006 (Issue 5)
- Nimmer, Raymond T. (2006): «Breaking the Barriers: The Relation Between Contract And Intellectual Property Law». I: *The Berkeley Technology Law Journal* Number 13:3 (Fall 1998)
- «Ninth Circuit Model Civil Jury Instructions» <http://www.ca9.uscourts.gov/web/sdocuments.nsf/civ?OpenView&Start=1&Count=250&Expand=19#19> (12. februar 2008)
- «The Open Source Definition» <http://www.opensource.org/docs/osd> (11. februar 2008)
- Percival, Colin (2003): «Naive differences of executable code», <http://www.daemonology.net/bsdif/> (13. april 2008)
- Raymond, Eric Steven (1999): *A brief History of Hackerdom*. I: «Open Sources: Voices from the Open Source Revolution» s. 19-30 Sebastopol, CA: O'Reilly & Associates, Inc.
- Raymond, Eric Steven (2000): *Homesteading the Noosphere* <http://www.catb.org/~esr/writings/cathedral-bazaar/homesteading/homesteading.ps> (23. nov 2007)
- Raymond, Eric Steven (2000): *The Cathedral and the Bazaar* <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/cathedral-bazaar.ps> (23. nov 2007)
- Raymond, Eric Steven (2005): *Open Minds, Open Source* <http://www.catb.org/~esr/writings/analog.html> (26. nov 2007)
- Rdir. 91/205: Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs.
- Riis, Thomas (2001): *Immaterialret & IT*, København: Jurist- og Økonomiforbundets Forlag
- Rognstad, Ole-Andreas (1999): *Spredning av verkseksemplar. Om konsumpsjon av rettigheter i eksemplar av vernede verk*. Oslo: Cappelen Akademisk Forlag

- Rognstad, Ole-Andreas (2001): «Distribusjon av opphavsrettsbeskyttet materiale i digital form». Foiler til foredrag på Nordisk konferanse i rettsinformatikk 2001. http://www.jus.uio.no/iri/om_iri/seminarer/nord_konf2001/rognstad.ppt (10. januar 2008)
- Rognstad, Ole-Andreas og Are Stenvik (2002): «Hva er immaterialretten verd?». I *Bonus Pater Familias: Festskrift til Peter Lødrup* s. 511-548 Oslo: Gyldendal Norsk Forlag
- Rognstad, Ole-Andreas (2004): *Fragmenter til en lærebok i opphavsrett*. Oslo: Institutt for privatrett. (Institutt for privatretts stensilserie; 165)
- Rognstad, Ole-Andreas (2007): «Rettsvalget i opphavsrettslige tvister» I: *Rett og toleranse. Festskrift til Helge Johan Thue* s. 411-422 Oslo: Gyldendal Norsk Forlag
- Rosen, Lawrence (2002): «Why the Public Domain Isn't a License» <http://www.linuxjournal.com/article/6225> (22. november 2007)
- Rosen, Lawrence (2002): «Allocation of the Risk» <http://www.linuxjournal.com/article/6155> (24. januar 2008)
- Samuels, Edward (2000): *The illustrated story of copyright*. New York: Tomas Dunne Books St. Martin's Press
- Samuelson, Pamela, Randall Davis, Mitchell D. Kapor og J. H. Reichman (1994) «A manifesto concerning the legal protection of computer programs» i *Columbia Law Review* Vol. 94 No. 8
- Samuelson, Pamela (1999): «Challenges for the World Intellectual Property Organization» i *European Intellectual Property Review* Gloucester: Sweet & Maxwell Ltd. s. 578-591
- St. Laurent, Andrew M. (2004): *Open Source & Free Software Licensing* Sebastopol, CA: O'Reilly Media. Inc.
- Stallman, Richard (1983): «New UNIX implementation» <http://groups.google.com/group/net.unix-wizards/msg/4dadd63a976019d7?dmode=sourc> (26. november 2007)
- Stallman, Richard (1992): «Why Software Should Be Free» <http://www.gnu.org/philosophy/shouldbefree.html> (11. februar 2008)
- Stallman, Richard (2002): *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Boston: Free Software Foundation
- Sterling, J.A.L. (2003): *World copyright law*. London: Sweet & Maxwell Ltd.

- Stoltz, Mitchell L. (2006) «The Penguin Paradox: How the Scope of Derivative Works in Copyright Affects the Effectiveness of the GNU GPL». I: *Boston University Law Review* Vol. 85 s. 1439 -1477
- Stray, Anne Lise Sijthoff Stray (1989): *Opphavsretten*
Oslo: Universitetsforlaget
- Svantesson, Dan Jerker B (2007): «Private International Law and the Internet» Alphen aan den Rijn: Kluwer Law International
- Tiemann, Michael (2006): «History of the OSI» <http://www.opensource.org/history> (27. november 2007)
- Torvalds, Linus (1991) «What would you like to see most in minix?» <http://groups.google.com/group/comp.os.minix/msg/b813d52cbc5a044b?dmode=source> (26. november 2007)
- Torvund, Olav (1999): «Opphavsrett i oppdragsforhold» I: *Nordiskt immateriellt rättsskydd*, 1999 s. 123-129
- Torvund, Olav (1997): *Kontraktsregulering -IT kontrakter*
Oslo: Tano Aschehoug
- Torvund, Olav (1997) «Opphavsrett til dataprogrammer utviklet i vertikalt samarbeid». I: *Ånd og rett: festskrift til Birger Stuevold Lassen*
Oslo: Universitetsforlaget
- Udsen, Henrik «Uagsomhet som aftalestiftende retsfaktum – Et bidrag til den aftaleretlige forpliktelseslære». I: *Tidsskrift for Rettsvitenskap*, 2006, nr. 1, s. 104 – 151
- Välimäki, Mikko (2005): *The Rice of Open Source Licensing*.
Helsinki: Turre Publishing
- «Vedtak 20. mars 1987 av Nemda til behandling av tvister vedrørende videresending av kringkastingsprogram». I: *Nordiskt immateriellt rättsskydd*, 1989, s. 369 - 373
- «Venezuela: Government migration to Open Source makes experts scarce» Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens <http://ec.europa.eu/idabc/en/document/6537/5934> (20. januar 2008)
- Vigen, Bernhard (2001): *Arbeidstakerens opphavsrett*.
Oslo: Universitetsforlaget

- Vinje, Thomas C. (1992): «The EC Software Directive and the Question of Interoperability». I: *Nordisk Tidskrift för immaterialrätt ovh angränsade ämnesområden*, 1992 s. 74-91
- Vyrje, Magnus Stray (1985): *Vanhjemmel, opphavsrett og datamaskinprogrammer*. Oslo: Universitetsforlaget (Complex 12/85)
- Wagle, Andreas og Ødegaard jr, Magnus (1997): *Opphavsretten i en digital verden* Oslo: Cappelen
- Williams, Sam (2002): «Free as in Freedom» <http://www.oreilly.com/openbook/freedom/> (11. februar 2008)
- What is Mac OS X?* <http://www.kernelthread.com/mac/osx/arch.html> (12. november 2007)

12 DOMMER

Høyesterett

- Rt. 2005 s. 41 «Napster»
- Rt. 1997 s.199 «Cirrus»
- Rt. 1993 s. 1583 «SACIS»
- Rt. 1992 s. 620
- Rt. 1980 s. 243 «Tampax»
- Rt. 1970 s. 1235
- Rt. 1970 s. 1172 «Diskotekdommen»
- Rt. 1953 s. 633 «Bedriftsmusikk»
- Rt. 1948 s. 370
- Rt. 1940 s. 48
- Rt. 1932 II s. 59 «Irma-Mignon»

Lagmannsretten

- RG 1999 s. 330 «MOCS»
- RG 1997 s. 390
- RG 1997 s. 338
- Gulating lagmannsretts dom av 14 november 1988 Studio 46 Reklamebyrå v/Egil Korsnes mot A/S Papir (Nordiskt immateriellt rättsskydd, 1991 s. 413-416)

Tingretten

- RG 2000 s. 1516 «Markus Data»
- Oslo byrett 6. desember 1996 Eik Computing AS vs Norsk AS Philips, Moberg Holding AS og EM Data AS
- Oslo byrett 1992, «Alcatel STK» (Lov & Data nr. 34, mars 1993, s. 2-3)

- Oslo byrett 28. september 1982, «Alle Menn» (Nordiskt immateriellt rätts-skydd, 1983 s. 138-144)
- RG 1982 s. 932
- RG 1967 s. 158

Utenlandske dommer

- Landgericht Frankfurt am Main (Tyskland) Saknr: 2-6 0 224/06 «D-link»
- Landgericht München I (Tyskland) Saknr: 21 O 6123/04 «netfilter/iptables»
- Vestre Landsret (Danmark) den 27. juni 1978, «Den lille havfruen» (Nordiskt immateriellt rättsskydd, 1982 s 105-111)

13 FORARBEIDER

Innstillinger til Odelstinget

Innst. O. XI (1960-61): *Om lov om opphavsrett til åndsverk*

Odelstingsproposisjoner

Ot .prp. nr. 44 (2001-2002): *Om lov om forbrukerkjøp
(forbrukerkjøpsloven)*

Ot. prp. nr.15 (1994-1995): *Om lov om endringer i åndsverkloven m.m*

Ot. prp. nr. 84 (1991-1992): *Om lov om endringer i lov 12 mai 1961 nr 2
om opphavsrett til åndsverk m.v. og i enkelte andre lover som følge av
EØS-avtalen*

Ot. prp. nr. 33 (1989-1990): *Om lov om endringer i åndsverkloven
(opphavsrett og edb)*

Ot. prp. nr. 26 (1959-1960): *Om lov om opphavsrett til åndsverk*

Ot. prp. nr. 5 (1982-1983): *Avtaleloven 31 mai 1918 nr 4, m.m. (generell
formuerettslig lempingsregel)*

Norske offentlige utredninger

NOU 2001:32B: *Rett på sak Lov om tvisteløsning (tvisteloven)*

NOU 1979:32: *Formuerettslig lempningsregel*

NOU 1986:18: *Opphavsrett og edb. Delutredning III*

14 FORKORTELSER

avtl.	Lov om avslutning av avtaler, om fuldmagt og om ugyldige viljeserklæringer av 31. mai 1918 nr. 4
ekstl.	Lov om godtroerverv av løsøre av 2. juni 1978 nr. 37
EPC	European Patent Convention
kjl.	Lov om kjøp av 13. mai 1988 nr. 27
kretsm.	Lov om vern av kretsmønstre for integrerte kretser av 15. juni 1990 nr. 27
NL	Kong Christian Den Femtis Norske Lov 14. april 1688
patl.	Lov om patenter av 15. desember 1967 nr. 9
U.S.C.	United States Code
åvl.	Lov om opphavsrett til åndsverk m.v. av 12 mai. 1961 nr. 2

TIDLIGERE UTGITT I COMPLEX-SERIEN

CompLex er Senter for rettsinformatikk skriftserie. Serien startet i 1981, og det har blitt utgitt mer enn hundre titler. Bøkene i CompLex-serien kan bestilles fra Akademika (se bestillingsskjema bak i boken eller www.akademika.no).

2008

- 8/08 Rett og rimelighet i moralsk belysning – og andre grunnproblemer i norsk rettsliv
Jens Petter Berg..... NOK 657,-
- 7/08 Vern av tekniske beskyttelsessystemer etter åndsverksloven §53A
Andreas Norum NOK 156,-
- 6/08 Grunnloven § 100 (4) som hinder for bruk av midlertidige forføyninger mot ytringer – med spesielt fokus på forestående, antatte opphavsrettskrenkelser
Frederik Langeland..... NOK 135,-
- 5/08 Telekirurgi i et rettslig perspektiv – med spesiell vekt på etikk, samtykke og ansvar
Bjørn Ivar Christie Østberg NOK 201,-
- 4/08 IT-støtte for arbeid med lovsaker
Dag Wiese Schartum..... NOK 144,-
- 3/08 Juristopia: Semantic Wiki for Legal Information
Ole Christian Rynning..... NOK 243,-
- 2/08 Electronic Contracting in Europe. Benchmarking of national contract rules of United Kingdom, Germany, Italy and Norway in light of the EU E-commerce Directive
Maryke Silalahi Nuth..... NOK 192,-
- 1/08 Internet search engines' collecting and processing of web pages – from the perspective of copyright law
Ingvild Jørgensen NOK 165,-

2007

- 5/07 Gjennomgang av arkivretten
Martin Rødland NOK 129,-
- 4/07 Privacy & Identity Management
Thomas Olsen, Tobias Mahler, et al. NOK 234,-
- 3/07 Personvern og transportsikkerhet
Dag Wiese Schartum NOK 306,-
- 2/07 ZEBLEX 06 - Tre avhandlinger om fildeling, IT-sikkerhet og e-postkontroll
*Ida Otterstad, René Stub-Christiansen
& Cecilie Wille Søvik* NOK 348,-
- 1/07 Kontraktsregulering av domstolens kompetanse ved elektronisk handel
Vebjørn Krag Iversen NOK 186,-

2006

- 6/06 Lover - fra kunngjøring til hyperstrukturer: to avhandlinger
Per Marius Slagsvold & Kirill Miazine NOK 222,-
- 5/06 Retten til eget bilde
Maria Jongers NOK 198,-
- 4/06 Legal, Privacy, and Security Issues
in Information Technology Vol. 2
Kierkegaard, Sylvia Mercado (editor) NOK 783,-
- 3/06 Legal, Privacy, and Security Issues
in Information Technology Vol. 1
Kierkegaard, Sylvia Mercado (editor) NOK 918,-
- 2/06 Rettslige reguleringer av informasjonssikkerhet
Are Vegard Haug NOK 420,-
- 1/06 Anti-spam Legislation Between Privacy And
Commercial Interest. An overview of the European
Union legislation regarding the e-mail spam
Dana Irina Cojocarasu NOK 155,-

2005

- 1/05 **Renessansen som unnfanget Corpus Iuris Civilis.
Keiser Justinians gjenerobring av Romerriket**
Halvor Manshaas..... NOK 249,-
- 2/05 **Personvern og ytringsfrihet. Fotografering av siktede
i straffesaker – et vern for ytringsfrihet?**
Anette Engum..... NOK 132,-
- 3/05 **Rettigheter til geografisk informasjon.
Opphavsrett, databasevern og avtalepraksis.**
Steinar Taubøll..... NOK 206,-
- 4/05 **«The Answer to the Machine is in the Machine» and Other Collected
Writings**
Charles Clark..... NOK 401,-
- 5/05 **Digital Rights Management - Promises, Problems and Alternative
Solutions**
Kristian Syversen..... NOK 201,-
- 6/05 **DRM og Demokrati. Argumentasjoner, rettferdiggjøringer og
strategier bak endringen av åndsverksloven 2003-2005**
Jan Frode Haugseth NOK 224,-

2004

- 1/04 **Opphavsrettslige problemstillinger ved universitetene og høyskolene.
Innstilling fra immaterialrettsutvalget, oppnevnt av Universitets-
og Høgskolerådet 31. januar 2000. Avgitt til universitets- og
høgskolerådet 8. oktober 2003**
Immaterialrettsutvalget..... NOK 341,-
- 2/04 **Ansvarsfrihet for formidler ved formidling av
informasjonssamfunnstjenester**
Bård Standal..... NOK 311,-
- 3/04 **Arbeidsgivers adgang til å kontrollere og overvåke sine ansatte
med hovedvekt på grunnvilkårene for behandling av
personopplysninger i arbeidslivet**
Stefan Jørstad..... NOK 191,-

140 Åpen programvare - noen rettslige problemstillinger

.....

- 4/04 Elektroniske spor fra mobiltelefoner – om politiets bruk og
teleoperatørenes lagring av trafikkdata.
Christian Dahlgren NOK 117,-
- 5/04 International Jurisdiction and Consumers Contracts – Section 4
of the Brussels Jurisdiction Regulation.
Joakim S. T. Øren NOK 172,50
- 6/04 Elektronisk dokumentfalsk.
Lars Christian Sunde..... NOK 60,502003

2003

- 1/03 IT i domstolene. En analyse av norske domstolars
teknologianvendelse fra 1970 til 2001
Even Nerskogen..... NOK 330,-
- 2/03 Hvorfor vokser Norsk Lovtidend? En empirisk analyse av veksten
Martin Støren..... NOK 87,-
- 3/03 Etableringslandsprinsippet. En analyse av e-handelsdirektivet art 3 og
prinsippet om fri bevegelighet av tjenester ved elektronisk handel
Jon Christian Thaulow..... NOK 213,-
- 4/03 The Law of Electronic Agents. Legal contributions to ALFEBIITE –
A Logical Framework for Ethical Behaviour between Infohabitants
in the Information Trading Economy of the Universal Information
Ecosystem, IST-1999-10298
Jon Bing and Giovanni Sartor (eds) NOK 351,-
- 5/03 LEA 2003: The Law and Electronic Agents Proceedings of the
Second LEA Workshop, 24th June 2003, in connection with the
ICAIL 2003 Conference (Ninth International Conference on Artificial
Intelligence and Law), Edinburgh, Scotland, UK
Seminarrapport)..... NOK 228,-
- 6/03 Opphavsrettslige aspekter ved nettbasert formidling av musikk
Stig Walle NOK 153,-
- 7/03 Sceneinstruktørens opphavsrettslige stilling
Edle Endresen NOK 119,-

- 8/03 User-Centred Privacy Aspects In Connection With Location Based Services
Christian B. Hauknes..... NOK 203,-

2002

- 1/02 Koblingshandel og forholdet til fysisk og teknologisk integrasjon i relasjon til EØS-avtalens art.54(d)
Ole Jacob Garder..... NOK 180,-
- 2/02 To opphavsrettslige arbeider:
 Bjarte Aambø – Opphavsrettslige rettsmangler
 Erlend Ringnes Efskind – Skjermbildets rettslige natur
Aambø / Ringnes Efskind..... NOK 201,-
- 3/02 Arbeidstakeroppfinnelser ved universiteter og høyskoler. Innstilling fra et utvalg oppnevnt av universitets- og høyskolerådet 31 januar 2000. Avgitt til universitets- og høyskolerådet i oktober 2001
 NOK 213,-
- 4/02 Utøvende kunstners direkteoverføringer på Internett – med hovedvekt på kringkastingsbegrepet
Irina Eidsvold Tøien NOK 225,-
- 5/02 Administrasjon av radiofrekvensspekteret. Rettslige problemstillinger knyttet til telemyndighetenes forvaltning av frekvensressursene
Øyvind Haugen NOK 177,-
- 6/02 Overføring av personopplysninger til tredjeland. Kravet til tilstrekkelig beskyttelse etter EU-direktivet om personvern art. 25
Mona Naomi Lintvedt og Christopher J. Helgeby..... NOK 198,-
- 7/02 Digitale mellomledds ansvar for videreformidling av yringer. E-handelsdirektivet art. 12-14
Just Balstad..... NOK 186,-
- 8/02 Platekontrakten. Eksklusive overdragelser av utøverens rettigheter til eksemplarframstilling og spredning
Øyvind Berge..... NOK 237,-
- 9/02 Varemerkerettslige konflikter under .no. I hvilken grad kan registrering og bruk av et domenenavn medføre inngrep i en

varemerkerett? Hvordan løses konflikter under .no i dag, og hva kan være en mer hensiktsmessig tvisteløsningsmekanisme i fremtiden?

Silje Johannessen..... NOK 192,-

- 10/02 Vegard Hagen – Pekeransvar. Spørsmålet om ansvar for publisering av pekere på verdensveven (World Wide Web)
Hans Marius Graasvold – Pekeransvaret. Straffe- og erstatningsansvar for publisering av pekere til informasjon på Internett

Vegard Hagen / Martin Grasvold..... NOK 234,-

- 11/02 Personopplysningsloven § 7. En analyse av forholdet mellom personvern og ytringsfrihet slik det er uttrykt i personopplysningsloven § 7

Karen Elise Haug Aronsen..... NOK 198,-

- 12/02 Databasevern. Sui generis-vern av sammenstillinger etter gjennomføringen av databasedirektivet i åndsverkloven § 43

Lisa Vogt Lorentzen..... NOK 210,-

2001

- 1/01 Internet and Choice-of-Law – The International Sale of Digitised Products through the Internet in a European Context

Peter Lenda..... NOK 275,-

- 2/01 Internet Domain Names and Trademarks

Tonje Røste Gulliksen..... NOK 227,-

- 3/01 Internasjonal jurisdiksjon ved elektronisk handel – med Luganokonvensjonen art 5 (5) og elektroniske agenter som eksempel

Joakim S. T. Øren..... NOK 204,-

- 4/01 Legal issues of maritime virtual organisations

Emily M. Weitzenböck..... NOK 164,-

- 5/01 Cyberspace jurisdiction in the U.S. – The International Dimension of Due Process

Henrik Spang-Hanssen..... NOK 685,-

- 6/01 Norwegian border control in a Europe without Internal Frontiers
– Implications for Data Protection and civil liberties
Stephen Kabera Karanja..... NOK 252,-

2000

- 1/00 Klassikervernet i norsk åndsrett
Anne Beth Lange NOK 268.-
- 2/00 Adgangen til å benytte personopplysninger. Med vekt på det opprinnelige behandlingsformålet som begrensningsfaktor
Claude A. Lenth..... NOK 248.-
- 3/00 Innsyn i personopplysninger i elektroniske markedsplasser.
Line Coll..... NOK 148.-

1999

- 1/99 International regulation and protection of Internet domain and trademarks
Tonje Røste Gulliksen..... NOK 248.-
- 2/99 Betaling via Internett
Camilla Julie Wollan..... NOK 268.-
- 3/99 Internett og jurisdiksjon
Andreas Frølich Fuglesang & Georg Philip Krog..... NOK 198.-

1998

- 1/98 Fotografiske verk og fotografiske bilder, åndsverkloven § 1 og § 43 a
Johan Krabbe-Knudsen NOK 198.-
- 2/98 Straffbar hacking, straffelovens § 145 annet ledd
Guru Wanda Wanvik NOK 238.-
- 3/98 Interconnection – the obligation to interconnect telecommunications networks under EC law
Katinka Mahieu NOK 198.-

1997

- 1/97 Eksemplarframstilling av litterære verk til privat bruk
Therese Steen NOK 158.-
- 2/97 Offentlige anskaffelser av informasjonsteknologi
Camilla Sivesind Tokvam NOK 175.-
- 3/97 Rettslige spørsmål knyttet til Oppgaveregisteret
Eiliv Berge Madsen NOK 170.-
- 4/97 Private pengespill på Internett
Halvor Manshaus NOK 160.-
- 5/97 Normative Structures in Natural and Artificial Systems
Christen Krogh NOK 255.-
- 6/97 Rettslige aspekter ved digital kringkasting
Jon Bing NOK 178.-
- 7/97 Elektronisk informasjonsansvar
Tomas Myrbostad NOK 148.-
- 8/97 Avtalelisens etter åndsverksloven § 36
Ingrid Mauritzen NOK 120.-
- 9/97 Krav til systemer for forvaltning av immaterielle rettigheter
Svein Engebretsen NOK 168.-
- 10/97 American Telephony: 120 Years on the Road to Full-blown
Competition
Jason A. Hoida NOK 140.-
- 11/97 Rettslig vern av databaser
Harald Chr Bjelke NOK 358.-

1996

- 1/96 Innsynsrett i elektronisk post i offentlig forvaltning
Knut Magnar Aanestad og Tormod S. Johansen NOK 218.-
- 2/96 Public Policy and Legal regulation of the Information Market in the
Digital Network Environment
Stephen John Saxby NOK 238.-

- 3/96 **Opplysning på spill**
Ellen Lange..... NOK 218.-
- 4/96 **Personvern og overføring av personopplysninger til utlandet**
Eva I. E. Jarbekk NOK 198.-
- 5/96 **Fjernarbeid**
Henning Jakhell NOK 235.-
- 6/96 **A Legal Advisory System Concerning Electronic Data Interchange within the European Community**
Andreas Mitakas..... NOK 128.-
- 7/96 **Elektronisk publisering: Utvalgte rettslige aspekter**
Jon Bing og Ole E. Tokvam NOK 186.-
- 8/96 **Fjernsynsovervåking og personvern**
Finn-Øyvind H. Langfjell..... NOK 138.-

1995

- 1/95 **Rettslige konsekvenser av digitalisering: Rettighetsadministrasjon og redaktøransvar i digitale nett**
Jon Bing..... NOK 368.-
- 2/95 **Rettslige spørsmål i forbindelse med utvikling og bruk av standarder innen telekommunikasjon**
Sverre Sandvik NOK 178.-
- 3/95 **Legal Expert Systems: Discussion of Theoretical Assumptions**
Tina Smith NOK 278.-
- 4/95 **Personvern og straffeansvar – straffelovens § 390**
Ole Tokvam..... NOK 198.-
- 5/95 **Juridisk utredning om filmen «To mistenkelige personer»**
Johs. Andenæs NOK 138.-
- 6/95 **Public Administration and Information Technology**
Jon Bing and Dag Wiese Schartum NOK 348.-
- 7/95 **Law and Liberty in the Computer Age**
Vittorio Frosini NOK 158.-

1994

- 1/94 Deon'94, Second International Workshop on Deontic Logic in
Computer Science
Andrew J. I. Jones & Mark Sergot (ed)..... NOK 358.-
- 2/94 Film og videogramrett. TERESA (60)
Beate Jacobsen..... NOK 318.-
- 3/94 Elektronisk datutveksling i tollforvaltningen – Rettslige spørsmål
knyttet til TVINN
Rolf Risnæs..... NOK 225.-
- 4/94 Sykepenges og personvern – Noen problemstillinger knyttet til
behandlingen av sykepenges i Infotrygd
Mari Bø Haugestad..... NOK 148.-
- 5/94 EØS, medier og offentlighet. TERESA (103)
Mads Andenæs, Rolf Høyer og Nils Risvand NOK 148.-
- 6/94 Offentlige informasjonstjenester: Rettslige aspekter
Jon Bing..... NOK 148.-
- 7/94 Sattelittfjernsyn og norsk rett. MERETE (3) IV
Nils Eivind Risvand NOK 138.-
- 8/94 Videogram på forespørsel. MERETE (14) IV
Beate Jacobsen (red)..... NOK 158.-
- 9/94 «Reverse engineering» av datamaskinprogrammer. TERESA (92) IV
Bjørn Bjerke..... NOK 198.-
- 10/94 Skattemessig behandling av utgifter til anskaffelse av
datamaskinprogrammer. TERESA (75)
Gjert Melsom..... NOK 198.-

1993

- 1/93 Artificial Intelligence and Law. Legal Philosophy and Legal Theory
Giovanni Sartor NOK 148.-
- 2/93 Erstatningsansvar for informasjonstjenester, særlig ved databaseydelses
Connie Smidt NOK 138.-

- 3/93 Personvern i digitale telenett
Ingvild Hanssen-Bauer..... NOK 178.-
- 4/93 Consumers Purchases through Telecommunications in Europe. –
Application of private international law to cross-border contractual
disputes
Joachim Benno..... NOK 198.-
- 5/93 Four essays on: Computers and Information Technology Law
Morten S. Hagedal..... NOK 218.-
- 6/93 Sendetidsfordeling i nærradio MERETE (3) III
Marianne Rytter Evensen..... NOK 148.-
- 7/93 Essays on Law and Artificial Intelligence
Richard Susskind NOK 158.-

1992

- 1/92 Avskriving av mikrodatamaskiner med tilbehør – en nordisk studie
TERESA (87)
Beate Hesseltvedt..... NOK 138.-
- 2/92 Kringkastingsbegrepet TERESA (78)
Nils Kr. Einstabland..... NOK 208.-
- 3/92 Rettskilderegistre i Helsedirektoratet NORIS (94) I & II
Maria Strøm..... NOK 228.-
- 4/92 Softwarepatent – Imaterialrettens enfant terrible. En redegjørelse
for patenteringen af softwarerelaterede opfindelser i amerikansk og
europæisk ret
Ditlev Schwanenfügel..... NOK 158.-
- 5/92 Abonnementskontrakter fro kabelfjernsyn TERESA (78II)
Lars Borchgrevink Grindal..... NOK 248.-
- 6/92 Implementing EDI – a proposal for regulatory form
Rolf Riisnæs..... NOK 118.-
- 7/92 Deponering av kildekode«escrow»-klausuler TERESA (79)
Morten S. Hagedal..... NOK 128.-

- 8/92 EDB i juridisk undervisning – med en reiserapport fra England og
USA
Ola-Kristian Hoff NOK 228.-
- 9/92 Universiteters ansvar for bruk av datanett TERESA (94)
Jon Bing & Dag Elgesem NOK 198.-
- 10/92 Rettslige sider ved teletorg
Andreas Galtung NOK 148.-

BESTILLING

Jeg bestiller herved følgende Complex-utgivelser:

Nummer / årgang: _____

Tittel: _____

Nummer / årgang: _____

Tittel: _____

Nummer / årgang: _____

Tittel: _____

Nummer / årgang: _____

Tittel: _____

Nummer / årgang: _____

Tittel: _____

Navn: _____

Adresse: _____

Postadresse: _____

Telefon: _____

Bestillingsskjemaet sendes pr.post eller telefaks til:

akademika

Avd. juridisk litteratur Aulabygningen

Karl Johansgt. 47, 0162 Oslo

Telefon: 22 42 54 50

Telefaks: 22 41 17 08

Complex kan også bestilles via nettbokhandelen www.akademika.no

