

UiO : **Det juridiske fakultet**

# Overtredelse av copyleft- klausulen

Opphavsrettslige og kontraktsrettslige sanksjoner

Kandidatnummer: 715

Leveringsfrist: 25.april 2018

Antall ord: 17 388



# Innholdsfortegnelse

<b>1</b>	<b>INNLEDNING.....</b>	<b>3</b>
1.1	Tema og problemstillinger .....	3
1.2	Avgrensning og presisering av problemstillingen .....	4
1.3	Rettskildebildet .....	5
1.3.1	Lover og forarbeider .....	5
1.3.2	Rettspraksis.....	6
1.3.3	Internasjonale kilder .....	6
<b>2</b>	<b>INTRODUKSJON TIL OPEN SOURCE SOFTWARE.....</b>	<b>8</b>
2.1	Dataprogrammer og kildekoden.....	8
2.1.1	Hva er et dataprogram?.....	8
2.1.2	Åpen kildekode og proprietær kildekode .....	9
2.2	Historie og filosofi .....	9
2.2.1	Ideene bak åpen kildekodebevegelsen.....	9
2.2.2	Open source software: betydning og påvirkning i IT-bransjen .....	11
2.2.3	Hva kjennetegner et åpen kildekodeprogram? .....	12
<b>3</b>	<b>OPPHAVSRETTLIGVERN AV ÅPEN KILDEKODEPROGRAM .....</b>	<b>14</b>
3.1	Innledning .....	14
3.2	Vilkår for opphavsrettsligvern .....	15
3.2.1	Frembringelse .....	15
3.2.2	Vernet verkstype.....	16
3.2.3	Kravet til verkshøyde- originalitetskrav .....	17
3.3	Opphavsrettens subjekter .....	21
3.3.1	Ansettelsesforhold .....	21
3.3.2	Oppdragsforhold .....	21
3.3.3	Fellesverk.....	21
3.3.4	Sammensatte verk og samleverk .....	22
3.3.5	Bearbeidede verk .....	22
3.4	Opphaverens eksklusive rettigheter .....	23
<b>4</b>	<b>INTRODUKSJON TIL GNU GENERAL PUBLIC LICENSE VERSJON 3.....</b>	<b>25</b>
4.1	Innledning .....	25
4.2	Jurisdiksjon og lovvalg .....	27
4.2.1	Jurisdiksjon.....	27
4.2.2	Lovvalg .....	27
4.3	Kontraksrettslige aspekter ved GPL .....	29
4.3.1	Avtaleslutning.....	29
4.3.2	Avtaletolkning .....	30
4.4	GPL versjon 3: Tillatelser og vilkår.....	31
4.4.1	Rettslig grunnlag.....	31
4.4.2	Eksemplarforestillingsretten.....	32
4.4.3	Tilgjengeligjøringsretten .....	32
<b>5</b>	<b>NÆRMERE OM COPYLEFT-KLAUSULEN.....</b>	<b>34</b>

5.1	Innledning .....	34
5.2	Tolkning av klausulen: Bestemmelsenes innhold .....	34
5.2.1	”Convey” .....	35
5.2.2	”... A work based on the Program” .....	39
5.2.3	”The entire work, as a whole” .....	41
<b>6</b>	<b>SANKSJONER VED OVERTREDELSE AV COPYLEFT-KLAUSULEN.....</b>	<b>43</b>
6.1	Innledning .....	43
6.2	Opphavsrettslig og kontraktsrettslig virkende vilkår i copyleft-klausulen .....	43
6.2.1	Opphavsrettslig virkende vilkår.....	43
6.2.2	Kontraktsrettslig virkende vilkår .....	43
6.3	Opphavsrettslige og kontraktsrettslige sanksjoner.....	44
6.3.1	Erstatning .....	45
6.3.2	Naturaloppfyllelse .....	46
<b>7</b>	<b>AVSLUTNING.....</b>	<b>49</b>
	<b>LITTERATURLISTE.....</b>	<b>51</b>

# 1 Innledning

## 1.1 Tema og problemstillinger

I den stadig mer digitaliserte hverdagen finner en dataprogrammer i alt fra smarttelefoner, klokker og biler, til tekstbehandlingsverktøy. IT-bransjen har i de siste årene økt bruken av åpen kildekodeprogrammer enten direkte, eller ved å inkorporere disse inn i sin egen programutvikling. Et studie fra 2015 viser at 78% av bedriftene brukte åpen kildekodeprogrammer i sin virksomhet<sup>1</sup>. Gigantselskapet Google® bruker for eksempel en bearbejdet versjon av Linux for serverne sine<sup>2</sup>. Tilbydere av åpen kildekodeprogrammer som Red Hat®, har profittert mye på å bygge sin virksomhet på dette. Den økende bruken av slike programmer gjør det interessant å se på de juridiske problemstillingene som kan oppstå.

Dataprogrammer er vernet som åndsverk etter lov 12 mai 1961 nr. 2 om opphavsrett til åndsverk m.v. Dataprogrammer er ikke et typisk åndsverk da disse historisk sett har vært frembringelser på det kunstneriske eller litterære området. Verkshøydeproblematikken for tekniske frembringelser, samt rekkevidden av vern for dataprogrammer, vil være av interesse.

Dataprogrammer distribueres under lisensavtaler som gir lisenstaker tillatelser til å gjøre bruk av programmet under nærmere angitte vilkår. Dersom programmet er et åndsverk, vil opphaver få eksklusiv råderett over sitt program. Tredjeparter som ønsker å gjøre bruk av opphaverens eksklusive rettigheter, må få tillatelse av denne. Lisensavtalen gir slik tillatelse, og opphaver bestemmer rammene og vilkårene rundt bruken. Åpen kildekode-lisenser bygger på et tilsvarende system, men stiller en del særegne vilkår. Avhandlingen tar sikte på å undersøke GNU General Public License versjon 3, som inneholder en sterk copyleft-klausul. Lisensen inneholder begreper som avviker fra de tradisjonelle opphavsrettslige, men som er inspirert av disse. Dette byr på tolkningsproblemer hva gjelder begrepenes innhold og rekkevidde.

Ettersom åpen kildekodeprogrammer ikke har en lukket og begrenset distribusjonskanal, er det vanskelig å holde oversikt over hvem som har eksemplarer av programmet og hvorvidt

---

<sup>1</sup> North Bridge & Black Duck (2015)

<sup>2</sup> IBM (2012)

alle lisenstakerne overholder vilkårene i lisensen. Ved brudd på avtalevilkår vil opphaver kunne gi en rekke sanksjoner. Vilårene kan være hjemlet i opphavsretten eller i kontrakt. Hvorvidt vilårene er opphavsrettslig eller kontraktsrettslig virkende, bestemmer hvilke sanksjoner som kan gis. Sanksjonene er forankret i ulike hensyn og har dermed også ulike virkninger. Det er usikkert hvilke sanksjoner som har de beste grunner for seg hva gjelder overtredelse av åpen kildekode-lisenser, og da særlig copyleft-klausulen.

## **1.2 Avgrensning og presisering av problemstillingen**

Avhandlingen vil ta sikte på å gi en oversikt over de opphavsrettslige spørsmålene knyttet til dataprogrammer som verkstype. Fokuset vil ligge på verkshøydevurderingen da dette vil by på særlige utfordringer ettersom dataprogrammer har en ren funksjonell hensikt hvilket skiller seg fra de typiske åndsverkene loven ellers gjelder. Åpen kildekodeprosjekter utvikles sjeldent kun av en opphaver alene. Utviklere kan være ansatt i et selskap, eller være leid inn som konsulent. Ettersom åpen kildekodeprosjekter er tilgjengelig på Internettet og åpent for alle, kan det også være privatpersoner som kommer med bidrag. Bidragsyterne kan derfor bli mange, og det kan bli krevende å holde oversikt over hvem som er opphaver av de ulike verksdeler. Et problem som kan oppstå er dersom en bidragsyter legger til kode den selv ikke har skrevet. Her kan det oppstå vanhjemmel hvilket kan gi store konsekvenser. En tilknyttet problemstilling vil være dersom bidragsyter legger ved kode som er lisensiert under en lisens som er inkompatibel med GPL. Disse problemstillingene skal jeg i midlertid ikke behandle inngående da dette er et tema i seg selv.

GPL inneholder bestemmelser om patent og tekniske beskyttelsessperrer. Lisensen inneholder også klausuler om garanti og ansvarsfraskrivelse. Det kan diskuteres hvorvidt det i norsk rett er anledning til å fraskrive seg et hvert ansvar, og om lisenstaker kan lempe avtalen etter avtaleloven § 36. Disse problemstillingene faller utenfor oppgaven og vil ikke bli behandlet.

Avhandlingen fokuserer på de praktiske opphavsrettslige og kontraktsrettslige sanksjonene som kan komme til anvendelse ved overtredelse av copyleft-klausulen. Jeg vil foreta en sammenligning av sanksjonene, samt foreta en de lege ferenda-vurdering av hvordan overtredelse kan sanksjoneres med hensyn til lisensgiver og lisenstakers interesser. Ettersom lisensen inneholder en del uklarheter hva gjelder begrepsbruken, vil det være nødvendig å tolke copyleft-klausulen inngående. Dette vil være viktig da en ikke kan sanksjonere

handlinger som ikke utløser copyleft-klausulen. Grensen mellom overtredelse og hva som regnes som ”frie” handlinger vil behandles.

Det nevnes forøvrig at jeg kommer til å bruke betegnelsene opphaver og lisensgiver. Lisensgiver kan være den opprinnelige opphaver, eller andre personer som den opprinnelige opphaveren har overdratt sine rettigheter til. Lisensgiver vil i denne avhandlingen inneha begge betydninger. Når det er tale om opphaver, er det kun tale om den opprinnelige opphaveren i lovens forstand.

### **1.3 Rettskildebildet**

Åpen kildekode-problematikken er relativt lite behandlet i norsk rett, og det finnes få tungtveiende rettskilder som tar opp disse spørsmålene. Jeg har derfor måttet se til internasjonale og utenlandske rettskilder i drøftingen av de relevante problemstillingene, og da særlig hva gjelder tolkning av lisensen og sanksjonsmuligheter. Dette gjør at de mulige tolkningsutfallene og løsningene står mer åpent enn ellers.

#### **1.3.1 Lover og forarbeider**

Det relevante lovgrunnlaget er lov 12 mai 1961 nr. 2 om opphavsrett til åndsverk m.v., heretter åndsverksloven. Loven er et resultat av tett nordisk samarbeid, hvilket innebærer at forarbeider og rettspraksis fra de øvrige nordiske landene vil ha relevans<sup>3</sup>. Høyesterett la i Rt. 1985 s. 883 betydelig vekt på de svenske lovforarbeidene ved tolkning av en bestemmelse i den norske åndsverksloven. I den senere tid er det imidlertid klart at EU-retten har fått en mye større plass i lovverket, slik at hensynet til europeisk rettsenhet står sterkt. Dette kommer jeg tilbake til under punkt 1.3.3.2.

Hva gjelder forarbeider, vil Ot. Prp. Nr. 26 (1959-1960) være en relevant kilde. Disse forarbeidene er relativt gamle, men må sies å ha beholdt sin rettskildemessige vekt da det generelt sett ikke finnes et stort omfang av utfyllende rettspraksis. Det kan også være grunner til å se til det nye lovforslaget i Prop. 104 L lov om opphavsrett til åndsverk mv., som på et vis vil være en form for etterarbeider.

---

<sup>3</sup> Rognstad (2009) s. 41. Selv om det er et felles mål om rettsenhet, betyr det ikke at ulikheter ikke kan oppstå i den senere rettsutviklingen. Se også Ot.prp. nr. 26 (1959-1960) s. 4-5

På det kontraktsrettslige og obligasjonsrettslige området vil de alminnelige obligasjonsrettslige prinsippene komme til anvendelse der annet ikke er avtalt mellom partene. De lovfestede reglene kommer ikke direkte til anvendelse da dette ikke passer for åpen kildekode-lisenser, og det må derfor tas utgangspunkt i de ulovfestede reglene. De alminnelige obligasjonsrettslige prinsippene er blitt til gjennom sedvane og er uttrykk for gjeldende rett. Dette vil være særlig relevant hva gjelder sanksjoner. Etter norsk rett råder det kontraktsfrihet, hvilket innebærer at de kontraherende parter kan avtale seg vekk fra disse alminnelige prinsippene. Disse er derfor bakgrunnsrett, og kommer til anvendelse der partene ikke er enige om annet.

### 1.3.2 Rettspraksis

På generell basis finnes det relativt få Høyesterettsdommer på det opphavsrettslige området. Ingen av disse tar opp åpen kildekode-problematikken, hvilket innebærer at det er usikkert hvordan domstolen hadde løst disse tvistene. Heller ikke i underrettene er disse spørsmålene direkte berørt. Tilsvarende gjelder for svensk og dansk rettspraksis.

### 1.3.3 Internasjonale kilder

Selv om opphavsretten er territoriell, har den vært gjenstand for internasjonal regulering og samarbeid, hvorav de viktigste er Bernkonvensjonen fra 1886, WIPO Copyright Treaty (WCT) fra 1996, og TRIPS-avtalen fra 1994. På regionalt nivå er opphavsretten regulert gjennom en rekke direktiver fra EU som Norge har gjennomført i forbindelse med harmoniseringsprosessen på det immaterielle området.

#### 1.3.3.1 Internasjonal rett

Norge er tiltrådt Bernkonvensjonen og TRIPS-avtalen, og disse traktatene regulerer i hovedsak medlemslandenes forpliktelser overfor opphavere fra andre unionsland og stadfester bestemmelser om verkstype og vernetid. Norge er ikke tiltrådt WCT, men traktaten er likevel relevant da EU har ratifisert denne. Denne traktaten er også grunnlaget for EUs generelle opphavsrettsdirektiv, hvilket Norge har gjennomført. Denne har derfor indirekte betydning for Norge<sup>4</sup>.

---

<sup>4</sup> Ibid s. 47

### *1.3.3.2 EU-retten og EU-domstolens praksis*

Gjennom lov 27 nov 1992 nr. 109 om gjennomføring i norsk rett av hoveddelen i avtale om Det europeiske økonomiske samarbeidsområde (EØS) m.v. (EØS-loven), er Norge forpliktet til å gjennomføre en rekke opphavsrettslige direktiver, hvilket innebærer at norsk lovgivning må være i overensstemmelse med disse og tolkes direktivkonformt, jf. Rt. 2000 s. 1811 (Finanser I). Videre, er Norge forpliktet til å tolke EØS-rettslige bestemmelser i samsvar med EU-domstolens som ble truffet før undertegningen av EØS-avtalen, jf. art 6. Senere rettspraksis fra EU-domstolen har likevel stor vekt ved tolkning av EØS-rettslige bestemmelser, jf. Rt. 2002 s. 391 (God Morgon). Grunnloven § 88 fastslår at ”Høyesterett dømmer i siste instans”, hvilket innebærer at Norge formelt sett ikke er bundet av EU-domstolens uttalelser. Det er likevel slik at disse uttalelsene blir tillagt stor vekt, og vil være en viktig rettskilde.

De mest relevante direktivene er Rdir. 2001/29/EF om harmonisering af visse aspekter af ophavsret og beslægtede rettigheder i informationssamfundet, heretter infosoc-direktivet, og Rdir. 1996/9/EF om retlig beskyttelse af databaser, heretter dataprogramdirektivet. Infosoc-direktivet er det generelle opphavsrettslige direktivet, hvilket betyr at denne også gjelder dataprogrammer. Dataprogramdirektivet vil være *lex specialis* og utfyller infosoc-direktivet. Direktivene har som formål å harmonisere opphavsrettslovgivningen mellom medlemsstatene da det på dette området har vært store forskjeller.

Eu-domstolen har også avsagt mange dommer på dette området, men ingen som direkte gjelder åpen kildekode-programmer.

### *1.3.3.3 Utenlandsk rettspraksis*

Utenlandsk rettspraksis har generelt liten rettskildemessig vekt, og særlig i de rettssystemer som er annerledes enn det norske. Grunnet mangelen på norsk rettspraksis, har det vært nødvendig å se til utenlandsk praksis. Selv om disse ikke kan tillegges noe særlig vekt, kan de belyse vurderingsmomenter og hensyn som kan være av betydning i norsk sammenheng.



## 2 Introduksjon til Open Source Software

I dette kapittelet gis det en kort og forenklet teknisk introduksjon til datamaskinprogrammets oppbygning. Det vil fokuseres på hensikten med, og betydningen av kildekoden, samt hva som menes med en åpen kildekode. Videre vil det gis en summarisk fremleggelse av historien og filosofien bak åpen kildekodebevegelsen, dens betydning for bransjen i dag, og hva som kjennetegner et åpen kildekodeprogram.

### 2.1 Dataprogrammer og kildekoden

#### 2.1.1 Hva er et dataprogram?

Et dataprogram kan beskrives som “en serie instruksjoner som forteller maskineriet i en datamaskin hva den skal gjøre. Et dataprogram kan integreres i selve maskinvaren, eller i en uavhengig form som må lastes inn i en datamaskins hukommelse og «kjøres».”<sup>5</sup>.

Dataprogrammer<sup>6</sup> skrives i et programmeringsspråk<sup>7</sup> i form av kildekode. Kildekoden ligner vårt vanlige språk da det består av menneskelig lesbar tekst, og er en serie med instruksjoner. Datamaskinen vet ikke mer enn man forteller den. Det er opp til programmereren å gi instruksjoner som maskinen skal utføre. Instruksjonene gis i kildekodeform og kompiles til maskinkode. Denne oversettelsesprosessen er nødvendig da maskinen kun forstår maskinkode. Maskinkode er en sekvens med binære tegn bestående av 0 og 1 (bits). Programmets funksjoner blir utformet og kontrollert gjennom kildekoden. Dersom en

---

<sup>5</sup> Rossen. I Store Norske Leksikon (2015 a), se også lignende definisjon i NOU 1986:18 s. 7, note 1. Et dataprogram kan også ses på som en ”formulering av algoritmer og tilhørende datastrukturer og konkretiseringen av disse i form av prosesser og objekter i et bestemt programmeringsspråk”, se Kielland (2017) s. 54. Hvilken definisjon en velger er ikke avgjørende i denne sammenheng, men det er verdt å merke seg at det kan defineres på ulike måter.

<sup>6</sup> Det finnes tre hovedkategorier av programmer, hhv. operativsystemer, applikasjonsprogrammer, og mellomvare. Systemprogrammer gjør maskinen i stand til å utføre oppgaver på seg selv, mens applikasjonsprogrammer gjør det mulig å bruke maskinen til å utføre oppgaver som ikke direkte har med maskinen i seg selv å gjøre. Mellomvare er derimot programvare som gjør det lettere å bruke maskinressurser i et nettverk og er i dag ofte en integrert del av operativsystemet. Se mer i Rossen. I Store Norske Leksikon (2015).

<sup>7</sup> Programmeringsspråket er et formalspråk i motsetning til naturlig språk som for eksempel norsk. Formalspråket gir lite rom for ”poesi” i den grad det er entydig i betydningen og har en streng og nøyaktig syntaktisk oppbygning. Det kan feks. sammenlignes med formalspråket kjemikere bruker når de skriver kjemiske former, hvilket er semantisk utvetydelige. Downey (2012) s. 5 flg. Se også Kielland (2017) s. 58 flg. Eksempler på vanlige programmeringsspråk er C++, Java, og Python.

ønsker å endre programmer, må en endre kildekoden. Kildekoden er derfor helt essensiell i programvareutviklingen.

### 2.1.2 Åpen kildekode og proprietær kildekode

Gjennomsnittlige brukere av programvarer møter som regel aldri på kildekoden. De programmene vi anvender til daglig distribueres både som proprietære og åpen kildekodeprogrammer. Proprietære programmer er programmer der kildekoden ikke er tilgjengelig for brukeren da programmene gis ut i maskinkodeform. Et eksempel på dette er Microsoft® Word. Brukeren får en rett til å anvende programmet mot betaling, men får ikke tilgang til kildekoden og kan heller ikke endre programmet. Proprietære programmer kan også være vederlagsfrie, som for eksempel Skype®. Det avgjørende er altså ikke om en må betale for bruksretten. I slike ”lukkede” programvarer, er det kun en avgrenset krets med personer som har tilgang til å kopiere, endre og undersøke kildekoden<sup>8</sup>. Oftest er det programvareutviklerne og rettighetsholderne. Åpen kildekodeprogrammer, vil som navnet tilsier, også gi tilgang til programmets kildekode. Dette innebærer at også brukeren kan kopiere, undersøke, modifisere, og forbedre kildekoden osv. Har en tilgang til kildekoden, kan en også inkorporere ønskede deler av et program inn i et annet. Et eksempel på et åpen kildekodeprogram er LibreOffice. For ordensskyld nevnes det at det avgjørende er at kildekoden gjøres tilgjengelig, og ikke hvorvidt programmet koster penger eller ei. ”When we speak of free software, we are referring to freedom, not price”<sup>9</sup>. Den praktiske forskjellen mellom proprietære og åpen kildekodeprogrammer, er at mens brukeren ved proprietære programmer kun har anledning til å bruke programmet slik det er, får han eller hun ved åpen kildekodeprogrammer større frihet til å råde over programmet.

## 2.2 Historie og filosofi

### 2.2.1 Ideene bak åpen kildekodebevegelsen

På 50- og tidlig 60-tallet var det ikke mulig å skille programvaren fra maskinvaren, så selve programvaren var ofte vederlagsfri da man tok betalt for maskinen. Det var først i 1969 at IBM® begynte å selge programvaren separat<sup>10</sup>. Dette var starten på kommersialiseringen av programvare. I miljøets tidlige dager var det ikke uvanlig å dele informasjon og kunnskap da

---

<sup>8</sup> Brukere må ofte signere en lisensavtale som forbyr disse å foreta de overnevnte handlingene. Grunnlaget for hvorfor rettighetshaverne av dataprogrammer kan forby slike handlinger, vil behandles nærmere i kapittel 3 og 4.

<sup>9</sup> GPL versjon 3 (2007), fortale tredje avsnitt.

<sup>10</sup> Wikipedia (2018 a)

forskerne ofte arbeidet sammen og måtte ha tilgang til kildekode for å undersøke, endre og forbedre programmene. I 1983 startet Richard Stallman ”the free software movement”, og to år senere ble Free Software Foundation grunnlagt<sup>11</sup>. Hensikten var å fremme bruk av åpne programmer, samt sikre utviklerne og brukernes frihet.

En kan tilnærme seg åpen kildekode på flere måter. Noen har ment at det kun er en form for metodologisk utvikling av programmer<sup>12</sup>, mens andre har sett på det som uttrykk for en filosofi. Legger en den første tilnærmingen til grunn, kan en si at åpen kildekode er en spesifikk måte å utvikle programmer på. Åpen kildeprogrammer utvikles gjerne i prosjekter der en gruppe utviklere har hovedansvaret for prosjektets fremgang, mens en rekke andre programmerere kan bidra med små og store innspill som for eksempel retting av feil (debugging), forslag til modifikasjoner, undersøkelser, testing, skrive dokumentasjoner osv. En fordel med at koden er tilgjengelig for alle og ikke bare en gruppe med utviklere, er at flere har muligheten til å oppdage feil og mangler ved programmet. Programmene kan på denne måten bli sikrere, bli kontinuerlig forbedret, og inneholde færre feil. Denne måten å utvikle programmer på skiller seg fra den tradisjonelle da sluttbrukere av programmet også får ta del i selve utviklingen. På mange måter viskes også den hierarkiske strukturen ut og utviklingen blir mer desentralisert. Den filosofiske siden ved åpen kildekode kan ses på som en reaksjon mot å monopolisere programvarer og informasjon. Åpen kildekode skal gi brukere frihet til å utnytte produktene slik de ønsker og gi brukeren vide rettigheter. Kildekoden er ment for å deles da dette kan fremme innovasjon og sikre fri informasjonsflyt. På denne måten kan man si at åpen kildekodebevegelsen både har en praktisk og en filosofisk side. Et forsøk på å definere åpen kildekodeprogrammer lyder som så:

”OSS is software whose source code has been published and is made available to the public. Anyone can copy, modify, and redistribute the source code without paying royalties or fees but must respect and quote the original contributors. Open source code is developed and maintained by open source communities. These communities

---

<sup>11</sup> Selv om begrepet ”open source” og ”free software” ikke er synonymer vil dette skillet ikke adresseres noe nærmere da det ikke har avgjørende betydning for de juridiske drøftelsene i denne avhandlingen. Se mer om dette i Stallman sin artikkel: <https://www.gnu.org/philosophy/open-source-misses-the-point.html>

<sup>12</sup> Shemtov (2013), s. 7

are virtual communities of practitioners who use the Internet to share protect artifacts.”<sup>13</sup>

Definisjonen dekker i midlertid ikke hva fenomenet ”open source” innebærer i dag<sup>14</sup>. Open source har på mange måter gått fra å være en sosial bevegelse til å bli en kommersiell utviklingsmetode av programmer. Stadig flere åpen kildekodeprosjekter har velfungerende forretningsmodeller, klare markedsstrategier og utviklere som blir betalt for å jobbe med prosjektene. I det følgende skal vi se på hvordan dette har påvirket utviklingen av IT-bransjen og hvordan åpen kildekodeprogrammer brukes i praksis.

### 2.2.2 Open source software: betydning og påvirkning i IT-bransjen

Det første eksempelet på at open source ble brukt som en kommersiell strategi, var i 1998 da Netscape® bestemte seg for å distribuere sin nettleser, nå kjent som Firefox®, som et åpen kildekodeprodukt for å kunne konkurrere med Microsofts® Internet Explorer®<sup>15</sup>. I senere tid har flere teknologikjempere frigitt kildekoden til noen av sine produkter<sup>16</sup>, slik som Googles® TensorFlow og Microsofts® CNTK. Kildekoden ble frigitt dels grunnet en dominoeffekt da konkurrentene friga sine koder, men også fordi open source har den fordel at en kan få hjelp til å utvikle og rette programmene av langt flere enn om koden skulle ha vært forbeholdt et mindre team med ansatte utviklere<sup>17</sup>. Open source er blitt en stor og praktisk viktig del av bransjen, særlig hva gjelder programvarebiblioteker. Et programvarebibliotek<sup>18</sup> kan defineres som en samling med data og programmeringskoder som brukes til å utvikle programvarer og applikasjoner. Disse komponentene har ofte enkle og vanlige funksjoner som for eksempel ”printing”, lese informasjon fra en disk, eller å åpne filer. Ved å bruke programbiblioteker med forhåndsskrevne koder, subrutiner, klasser osv. bruker man mindre tid på å utvikle programmene. Utvikleren kan inkorporere de ønskede funksjonene fra programbiblioteket inn i sitt eget program og slipper dermed å skrive disse kodene selv. Eksempler på slike

---

<sup>13</sup> Fitzgerald (2011), s. 1. Det nevnes også at open source ikke er det samme som gratis programvare (freeware). Freeware kan godt være vederlagsfri, men det er ikke gitt at kildekoden også er tilgjengelig.

<sup>14</sup> Fitzgerald (2006)

<sup>15</sup> Dixon (2004) s. 7

<sup>16</sup> Furness (2017)

<sup>17</sup> Disse programmene gjelder i hovedsak kunstig intelligens, og en rekke store aktører som Facebook®, Google®, Microsoft® og Baidu® frigjorde deler av kildekodene til disse prosjektene. Se Furness (2017)

<sup>18</sup> Med hensyn til avhandlingens omfang henvises leseren til eksterne kilder for en mer detaljert fremstilling av de tekniske og interoperasjonelle aspektene ved programbiblioteker. Se mer om dette i feks. Wikipedia (2018 b).

biblioteker er Boost® og QT®. Et annet praktisk eksempel på programvarebiblioteker er de som tilbyr grafisk brukergrensesnitt<sup>19</sup>. Programmereren kan hente ut forhåndsskrevne koder eller ”pakker” fra bibliotekene for å gjøre programmet kompatibelt med operativsystemet som programmet skal kjøres på i stedet for å skrive disse kompliserte instruksjonene fra bunn av. Slike grafisk brukergrensesnitt-pakker kan også være lisensiert som åpen kildekodeprogram. Eksempler<sup>20</sup> på slike biblioteker er PyQT og JAVA®.

Selv om åpen kildekodeprogrammer ofte distribueres vederlagsfritt, betyr ikke dette at åpen kildekodeprosjekter ikke er profittgenererende. Utviklere benytter ofte en forretningsmodell basert på ”support and service”. Programmet distribueres åpent og vederlagsfritt, men brukere av programmet må betale for supporttjenester, service og oppgraderinger osv. Red Hat® er et firma som anvender denne modellen. Andre utviklere som MYSQL® tilbyr premium-versjoner av programmet som inneholder tilleggselementer. En kan også tilby programmet som åpen kildekodeprogram, men ta betalt for adgangen til å relisensiere programmet som proprietær programvare. Software as a Service (Saas) er også en modell mange utviklere bruker sammen med åpen kildekodeprogrammer. Wordpress® har til eksempel tjent store summer på å tilby slike skyløsninger. Dette kommer jeg tilbake til i kapittel 5.

### 2.2.3 Hva kjennetegner et åpen kildekodeprogram?

I lys av de overnevnte filosofiske ideene ser vi nå på hvilke krav som stilles til et program for at det kan defineres som et åpen kildekodeprogram slik det forstås som et fenomen. Dette har betydning ettersom det kun er programmer som tilfredsstiller disse kravene, som kan lisensieres under åpen kildekode lisenser. Det nevnes innledningsvis at open source software og free software retter seg mot to like, men ikke identiske fenomener. Vilklårene for hva som kvalifiserer seg som open source og free software overlapper hverandre i sin helhet, og jeg finner det derfor tilstrekkelig å gi en generell sammenfatning av hovedvilklårene<sup>21</sup> med hensyn

---

<sup>19</sup> ”Brukergrensesnitt, betegnelse på kontaktflaten mellom brukeren og datamaskinens operativsystem og programmer, avgjør hvordan brukeren styrer programmene.”, Rossen (2018 b). Eksempler på grafisk brukergrensesnitt er styring ved bruk av pekeredskap, åpne og lukke programvinduer etc.

<sup>20</sup> Wikipedia (2017 c)

<sup>21</sup> Se nærmere om kriteriene i <https://opensource.org/osd-annotated> og <https://www.gnu.org/philosophy/philosophy.html>

til avhandlingens omfang. Open source software stiller noen flere vilkår enn free software, men essensen i filosofien kan oppsummeres slik<sup>22</sup>:

*Åpen kildekode:* Brukeren må få tilgang til programmets kildekode.

*Bruksfrihet:* Brukeren må kunne anvende programmet uten restriksjoner hva gjelder formål, antall bruksenheter osv.

*Bearbeidelse:* Brukeren må kunne endre eller inkorporere programmet med andre programmer.

*Eksemplarfremstilling:* Brukeren må kunne fremstille eksemplarer av programmet i uendret og endret tilstand.

*Tilgjengeliggjøring:* Brukeren må kunne gjøre programmet, i endret eller opprinnelig tilstand, tilgjengelig for allmennheten.

*Vederlagsfri:* En kan kreve vederlag for nedlastning eller distribusjon av en kopi av programmet, men brukeren må kunne utøve de overnevnte handlingene vederlagsfritt.

I det følgende skal vi se på den rettslige beskyttelsen av dataprogrammer og de særlige problemstillingene som gjør det gjeldende for denne verkstypen etter åndsverksloven.

---

<sup>22</sup> Min oppsummering av filosofien. Utfyllende tekst er tilgjengelig på <https://www.gnu.org/philosophy/philosophy.en.html>

### 3 Opphavsrettsligvern av åpen kildekodeprogram

#### 3.1 Innledning

Opphavsretten gjelder på det kunstneriske og det litterære felt, og omfatter det vi tradisjonelt forbinder det med, slik som skjønnlitteratur, malerier, skulpturer osv. I dag faller også møbler, utformingen av varemerker, manualer, og dataprogrammer inn under opphavsrettens verneområde. Dataprogrammer vernes som litterære verk etter Bernkonvensjonen artikkel 2, jf. TRIPS artikkel 10, og på europeisk nivå etter dataprogramdirektivets artikkel 1. Etter norsk rett er dataprogrammer nevnt som eksempel på vernet verkstype etter åvl. § 1 (1) punkt 12. Begrunnelsen for opphavsrettslig vern bunner i at dataprogrammer skrives i kildekode som ligner alminnelige litterære tekster selv om de fragmenterte ordene og tallene ikke har et klart betydningsinnhold for andre enn de som forstår programmeringsspråket. Ettersom opphavsretten opprinnelig ble etablert for å verne tradisjonelle kunstneriske og litterære verk, vil en del av bestemmelsene og insentivene bak disse, passe dårlig for dataprogrammer som er nokså ulik i natur og frembringelsesmåte. Det kan diskuteres hvorvidt opphavsrett er den egnede måten å verne dataprogrammer på, men anvendelse av patentretten slik den er formulert i dag, har vist seg å by på utfordringer<sup>23</sup>. Dataprogrammer er dynamiske og har en iboende funksjonell side hvis utforming og kreasjon er betinget på en helt annen måte enn litterære verk. Dette vil bringe utfordringer når en skal vurdere hvorvidt et dataprogram har verkshøyde eller ei.

Opphavsretten begrunnes blant annet i et rettferdighetshensyn der opphaveren skal få nyte frukter av sitt arbeid. ”Opphavsretten skal gi opphavsmannen kompensasjon for den innsats som ligger til grunn for det åndsverket hun har skapt.”<sup>24</sup> Ettersom opphaveren gis belønning i form av eiendomsrett og enerettsbeføyelser, må det også stilles krav til verket for at forutsetningene for belønningssynspunktet slår til. Åndsverk er således et rettslig begrep for verk som tilfredsstillers lovens krav. I dette kapitlet skal vi se på de alminnelige vilkår for vern etter åndsverksloven, hvem som har opphavsrett til verket, og hvilke eksklusive rettigheter opphaveren har.

---

<sup>23</sup> Dataprogrammer som sådan, jf. EPC art 52 (3) ”as such”, kan ikke patenteres da disse anses for å være ”oppskrifter” eller en metode, hvilket er ekskludert patentering. Se mer om dette i Kielland (2017).

<sup>24</sup> Rognstad (2009) s. 31

## 3.2 Vilkår for opphavsrettsligvern

Det nevnes innledningsvis at det etter norsk rett ikke kreves noen form for fiksering<sup>25</sup> av åndsverket for at det skal ha opphavsrettslig vern, ei eller at rettigheten må registreres for å være vernet<sup>26</sup>. Det er ikke nødvendig å bruke ©-symbolet, og det er heller ikke påkrevd at opphaveren oppgir sitt navn på verket, se åvl. § 7. Opphaveren får automatisk vern for sin frembringelse etterhvert som det skapes, og den alminnelige vernetiden er 70 år etter utløpet av opphaverens dødsår, se § 40.

Det stilles tre vilkår for opphavsrettsbeskyttelse som kan utledes fra åndsverksloven. For det første må det foreligge en frembringelse, jf. § 1 (1) ”den som skaper et åndsverk”. For det andre må det foreligge en frembringelse på det kunstneriske eller litterære området, jf. § 1 (2), og for det tredje må frembringelsen ha verkshøyde. Det siste kravet følger av forarbeidene så vel som rettspraksis og er etablert gjeldende rett.

### 3.2.1 Frembringelse

Dagens åvl. § 1 (1) taler om skapelse og ikke frembringelse, men uttrykket er ikke ment til å ha en annen rettslig betydning enn den hadde i tidligere lov<sup>27</sup>. I frembringelseskravet ligger det at verket må ha en ytre realitet, dvs. på en slik måte at ”Trediemand kan gøre sig bekendt med det.”<sup>28</sup> Videre må det sies at ”beskyttelsen gjelder åndsverket i den individuelle form og skikkelse som opphavsmannen har gitt det.”<sup>29</sup> Dette innebærer at verket i endret skikkelse også er vernet så fremt det beholder den identitet opphaveren har gitt det. Det er kun uttrykket eller formen verket har fått som kan vernes, ikke ideene som ligger til grunn. Dette kommer uttrykkelig frem i TRIPS artikkel 9 (2), og er et grunnleggende prinsipp i opphavsretten.

---

<sup>25</sup> Bernkonvensjonen art 2 (2) gir statene frihet til å stille dette som vilkår, men hovedregelen er altså at det ikke kreves.

<sup>26</sup> Registrering kreves for eksempel for varemerker eller patentrettigheter om en skulle håndheve disse overfor andre.

<sup>27</sup> Innst. O. XI (1960-61) s. 14 flg.

<sup>28</sup> Jacobsen (1941) s. 56-57

<sup>29</sup> Knoph (1936) s. 101



For dataprogrammers vedkommende vil det si at ideene bak programmet ikke vil være vernet, ei heller algoritmene<sup>30</sup> i den grad de fremstår som programmets underliggende ideer<sup>31</sup>, da disse anses for å være metoder eller fremgangsmåter. Dette er frie elementer som kan brukes av andre uten at en trenger å innhente samtykke fra opphaveren. Kildekoden vil på en annen side ha en ytre realitet og vil kunne vernes. Det samme gjelder objektkoden selv om denne fremstår som ganske ulik kildekoden. Ettersom objektkoden er en ren oversettelse av kildekoden, vil denne vernes på lik linje da den har samme verksidentitet. Dette kan sammenlignes med musikknoter som spilles, og musikk som tas opp gjennom et digitalt medium. Musikkverket er identisk selv om representasjonsformen er ulik.

### 3.2.2 Vernet verkstype

Lovens § 1 gir en ikke-uttømmende liste over typer av verk som kan vernes som åndsverk.

Dataprogrammer er eksplisitt nevnt i punkt 12, og omfatter i hovedsak kodene<sup>32</sup>. Etter dataprogramdirektivet artikkel 1, er i midlertid også forberedende designmateriale<sup>33</sup> en del av begrepet ”datamaskinprogram”. Disse materialene vernes derfor som dataprogrammet og lovens særregler gjelder tilsvarende<sup>34</sup>. Øvrige materialer som er nært knyttet til programmet slik som programdokumentasjon, manualer etc. vernes ikke som dataprogram.

Brukergrensesnitt eller ”the look and feel” av programmet synes i nordisk rett ikke å være vernet som en del av programmet, selv om dette ikke er helt avklart<sup>35</sup>. Programmeringsspråket

---

<sup>30</sup> Defineres som ” fullstendig og nøyaktig beskrivelse av fremgangsmåten for løsning av en beregningsoppgave eller annen oppgave.” Aarnes (2015).

<sup>31</sup> I litteraturen har det blitt uttrykt at det å si at de underliggende algoritmene er programmets ide, er en overforenkling av forholdet mellom algoritme og program, og skillet mellom ide og uttrykk. Se mer om dette i Swinson (1991) s. 149 flg. Rognstad gir også uttrykk for at begrepet algoritme kan ha flere betydninger og at det derfor ikke alltid er treffende å si at algoritmer er ideer som ikke kan vernes, se Rognstad (2009) s. 114. Lagmannsretten har i RG 1999 s. 330 gitt uttrykk for at ”gitte sekvenser av meningsinnhold i programformat” også kan vernes. Beskyttelsen av verkets uttrykk fremkommer her altså i en mer abstrakt form. Se mer om dette i Kleiva (2009) s. 35

<sup>32</sup> Se mer om definisjoner som leder til denne tolkningen i Rognstad (2009) s. 112

<sup>33</sup> Dette er feks. skisser, diagrammer, flow-charts etc. som brukes i skapelsesprosessen og som resulterer i programmet.

<sup>34</sup> Det har vært diskutert hvor hensiktsmessig det er å inkludere dette materialet i beskyttelsen av selve programmet, og ikke som et eget adskilt verk. Dette kan ha betydning for originalitetsvurderingen. Om de forberedende designmaterialene også skal vurderes som en del av programmet, kan et uoriginalt program få ”avhjelp” av et originalt designmateriale slik at verket i sin helhet får verkshøyde. Det kan også reises spørsmål om hvorvidt materialene i seg selv også er beskyttet som tekster og tegninger i tillegg til å være beskyttet som dataprogrammer, altså få et dobbeltvern. Se mer om disse problemstillingene i Bainbridge (2008) s. 59 flg.

<sup>35</sup> Se Rognstad (2009) s. 113. I sak C-393/09 *Bezpečnostní softwarová asociace – Svaz softwarové ochrany v Ministerstvo kultury*, utaler EU-domstolen at brukergrensesnittet ikke er en del av programmet.

og programmets funksjonalitet er, som utgangspunkt, heller ikke vernet som dataprogram<sup>36</sup>. Det nevnes i denne sammenheng at også deler av et større program kan være opphavsrettslig beskyttet gitt at de oppfyller de øvrige kriteriene<sup>37</sup>.

### 3.2.3 Kravet til verkshøyde- originalitetskrav

Det nevnes innledningsvis at det ikke stilles et nyhetskrav for verket. Utgangspunktet er videre at det ikke skal gjøres en vurdering av verkets kvantitative og kvalitative verdi, men at det avgjørende er hvorvidt verket må ”iallfall i noen grad være uttrykk for original og individuelt preget åndsvirksomhet fra opphavsmannens side.”<sup>38</sup> Høyesterett uttaler noe lignende i Rt. 2007 s. 1329: ”For at en frembringelse skal ha karakter av «åndsverk» i åndsverklovens forstand, må den være resultat av en individuelt preget skapende innsats, og ved denne innsatsen må det være frembrakt noe som fremstår som originalt.” Et lignende kriterium finner man i dataprogramdirektivet artikkel 1, der det fremgår at verket må være et resultat av ”the authors own intellectual creation”<sup>39</sup>. Ved vurdering av originalitetskravet skal en både ta hensyn til enkeltelementer ved verket, så vel som helhetsinntrykket<sup>40</sup>. På en annen side skal det ikke legges vekt på ”frembringelsens kunstneriske, estetiske eller litterære verdi”<sup>41</sup>.

Det har vært behov for veiledende retningslinjer for hva som ligger i uttrykket ”original og individuelt skapende åndsinnsetning.” I Norden har det vært to synspunkter som har vært fremmet: dobbeltfrembringelses-, og valgfrihetssynspunktet<sup>42</sup>. Det første går ut på å hvor stor risikoen er for at noen andre skaper et identisk verk, og det andre beror på hvorvidt opphaveren har blitt stilt overfor flere valgmuligheter hva gjelder verkets utforming. Om det kun finnes en eller få muligheter, kan det tale for at frembringelsen ikke er et resultat av en

---

<sup>36</sup> Se sak 406/10 SAS Institute Inc v World Programming Ltd, der EU-domstolen slutter seg til generaladvokatens uttalelser, para 52 i den rådgivende uttalelsen: “[t]o accept that a functionality of a computer program can be protected as such would amount to making it possible to monopolise ideas, to the detriment of technological progress and industrial development”. Generaladvokat Bot skiller mellom ikke-kreative og kreative funksjonaliteter, hvorav det sistnevnte *kan*, hvis tilstrekkelig originalt, bli vernet etter dataprogramdirektivet.

<sup>37</sup> Se Ot. Prp. Nr. 33 (1989-90) s. 14 og nordisk rettspraksis NJA 2995 s. 256 og UfR 1951 s. 725. Se også EU-praksis i sak C-5/08 Infopaq v. Danske Dagblades forening, der elleve ord fra en nyhetsartikkel ble ansett for å være opphavsrettslig vernet.

<sup>38</sup> Ot. Prp. Nr. 26 (1959-60) s. 12. Det er i den nye åndsverksloven fremmet et forslag om å lovfeste verkshøydekravet, se Prop. 104 L (2016-2017) s. 27 flg.

<sup>39</sup> Dette kravet er ikke å finne i det generelle opphavsrettsdirektivet, men gjelder på ulovfestet grunnlag gjennom en rekke dommer, se f.eks. Sak C-5/08 Infopaq International A/S mot Danske Dagblades Forening

<sup>40</sup> HR-2017-2165-A premiss 67, jf. Rt. 2007 s. 1329 premiss 45.

<sup>41</sup> Rognstad (2009) s. 87, se også dataprogramdirektivets fortale 8. Ledd

<sup>42</sup> Ibid.

original og skapende åndinnsats. Høyesterett har i Rt. 2007 s. 1329 uttalt at dette kun er momenter i den samlede vurderingen og er ikke alene avgjørende<sup>43</sup>.

Før vi går over på vurderingen av verkshøydekravet for dataprogrammer, er det hensiktsmessig å si noe om terskelen for når et verk kan sies å være et åndsverk. Den generelle terskelen etter norsk rett i dag er ikke høy, men kravene kan variere fra verkstype til verkstype<sup>44</sup>. Terskelen etter EU-rettspraksis er tilsynelatende heller ikke særlig høy. Det er likevel vanskelig å si noe konkret om hvor høy terskelen er ettersom dette skjønsmessige kravet i sin helhet ikke er harmonisert i EU. I sak C-5/08 Infopaq v. Danske Dagblades Forening, ble elleve ord fra en nyhetsartikkel etter en helhetsvurdering ansett for å ha verkshøyde. Det avgjørende var ikke antall ord, altså en kvantitativ vurdering, men at ordenes sammensetning hadde verkshøyde. I utgangspunktet er terskelen den samme for alle verkstyper som inngår i det generelle opphavsrettsdirektivet<sup>45</sup>. Hva gjelder dataprogrammer fremgår det av dataprogramdirektivet av originalitetskravet er hvorvidt programmet er uttrykk for ”the author’s own intellectual creation”. Det er i litteraturen antydning at dette setter en lav terskel for originalitet, selv om det ikke kan utledes fra ordlyden i seg selv<sup>46</sup>. Tradisjonelt sett har Tyskland praktisert en streng originalitetsnorm<sup>47</sup>, mens det etter engelsk rett har vært et lavt krav<sup>48</sup>. Det fremgår av Infosoc-direktivets forord punkt 7, at direktivet har som formål å harmonisere opphavsretten. Det må derfor kunne utledes at terskelen vil ligge et sted mellom de to ytterpunktene<sup>49</sup>. Dette er likevel ikke bindende for tolkningen av ordlyden. Ettersom det er mangel på klare holdepunkter, antas det at det ikke stilles særskilte krav til verkshøyde for

---

<sup>43</sup> EU-domstolen har i sak 145/10 Painer v. Standard VerlagsGmbH and Others, gitt uttrykk for at opphaverens frie og kreative valg er utgangspunktet for selve originalitetsvurderingen, og legger vekt på dette. I Rt. 2012 s. 1062 følger Høyesterett opp utgangspunktet om at momentene ikke er avgjørende, selv etter Painer-avgjørelsen<sup>43</sup>. Da Norge etter EØS-avtalens artikkel 1 er forpliktet til å fremme rettsenhet, kan det stilles spørsmål ved hvordan domstolene nå må forholde seg til valgfrihetssynspunktet som avgjørende moment for originalitetsvurderingen. Dette skal imidlertid ikke drøftes nærmere her.

<sup>44</sup> Se f.eks. Rt. 1997 s. 199 som gjaldt byggespesifikasjoner som språkverk, Rt. 2007 s. 1329 som gjaldt sceneverk, Rt. 2013 s. 822. Se også Rt. 1962 s. 964 om brukskunst.

<sup>45</sup> Se sak 145/10 Painer. Høyesterett har i Rt. 2012 s. 1062 begrenset EU-domstolens uttalelser om at det settes en lav terskel for åndsverk med anvendelse for brukskunst da det etter norsk praksis er stilt strengere krav for originalitet for denne type verk.

<sup>46</sup> Se Rognstad (2009) s. 114

<sup>47</sup> Se tysk høyesterettsdom I ZR 52/83 – Inkasso-Programm

<sup>48</sup> For å innfri originalitetskravet i Copyright Designs and Patent Act 1988 artikkel 1, er det nok å vise til ”labor, skill or effort”. Dette kravet er utviklet gjennom rettspraksis.

<sup>49</sup> Rosenmeier (2001) s. 206 og Kielland (2005) s. 108-109.

dataprogrammer<sup>50</sup>.

Hva gjelder selve verkshøydevurderingen er valgfrihets- og dobbeltfrembringelsessynspunktene sentrale vurderingsmomenter<sup>51</sup>. Dataprogrammer er i sin natur ment til å løse bestemte oppgaver og er funksjonelt betinget i sin oppbygning og måten en velger å skrive kodene på, samt de valgene en tar i prosessen. Om programmets uttrykk og utforming er bundet av programmeringsspråkets strenge formalitet, interoperabilitet, og andre tekniske begrensninger, kan man si at programmet er funksjonalitetsavhengig. Selv om det rent matematisk sett kan foreligge svært mange valgkombinasjoner i løpet av utviklingsprosessen, vil funksjonalitetsavhengigheten kunne føre til at valgene ikke vil anses for å være frie og individuelle. Dersom valgene som foretas er styrt av tekniske begrensninger eller med hensyn til oppgavene programmet skal løse, kan det tale i mot at det foreligger en individuell og skapende innsats<sup>52</sup>. Om det kun er svært få valgmuligheter kan det føre til risiko for dobbeltfrembringelse av identiske verk.

”Dobbeltfrembringelseskriteriet er imidlertid bedre egnet ved verkshøydevurderingen for andre verk enn datamaskinprogrammene. Til tross for programmeringsspråkernes begrensninger er det mulig å skrive tilsvarende funksjoner for et program på et utall forskjellige måter, med for så vidt unødvendige kodesnutter”.<sup>53</sup>

Programmereren kan også velge å bruke ulike variabler og headinger. Faren for identiske frembringelser er derfor forholdsvis liten.

Som nevnt kan også deler av et program være vernet som åndsverk dersom disse delene i seg selv oppfyller originalitetskravet. Svært små verksdeler slik om enkeltord i en linje med kode vil ikke kunne ha verkshøyde på samme måte som enkeltord i en roman ikke vil være et åndsverk i seg selv. Disse elementene kan fritt benyttes av andre. Spørsmålet blir så om og hvordan et program kan få verkshøyde dersom en setter sammen ulike verksdeler som i seg

---

<sup>50</sup> Rognstad (2009) s. 115.

<sup>51</sup> Rosenmeier (2001) s. 209, og Rognstad (2009) s. 115.

<sup>52</sup> Rene funksjonelle elementer vil ikke ha verkshøyde da de ikke vil være uttrykk for en individuell skapende innsats. For å vurdere hvorvidt noe er rent funksjonelt, kan en se på om elementet følger bransjestandard eller praksis. Om dette besvares bekreftende kan det tale for at elementet er funksjonelt og derfor ikke har verkshøyde. Jonny (2010) s. 28.

<sup>53</sup> Kielland (2005) s. 111

selv ikke har verkshøyde. Dette kan være tilfellet for programmer utviklet med CASE-verktøy<sup>54</sup>. Noen typer av CASE-verktøy kan automatisere programutviklingsprosessen ved å lage ”flowcharts”, modeller, systemspesifikasjoner, generere koder etc. Visse CASE-tools tilbyr også ferdige moduler der en kan klikke og dra ønskede funksjoner inn i sitt eget program ved hjelp av et grafisk brukergrensesnitt. Eksempler på dette er Rational Rose og MetaEdit+. Det kan derfor spørres om og på hvilken måte det foreligger skapende åndsinnsetts fra opphaverens hånd dersom en bruker slike automatiserte verktøy til å skape programmer. Om programmereren setter sammen ferdige moduler, kan han eller hun få opphavsrett dersom sammenstillingen av delene er uttrykk for individuell og skapende åndsinnsetts. Her kan det imidlertid også oppstå problemer hva gjelder hvem som har opphavsretten til de ulike delene og det ”endelige” programmet som helhet.

Det skal gjøres en konkret helhetsvurdering med valgfrihets- og dobbeltfrembringelsessynspunktet som hjelpemomenter. Det avgjørende er om programmet fremstår som originalt og er et resultat av en skapende og individuell innsats. Den skapende og individuelle innsatsen ligger her i den kombinasjonen av valg som er foretatt. Valgmuligheter som gjelder programmets struktur kan tilføre originalitet. For eksempel er det i litteraturen hevdet at et ”dårlig” program lettere vil kunne oppfylle originalitetskravet enn de ”gode”, da de effektive programmene vil ”i likhet med den rette linje mellom to punkter, nettopp være helt uten preg av en individuell skapende opphavsmannsinnsats.”<sup>55</sup> Likeledes vil svært enkle programmer også være uten verkshøyde da det ikke har krevd noen form for kreativ åndsinnsetts. Et banalt eksempel på dette er det første programmet de fleste programmerere lærer seg å skrive: ”Hello, World”. Det gjenstår å se hvordan domstolene vil foreta verkshøydevurderingen for dataprogrammer, men forarbeidene gir antydning til at det mest sannsynlig ikke skal veldig mye til: ”det typiske... være at et program av noen kompleksitet, og som har interesse for markedet, er så preget av den eller de som står bak programmet, at det ikke vil være tvilsomt at det tilfredsstillende kravene for verkshøyde.”<sup>56</sup>

---

<sup>54</sup> CASE står for ”computer-aided software engineering” og er automatiserte programutviklingsverktøy som brukes for å designe, implementere og utvikle programmer. Se mer om dette i [https://en.wikipedia.org/wiki/Computer-aided\\_software\\_engineering#CASE\\_Software](https://en.wikipedia.org/wiki/Computer-aided_software_engineering#CASE_Software)

<sup>55</sup> Rognstad (2009), s. 86

<sup>56</sup> Ot. Prp. Nr. 33 (1989-90) s. 14.

### 3.3 Opphavsrettens subjekter

Utgangspunktet i norsk rett er at skaperen av et åndsverk får opphavsretten, jf. åvl. § 1 (1). Som utgangspunkt er det altså programmereren som får opphavsretten til programmet.

#### 3.3.1 Ansettelsesforhold

I ansettelsesforhold vedrørende dataprogrammer, er det imidlertid et unntak, jf. åvl. § 39g<sup>57</sup>. Dersom programmet skapes av en arbeidstaker ”under utførelsen av oppgaver som omfattes av arbeidsforholdet eller etter arbeidsgivers anvisninger”, går opphavsretten over til arbeidsgiveren. Programmereren beholder likevel sine ideelle rettigheter etter § 3. Her kan det imidlertid oppstå problemer dersom den ansatte programmereren skriver programmer eller bidrar til åpen kildekodeprosjekter på siden uten å ha avtalt dette med sin arbeidsgiver, og at disse bidragene kan sies å falle inn under arbeidsforholdet<sup>58</sup>. Dette vil ikke bli drøftet nærmere med hensyn til avhandlingens omfang.

#### 3.3.2 Oppdragsforhold

Ved oppdragsforhold er det ofte flere eksterne konsulenter som er innom prosjektet på ulike stadier. Opphavsretten vil tilligge den som skaper programmet eller kommer med bidrag som kan sies å ha verkshøyde. Oftest vil det være tale om medopphavsrett der flere opphavere har opphavsrettigheter i sluttresultatet. Oppdragsgiver vil, uten nærmere avtale, som utgangspunkt få den rett som er rimelig og nødvendig i forhold til oppdragets formål<sup>59</sup>, altså i hovedsak en bruksrett. Det kan dog være gode grunner som taler for andre løsninger, se fotnote 59.

#### 3.3.3 Fellesverk

Åpen kildekodeprosjekter utvikles gjerne i samarbeid. Om bidragene ikke kan skilles fra hverandre som ”særskilte verk”, er det tale om et fellesverk, jf. åvl. § 6. Ettersom disse prosjektene er under kontinuerlig utvikling, kan det være krevende å skille et bidrag fra et annet. Dersom åpen kildekodeprosjektet er et fellesverk, får dette betydning for den enkelte bidragsyters råderett, se § 6 (2) og (3). I praksis skrives det gjerne avtaler der bidragsyterne overfører sine rettigheter til et sentralt organ eller den originale opphaveren, som forvalter opphavsrettighetene på vegne av prosjektet. Dette gjør det mer effektivt å håndheve opphavsrettslige sanksjoner mot tredjeparter. Eksempler<sup>60</sup> på slike avtaler er CLA

---

<sup>57</sup> Se også dataprogramdirektivets artikkel 2 (3)

<sup>58</sup> Se mer om dette i Torvund (1999) s. 124-125

<sup>59</sup> Se Torvund (1997)

<sup>60</sup> Civil Commons (udatert)

(Contributor License Agreement) og CAA (Copyright Assignment Agreement) som brukt av Oracle® og Freedom Software Foundation<sup>61</sup>.

### 3.3.4 Sammensatte verk og samleverk

Sammensatte verk inneholder verk som er satt sammen av flere deler der hver av delene har verkshøyde og er selvstendige verk. Disse verkene omtales også som ”uekte samarbeidsverk”. Et eksempel på sammensatte verk, er bruk av ferdige programbiblioteker i ens eget program. Åpen kildekode-prosjekter er ofte slike programbiblioteker. Opphaverne vil da ha opphavsrett til sine respektive verk i det sammensatte programmet. Om en skulle ønske å råde over verket i sin helhet, må alle opphaverne gi samtykke til dette. Annerledes vil det være dersom et av verkene er falt i det fri. Da kan en fritt råde over hele verket uten å innhente samtykke. Dersom verkene ikke er falt i det fri, kan det oppstå problemer hvis en ønsker å lisensiere sitt verk som åpen kildekodeprogram og har anvendt proprietære programvarebiblioteker i det sammensatte verket. Tilsvarende gjelder dersom en ønsker å kombinere sterke og svake copyleft-lisenser. Dersom det er to inkompatible lisenser som regulerer to selvstendige verk som er satt sammen til et program, vil det dukke opp problemer dersom en av lisensene krever at en råder over det sammensatte verket på en måte som er i strid med den andre lisensen.

Samleverk er det samme som sammensatte verk bare at det er sammensetningen av verkene som har verkshøyde.

### 3.3.5 Bearbeidede verk

Endring og bearbeidelse skiller seg fra fellesverk da disse handlingene oppstår *etter* at verket allerede er frembrakt. En bearbeidelse av et verk er et verk som ligger ”oppå” eller ”inni” et annet verk<sup>62</sup>. Et eksempel på dette er oppgraderinger av programmet. For at bearbeideren skal kunne få opphavsrett, må en ”klarlegge hvilke selvstendige *endringer og tillegg* bearbeideren har tilført originalverket, og så vurdere om *disse* fyller kravet til verkshøyde, hver for seg eller etter en helhetsbedømmelse.”<sup>63</sup> Den nedre grensen for bearbeidelse er endringer som er av ren teknisk eller håndverksmessig arbeid som for eksempel små feilrettinger i et program.

Bearbeiderens råderett er avhengig av den opprinnelige opphaverens samtykke da denne ikke kan disponere over verket uten samtidig å også disponere over det originale verket, jf. åvl. § 2

---

<sup>61</sup> Se <http://www.oracle.com/technetwork/community/oca-486395.html> og <https://www.fsf.org/bulletin/2014/spring/copyright-assignment-at-the-fsf>

<sup>62</sup> RG 1999 s. 330

<sup>63</sup> HR-2017-2165-A premiss 83.

og § 5 (2). Opphaveren av det opprinnelige programmet har vern for sitt verk uavhengig av endringene eller bearbeidelsene som er gjort, jf. åvl. § 1 og § 5 (2), men kan ikke disponere over det bearbeidede verket uten samtykke fra bearbeideren. Det foreligger derfor et gjensidig avhengighetsforhold<sup>64</sup>. Om en programutvikler tar deler fra et annet program, om det så er noen linjer med koder, moduler eller individuelle trekk, og bygger sitt verk på dette, kan utvikleren bli avhengig av samtykke fra opphaveren dersom disse delene har verkshøyde. Dersom det oppstår et nytt og selvstendig verk, kreves det ikke samtykke fra den opprinnelige opphaveren da denne ikke har opphavsrettigheter i dette verket.<sup>65</sup> Det oppstår et selvstendig verk dersom en kun bruker elementer som er frie, som ideer eller enkelte linjer med koder uten verkshøyde, og der det opprinnelige verket ikke går igjen i det nye verket på en slik måte at det bevarer sin identitet. Her kan det oppstå et vanskelig spørsmål. Ettersom åpen kildekodeprosjekter er under kontinuerlig utvikling, kan det spørres om hvorvidt det er mulig å endre eller bearbeide verket så mye at det opprinnelige verket eller uttrykket blir ”vannet ut”. Altså, at det skjer så mange bearbeidelser at det til slutt oppstår et nytt og selvstendig verk. Hvorvidt dette er tilfellet beror på en konkret vurdering: ”bearbeidelsen må ved denne vurderingen holdes opp mot alle sider av originalverket som er uttrykk for originalopphavsmannens åndsinnsetts - detaljer så vel som helhet.”<sup>66</sup>

### 3.4 Opphaverens eksklusive rettigheter

For sin åndsinnsetts får opphaveren en tidsbegrenset rett til å utnytte sitt verk. Opphaveren får både økonomiske og ideelle rettigheter, jf. åvl. § 2 og § 3. De økonomiske rettighetene har en positiv og en negativ side- en får rett til å selv utnytte disse rettighetene, men også hindre andre i å gjøre det samme. Selv om det er opphaveren som får disse eksklusive rettighetene, er det ikke noe i veien for at han eller hun gir andre rett til å utnytte verket, eller overdra rettighetene helt eller delvis, jf. § 39. Eksemplarframstillingsretten er vidt definert, og omfatter varige så vel som midlertidige framstillinger av verket. Når det kommer til dataprogrammer, vil det framstilles midlertidige eksemplarer hver gang programmet ”kjøres”,

---

<sup>64</sup> Ibid premiss 77 flg.

<sup>65</sup> Grensen mellom avledede og selvstendige verk (eller verksdeler) får betydning ved relisensiering av programmer. Om en f.eks. skulle ønske å gå fra en proprietær lisens til GPL kan det oppstå problemer dersom det aktuelle programmet er et avledet verk. Slik relisensiering krever derfor samtykke fra den opprinnelige opphaveren, og i programutviklingssammenheng der det kan være mange medopphavere, vil dette kunne by på utfordringer.

<sup>66</sup> HR-2017-2165-A Premiss 83



overføres, lastes ned fra nettet etc. Tilgjengeliggjøringsretten gjelder utenfor den private sfæren da det må være overfor ”allmennheten”. Hva som menes med allmennheten har vært praktisert noe ulikt i Norge og i EU. I Norge har man i rettspraksis lagt vekt på personlige bånd der allmennheten anses for å være utenfor ”den forholdsvis snevre krets som familie-, vennskaps- og omgangsbånd skaper”<sup>67</sup>, mens det i EU har vært avgjørende med hvorvidt en ubestemt krets av personer har hatt *mulighet* for tilgang til verket<sup>68</sup>.

De ideelle rettighetene er uoverdragelige etter norsk rett, jf. § 3 (3) og Bernkonvensjonen artikkel 6bis (1). Opphaveren beholder disse rettighetene selv om rettighetene til verket er helt overdratt. For åpen kildekodeprogrammer blir de ideelle rettighetene viktige da brukeren får en rett til å endre programmet, og er ofte vederlagsfrie. Navngivelsesretten blir derfor sentral både for at opphaveren får høste heder og ære, men også for å kunne spore hvem som har gjort hvilke endringer i programmet.

Opphaverens enerettigheter er ikke ubegrensede. Han eller hun kan ikke forby handlinger som loven gir brukeren rettigheter til. Eksempler på dette er § 39h og § 39i som gjelder spesielt for dataprogrammer. Opphaveren har heller ikke eksklusiv råderett over elementer i verket som ikke oppfyller kravene til verkshøyde, ei heller ideene bak verket. Det innebærer at bruk av uvernede elementer eller ideer bak et program ikke vil føre til opphavsrettslige inngrep. Dette vil heller ikke være et kontraktsbrudd da opphaveren uansett ikke har noen rettigheter i disse elementene med mindre dette er særlig avtalt.

---

<sup>67</sup> Knoph (1936) s. 89. Se også Rt. 1970 s. 1172, UfR 1999 s. 2011, og UfR 2007 s. 1581.

<sup>68</sup> Se sakene C-306/05 SGAE, C-351/12 OSA, C-466/12 Svensson, C-348/13 BestWater.

## 4 Introduksjon til GNU General Public License versjon 3

### 4.1 Innledning

I dataprogramindustrien er det vanlig å bruke lisenser som gir brukeren en tillatelse til å anvende programmet fremfor å selge dem en kopi av selve programmet<sup>69</sup>. Lisensen består av betingelser og begrensninger som brukeren må godta for å kunne utnytte de ulike tillatelsene. Det finnes en rekke forskjellige typer lisenser som stiller ulike vilkår. Visse lisenser stiller strenge bruksvilkår, slik som Microsoft® License for Office-pakken, mens andre lisenser, typisk de akademiske lisensene, gir brukeren tilgang til programmets kildekode m.m., og vilkårsløs adgang til blant annet å fremstille eksemplarer eller distribuere programmet til allmennheten med valgfritt formål. Eksempler på en slik lisens er MIT License og 3-clause BSD License.

Det finnes også en rekke lisenser som gir tilsvarende tillatelser, men som inneholder en copyleft-klausul. Felles for disse lisensene er at viderelicensiering av programmet, i opprinnelig eller i endret forstand, må gjøres på samme vilkår som i den opprinnelige lisensen. Copyleft-lisensene deles inn i svak-copyleft og sterk-copyleft. Svake copyleft-lisenser gir lisenstaker flere unntak fra hovedregelen om at viderelicensieringen må gjøres på samme vilkår. Mozilla Public License 1.1 (MPL) gir for eksempel lisenstaker anledning til å linke programmer til hverandre uten at copyleft-klausulen utløses<sup>70</sup>. GNU Lesser General Public License<sup>71</sup> (LGPL) er et annet eksempel. GNU General Public License (GPL)<sup>72</sup> er et eksempel på en sterk copyleft-lisens. Kort oppsummert går denne copyleft-klausulen ut på at dersom en ønsker å utnytte det copyleft-lisensierte programmet med andre verk, enten i uendret eller i bearbeidet form, må en frigi kildekode for den uendrede versjonen, den bearbejdede versjonen av det GPL-lisensierte programmet, samt kildekode for verket som det GPL-lisensierte programmet er linket til. Dette behandles nærmere i kapittel 5. Noen av

---

<sup>69</sup> Dette henger sammen med spredningsrettens konsumpsjonsprinsipp som i hovedsak gjelder salg av fysiske eksemplarer etter norsk rett, se Rognstad (2009) s. 160. Hva gjelder konsumpsjon av programmer som er lastet ned fra nettet, vises det til sak C-128/11 Usedsoft v. Oracle International Corp.

<sup>70</sup> Shemtov (2013) s. 79

<sup>71</sup> Denne lisensen er mest praktisk for de som utvikler biblioteker og ønsker at disse skal kunne brukes sammen med proprietære programmer. Lisensen er tilgjengelig her: <https://www.gnu.org/licenses/lgpl-3.0.en.html>

<sup>72</sup> Lisensteksten i sin helhet kan leses her: <https://www.gnu.org/licenses/gpl-3.0.en.html>

tillatelsene i GPL er vilkårsløse<sup>73</sup>, mens andre tillatelser stiller andre vilkår<sup>74</sup>. Versjon 3 av GPL ble lansert i 2007 og er den nyeste. Denne versjonen introduserer en rekke nye begreper som ikke har paralleller til noen jurisdiksjoners opphavsrettslige reguleringer<sup>75</sup>. Dette var et bevisst valg fra forfatterens side for å forhindre at lisensens begreper og dets innhold tolkes i lys av de etablerte opphavsrettslige begrepene fremfor at de tolkes etter lisensens egne definisjoner samt sikre like resultater i eventuelle rettsaker. Problemer med dette utgangspunktet kommer jeg tilbake til under punkt 4.4.1.

Ettersom mange bedrifter bruker store ressurser på å utvikle programmene sine, ønsker de å holde kildekoden hemmelig. Dersom en for eksempel ønsker å benytte seg av åpen kildekodebiblioteker lisensiert under GPL, kan det oppstå interessekonflikter. Det kan også være programvareutviklere som ser fordelene av å utgi programmene med åpen kildekode, men som også ønsker at disse skal kunne brukes sammen med proprietære programmer. Det er derfor ikke uvanlig at utviklere tilbyr dobbellisensiering<sup>76</sup>. Utviklere som MySQL® og Qt® tilbyr programmer under flere lisenser, proprietære så vel som åpen kildekode under blant annet GPLv3.

I dette kapittelet skal jeg se på de kontraktsrettslige aspektene ved GPL, hvordan denne skal tolkes, samt gi en grunnleggende oversikt over lisensens tillatelser og vilkår. Først skal jeg i middeletid gi en kortfattet introduksjon til internasjonal privatrett hva gjelder jurisdiksjon og lovvalg ved anvendelse av GPL. En inngående behandling av de omdiskuterte spørsmål som måtte oppstå, vil ikke behandles.

---

<sup>73</sup> Se f.eks. GNU GPL versjon 3 artikkel 2 annet avsnitt, se også artikkel 9, som gir en ubegrenset og vilkårsløs tillatelse til å "make, run and propagate" verk. Lisensen stiller heller ikke vilkår for rettigheter som brukeren får etter loven.

<sup>74</sup> Se f.eks. artikkel 4

<sup>75</sup> Versjon 2 av GPL brukte en rekke begreper som kunne gjenfinnes i amerikansk opphavsrett, se <https://www.law.cornell.edu/uscode/text/17/106> og <https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html>

<sup>76</sup> Her kan det oppstå problemer dersom en bidragsyter ved et åpen kildekodeprosjekt ikke ønsker at sitt bidrag brukes i proprietære programmer. Utviklerne som mottar bidragene må derfor på forhånd sikre seg rettigheter til å lisensiere programmet proprietært. Dette kan gjøres gjennom egne avtaler, se f.eks. Qt contributor license agreement: <https://www1.qt.io/contributionagreement/>

## 4.2 Jurisdiksjon og lovvalg

Ettersom åpen kildekodeprosjekter i hovedsak distribueres via det globale nettet, vil det kunne oppstå en rekke internasjonale spørsmål i forbindelse med bidragsytere til prosjektet og brukere av programmet. Spørsmål om jurisdiksjon går ut på hvilket lands domstol en sak skal føres for, mens lovvalgsspørsmålet oppstår når den juridiske tvisten kan avgjøres etter flere lands lover. Uavhengig av hvilket lands rett saken skal løses etter, er det vernetingsprosesser som følges. Det er derfor også domstolslandets lovvalsregler som legges til grunn<sup>77</sup>. Det finnes både nasjonale og internasjonale regler for hvordan en bestemmer hvilken jurisdiksjon og hvilket lands rett som legges til grunn. I tillegg kan partene som utgangspunkt avtale dette seg i mellom.

### 4.2.1 Jurisdiksjon

Partene kan som hovedregel avtale hvilket vernetingsprosesser saken skal gå for, jf. tvl. § 4-6. Om dette ikke er gjort, faller en tilbake på bakgrunnsretten. Utgangspunktet etter norsk rett er at saken kan føres for norske domstoler dersom ”saksforholdet har tilstrekkelig tilknytning til Norge”, jf. tvl. § 4-3. Hvis så er tilfellet, vil bestemmelsene om alminnelig vernetingsprosesser utenfor kontraktsforhold gjelde tilsvarende, jf. § 4-4 og § 4-5 (3). Den EØS-gjeldende konvensjonen om domsmyndighet vedtatt i Lugano i 2007, ”gjelder som lov” i Norge, jf. tvl. § 4-8. Luganokonvensjonen skiller mellom saker i og utenfor kontraktsforhold. For saker i kontraktsforhold gjelder domisilprinsippet, jf. artikkel 2 (1)<sup>78</sup> og 5 (1). For saker utenfor kontrakt, f.eks. erstatning etter åndsverksloven, kan saken føres der skaden ble voldt<sup>79</sup>, jf. artikkel 5 (3). GPL versjon 3 inneholder ingen klausul om valg av jurisdiksjon. Vernetingsprosesser bestemmes derfor av det konkrete saksforholdet.

### 4.2.2 Lovvalg

Når det kommer til lovvalg, er det ulike lovvalsregler for ulike rettsområder. Ettersom GPL både har en opphavsrettslig og en kontraktsrettslig side, kan det være at lovvalget blir forskjellig alt ettersom hva den konkrete saken gjelder. Det nevnes også at GPL er bygget opp etter amerikansk rett, noe som kan by på problemer når en skal løse tvisten etter norsk rett da regelsettene er nokså forskjellige.

---

<sup>77</sup> Rognstad (2009) s. 427.

<sup>78</sup> En mer detaljert forklaring av de alminnelige og de spesielle reglene finnes i Ot.prp. nr. 89 (2008-2009) punkt 5.2.1 og 5.2.2.

<sup>79</sup> Dette inkluderer også de stater der den skadevoldende handling får virkning. For handlinger utført på nettet kan dette være av praktisk betydning.

#### 4.2.2.1 *Opphavsrettslige tvister*

Opphavsretten er i hovedsak et nasjonalt anliggende, og reguleres etter nasjonale regler, jf. åvl. § 57. Selv om det finnes en rekke internasjonale traktater på det opphavsrettslige området, er det nasjonalstaten som gir rettigheter og håndhever sanksjoner. Det nevnes også at opphavsretten har et strafferettslig tilsnitt. Med dette som bakgrunn, er det derfor det landets lov hvis handling har funnet sted, som skal anvendes. Dette kommer til uttrykk i Bernkonvensjonen artikkel 5 (2) som sier at "the extent of protection, as well as the means of redress afforded to the author to the protect its rights, shall be governed by the laws of the country where the protection is claimed." Dette grunnprinsippet bør suppleres av det alminnelige tilknytningskriteriet etter norsk rett, jf. Rt. 1923 II s. 59<sup>80</sup>. "Et rettsforhold bør bedømmes etter rettsreglene i det landet det har sin sterkeste tilknytning"<sup>81</sup>.

#### 4.2.2.2 *Kontraksrettslige tvister*

Hovedregelen er også her at partene kan avtale hvilket lands rett tvisten skal løses etter. Dersom slik avtale ikke foreligger, anvender man i norsk rett den såkalte individualiserende metode. Denne metoden ble først introdusert i Rt. 1923 II s. 59, og er fortsatt uttrykk for gjeldende rett i dag. Utgangspunktet er altså at en skal velge det lands rett hvis sak har sterkeste tilknytning<sup>82</sup>. Det skal foretas en helhetsvurdering av alle relevante momenter, subjektive som objektive. GPL har ingen lovvalgsklausuler, jf. GPLv3 artikkel 2 "(...)as provided by copyright law", og artikkel 7 "(...) under applicable law". Hvilket lands rett som legges til grunn må avgjøres konkret, og her kan det oppstå en del spørsmål knyttet til distribusjon av programmer via Internett. Dette lar jeg imidlertid være å gå inn på.

Oppsummeringsvis kan det utledes at det er av betydning hvorvidt tvisten gjelder de opphavsrettslige eller de kontraksrettslige vilkårene i GPL da dette vil ha direkte utslag på hvordan saken løses og overfor hvilken domstol.

---

<sup>80</sup> Rognstad (2009) s. 429

<sup>81</sup> Ibid.

<sup>82</sup> Cordero-Moss (2011) s. 1

## 4.3 Kontraktsrettslige aspekter ved GPL

### 4.3.1 Avtaleslutning

GPLv3 er utformet som en standardavtale da den ikke er individuelt fremforhandlet. Lisensen kan heller ikke endres, jf. :”everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed”<sup>83</sup>. Dersom en ønsker å utnytte lisensgjensstanden, må en godta GPL slik den fremkommer. Selv om lisensen ble sendt ut på høring under utformingsprosessen, vil det ikke foreligge noen form for forhandling mellom lisensgiver og lisenstaker før avtaleinngåelsen. Lisensen er derfor ensidig utformet av lisensgiver, hvilket kan ha betydning for hvordan lisensen skal tolkes.

Åpen kildekodeprogrammer lastes i hovedsak ned fra Internett. Visse leverandører bruker såkalte clickwrapavtaler<sup>84</sup>, mens andre gjør programmene fritt tilgjengelige for nedlastning uten noen form for avtaleinngåelse før en får tilgang til programmet. Free Software Foundation, som har utgitt lisensen, anbefaler at en legger inn en lisensnotis i starten av hver kildefil. Dette ligner mer på en form for browswrapavtale. Da dette kun er anbefalinger og ikke en del av lisensen i seg selv, er det intet som hindrer utvikleren å gjøre lisensen tilgjengelig på andre måter<sup>85</sup>. Det er heller intet krav om at brukeren må ha uttrykkelig godtatt avtalen<sup>86</sup>, inngåelsen er snarere basert på en presumsjon av aksept<sup>87</sup>. Dette henger sammen med at åpen kildekodeprogrammer distribueres fritt og ikke gjennom en fast kanal som lisensgiver har kontroll over. Overnevnte avtaleinngåelsesformer vil være nokså illusorisk når programmene likevel er ment til å spres helt fritt. GPLv3 artikkel 9 gir uttrykk for at en aksepterer lisensen ved utnyttelse av programmet gjennom visse handlinger: ”by modifying or propagating a covered work, you indicate your acceptance of this License to do so”. Det har

---

<sup>83</sup> Se GNU GPLv3 Copyright notice. Det kan spørres om dette har betydning ved en evt. kontraktsrevisjon.

Dersom lisensen skulle ha svært urimelige virkninger slik at avtl. § 36 kommer til anvendelse, er det gode grunner for å anta at notisen om at lisensen ikke kan endres, ikke vil kunne opprettholdes. Det kan også spørres om hvilke konsekvenser, hvis noen, det får dersom en endrer avtalen. Dette skal jeg imidlertid ikke adressere.

<sup>84</sup> Se f.eks. Qt: <https://www.qt.io/download>. En Clickwrapavtale er elektroniske avtaler der brukeren kan klikke seg inn på avtalen og godta denne før en får programmet. Et slikt elektronisk ”klikk” fungerer således som en signatur i tradisjonell forstand.

<sup>85</sup> Se også GPLv3 artikkel 0 åttende ledd.

<sup>86</sup> Ser en f.eks. til Open Source Initiative og deres definisjon av ”open source” punkt 10, kan det snarere utledes at clickwrapavtaler ikke er den optimale avtaleinngåelsesformen: <https://opensource.org/osd-annotated>

<sup>87</sup> Zhu (2013) s. 291

vært argumentert at disse faktorene gjør at GPL kontraktsrettslig står på svak grunn<sup>88</sup>. På en annen side er GPL blitt håndhevet som gyldig inngått avtale i andre jurisdiksjoner<sup>89</sup>. Ved vurdering av hvorvidt avtalen er gyldig inngått, kan det være grunner for å se til ehandelsloven § 11 som oppstiller en rekke momenter som kan ha betydning for avtalens gyldighet<sup>90</sup>.

#### 4.3.2 Avtaletolkning

Avtaler tolkes med utgangspunkt i partenes felles subjektive forståelse. Dersom slik forståelse ikke foreligger, skal avtalen tolkes objektivt. Ved skriftlige avtaler betyr dette at en tar utgangspunkt i den naturlige språklige forståelse av avtalens ordlyd<sup>91</sup>, jf. Rt. 1997 s. 1807. GPL er en standardavtale, og det har hersket uenighet om hvordan slike avtaler skal tolkes. Visse har hevdet at en kan se til avtalens tilblivelseshistorie, formål etc., mens andre har argumentert for en strengere objektiv tolkning<sup>92</sup>. Det kan være grunner for å se til avtalens forarbeider, men dette skal gjøres med stor varsomhet, særlig hvis det er vilkår i avtalen som tilgodeser en parts interesser. ”I slike tilfeller kan det være mer nærliggende å tolke standardvilkårene innskrenkende i favør av den underlegne part”<sup>93</sup>, altså benytte uklarhetsregelen. GPL går langt i å gi brukere tillatelser, og i visse sammenhenger kan dette være på bekostning av lisensgivers handlefrihet, se f.eks. GPLv3 artikkel 5. Hvor ensidig eller balansert vilkårene, samt formålet med de enkelte vilkårene og lisensen som helhet, vil kunne ha betydning for tolkningsutfallet. Det kan også tenkes at GPL fremstår som et gaveløfte der lisensgiver gir lisenstaker en fordel. I så fall kan det tale for at avtalen skal tolkes til fordel for lisensgiver. Det må dog nevnes at ikke alle åpen kildekodeprogrammer er vederlagsfrie. Det kan derfor spørres om det er en fordel som gis dersom en må betale for selve programmet selv om en får tilgang til kildekodeprogrammet, samt om det foreligger en gavehensikt. GPL retter seg mot både profesjonelle så vel som private brukere. Hvem den kontraherende part er, og vilkårenes

---

<sup>88</sup> De Alaiza Cardona (2006) s. 60 flg. I nyere rettspraksis fra USA har domstolene i midlertid kommet frem til at GPL er en gyldig inngått avtale, se f.eks. *Artifex v. Hancom*.

<sup>89</sup> Se distriktsdommer fra München, *Welte v. Sitecom Deutschland GmbH* og *Welte v. Skype Technologies S.A.* Det finnes også en rekke dommer fra USA, se f.eks. *Artifex v. Hancom*. Selv om disse dommene har liten rettskildemessig vekt, er de likevel illustrerende. Se også Jaeger (2010).

<sup>90</sup> Med hensyn til oppgavens omfang vil jeg ikke gå nærmere inn på dette. Det nevnes for såvidt at ehandelsloven gjelder for informasjonssamfunnstjenester jf. ehl. § 1 (2) hvis definisjon favner vidt. Det kan derfor godt tenkes at leverandører av åpen kildekodeprogrammer som formidler sine produkter på Internett, faller inn under lovens virkeområde.

<sup>91</sup> Selv om avtalen er forfattet på engelsk vil de norske tolkningsreglene gjelde dersom det er norsk rett som anvendes.

<sup>92</sup> Woxholth (2014) s. 419

<sup>93</sup> *Ibid* s. 420

virkning i de enkelte tilfellene (reelle hensyn), vil også kunne spille inn som momenter. FSF som har forfattet avtalen, har også opprettet en FAQ-side der de forklarer hensyn bak de enkelte bestemmelsene, samt forklarer hvordan lisensen skal tolkes og hvilket innhold bestemmelsene har etter deres mening. Ettersom dette er en standardavtale, vil forfatterne sjeldent være en av de kontraherende parter. Å legge mye vekt på hvordan FSF mener lisensen skal tolkes, vil ikke være hensiktsmessig.

Selv om lisensen er forfattet på engelsk, er hovedsynspunktet at slike kontrakter skal behandles på samme måte som de norske, altså etter nasjonalrett. Noen av begrepene som brukes i lisensen er inspirert av og basert på amerikansk opphavsrett. Selv om tolkning av utenlandske lover og rettspraksis ikke har noen særlig rettskildemessig vekt, kan det likevel være grunner til å se på bestemmelsenes innhold ettersom GPL i stor grad er inspirert av den amerikanske opphavsrettslovgivningen<sup>94</sup>.

Utgangspunktet må likevel være at avtalen tolkes objektivt, med vekt på lisensens formål og reelle hensyn.

#### **4.4 GPL versjon 3: Tillatelser og vilkår**

##### **4.4.1 Rettslig grunnlag**

Dersom det aktuelle dataprogrammet er et åndsverk, vil det kun være opphaveren som har rett til å foreta visse handlinger, se f.eks. åvl. § 2. Skulle opphaveren ønske å gi andre tilsvarende rett, har han eller hun anledning til dette, jf. åvl. § 39 (1). En slik hel eller delvis overdragelse av rettigheter kan gjøres i form av en lisensavtale. Det finnes ulike typer av lisenser, men den mest aktuelle lisensformen for åpen kildeprogrammer er ”enkle lisenser”<sup>95</sup>. Slike lisenser gir brukeren en ikke-eksklusiv positiv rett til å utnytte programmet. I utgangspunktet kan brukeren ikke viderelisensiere programmet ved enkle lisenser, jf. § 39b, men GPL gir en slik tillatelse, jf. artikkel 10. GPL bygger dermed på opphavsretten, selv om visse vilkår kan være rent kontraktsrettslige.

---

<sup>94</sup> Se mer om tolkning av engelskspråklige standardavtaler i Selvig (1986) s. 4

<sup>95</sup> Rognstad (2009) s. 340 flg.



GPL forsøker å løsrive seg fra alle jurisdiksjoner og deres opphavsrettslige tradisjoner<sup>96</sup>. På en annen side er det heller ikke mulig å tolke den helt uavhengig av opphavsretten. Lisensen bygger på og bruker opphavsrettslige begreper og opphavsrettens systematiske inndelinger av enerettighetene. Selv om begrepene i lisensen kan ha et annet innhold enn de tradisjonelle begrepene, finner jeg det likevel hensiktsmessig å ta utgangspunktet i disse da begrepene i lisensen ikke er klart nok definert.

#### 4.4.2 Eksemplarfrestillingsretten

Når en laster ned et åpen kildekodeprogram fra nettet, eller anvender (kjører) programmet, lages det en kopi i datamaskinens internminne. Slik eksemplarfremstilling faller inn under opphaverens enerett etter åvl. § 2 (1) som dekker fysiske, digitale, og varige eksemplarer. Midlertidige eksemplarer faller inn under unntaket fra eneretten etter § 11 a. Da slike handlinger i utgangspunktet krever samtykke, kan opphaveren også stille vilkår for hvordan disse handlingene skal utnyttes. Lisensen gjør imidlertid slike handlinger vilkårsløse, jf. GPLv3 artikkel 2 (2), og 9. Formålet med åpen kildekodeprogrammer er nettopp å dele kunnskap og gi brukeren frihet til å utnytte det slik en vil, jf. fortalens tredje ledd. Laster man ned og lager en kopi av programmet, kan man gjøre dette uten å godta lisensen. Annerledes arter det seg dersom en også vil gi andre tilgang til programmet.

#### 4.4.3 Tilgjengeligjøringsretten

Lisensen introduserer begrepet "convey" og defineres i artikkel 0 (7) som "any kind of propagation that enables other parties to make or receive copies." Med "propagation" menes alle type handlinger som faller inn under opphaverens enerett, se artikkel 0 (6). Definisjonen av "convey" gir uttrykk for at begrepet retter seg mot tilgjengeligjøringsretten, jf. åvl. § 2 (1) og deriblant spredning-, visnings-, og fremføringsretten, jf. § 2 (2). Dette kommer også til uttrykk i fortalens sjette ledd der det uttales at lisensen gir tillatelse til å "copy, distribute and/or modify". Selv om de nye begrepene må tolkes autonomt, kan det likevel sies at de er inspirert av amerikansk opphavsrett da de tidligere versjonene av lisensen brukte de opphavsrettslige begrepene i loven. Det er også uttalt fra lisensgiverens side at "convey" mer eller mindre betyr det samme som "distribute" i versjon 2<sup>97</sup>. "Distribute" etter The Copyright

---

<sup>96</sup> Til sammenligning har UNIDROIT og PECL lignende formål da de angir prinsipper for kontraktsretten uavhengig av de ulike nasjonale jurisdiksjonene.

<sup>97</sup> se <https://www.gnu.org/licenses/gpl-faq.html#ConveyVsDistribute>

Act 17 U.S.C. § 106 (3), favner videre enn tilgjengeliggjøringsretten etter norsk rett<sup>98</sup>. Likevel kan det utledes at lisensen sikter til en eller annen form for tilgjengeliggjøring av programmet.

Lisensen lister opp ulike tilfeller av tilgjengeliggjøring med varierende vilkår.

Tilgjengeliggjøring av en ren kopi av programmet er betinget av at brukeren navngir opphaverne og videreformidler lisensvilkårene, jf. artikkel 4. Dersom en ønsker å tilgjengeliggjøre en endret versjon av programmet, følger det av artikkel 5 at en i tillegg til vilkårene i artikkel 4, må oppgi at en har endret programmet samt frigi kildekoden for "the entire work". Dersom programmet er utgitt i objektkodeform, må en i tillegg til de øvrige vilkår, også gi brukeren tilgang til programmet i kildekodeform.

I det følgende kapittel skal jeg se nærmere på copyleft-klausulen i artikkel 5.

---

<sup>98</sup> Gue (2012) s. 107. Denne inndelingen i loven er foreslått videreført i den nye åndsverksloven, se proposisjonens side 30 flg.

## 5 Nærmere om copyleft-klausulen

### 5.1 Innledning

Bruk av åpen kildekodeprogrammer kan få stor betydning for lisenstakere dersom copyleft-klausulen kommer til virkning. Særlig vil det være viktig for selskaper som utvikler egne programmer. For det første kan det være avgjørende ved salg av virksomheten hvorvidt en har brukt GPL-lisensiert kode i sine egne programmer. Oppkjøpssiden vil trolig ikke betale meget for et selskap hvis program må gjøres tilgjengelig for alle i form av kildekode. For det andre vil det også dukke opp problemer dersom en ønsker å utvikle programmer i samarbeid med andre virksomheter. Dersom en part bruker åpen kildekodeprogrammer i sin utvikling, kan dette smitte over på den andre parts bidrag. Den ”virale” klausulen i GPL har av mange blitt omtalt som en infeksjon. Dersom en del av programmet er blitt ”smittet” av GPL, blir resten av programmet også infisert. Det er likevel ikke slik at absolutt alle verk som kommer i kontakt med programmer lisensiert under GPL, blir rammet av copyleft-klausulen. I det følgende skal jeg tolke denne klausulen med den hensikt å klarlegge copyleft-bestemmelsens materielle innhold med fokus på hvilke scenarioer som faller inn under dennes virkeområde. Dersom et scenario går fri av klausulen, trenger en ikke å frigi kildekode for hele programmet.

### 5.2 Tolkning av klausulen: Bestemmelsenes innhold

GPL er forfattet med tanke på brukerne av lisensen, og ikke for jurister. Selv om hensikten er å gjøre lisensen enkel og forståelig, blir den også mer uklar. I tillegg til å introdusere nyoppfunne begreper, bruker lisensen en del praktiske eksempler slik at den lettere kan forstås av brukerne, se f.eks. artikkel 6. Dette gjør at lisensen til tider blir ordrik og noe upresis i sin gjengivelse av de opphavsrettslige begrepene, se f.eks. begrepet ”convey” i artikkel 0 (7). Disse momentene gjør at lisensen blir vanskelig å tolke. Artikkel 5 består av to ledd hvis første ledd inneholder vilkårene der bokstav c er av særlig interesse. Her fremgår det at ”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... to *the whole of the work, and all its*

*parts*<sup>99</sup>. Dette gjelder dersom du “*convey a work based on the Program, or the modifications to produce it from the Program*”<sup>100</sup>.

### 5.2.1 “Convey”

Da ”convey” retter seg mot en form for tilgjengeliggjøring, er spørsmålet hvilke typer av tilgjengeliggjøring som vil utløse copyleft-klausulen. Det norske tilgjengeliggjøringsbegrepet omfatter spredning-, visning- og fremføringsrett. Etter GPL sin ordlyd i artikkel 0 (6) og (7), vil ”convey” omfatte ”distribution”, ”making available to the public” og andre handlinger.

#### 5.2.1.1 “Distribution”

Etter amerikansk opphavsrett 17 USC § 106 (3), faller spredningsrett inn under hovedbegrepet ”distribution”. Spredning av fysiske kopier av et program, f.eks. på en CD, vil kunne utløse copyleft. Tilsvarende gjelder etter norsk rett. Det fremgår ikke av GPL om begrepet også gjelder elektroniske eksemplarer av programmet overført via nettet, men det er klargjort gjennom amerikansk rettspraksis at slike handlinger faller inn under ”distribution”<sup>101</sup>. Etter norsk rett faller digital overføring inn under fremføringsretten<sup>102</sup>. Det at handlingen er klassifisert ulikt, har ikke betydning i denne sammenheng. Videre er det etter norsk rett tilstrekkelig at eksemplaret frembys for salg. Det må altså ikke ha skjedd en besittelsesovergang da annonsering eller tilbud på nett kan være nok. Se f.eks. C-516/13 Knoll, der EU-retten drøfter nettopp dette. På en annen side fremgår det av artikkel 5 (1) bokstav c at copyleft utløses hvis noen ”comes into possession of a copy”. Ordlyden gir uttrykk for at det faktisk må ha skjedd en overdragelse, og ikke bare fått programmet tilbudt. Det kan dog tenkes at ”possession” kan tolkes som at det inkluderer de tilfeller der man reelt sett har mulighet til å skaffe seg en kopi som i Knoll. Etersom ”convey” også inkluderer tilrådighetsstillelsesretten, kan det etter en helhetstolkning av GPL være grunner for å si at ”possession” også inkluderer det overnevnte tilfellet.

#### 5.2.1.2 “Making available to the public”

Når det kommer til ”making available to the public” er det etter amerikansk rett omdiskutert om og når en handling kan sies å omfattes av dette alternativet. En mulig slutning av

---

<sup>99</sup> Min utheving

<sup>100</sup> Min utheving

<sup>101</sup> Gue (2012) s. 107

<sup>102</sup> Rognstad (2009) s. 160. I den nye åndsverksloven er ”offentlig overføring” blitt foreslått som egen underkategori under tilgjengeliggjøringsretten, se Prop. 104 L (2016-2017) side 30 flg.

sprikende rettspraksis kan være at det er tale om ”making available to the public”, dersom noen faktisk har lastet ned programmet som er blitt lagt ut for nedlastning på nett<sup>103</sup>. I EU er det etablert at handlinger omfattes av infosoc artikkel 3 (1) dersom ”almenheten får adgang til dem på et individuelt valgt sted og tidspunkt”. Tilsvarende formulering finnes i åvl. § 2 (3). Om en tilrådighetsstiller et program på Internett slik at brukeren kan laste dette ned på et valgfritt sted og tidspunkt, er det tale om ”making available to the public”<sup>104</sup>. Det kan så spørres hva som ligger i ”almenheten”. Som adressert i kapittel 3 er det en noe ulik tilnærming til hva som regnes for allmennheten i EU og i Norge. Det er usikkert hvilket innhold en skal legge til grunn hva gjelder denne formen for ”conveying”. Ettersom dette alternativet ble tilføyd i GPLv3, kan det tale for at lisensen forsøker å fange opp alle situasjoner av tilgjengeliggjøring, og at ”making available to the public” derfor burde tolkes med dette i mente.

### 5.2.1.3 ”Other parties”

Det er bare tale om ”conveying” dersom dette skjer overfor ”other parties”, jf. artikkel 0 (7). Det vil si at ”distribution” og ”making available to the public” av et program ikke utløser copyleft om det ikke er tale om ”other parties”. Ettersom lisenstakere av GPL både kan være individer og organisasjoner, jf. artikkel 0 (3), må det kunne utledes at en kan spre eksemplarer og gjøre det tilgjengelig for en selv, f.eks. spre eksemplarer til flere enheter en måtte disponere, eller innad i en organisasjon. I USA har de noe som kalles ”the limited publication doctrine”. Dersom en gjør noe tilgjengelig for en begrenset krets, vil det ikke være snakk om ”distribution” overfor allmennheten. For at det skal være tale om en begrenset krets, må verket være ”distributed to (1) a select group, (2) for a limited purpose, and (3) with restrictions on the recipients’ ability to distribute, reproduce, sell and diffuse the copyrighted works”<sup>105</sup>. Det er usikkert hvordan dette kan brukes på GPL, men doktrinen gir i alle fall noen veiledende momenter som kan vurderes. Etter norsk rett er det kun tilgjengeliggjøring overfor ”allmennheten”<sup>106</sup> som faller inn under opphaverens enerett.

---

<sup>103</sup> Se f.eks. Ford Motor Co. V. Smmit Motor Prods., Inc., 930 F. 2d 277, 299 (3d Cir. 1991), London-Sire Records, Inc. v. Doe, 542 F. Supp. 2d 153, 169 (D. Mass. 2008), og Nat’l Car Rental Sys., Inc. v. Computer Assocs. Int’l, Inc., 991 F. 2d 426, 430-35 (8<sup>th</sup> Cir. 1993). Se også Gue (2012) s. 109

<sup>104</sup> Pila (2016) s. 319

<sup>105</sup> Gue (2012) s. 112. Se også White v. Kimmell, 193 F. 2d 744, 746-47 (9<sup>th</sup> Cir. 1952).

<sup>106</sup> Se kapittel 3.4

Særlig kan det være interessant å se på hva som regnes for å være tilgjengeliggjøring innad i en ”organization” etter artikkel 0 (3). I den forbindelse er det nødvendig å se på hva som er å regnes som en og samme organisasjon. Ettersom det i praksis er mange kommersielle aktører som benytter seg av åpen kildeprogrammer, kan dette komme på spissen. For eksempel bruker gigantselskapet Google® Goobuntu internt, hvilket er lisensiert under GPL<sup>107</sup>. Dette programmet har de modifisert og tilpasset deres behov. Ettersom det kun brukes internt, trenger de ikke å lisensiere den endrede versjonen under GPL.

Dersom det kun er et enkelt firma som ikke er eid av andre juridiske personer, må det kunne sies at tilgjengeliggjøring innad i dette firmaet ikke kan regnes som ”convey” i lisensens forstand. En slik tolkning stemmer overens med lisensens ordlyd og formål. Vanskeligere blir det dersom et firma er en del av et konsern. Selskapsrettslig vil firmaene som inngår i et konsern være selvstendige, mens de skatterettslig ses på som en økonomisk enhet. Vi ser for oss at firma A og B eies av morselskapet C. Vil det være snakk om tilgjengeliggjøring innad i en organisasjon dersom firma A gjør programmet tilgjengelig for firma B eller for morselskapet C? Denne problemstillingen gir lisensen ingen svar på. Formålet med lisensen er å gi lisenstakeren bruksfrihet. Dersom en ønsker å bruke åpen kildekodeprogrammer til å utvikle programmet til eget bruk, vil dette falle inn under friheten som lisensen gir. Hvis firma A har utviklet en ny versjon av et GPL-lisensiert program som f.eks. skal effektivisere regnskapsrutiner innad i firmaet, kan det være grunner for å si at de kan gjøre dette tilgjengelig for firma B med tilsvarende formål. På en annen side er det forklart i GPL FAQ at et firma kan endre og installere versjoner av programmet ”through its own facilities”<sup>108</sup>. Ut i fra dette kan det virke som at tilgjengeliggjøring overfor firma B, som mest sannsynlig ikke deler intern infrastruktur med firma A, er tilgjengeliggjøring utenfor organisasjonen. Hva som menes med ”organization” har altså ikke et klart svar.

#### *5.2.1.4 Særlige problemstillinger*

I forbindelse med tilgjengeliggjøring, dukker det opp to praktiske tilfeller: cloud-computing og outsourcing.

---

<sup>107</sup> Shemtov (2013) s. 230

<sup>108</sup> <https://www.gnu.org/licenses/gpl-faq.html#InternalDistribution>

Cloud-computing er en distribusjonsmodell der leverandørene tilbyr datarelaterte tjenester via nettet<sup>109</sup>. Brukere av ”skyen” betaler for tilgangen til tjenestene og slipper å bruke ressurser på å ha sin egen datainfrastruktur. En leier altså tilgang til ulike datatjenester som kan brukes fra skyen. Et eksempel på en tilbyder av slike tjenester er IBM®. En slik tjeneste kan være software as a service (SaaS). Leverandørene lager en digital plattform der brukeren kan anvende programmet uten at de mottar en kopi av programmet eller får dette ”made available”. De får kun mulighet til å bruke det som en tjeneste. En annen variant er application as a service (ASP), som f.eks. Gmail. Ettersom brukeren aldri får overført programmet, vil slike skyløsninger tilsynelatende ikke utløse copyleft da det ikke er snakk om tilgjengeliggjøring etter GPL<sup>110</sup>. Dette kommer også til uttrykk i artikkel 0 (7) der det står at ”mere interaction with a user through a computer network, with no transfer of a copy, is not conveying”. Videre kan en tenke seg et annet scenario. Firma A bruker en skyløsning fra firma B som ”host” for sitt program. Vil en slik handling være tilgjengeliggjøring? Under limited publication-doktrinen kan man tenke seg at det ikke vil være tilgjengeliggjøring da det er gjort overfor en begrenset krets (ansatte i firma B), med en begrenset hensikt, og sannsynligvis med begrensninger for hva firma B kan gjøre med programmet. Etter norsk rett faller dette mest sannsynlig utenfor den snevre krets av familie og venner, men firma B får nok ikke ”stilt verket til rådighet på en slik måte at den enkelte selv kan velge tid og sted for tilgang” til programmet, jf. åvl. § 2 (3). I EU-retten kan det tenkes en tilsvarende vurdering. I vårt scenario hoster firma B programmet i sin sky på vegne av firma A. Dersom de i tillegg til å hoste programmet, også får i oppdrag å endre eller forbedre programmet, vil dette bringe oss videre til outsourcing.

Outsourcing er når ”et foretak setter deler av sin produksjon til underleverandører”<sup>111</sup>. Scenarioet over vil derfor være outsourcing. Spørsmålet er om outsourcing utløser copyleft. Her kan det være tale om tilgjengeliggjøring i flere ledd: (1) fra firma A til firma B, (2) fra firma B til andre underleverandører, og (3) frem og tilbake mellom firma A og B i utviklingsprosessen. GPL gir noe veiledning i disse spørsmålene. Artikkel 2 (2) sier at det ikke er tale om tilgjengeliggjøring dersom du gjør dette overfor andre hvis deres eneste hensikt er å ”make modifications exclusively for you”. Det er gitt at firma B må gjøre dette

---

<sup>109</sup> Ranger (2018)

<sup>110</sup> For å sette smutthullet har det blitt opprettet en annen lisens der slike skyløsninger også er å anse som tilgjengeliggjøring, se GNU Affero GPL <https://www.gnu.org/licenses/agpl-3.0.en.html>

<sup>111</sup> Vikøren (2018)

”exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you”. Det står ikke eksplisitt at dette dekker outsourcing, men ettersom GPL er en praktisk lisens, kan det tenkes at denne ordlyden er ment til å dekke disse scenarioene. Går en til FAQ sier de i midlertid at ”providing copies to contractors for use off-site is distribution”<sup>112</sup>. Da firmaer som det outsources til som oftest ikke er on-site, vil outsourcing etter FAQ-en være tilgjengeliggjøring. Det må dog være slik at lisensens ordlyd går foran en FAQ. Mest sannsynlig vil ikke outsourcing som tilfredsstillende kriteriene i artikkel 2 være tilgjengeliggjøring som utløser copyleft.

### 5.2.2 ”... A work based on the Program”

Det er kun tilgjengeliggjøring av verk som er ”based on” det GPL-lisensierte programmet som vil utløse copyleft. Ut i fra ordlyden alene vil ”based on” kunne forstås som ”basert på” eller ”inspirert av”. En slik tolkning vil kunne føre til at verk som kun er idémessig inspirert av det GPL-lisensierte programmet rammes av bestemmelsen. Imidlertid følger det av artikkel 0 (4), at et verk vil være ”based on” det tidligere verket dersom en ”copy from or adapt all or part of the work”. 17 U.S.C § 101 og § 103 taler om ”derivative works”. Som etter norsk rett vil det avledede verket basere seg på et eksisterende åndsverk, jf. åvl. § 4 (2). Ettersom bearbeiderens råderett over det avledede verket er avhengig av opphaveren, er det nødvendig å se på hva som regnes for å være et verk ”based on” programmet. Dersom det ikke er et slikt verk, vil ikke bearbeiderens råderett være betinget av samtykke fra opphaveren, og derfor heller ikke underlagt copyleft da dette vil falle utenfor opphaverens enerettsbeføyelser.

Dersom en endrer ved å tilføye eller slette linjer med koder fra programmet, vil dette være å bearbeide programmet på en slik måte som utløser copyleft. Ut i fra ordlyden i artikkel 0 (4), vil inkorporering av det opprinnelige programmet, enten hele eller deler, inn i et annet program, falle inn under bestemmelsenes virkeområde. Her er det viktig å merke seg at det kun gjelder inkorporering av verksdeler med verkshøyde. Verksdeler uten opphavsrettslig beskyttelse kan fritt benyttes uten at dette utløser copyleft. Etter amerikansk rett finnes det også unntak i form av *scènes à faire*-doktrinen<sup>113</sup>, så vel som fair use. I Norge og i EU er

---

<sup>112</sup> <https://www.gnu.org/licenses/gpl-faq.html#InternalDistribution>

<sup>113</sup> ”Scènes à faire... applies to expression that is so associated with a particular genre, motif, or idea that one is compelled to use such expression... such scenes are unpredictable by copyright law”, se *Tetris Holdings v. Xio Interactive* s. 16



loven mer rigid, og tilsvarende unntak finnes ikke. Når det kommer til vurdering av bearbeidelser vises det til fremstillingen i kapittel 3.

Hva gjelder inkorporering av verksdeler dukker det opp noen særskilte spørsmål knyttet til statisk og dynamisk linking.

”Static linking means embedding a subcomponent’s code into a larger program-literally, copying and pasting lines of code from one program to another. This is akin to copying chapters out of Lewis Carroll’s *Through the Looking Glass* into another book. Dynamic linking means referencing object code that separately exist. This would be akin to mentioning Alice and expecting readers to consult the other work.”<sup>114</sup>

Som utgangspunkt kan det tenkes at statisk linking vil kunne utløse copyleft ettersom en del av et verk blir kopiert og ”limt inn” i et annet. Dynamisk linking inkorporerer ingen konkrete verksdeler på en permanent måte inn i det andre programmet, snarere kan en si at programmene kommuniserer med hverandre på avstand. Stallman har blant annet uttalt at typen av linking ikke er avgjørende da en heller burde se til hvorvidt programmene er ment til å fungere med hverandre slik at en kan behandle dem som et samlet program<sup>115</sup>. Dette synspunktet kan på ingen måte utledes av ordlyden i lisensen, så løsningen fremstår som usikker. Dynamisk linking brukes ofte i forbindelse med programbiblioteker. Vi ser for oss to scenarier: (1) leverandør A utvikler et proprietært program som linkes til et GPL-lisensiert bibliotek. Må leverandøren frigi kildekode for det proprietære programmet? Og omvendt: (2) Kan et GPL-lisensiert program linke til et proprietært bibliotek? FAQ-en svarer bekreftende på det første spørsmålet<sup>116</sup>. Hva gjelder det andre, kommer det an på hvorvidt det proprietære biblioteket faller inn under systembibliotek-unntaket som fremgår av artikkel 1 tredje ledd. En kan også dynamisk linke til plug-ins. Plug-ins er en tilleggskomponent som utvider funksjonaliteten til et eksisterende program. For eksempel er Adobe® Flash Player en plug-in som utvider nettleserens funksjonalitet. Spørsmålet blir om en leverandør må frigi kildekode for sine plug-ins dersom de linker dette til et GPL-lisensiert program. Lisensen gir ingen svar på dette, og FAQ-en er fragmentert hva gjelder plug-ins. Det kan virke som at dersom

---

<sup>114</sup> Gue (2012)

<sup>115</sup> Stallmann i en tale fra 2006, <https://fsfe.org/campaigns/gplv3/torino-rms-transcript.en.html>

<sup>116</sup> <https://www.gnu.org/licenses/gpl-faq.html#IfLibraryIsGPL>

hovedprogrammet og plug-inen anses for å være et kombinert program, må en frigi kildekode for plug-inen. Dersom de anses for å være separate programmer, utløses ikke copyleft<sup>117</sup>. Wordpress® og Drupal, hvilket er to store CMS-tilbydere, gir uttrykk for at plug-ins er så nært tilknyttet hovedprogrammet at disse bør GPL-lisensieres<sup>118</sup>. På en annen side kan nok ikke en GPL-lisensiert plug-in utløse copyleft for et proprietært hovedprogram da hovedprogrammet ikke vil være et avledet verk i forhold til plug-inen ettersom hovedprogrammet vil ha eksistert før plug-inen.

Hvordan spørsmål knyttet til linking vil løses i praksis, er vanskelig å si. En kan foreta en opphavsrettslig krenkelsesvurdering og vurdere om det nye verket ville ha blitt ansett for å gjøre inngrep i originalopphaverens rettigheter. Dersom dette besvares bekreftende, kan det være grunner for å si at det nye verket er et avledet verk som kan utløse copyleft.

### 5.2.3 "The entire work, as a whole"

Dersom en har tilgjengeliggjort et bearbeidet verk, må en lisensiere "the entire work, as a whole" under GPL. Spørsmålet blir så hva som regnes for å være hele verket. Artikkel 5 (1) bokstav c sier at copyleft vil gjelde "whole of the work, and all its parts, regardless of how they are packaged". Ved første øyekast virker det som at bestemmelsen er vidt, og at alle verk som kommer i kontakt med et GPL-lisensiert verk, omfattes av denne. Annet ledd gir i midlertid et unntak for "aggregates". Dette defineres som en samling av andre separate og uavhengige verk som ikke i sin natur er en forlengelse av det GPL-lisensierte programmet, og ei heller knyttet til det som en del av et større program. I prinsippet vil det si at dersom to uavhengige verk distribueres på samme CD, kan et program være GPL-lisensiert uten at det andre blir infisert. Dette bunner i at opphaveren av et verk ikke har råderett over et annet selvstendig verk, jf. åvl. § 5 (1). Bestemmelsen sier i midlertid at også slike selvstendige verk kan rammes av copyleft. Det kan derfor settes opp tre scenarier: (1) A endrer og inkorporerer et GPL-lisensiert program inn i sitt hovedprogram. Etter hovedregelen i bokstav c, må hele programmet lisensieres under GPL. (2) A setter sammen to separate og uavhengige verk, der verk 1 er GPL-lisensiert og verk 2 kan sies å være en forlengelse av verk 1. Etter ordlyden i annet ledd vil slik sammenstilling ikke falle inn under aggregate-unntaket. Verk 2 må derfor lisensieres under GPL. (3) A setter sammen to verk på en slik måte som faller inn under aggregate-unntaket. Verk 2 går fri av GPL.

---

<sup>117</sup> <https://www.gnu.org/licenses/gpl-faq.html#GPLAndPlugins>

<sup>118</sup> Se <https://www.drupal.org/about/licensing#project-licensing> og <https://wordpress.org/about/license/>

Det avgjørende blir å vurdere hva som regnes for å være et virkelig selvstendig program, og hva som er et selvstendig program som likevel er knyttet til det GPL-lisensierte programmet. FAQ-en gir uttrykk for at det ikke er noe klart svar, men at en må se på hvordan programmene kommuniserer med hverandre. Dersom modulene er i samme exe-fil er de kombinerte programmer som ikke er aggregates. Motsetningsvis vil sockets, pipes og command-line arguments være kommunikasjonsformer som vanligvis benyttes mellom to separate programmer. Annerledes kan det bli dersom programmene utveksler informasjon som er intime, f.eks. interne datastrukturer. Da kan det være at programmene anses for å være satt sammen som et stort program<sup>119</sup>. Disse synspunktene er FSF sine egne, og vil ikke ha avgjørende vekt. GPL gjør unntak for systembiblioteker slik at en ikke må frigi kildekoden for disse selv om de distribueres sammen gitt at det faller inn under definisjonen i artikkel 1 (3) og (4). Ettersom systembiblioteker har standardiserte og generelle funksjoner, kan det muligens utledes at dersom programmene kommuniserer med hverandre på en mer tilpasset og kompleks måte, vil de lettere kunne anses for å være kombinerte programmer. Dette kommer på spissen når det kommer til linking. Artikkel 1 (4) gir uttrykk for at dynamisk linkede subprogrammer vil være en del av et sammenstilt verk dersom informasjonsutvekslingen og kontrollflyten er ”intimate”. Hva som ligger i dette er usikkert. GPL adresserer heller ikke konsekvensen av at en kombinerer GPL-lisensiert program med kjernemoduler. Dersom en finner kommunikasjonen mellom kjernen og kjernemodulen for å være intim, kan det være at en proprietær kjernemodul blir infisert av en GPL-lisensiert kjerne som f.eks. Linux.

---

<sup>119</sup> <https://www.gnu.org/licenses/gpl-faq.html#MereAggregation>

## **6 Sanksjoner ved overtredelse av copyleft-klausulen**

### **6.1 Innledning**

Dersom en begår en handling som er i strid med copyleft-klausulen, vil det være en overtredelse av lisensen. Spørsmålet blir hvilke sanksjoner misligholdet vil utløse. Noen tilfeller vil være alminnelige kontraktsbrudd, andre vil være opphavsrettskrenkelser. Lisensen som kontrakt gjelder *inter partes*, dvs. at vilkårene kun retter seg mot lisenstakerne, mens opphavsretten gjelder *erga omnes*, altså overfor enhver. Om overtredelsen er det ene eller det andre har ikke bare betydning hva gjelder sanksjoner, men også håndhevelsesmuligheten. Opphaveren kan gjøre krav gjeldende overfor alle, mens lisensgiver kun kan gjøre krav gjeldende overfor lisenstakeren. Denne forskjellen gjør at sanksjonsmulighetene etter opphavsretten og kontraktsretten varierer. Først må det klargjøres hvorvidt overtredelsen har opphavsrettslige eller kontraktsrettslige følger, for så å se på forskjellene mellom sanksjonsmulighetene.

### **6.2 Opphavsrettslig og kontraktsrettslig virkende vilkår i copyleft-klausulen**

#### **6.2.1 Opphavsrettslig virkende vilkår**

Bearbeideren vil få opphavsretten til den bearbejdede delen av programmet, men vil forøvrig være avhengig av den opprinnelige opphaver dersom vedkommende ønsker å tilgjengeliggjøre verket. Dette fremgår av avl. § 4 (2), og bearbejderens rett springer derfor ut av opphavsretten. Opphaveren har anledning til å overdra sine rettigheter helt eller delvis, og må i den anledning kunne stille nærmere vilkår til hvilke rettigheter det er snakk om, og hvordan disse kan benyttes. Spørsmålet blir om opprinnelig opphaver kan stille vilkår for den delen av verket som kun bearbejderen har rettigheter til. Ettersom verket ikke kan tilgjengeliggjøres uavhengig av det opprinnelige verket, må dette besvares bekræftende. Dersom bearbejderen ikke følger de vilkår som opphaveren har satt for tilgjengeliggjøringen, vil det innebære et inngrep i opphavsretten.

#### **6.2.2 Kontraktsrettslig virkende vilkår**

Copyleft-klausulen gjelder som nevnt også visse separate og uavhengige verk. Ettersom den opprinnelige opphaverens rettigheter ikke strekker så langt som å gjelde for andres åndsverk, vil vilkårene ikke bunne i opphavsretten. Det må derfor sies at overtredelse av vilkår som gjelder disse verkene, vil være rent kontraktrettslige.

### 6.3 Opphavsrettslige og kontraktsrettslige sanksjoner

Sanksjonsapparatene etter åndsverksloven og kontraktsretten er nokså ulike. Opphaveren kan etter åndsverksloven gjøre sine krav gjeldende for alle som krenker dennes opphavsrettigheter, det vil si både lisenstaker og tredjeparter. Ved brudd på de opphavsrettslig relevante vilkårene i lisensen, vil opphaver kunne ha erstatning- og berikelseskrav, jf. åvl. § 55. I visse tilfeller kan det også være snakk om strafferettslige sanksjoner, jf. åvl. § 54. En annen sentral sanksjon, er at opphaver kan kreve nedleggelse av forbud mot bruk, dvs. stanse den krenkende handlingen.

Et rent kontraktsbrudd medfører en automatisk og umiddelbar avslutning av lisensavtalen etter GPLv3 artikkel 8 (1). Lisenstaker kan under visse forutsetninger ”reaktivere” lisensavtalen dersom en ”cease all violation of this License”, jf. artikkel 8 (2). Hva gjelder copyleft-klausulen betyr dette i praksis at dersom lisenstaker i etterkant følger vilkårene og frigir kildekode for de relevante verkene, blir lisensavtalen ”reaktivert”.

Spørsmålet blir i denne sammenheng om lisensgiver kan kreve å få kildekode utlevert, altså fremme et krav om naturaloppfyllelse i de tilfeller der lisenstaker ikke vil gjøre dette på eget initiativ. Det kan da spørres om den obligasjonsrettslige bakgrunnsretten kommer til anvendelse. Ettersom kjøpsloven ikke dekker kontraktsforhold om rene bruksrettigheter, må en i så fall ta utgangspunkt i de alminnelige obligasjonsrettslige prinsippene. Hva gjelder naturaloppfyllelse er det klare utgangspunktet at kreditor kan kreve å få kontrakten gjennomført *in natura*<sup>120</sup>. Lisensgiver av åpen kildekodeprogrammer vil ha særlig interesse i naturaloppfyllelse da tilgang til kildekode er nødvendig for å innfri det filosofiske grunnlaget om brukerfrihet. Kildekode vil derfor kunne anses som en ikke-fungibel ytelse. Lisensgiver vil også kunne kreve erstatning i kontraktsforhold hvis vilkår i stor grad tilsvarer den alminnelige erstatningsretten. Hvorvidt bakgrunnsretten kommer til anvendelse, beror på om og hvordan misligholdsstrafkjonene er regulert i kontrakten. Dersom lisensen uttømmende angir de aktuelle sanksjonene, vil dette regulere kontraktsforholdet. Dersom dette ikke kan utledes av lisensen, vil bakgrunnsretten være supplerende.

---

<sup>120</sup> Hagstrøm (2013) s. 379

De overnevnte sanksjoner vil være de mest sentrale og praktiske hva gjelder brudd på copyleft. Jeg skal foreta en sammenligning av erstatning som sanksjon i og utenfor kontraktsforhold, samt drøfte hvorvidt lisensgiver kan kreve naturaloppfyllelse etter norsk rett.

### 6.3.1 Erstatning

Erstatning som opphavsrettslig sanksjon finnes i åvl. § 55 (1). Bestemmelsen viser imidlertid til de alminnelige erstatningsregler, hvilket innebærer at opphaver vil ha et erstatningskrav dersom det foreligger culpa-ansvar. Det vil foreligge et slikt ansvar dersom (1) skadevolder har handlet culpøst (utvist skyld), (2) det har oppstått et økonomisk tap, og (3) det foreligger adekvat årsakssammenheng mellom handling og skade. I utgangspunktet er simpel culpa (uaktsomhet) nok til å konstatere skyld. Hva gjelder erstatning i kontraktsforhold er grunnvilkårene de samme som i den alminnelige erstatningsretten, selv om hensynene bak reglene er ulike. Erstatning for opphavskrenkelses har en gjenopprettelses- og prevensjonsfunksjon der opphaver skal bli økonomisk kompensert for den skadevoldende handling, samt forhindre ytterligere opphavskrenkelses. I kontraktsretten vil erstatning være en misligholdsbeføyelse som skal presse partene til å oppfylle sine obligasjoner i henhold til kontrakten<sup>121</sup>. Videre må det nevnes at erstatning som kontraktsrettslig misligholdsbeføyelse kan reguleres ved avtale, se f.eks. GPLv3 artikkel 16, mens erstatning etter åndsverksloven er preseptorisk. Det nevnes i denne sammenheng at artikkel 16 utelukker erstatningskrav som kommer *fra* lisenstaker. Lisensen utelukker altså ikke at lisensgiver kan fremme slike krav *mot* lisenstaker.

En annen forskjell er at erstatning i opphavsretten også kan kreves for skader av ikke-økonomisk art, mens erstatning som misligholdsbeføyelse kun gjelder formuestap. Opphaver kan kreve erstatning for ikke-økonomisk tap, jf. åvl. § 55 (1) annet punktum dersom det er utvist forsett eller grov uaktsomhet. Hva gjelder krenkelse av åpen kildeprogrammer, kan det være vanskelig å si at det foreligger et økonomisk tap da programmene ofte er vederlagsfrie. Spørsmålet blir om opphaver av åpen kildeprogrammer kun kan kreve erstatning etter særregelen i § 55 (1) annet punktum med skjerpet culpa-vurdering, og ikke alminnelig erstatningsrett som forutsetter et økonomisk tap og simpel uaktsomhet. På en side kan en tenke seg at lisenstaker som benytter et åpent og gratis program får en eller annen form for

---

<sup>121</sup> Ibid s. 466

begunstigelse. Lisenstaker kan ha sluppet å betale for en proprietær variant av programmet, eller bespart seg merkostnader de hadde pådratt dersom de hadde utviklet programmet på egenhånd. Utnyttelsen vil derfor ikke ha vært gratis i vid forstand. Det økonomiske tapet hadde i så fall kunne vært fastsatt til kostnadene lisenstaker hadde pådratt seg dersom de hadde benyttet andre løsninger. Det nevnes også at det i teorien er gitt uttrykk for at det kan oppstilles en ulovfestet vederlagsregel der opphaver kan kreve vederlag selv om det ikke påvises et økonomisk tap<sup>122</sup>. Opphaver kan etter åndsverksloven også kreve restitusjon av berikelse etter § 55 (2) uavhengig av skyld. Størrelsen på kravet vil være nettofortjenesten av den skadevoldende handling. Dette innebærer at dersom det kun er brukt noen verksdeler, er det nettofortjenesten av denne delen som skal kompenseres.

Erstatning er en økonomisk kompensasjon, og for åpen kildekodeprogrammer vil ikke økonomisk godtgjørelse alltid være tilstrekkelig. For opphaver vil ikke gjenopprettings- og prevensjonsfunksjonen slå til ved erstatning i form av penger, da det avgjørende for opphaver, og andre brukere, vil være tilgang til kildekoden. På dette grunnlag vil naturaloppfyllelse være en bedre sanksjon. Dette gjelder dog bare for kontraktsrettslige forhold og ikke rene opphavsrettskrenkelser.

### 6.3.2 Naturaloppfyllelse

Naturaloppfyllelse som kontraktsrettslig sanksjon for brudd på copyleft-klausulen er lite behandlet i norsk rett, og det er usikkert hvorvidt et slikt krav hadde fått medhold dersom det hadde blitt prøvd for domstolen. Selv om slike saker ofte blir forlikt, er det noen som er kommet opp for domstolene i Tyskland og USA. Dommene har liten rettskildemessig vekt, men illustrerer likevel mulige utfall av krav på naturaloppfyllelse. Tysk rett, som norsk rett, bygger på sivilrett, mens den amerikanske rettstradisjonen bygger på common law.

I en rekke distriktsdommer fra Tyskland, har retten konkludert med at GPL er en håndhevbar lisens der lisenstaker må følge vilkårene i GPL. Dette kommer til uttrykk i blant annet *B.V. vs. University of Germany*<sup>123</sup> og *Harald Welte vs. D-Link Deutschland GmbH*<sup>124</sup>. Selv om dommene ikke fastsetter at saksøkte må utlevere kildekoden, gir de i alle fall indikasjoner på at det er et vilkår som saksøkte må imøtekomme. I førstnevnte dom ble det også diskutert

---

<sup>122</sup> Rognstad (2009) s. 403. Se også Rt. 2009 s. 1568.

<sup>123</sup> Regional domstol Halle, saksnummer 4 O 133/15

<sup>124</sup> Regional domstol Frankfurt, saksnummer 2-6 O 224/06

hvorvidt artikkel 8 i GPLv3 skulle tolkes slik at en kan unngå sanksjoner dersom en ”cease all violations”. Dersom en legger dette til grunn, vil overtredelse av lisensen ikke få noen store konsekvenser for lisenstaker. Uten reelle sanksjonsmuligheter vil det være vanskelig å håndheve vilkårene i lisensen. Domstolen kom derfor frem til at det ikke skulle tolkes slik at lisensgiver sa fra seg sanksjonsmulighetene. I denne saken ble lisenstaker ilagt en bot.

I USA har det versert flere saker som går på overtredelse av copyleft-klausulen. I *Versata vs. Ameriprise*<sup>125</sup> ble Ameriprise saksøkt av Versata for brudd på en inter partes lisensavtale. Ameriprise fant ut at Versata hadde brukt GPL-lisensiert kode uten å frigi kildekode, og gikk til motsøksmål. Denne saken ble forlikt før domstolen fikk tatt stilling til utfallet. I *Artifex vs. Hancom*<sup>126</sup>, ble Hancom saksøkt for ikke å ha frigitt kildekode i tråd med GPLv3 da de benyttet seg av koden til Artifex som var lisensiert under overnevnte. Domstolen kom frem til at copyleft-klausulen er et kontraktsrettslig vilkår hvilket innebærer at det i teorien kan kreves naturaloppfyllelse. I amerikansk rett er det som alminnelig regel ikke adgang til å kreve ”specific performance”. Unntaket er når dette vil være særlig urimelig overfor kreditor<sup>127</sup>. Dommeren uttaler så at GPLv3 artikkel 8 ikke utelukker naturaloppfyllelse, men drøfter ikke dette som sanksjonsmulighet nærmere. Også denne saken ble forlikt.

Spørsmålet er så hvordan dette kan tenkes løst etter norsk rett. Ettersom retten til naturaloppfyllelse er hovedregelen snarere enn unntaket, kan det tenkes at det er større mulighet for å få medhold i et slikt krav etter norsk rett enn etter amerikansk rett. Det kan spørres hvorvidt artikkel 8 skal forstås som en uttømmende regulering av mislighold der avslutning av lisensavtalen er eneste sanksjon. Dette kommer i midlertid ikke eksplisitt frem i lisensen, og lisensgiver vil ha interesse i at lisenstaker innfrir vilkårene. Uten pressmidler i form av håndhevbare sanksjoner fra lisensgiver side, vil muligheten til å stoppe krenkelser være en illusjon. Det vil ikke være tilstrekkelig å gi notis til lisenstaker om de må opphøre den krenkende handlingen da dette lett kan overses. Artikkel 8 må derfor kunne suppleres med bakgrunnsretten.

---

<sup>125</sup> Case no.: D-1-GN-12-003588; 53rd Judicial District Court of Travis County, Texas

<sup>126</sup> Case no.: 16-cv-06982-JSC US District Court of Northern District of California

<sup>127</sup> Hagstrøm (2013) s. 377



Et sentralt hensyn som taler for at lisensgiver burde få medhold i sitt krav, er at økonomisk kompensasjon ikke vil bøte på skaden som er skjedd. Forutsetningen for distribuering av åpne kildekodeprogrammer er at kildekode skal være tilgjengelig for tredjeparter slik at deres bruksfrihet blir i varetatt. Fri informasjonsflyt og deling av kunnskap er essensiell for slike lisenser, og erstatning alene vil ikke gjenopprette tapt kunnskapsutveksling. Lisensgiver vil derfor ha særlig interesse i at lisenstaker frigir kildekode. Dersom det gis oppfyllelsedom om ytelse in natura, vil det kunne ha en preventiv effekt slik at lisenstakere som vurderer å bruke GPL-lisensierte programmer, blir mer oppmerksom på konsekvensene. Hensyn som taler i mot en slik løsning kan være at det oppstår et misforhold i ytelse og motytelse. Dersom en kun har brukt noen få linjer med kode som er GPL-lisensiert og inkorporert det i et omfattende og kostbart program, kan det etter en interesseavveining fremstå som ubalansert å måtte frigi kildekode for hele verket. Skulle en komme frem til at det i det konkrete tilfellet er adgang til å kreve naturaloppfyllelse, må en i så fall også foreta en interesseavveining av lisensgivers behov for ytelse in natura opp mot hvor urimelige dette vil være overfor lisenstaker.

Alternativt vil det heller ikke kunne kreves naturaloppfyllelse dersom det er umulig for lisenstaker å innfri forpliktelsene etter GPL. Umulighet kan være fysisk umulighet, men også rettslig umulighet, f.eks. ved at det vil være ulovlig eller at en ikke har hjemmel til å innfri forpliktelsen. Dette kan tenkes å være tilfellet dersom en benytter lisenser som er inkompatible. For eksempel er hovedprogrammet lisensiert under GPL, mens det linkede biblioteket er lisensiert under en inkompatibel lisens. Dersom lisenstaker er forpliktet under en lisens til ikke å frigi kildekode, vil det kunne argumenteres for at det foreligger rettslig umulighet for lisenstaker å samtidig innfri sine forpliktelser etter GPL. På en annen side er det ikke tale om rettslig umulighet i form av at det er ulovlig å bryte den inkompatible lisensen da det kun vil være tale om mislighold<sup>128</sup>. Det er usikkert hvorvidt motstridende kontraktsforpliktelser i slike tilfeller vil kunne anses for å være umulighet i obligasjonsrettsligforstand.

---

<sup>128</sup> Hagstrøm (2013) s. 384

## 7 Avslutning

Utformingen av GNU GPL versjon 3 skaper en rekke usikkerhetsmomenter da ordlyden i lisensen er uklar på flere punkter, særlig hva gjelder innholdet i copyleft-klausulen. Et av vilkårene som utløser copyleft, er at programmet må være "conveyed". Begrepet "convey" omfatter blant annet handlinger som "distribution" og "making available to the public". Særlig interessant er spørsmålene om hvorvidt og i hvilken utstrekning cloud-computing og outsourcing faller inn under disse kategoriene. Det andre vilkåret som utløser copyleft, går ut på at programmet må være "a work based on the program". Hva som ligger i dette, er heller ikke klart. Her dukker det opp spørsmål knyttet til inkorporering av verksdeler ved linking. Dersom det er tale om "conveying" av et "work based on the program", må "the entire work, as a whole" lisensieres under GPL, hvilket innebærer at en må gjøre kildekoden tilgjengelig for alle. Det er derfor viktig å klarlegge hva som regnes for å være "the entire work, as a whole". Ordlyden i lisensen gir uttrykk for at bearbejdede verk, samt uavhengige og selvstendige verk kan falle inn under bestemmelsens virkeområde. Dette beror på om det uavhengige og selvstendige verket kan anses for å være en forlengelse av det GPL-lisensierte programmet. Hvor denne grensen går er usikkert. Når det kommer til åpen kildekodeproblematikken, er det få rettskilder å lene seg på, og grensene for hvilke tilfeller som faller inn under GPL, og hvilke som går fri er uavklart.

Sanksjonsapparatene for krenkelse etter åndsverksloven og mislighold av kontrakt er ulike. Visse vilkår i copyleft-klausulen vil ha opphavsrettslige virkninger, mens andre vil ha kontraktsrettslige virkninger. Bestemmelser i GPL som gjelder verk opphaveren etter åndsverksloven ikke har råderett over, vil være en kontraktsrettslig regulering hvis overtredelse vil utløse misligholdsbeføyelser. Opphavsrettens sanksjonssystem vil ikke anvendes for disse tilfellene. Erstatning etter åndsverksloven har ulikt formål og er begrunnet i andre hensyn enn erstatning som misligholdsbeføyelse. Erstatning etter åndsverksloven kan også gis der det ikke foreligger et økonomisk tap, mens erstatning ved mislighold av kontrakt forutsetter et formuestap. Det er gode grunner for at naturaloppfyllelse burde kunne kreves ved mislighold av copyleft da lisensgiver vil ha stor interesse i at kildekoden faktisk utleveres slik at lisensens formål blir oppfylt. En ren økonomisk erstatning vil ikke være tilstrekkelig i dette tilfellet.

Ettersom det råder usikkerhet rundt innholdet i GPL, vil lisenstaker være tjent med å utarbeide en åpen kildekodestrategi. Dette innebærer at lisenstaker, dersom dette er et firma, innfører gode rutiner ved utvikling av programvarer. Virksomheten burde vurdere hvilke risikoer firmaet tar ved å bruke åpen kildekode, foreta juridiske vurderinger samt ta stilling til hvordan åpen kildekode faktisk skal brukes i virksomheten. Dersom det bearbeidede programmet skal brukes internt, må en passe på at det ikke spres utenfor virksomheten. Dersom en bruker åpen kildekode i utvikling av egne programmer, må en se til at lisensvilkårene blir overholdt. Typisk er det at virksomheter bruker en rekke ulike lisenser, som f.eks. Instagram®. I slike tilfeller vil det være viktig å ikke kombinere lisenser som ikke er kompatible. Det vil også være hensiktsmessig for virksomheten å utarbeide lister over ”godkjente” åpen kildekodeprogrammer som ikke er lisensiert under lisenser med vilkår en ikke ønsker å innfri.

## Litteraturliste

### Norske lover

- 1814 Kongeriket Norges Grunnlov av 17 Mai 1814
- 1918 Lov om avslutning av avtaler, om fullmakt og om ugyldige viljeserklæringer (avtaleloven) av 31 mai 1918 nr. 4
- 1961 Lov om opphavsrett til åndsverk m.v. (åndsverkloven) av 12 mai 1961 nr. 2
- 1992 Lov om gjennomføring i norsk rett av hoveddelen i avtale om Det Europeiske økonomiske samarbeidsområde (EØS) mv.v. (EØS-loven) av 27 nov 1992 nr. 109
- 2003 Lov om visse sider av elektronisk handel og andre informasjonssamfunnstjenester (ehandelsloven) av 23 mai 2003 nr. 35
- 2005 Lov om mekling og rettergang i sivile tvister (tvisteloven) av 17 juni 2005 nr. 90

### Forarbeider

- Innst. O. Nr. XI (1960-61) Innstilling fra kirke- og undervisningskomiteen om lov om opphavsrett til åndsverk.
- NOU 1986:18 Opphavsrett og edb.
- Ot. Prp. Nr. 26 (1959-90) om lov om opphavsrett til åndsverk
- Ot. Prp. Nr. 33 (1989-90) om lov om endringer åndsverksloven (opphavsrett og edb)
- Ot. Prp. Nr. 89 (2008-09) Om lov om endringer i tvisteloven m.m. og om samtykke til ratifikasjon av Luganokonvensjonen 2007 om domsmyndighet og om anerkjennelse og fullbyrdelse av dommer i sivile og kommersielle saker
- Prop. 104 L (2016-17) om lov om opphavsrett til åndsverk mv.

## **Internasjonale konvensjoner**

1886	Bernkonvensjonen om vern av litterære og kunstneriske verk
1926	International Institute for the Unification of Private Law (UNIDROIT)
1973	Europeisk patent konvensjon (EPC)
1994	Agreement on Trade-related aspects of Intellectual Property rights (TRIPS)
1996	WIPO Copyright Treaty
2002	Principles of European Contract Law (PECL)
2007	Luganokonvensjonen om domsmyndighet og om anerkjennelse og fullbyrdelse av dommer i sivile og kommersielle saker

## **EU-direktiver**

Rdir. 2001/29/EF	om harmonisering av visse aspekter af ophavsret og beslægtede rettigheter i informationssamfundet
Rdir. 1996/9/EF	om retlig beskyttelse af databaser

## **Utenlandske lover**

1947	U.S. Code: Title 17 Copyrights
1988	Copyright, Designs & Patents Act

## **Rettsavgjørelser**

### *Norsk rettspraksis*

Rt. 1923 II s. 59	
Rt. 1962 s. 964	Wegner
Rt. 1970 s. 1172	Roys Place
Rt. 1997 s. 199	Cirrus
Rt. 1997 s. 1807	Cigna

Rt. 1985 s. 883      Electric Circus  
Rt. 2000 s. 1811      Finanger I  
Rt. 2002 s. 391      God Morgon  
Rt. 2007 s. 1329      Huldra  
Rt. 2009 s. 1568      Andy Finch  
Rt. 2012 s. 1062      Tripp Trapp  
Rt. 2013 s. 822      Ambassadør

HR-2017-2165-A      Il tempo gigante  
RG 1999 s. 330      MOCS

*Avgjørelser fra EU-domstolen*

Sak C-306/05      Sociedad General de Autores y Editores de España (SGAE) v Rafael  
                         Hoteles SA  
                         ECLI:EU:C:2006:764

Sak C-5/08      Infopaq v. Danske Dagblades forening  
                         ECLI:EU:C:2009:465

Sak C-393/09      Bezpečnostní softwarová asociace – Svaz softwarové ochrany v  
                         Ministerstvo kultury  
                         ECLI:EU:C:2010:816

Sak C-145/10      Eva-Maria Painer v Standard VerlagsGmbH and Others  
                         ECLI:EU:C:2011:798

Sak C-406/10      SAS Institute Inc. v World Programming Ltd  
                         ECLI:EU:C:2012:259

Sak C-128/11      UsedSoft GmbH v Oracle International Corp.  
                         ECLI:EU:C:2012:407

Sak C-351/12	OSA – Ochranný svaz autorský pro práva k dílům hudebním o.s. v Léčebné lázně Mariánské Lázně a.s.  ECLI:EU:C:2014:110
Sak C-466/12	Nils Svensson and Others v Retriever Sverige AB ECLI:EU:C:2014:76
Sak C-348/13	BestWater International GmbH v Michael Mebes and Stefan Potsch ECLI:EU:C:2014:2315
Sak C-516/13	Dimensione Direct Sales srl and Michele Labianca v Knoll International Spa. ECLI:EU:C:2015:315

*Utenlandsk rettspraksis*

Nordisk

NJA 1995 s. 256	Nummerbanken
UfR 1951 s. 725	Hvem ringer klokkerne for?
UfR 1999 s. 2011	Ungdomsklubb
UfR 2007 s. 1581	182

Tyskland

I ZR 52/83	Inkasso programm
Sak 4 O 133/15	B.V. v. Universitat
Sak 2-6 O 224/06	Harald Welte v. D-Link Deutschland GmbH
Sak 21 O 6123/04	Harald Welte v. Sitecom Deutschland GmbH
Sak 7 O 5245/07	Harald Welte v. Skype Technologies S.A

## USA

193 F. 2d 744, 746-47 (9th Cir. 1952)	White v. Kimmell
930 F. 2d 277, 299 (3d Cir. 1991)	Ford Motor Co. V. Smmit Motor Prods., Inc.
991 F. 2d 426, 430-35 (8 <sup>th</sup> Cir. 1993)	Nat'l Car Rental Sys., Inc. v. Computer Assocs. Int'l, Inc.
542 F. Supp. 2d 153, 169 (D. Mass. 2008)	London-Sire Records, Inc. v. Doe
863 F.Supp.2d 394 (D.N.J. 2012)	Tetris Holding LLC v. Xio Interactive Inc.
Versata v. Ameriprise	Case no.: D-1-GN-12-003588; 53rd Judicial District Court of Travis County, Texas
Artifex v. Hancorn	Case no.: 16-cv-06982-JSC US District Court of Northern District of California

## **Bøker**

Bainbridge, David. *Legal Protection of Computer Software*, 5. Utg., Haywards Heath: Tottel Publishion Ltd., 2008

Dixon, Rod. *Open Source Software Law*, 1. Utg., Norwood: Artech House Inc., 2004

Downey, Allen B. *Think Python: How to think like a computer scientist*, 1. Utg., Needham: Green Tea Press, 2012. <http://greenteapress.com/thinkpython/thinkpython.pdf>

Fitzgerald, Brian, Kesan, Jay P., og Russo, Barbara m.fl.. *Adopting Open Source Software- a Practical Guide*, 1. Utg., Cambridge: The MIT Press, 2011

Hagstrøm, Viggo. *Obligasjonsrett*, 2. Utg., Oslo: Universitetsforlaget, 2013



Jacobsen, J. Hartvig. *Ophavsretten. Forfatter og- Kunstnerretten. National Ret.*, København: Gyldendal, 1941

Kielland, Torger. *Copyleft- en analyse av rekkevidden av gjensidighetsvilkår i åpne programvarelisenser i norsk rett*, Oslo: Institutt for rettsinformatikk Complex nr. 7/2005, 2005

Kielland, Torger. *Patentering av informasjonsteknologiske oppfinnelser*, 1. Utg., Oslo: Universitetsforl., 2017

Kveiva, Odd Randgaard. *Åpen programvare- noen rettslige problemstillinger*, Oslo: Institutt for rettsinformatikk Complex nr. 1/2009, 2009

Knoph, Ragnar. *Åndsretten*, Oslo: Nationaltrykkeriet, 1936

Liin, Birgit. *Immaterialretsaftaler*, 1. Utg., København: Jurist- og økonomforbundets Forlag, 2009

Pila, Justine, og Torremans, Paul. *European Intellectual Property Law*, 1. Utg., Oxford: Oxford University Press, 2016

Rosenmeier, Morten, *Verkslæren i ophavsretten*, Jurist- og Økonomforbundets Forlag, 2001.

Shemtov, Noam og Walden, Ian (red). *Free and Open Source Software*, 1. Utg., Oxford: Oxford University Press, 2013

Woxholth, Geir, *Avtalerett*, 9. Utg., Oslo: Gyldendal norsk forlag, 2014

## **Tidsskrifter**

Arendt, Torstein, “Fortolkning av “shrinkwrap”- og “clickwrapavtaler” med eksempler fra amerikansk rettspraksis”, *Tidsskrift for Forretningsjus* 03/1999 volum 5 (1999), s. 341-350

Cordero-Moss, Guiditta, "Behov for kodifisering av den internasjonale privatretten", *Lov og Rett* volum 52 (2013), s. 1-2

Fitzgerald, Brian. "The Transformation of Open Source Software", *Management Information Systems Quarterly* 30 (3) (2006), s. 587-598.

Gue, Theresa, "Triggering Infection: Distribution and Derivative Works Under the GNU General Public License", *Journal of Law, Technology and Policy* volume 2012 (2012) s. 97-140

Humphrey, Watts S., "Software Unbundling: A Personal Perspective", *IEEE Annals of the History of Computing* volume 24 issue 1 (2002), s. 59-63

Johnny, Omar, Miller, Marc, Webbink, Mark. "Copyright in Open Source Software- Understanding the Boundaries", *International Free and Open Source Software Law Review* volume 2 issue 1 (2010), s. 13-38

Selvig, Erling. "Tolking etter norsk eller annen skandinavisk rett av certerpartier og andre standardvilkår utformet på engelsk", *Tidsskrift for Rettsvitenskap* årgang 99 (1986) s. 1-26

Swinson, John. "Copyright or Patent or Both: An Algorithmic Approach to Computer Software Protection", *Harvard Journal of Law and Technology* volume 5 fall issue (1991), s. 145-214.

Torvund, Olav. "Opphavsrett til dataprogrammer utviklet i vertikalt samarbeid", *Ånd og rett: festskrift til Birger Stuevold Lassen* (1997), s. 1027-1032

Torvund, Olav. "Opphavsrett i oppdragsforhold", *Nordiskt immateriellt rättsskydd* (1999), s. 123-129

Tsai, John, "For Better or Worse: Introducing the GNU General Public License Version 3", *Berkeley Technology Law Journal* volume 23 issue 1 (2008), s. 547-581

Zhu, Chen Wei, "Copyleft Reconsidered: Why Software Licensing jurisprudence Needs Insights from Relational Contract Theory", *Social & Legal Studies* volume 22 nr. 3 (2013), s. 289-308

## Internettsider

Aarnes, Johan F. *Algoritme*. (2015), <https://snl.no/algoritme> [10.02.2018]

Civil Commons. *Contributor agreements*. (udatert),  
[http://wiki.civiccommons.org/Contributor\\_Agreements/](http://wiki.civiccommons.org/Contributor_Agreements/) [26.02.2018]

De Alaiza Cardona, José J. González. *Open Source, Free Software, and Contractual Issues*. (2006),  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1031.6788&rep=rep1&type=pdf>,  
[03.03.2018]

Drupal. *Licensing*. (udatert), <https://www.drupal.org/about/licensing#project-licensing>  
[13.04.2018]

Free Software Foundation. *Copyright assignment at the FSF*. (2014),  
<https://www.fsf.org/bulletin/2014/spring/copyright-assignment-at-the-fsf> [28.02.2018]

Furness, Dyllan. *Top tech companies open source their AI secrets*. (2017),  
<https://www.techemergence.com/top-tech-companies-open-source-their-ai-secrets/>  
[02.02.2018]

GNU. *Frequently asked questions about the GNU licenses- compatible licenses*. (2018),  
<https://www.gnu.org/licenses/gpl-faq.html#IfLibraryIsGPL> [09.03.2018]

GNU. *Frequently asked questions about the GNU licenses- internal distribution*. (2018),  
<https://www.gnu.org/licenses/gpl-faq.html#InternalDistribution> [08.03.2018]

GNU. *Frequently asked questions about the GNU licenses- plug-ins*. (2018),  
<https://www.gnu.org/licenses/gpl-faq.html#GPLAndPlugins> [09.03.2018]

GNU. *Frequently asked questions about the GNU licenses*. (2018)

<https://www.gnu.org/licenses/gpl-faq.html> [25.02.2018]

GNU. *GNU General Public License, version 2*. (2017), <https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html> [28.02.2018]

GNU. *Philosophy of the GNU project*. (2018 a),

<https://www.gnu.org/philosophy/philosophy.html> [02.02.2018]

IBM. *Why big companies are embracing open source*. (2012),

[https://www.ibm.com/developerworks/community/blogs/6e6f6d1b-95c3-46df-8a26-b7efd8ee4b57/entry/why\\_big\\_companies\\_are\\_embracing\\_open\\_source119?lang=en](https://www.ibm.com/developerworks/community/blogs/6e6f6d1b-95c3-46df-8a26-b7efd8ee4b57/entry/why_big_companies_are_embracing_open_source119?lang=en)

[30.03.2018]

Jager, Till. *Enforcement of the GNU GPL in Germany and Europe*. (2010),

<https://www.netfilter.org/documentation/licensing/dippadm1268746871.43.pdf> [03.03.2018]

North Bridge & Black Duck. *2015 Future of Open source survey results*. (2015),

[https://www.slideshare.net/North\\_Bridge/2015-future-of-open-source-study](https://www.slideshare.net/North_Bridge/2015-future-of-open-source-study) [30.03.2018]

Open source Initiative. *The open source definition (annotated)*. (udatert),

<https://opensource.org/osd-annotated> [03.03.2018]

Oracle. *The Oracle contributor agreement*. (udatert),

<http://www.oracle.com/technetwork/community/oca-486395.html> [28.02.2018]

Ranger, Steve. *What is cloud computing and everything you need to know about the cloud, explained*. (2018), <http://www.zdnet.com/article/what-is-cloud-computing-everything-you-need-to-know-from-public-and-private-cloud-to-software-as-a/> [08.03.2018]

Rossen, Eirik. *Brukergrensesnitt*. (2018 b), <https://snl.no/brukergrensesnitt> [26.02.2018]

Rossen, Eirik. *Dataprogram*. (2015 a), <https://snl.no/dataprogram> [28.01.2018]

Stallman, Richard. *Why open source misses the point of free software*. (2016), <https://www.gnu.org/philosophy/open-source-misses-the-point.html> [31.01.2018]

Vikøren, Birger M. *Outsourcing*. (2018), <https://snl.no/outsourcing> [08.03.2018]

Wikipedia. *Computer-aided software engineering*. (2018), [https://en.wikipedia.org/wiki/Computer-aided\\_software\\_engineering#CASE\\_Software](https://en.wikipedia.org/wiki/Computer-aided_software_engineering#CASE_Software) [26.02.2018]

Wikipedia. *Free software movement*. (2018), [https://en.wikipedia.org/wiki/Free\\_software\\_movement](https://en.wikipedia.org/wiki/Free_software_movement) [31.01.2018]

Wikipedia. *History of IBM*. (2018 a), [https://en.wikipedia.org/wiki/History\\_of\\_IBM#1969:\\_Antitrust,\\_the\\_Unbundling\\_of\\_software\\_and\\_services](https://en.wikipedia.org/wiki/History_of_IBM#1969:_Antitrust,_the_Unbundling_of_software_and_services) [26.03.2018]

Wikipedia. *Library (computing)*. (2018 b), [https://en.wikipedia.org/wiki/Library\\_\(computing\)](https://en.wikipedia.org/wiki/Library_(computing)) [02.02.2018]

Wikipedia. *List of platform-independent GUI libraries*. (2018 c), [https://en.wikipedia.org/wiki/List\\_of\\_platform-independent\\_GUI\\_libraries](https://en.wikipedia.org/wiki/List_of_platform-independent_GUI_libraries) [26.02.2018]

Wikipedia. *Open source software*. (2018), [https://en.wikipedia.org/wiki/Open-source\\_software#History](https://en.wikipedia.org/wiki/Open-source_software#History) [31.01.2018]

Wordpress. *License*. (udatert), <https://wordpress.org/about/license/> [13.04.2018]