

# **Avoimen lähdekoodin lisenssit kaupallisessa liiketoiminnassa**

Matti Saastamoinen

Tampereen yliopisto  
Tietojenkäsittelytieteiden laitos  
Tietojenkäsittelyoppi  
Pro gradu -tutkielma  
Toukokuu 2006

Tampereen yliopisto

Tietojenkäsittelytieteiden laitos

Tietojenkäsittelyoppi

Matti Saastamoinen: Avoimen lähdekoodin lisenssit kaupallisessa liiketoiminnassa

Pro gradu -tutkielma, 95 sivua, 4 liitesivua

Toukokuu 2006

---

Avoimen lähdekoodin merkitys ohjelmistoteollisuudessa kasvaa jatkuvasti, kun yritykset hyödyntävät avointa lähdekoodia enemmän omassa toiminnassaan. Tässä tutkimuksessa vertaillaan yleisimpiä avoimen lähdekoodin lisenssejä ja tutkitaan niiden soveltuvuutta osana kaupallista liiketoimintaa. Tutkimuksessa esitellään keinoja hyödyntää avointa lähdekoodia ohjelmistokehityksessä ja tutkitaan, mitä rajoituksia lisenssit asettavat niillä lisensoitujen ohjelmistojen ja komponenttien hyödyntämiselle yksinoikeusohjelmissa. Vastaavaa tutkimusta avoimen lähdekoodin lisenssien soveltuvuudesta kaupallisen liiketoiminnan tarpeisiin ei ole aikaisemmin tehty tässä mittakaavassa. Tutkimuksessa käy ilmi, että avoimen lähdekoodin hyödyntäminen kaupallisesti on mahdollista, mutta ei aina riskitöntä. Yritysten on oltava erityisen tarkkoina velvoittavien avoimen lähdekoodin lisenssien, kuten tutkimuksessa käsiteltävien GPL:n, LGPL:n ja MPL:n, kanssa. Sallivat avoimen lähdekoodin lisenssit rajoittavat hyvin vähän, jos lainkaan, niillä lisensoitujen teosten kaupallista hyödyntämistä. Sallivista lisensseistä tutkimuksessa käsitellään BSD-, MIT-, PHP-, Apache- ja Artistic-lisenssi. Riskien välttäminen edellyttää mm. avoimen lähdekoodin idean ja lisenssiehtojen ymmärtämistä, ja tämä tutkimus antaa tietoa juuri näistä asioista.

Avainsanat ja -sanonnat: avoin lähdekoodi, kaupallinen liiketoiminta, lisenssi, Open Source, tietokoneohjelma, vapaa ohjelmisto

CR-luokat: K.5.1

# Sisältö

<b>1.</b>	<b>JOHDANTO .....</b>	<b>1</b>
<b>2.</b>	<b>OHJELMISTOJEN VAPAAUS JA AVOIMUUS .....</b>	<b>5</b>
2.1.	TIETOKONEOHJELMA JA LÄHDEKODI .....	5
2.2.	AVOIMEN LÄHDEKODIN HISTORIA .....	6
2.3.	VAPAA OHJELMISTO .....	8
2.3.1.	<i>Vapaan ohjelmiston määritelmä.....</i>	<i>9</i>
2.4.	AVOIN LÄHDEKODI.....	11
2.4.1.	<i>Avoimen lähdekoodin määritelmä.....</i>	<i>11</i>
2.5.	AVOIMEN LÄHDEKODIN HYÖDYT JA RISKIT HYÖDYNTÄJILLE .....	14
2.5.1.	<i>Avoimen lähdekoodin hyödyt.....</i>	<i>14</i>
2.5.2.	<i>Avoimen lähdekoodin riskit .....</i>	<i>15</i>
<b>3.</b>	<b>OHJELMISTOLISENSSIT.....</b>	<b>20</b>
3.1.	TIETOKONEOHJELMIEN TEKIJÄNOIKEUS.....	20
3.2.	JOHDANNAINEN TEOS .....	21
3.3.	YKSINOIKEUSLISENSSIT .....	22
3.4.	AVOIMEN LÄHDEKODIN LISENSSIT .....	22
3.4.1.	<i>Avoimen lähdekoodin lisenssin pysyvyys, tarttuvuus ja sallivuus.....</i>	<i>22</i>
3.4.2.	<i>Avoimen lähdekoodin lisenssien luokittelu.....</i>	<i>26</i>
3.4.3.	<i>Avoimen lähdekoodin lisenssien yhteensopimattomuus .....</i>	<i>27</i>
3.5.	MONILISENSOINTI.....	29
3.6.	MUITA LISENSOINTI- JA LEVITYSMALLEJA.....	31
3.6.1.	<i>Freeware.....</i>	<i>31</i>
3.6.2.	<i>Shareware.....</i>	<i>32</i>
3.6.3.	<i>Public domain.....</i>	<i>33</i>
3.6.4.	<i>Creative Commons .....</i>	<i>34</i>
<b>4.</b>	<b>AVOIMEN LÄHDEKODIN LISENSSIEN ANALYYSI.....</b>	<b>35</b>
4.1.	LISENSSIEN VALINTAPERUSTEET JA YLEISYYS .....	35
4.2.	VELVOITTAVAT LISENSSIT .....	39
4.2.1.	<i>GPL.....</i>	<i>39</i>
4.2.2.	<i>LGPL .....</i>	<i>47</i>
4.2.3.	<i>MPL.....</i>	<i>55</i>
4.3.	SALLIVAT LISENSSIT .....	58
4.3.1.	<i>BSD-lisenssi.....</i>	<i>58</i>
4.3.2.	<i>MIT-lisenssi .....</i>	<i>61</i>

4.3.3.	<i>Apache-lisenssi</i> .....	62
4.3.4.	<i>PHP-lisenssi</i> .....	67
4.3.5.	<i>Artistic-lisenssi</i> .....	69
<b>5.</b>	<b>AVOIMEN LÄHDEKOODIN LISENSSIT HYÖDYNTÄMISTILANTEISSA</b> .....	<b>73</b>
5.1.	AVOIMEN LÄHDEKOODIN HYÖDYNTÄMINEN .....	73
5.1.1.	<i>Käyttö työkaluna</i> .....	74
5.1.2.	<i>Rajapintakäyttö ja linkitys</i> .....	76
5.1.3.	<i>Lähdekoodin kopiointi</i> .....	80
5.2.	LISENSSIEN VAIKUTUS MUUNTELUTILANTEISSA .....	81
5.2.1.	<i>Muunnellun ohjelmiston sisäinen käyttö</i> .....	81
5.2.2.	<i>Muunnellun ohjelmiston levitys</i> .....	81
5.2.3.	<i>Muunnellun ohjelmiston hinnoittelu</i> .....	82
5.2.4.	<i>Muunnellun ohjelmiston lisenssin muuttaminen</i> .....	83
<b>6.</b>	<b>YHTEENVETO</b> .....	<b>85</b>
	<b>LÄHDELUETTELO</b> .....	<b>89</b>
	<b>LIITE 1. AVOIMEN LÄHDEKOODIN LISENSSIEN VERTAILUTAULUKKO</b> .....	<b>96</b>
	<b>LIITE 2. THE OPEN SOURCE DEFINITION</b> .....	<b>98</b>

# 1. Johdanto

Immateriaalioikeudet (Intellectual Property Rights, IPR) on aineettomiin omistuksiin liittyvä oikeudenala. Perinteisesti immateriaalioikeudet jaetaan kahteen pääalueeseen, tekijänoikeuteen ja teollisoikeuteen [KTM, 2005]. Tekijänoikeuden alle kuuluvat teoskynnyksen ylittävät teokset, kuten tietokoneohjelmat, sekä tekijänoikeuden ns. lähioikeudet, jotka eivät vaadi teoskynnyksen ylittämistä, kuten oikeus valokuvaan [Wikipedia, 2006]. Teollisoikeuksiin kuuluvat mm. patenttisuoja, mallisuoja ja oikeus tavaramerkkiin. IPR-oikeuksilla on yleensä taloudellista arvoa, ja oikeuksien loukkaamisesta voi olla sanktiona mm. vahingonkorvaus. Oikeuksien haltija voi luovuttaa oikeudet toiselle osapuolelle tai myöntää niihin käyttöoikeuksia eli lisenssejä. [Nykänen, 2004]

Tietokoneohjelmia ei yleensä myydä, vaan asiakas ostaa ohjelman käyttöön oikeuttavan lisenssin. Myös ilmaiseksi levitettävän ohjelman mukana seuraa käyttölisenssi. Lisenssissä voidaan määritellä esimerkiksi ohjelman käyttöaika, kuinka moneen tietokoneeseen ohjelman saa asentaa tai kuinka monta samanaikaista käyttäjää ohjelmalla saa olla. Ohjelman lisenssillä voidaan vähentää monia ohjelman omistajan oikeudellisia velvollisuuksia, kuten velvollisuus korvata ohjelmavikojen aiheuttamat tuhot, ja rajoittaa ohjelman hyödyntämismahdollisuuksia esimerkiksi kieltämällä ohjelman levitys, muuntelu, kopiointi tai jälleenmyynti.

Yksinoikeudella lisensoidun tietokoneohjelman, nk. omistusohjelman, käyttö, levitys tai muuntelu on kiellettyä, se vaatii erillisen luvan tai on muuten niin rajoitettua, ettei käyttäjä voi tehdä sitä vapaasti ja tehokkaasti [FSF, 2005c]. Yksinoikeuslisenssin avulla tekijänoikeuden tai patentin haltija antaa toisille luvan käyttää hänen aineetonta omaisuuttaan rajoitetulla tavalla niin, että ohjelman vapautta ei erikseen suojella [Rosen, 2004]. Yksinoikeusohjelmien tekijät ovat perinteisesti antaneet käyttäjälle hyvin rajoitetut oikeudet ohjelmiinsa. Yksinoikeuslisenssit sallivat ohjelmiston kopioinnin yleensä vain käytön yhteydessä,

eikä ohjelmiston lähdekoodia toimiteta ohjelmiston mukana [Välimäki, 2002]. Yksinoikeusohjelmaa ei siis tavallisesti saa levittää eteenpäin, eikä sen toimintaa saa muuttaa, vaikka ohjelmasta löytyisi virheitä. Yksinoikeuslisenssejä hinnoitellaan mm. käyttäjien ja käyttökopioiden lukumäärän mukaan [Välimäki, 2002].

Avoimen lähdekoodin lisenssit poikkeavat merkittävästi ajatusmalliltaan yksinoikeuslisensseistä. Ne antavat lisenssin saajalle oikeuksia, joita ohjelmistoyritykset ovat perinteisesti pidättäneet itsellään. Näitä ovat mm. ohjelmiston vapaa kopiointi- ja levitysoikeus, pääsy ohjelmiston lähdekoodiin ja ohjelmiston muuntelu-oikeus. Avoin lähdekoodi on mahdollistanut ohjelmien avoimen kehitysprosessin, joka on synnyttänyt yhteisöjä, joissa ohjelmia kehitetään yhteistyössä vapaaehtoisvoimin.

Ajatus lähdekoodin vapaasta jakelusta on saanut alkunsa tutkijoiden ja innokkaiden harrastajien, hakkereiden, keskuudessa [Williams, 2002]. Viime vuosina myös yritykset ovat alkaneet hyödyntää avointa lähdekoodia omissa liiketoiminnoissaan. Varsinkin Internetin kehitys ja yleistyminen ovat vaikuttaneet siihen, että avoimen lähdekoodin ohjelmistoja kehittävät ja markkinoivat yritykset ovat haastaneet yksinoikeuslisensseillä varustettujen tietokoneohjelmien liiketoimintamallin [Välimäki, 2002]. Eräät avoimen lähdekoodin lisenssit tarjoavat lisenssin saajalle huomattavia vapauksia, jotka voivat mahdollistaa avoimen lähdekoodin monipuolisen hyödyntämisen myös kaupallisessa liiketoiminnassa. Arvostetussa amerikkalaisessa Business Week -lehdessä on todettu [Lacy, 2005], että vuosi 2005 jää historiaan avoimen lähdekoodin liiketoiminnan vedenjakajana. Vuonna 2005 myös sijoittajat kiinnostuivat avoimen lähdekoodin tarjoamista uusista liiketoimintamahdollisuuksista, ja suurten organisaatioiden tietohallintojohtajat alkoivat nähdä avoimet ohjelmistot strategisena osana IT-arkkitehtuuria. Konsultointiyritys Optarosin tekemän tutkimuksen [Walli *et al.*, 2005] mukaan 87 % yhdysvaltalaisista yrityksistä, hallintoelimistä ja organisaatioista käyttää avointa lähdekoodia jollain tavoin.

Jotta avoimen lähdekoodin mahdollisuudet ja riskit pystyttäisiin tunnistamaan, on ymmärrettävä avoimen lähdekoodin erityispiirteet. Yksi perusedellytys on avoimen lähdekoodin lisenssien tuntemus. Tämä on haastava tehtävä jo avoimen lähdekoodin lisenssien suuren määrän vuoksi, mutta myös lisenssien sisältämien tulkinnanvaraisuuksien sekä kömpelöiden sanamuotojen ja rakenteiden vuoksi.

Tässä tutkimuksessa käsitellään avoimen lähdekoodin lisensointia, syvennyttään kahdeksaan suosittuun avoimen lähdekoodin lisenssiin ja tutkitaan, miten ne suhtautuvat niillä lisensoitujen ohjelmien ja ohjelmakomponenttien erilaisiin hyödyntämismenetelmiin kaupallisessa liiketoiminnassa ja yksinoin-

keusohjelmissa. Tarkoituksena ei ole selvittää mitä lisenssejä yritykset voisivat valita omien avoimen lähdekoodin ohjelmistojensa lisensoimiseksi, vaan tarkastella avointa lähdekoodia hyödyntäjänäkökulmasta. Karl Fogelin kirjassa *Producing Open Source Software: How to Run a Successful Free Software Project* [Fogel, 2005] käsitellään avointa lähdekoodia kehittäjänäkökulmasta. Kirjassa on tietoa projektin perustamisesta, hallinnasta, yhteisön kanssa toimimisesta ja myös lisenssin valitsemisesta.

Open Source Initiativen (OSI) *avoin lähdekoodi* ja Free Software Foundationin (FSF) *vapaa ohjelmisto* edistävät kumpikin lähdekoodin avoimuutta, mutta viittaavat erillisiin sosiaalisiin liikkeisiin ja ideologioihin, joilla on erilaiset näkökulmat ja päämäärät [Stallman, 2005]. Tässä tutkimuksessa ei keskitytä liikkeiden ideologioihin eikä oteta kantaa moraalisiin kysymyksiin, joita voidaan esittää muiden ilmaiseksi levittämien ohjelmistojen hyödyntämisestä yksinoikeusohjelmissa ja ylipäänsä kaupallisessa liiketoiminnassa. Lisäksi muut IPR-riskit, kuten patentit ja tavaramerkit, ovat tutkimuksen laajuuden ulkopuolella. Jatkossa puhuttaessa avoimesta lähdekoodista tarkoitetaan Open Source Initiativen hyväksymien lisenssien alaisia ohjelmistoja tai niiden osia.

Tutkimus suoritettiin kirjallisuustutkimuksena. Suomenkielistä kirjallisuutta avoimen lähdekoodin lisensseistä on saatavilla hyvin niukalti. Englanninkielistä painettua kirjallisuutta on tarjolla kohtalaisesti, joskin suuri osa siitä käsittelee lisenssejä yhdysvaltalaisen lainsäädännön pohjalta, mikä voi vaikeuttaa joitain tulkintatilanteita. Mikko Välimäen toukokuussa 2005 julkaistu väitöskirja [Välimäki, 2005a] on suomalainen ja eurooppalainen perehtyminen aiheeseen. Open Source Initiativen entinen oikeuskonsultti Lawrence Rosen on kirjoittanut kirjan avoimen lähdekoodin lisensseistä [Rosen, 2004], kuten myös ohjelmistolisensseihin erikoistunut lakimies Andrew Laurent [2004], mutta molemmat kirjat käsittelevät lisenssejä Yhdysvaltojen tekijänoikeuslain näkökulmasta. Kirjat ovat silti erinomaisia lähteitä lisenssien ehtojen ymmärtämiseksi, ja ne sisältävät myös huomioita lisenssien heikkouksista. Painetun materiaalin lisäksi Internetissä on tarjolla suuri määrä tietoa ja tulkintoja aiheesta.

On huomattava, että avoimen lähdekoodin lisenssien ehdoissa on paljon tulkinnanvaraisuuksia ja epäselvyyksiä, ja tähän mennessä on kertynyt hyvin vähän näyttöä siitä, miten avoimen lähdekoodin lisenssejä on tulkittu riita- ja ongelmatilanteissa oikeusistuimissa [Välimäki, 2005a; LG München, 2004]. Tämä tutkimus sisältääkin kirjoittajan ja viitattujen henkilöiden omia näkemyksiä, joihin pitää suhtautua tulkintoina eikä absoluuttisina totuuksina.

Tutkimus on pyritty kirjoittamaan niin, että sekä ohjelmistoteollisuuden ammattilaiset että teknisistä yksityiskohdista tietämättömät pystyisivät ymmärtämään esitetyt asiat. Tutkimus on jaettu johdannon lisäksi viiteen lukuun.

Luvussa 2 esitetään tietokoneohjelman ja lähdekoodin määritelmä ja käsitellään lähdekoodin saatavuuden merkitystä tietokoneohjelman muuntelulle. Lisäksi tutustutaan avoimen lähdekoodin historiaan sekä vapaan ohjelmiston ja avoimen lähdekoodin määritelmiin. Lopuksi pohditaan avoimen lähdekoodin hyötyjä ja riskejä hyödyntämistilanteissa.

Luvussa 3 käsitellään tietokoneohjelmien lisensointia. Aluksi käydään läpi välttämättömät tiedot tekijänoikeudesta ja johdannaisesta teoksesta myöhemmin esitettävien asioiden ymmärtämiseksi. Perinteisen yksinoikeuslisensoinnin käsittelyn jälkeen paneudutaan avoimen lähdekoodin lisenssien erityispiirteisiin. Luvun lopussa esitellään muita lisensointi- ja levitysmalleja.

Luvussa 4 keskitytään syvällisesti kahdeksaan merkittävään avoimen lähdekoodin lisenssiin. Jokaisen lisenssin kohdalla esitellään lisenssin alkuperä, toiminnallisuus ja keskeisimmät ehdot. Lisenssiehdot on käännetty suomen kielelle.

Luvussa 5 pohditaan luvussa 4 käsiteltyjen lisenssien ehtojen vaikutusta erilaisissa avoimen lähdekoodin hyödyntämistilanteissa. Erityisesti kiinnitetään huomiota lisenssien vaikutukseen avoimen lähdekoodin ohjelmistojen muuntelutilanteissa ja yhdistettäessä avointa lähdekoodia muuhun koodiin.

Luvussa 6 kootaan yhteen tutkimuksen keskeisimmät tulokset ja esitellään uusia tutkimusaiheita.



## 2. Ohjelmistojen vapaus ja avoimuus

Tässä luvussa kerrotaan, miten vapaa ohjelmisto ja avoin lähdekoodi ovat syntyneet, mitkä ovat näiden käsitteiden määritelmät, ja miten ne poikkeavat toisistaan. Luvun lopussa käsitellään avoimen lähdekoodin hyötyjä ja riskejä kaupallisessa liiketoiminnassa. Aivan aluksi tutustutaan tietokoneohjelman määritelmään ja lähdekoodin rooliin tietokoneohjelmassa.

### 2.1. Tietokoneohjelma ja lähdekoodi

Tietokoneohjelma on joukko koodattuja toimintaohjeita, jotka syötettynä tietokoneeseen saavat tietokoneen automaattisesti ohjaamaan toimintaansa määritellyn tehtävän suorittamiseksi [Oxford, 1989]. Ilman ohjelmia tietokoneet olisivat hyödyttömiä.

Tietokoneohjelmaa ei ole erikseen määritelty Suomen tekijänoikeuslaissa. Euroopan Unionin direktiivissä tietokoneohjelmien oikeudellisesta suojasta [ETY, 1991] tietokoneohjelmalla tarkoitetaan missä tahansa muodossa olevaa ohjelmaa, laitteistoon sisältyvät ohjelmat mukaan lukien. Tämä käsite sisältää myös tietokoneohjelman kehittämiseen tähtäävän valmisteleavan suunnittelutyön, jos suunnittelutyön tuloksena voi myöhemmässä vaiheessa olla tietokoneohjelma [ETY, 1991].

Tietokoneohjelman toimintaohjeet sisältäviä tiedostoja kutsutaan ohjelman *lähdekoodiksi*. Lähdekoodi kirjoitetaan jollakin ohjelmointikielellä, joka muutetaan *objektikoodiksi* kääntäjän tai tulkin avulla [Oxford, 1989]. Objektikoodi sisältää tietokoneen ymmärtämää *konekieltä*, ja se voidaan suorittaa tietokoneessa joko suoraan tai virtuaalikoneen avulla.

Tietokoneohjelman lähdekoodi on ohjelman resepti, josta selviää ohjelman toiminta, ja jota muuttamalla myös ohjelman toiminta muuttuu. Näin ollen, jotta tietokoneohjelmaan voitaisiin tehdä pieniä tai isoja muutoksia, muuntaa ohjelma täysin uuteen käyttötarkoitukseen tai tutkia sekä opiskella ohjelman

rakennetta ja oppia siitä jotain uutta, on pääsy ohjelman lähdekoodiin välttämätöntä.

Avoimen lähdekoodin lisenssit määrittelevät monin eri tavoin lähdekoodin käsitteen, ja niiden määritelmät sisältävät monesti paljon muutakin kuin pelkän ohjelmiston lähdekoodin. Esimerkiksi GNU Lesser General Public Licensen (LGPL) mukaan ohjelmistokirjaston täydellinen lähdekoodi sisältää kaiken lähdekoodin kaikkiin kirjaston sisältämiin moduuleihin ja lisäksi kaikki rajapintojen määrittelytiedostot sekä kirjaston kääntämiseen ja asentamiseen tarvittavat skriptit [OSI\_LGPL, 1999]. Skriptit ovat jollakin kuvauskielellä kirjoitettuja ohjelmapätkiä tai kokonaisia ohjelmia, jotka voidaan suorittaa välittömästi jollakin ohjelmalla ilman, että ne pitäisi ensin kääntää objektikoodiksi. Tulkintaroista johtuen jokaisen avoimen lähdekoodin lisenssin kohdalla kannattaa tarkistaa lisenssin oma lähdekoodin määritelmä, jos lisenssi sisältää sellaisen.

Useimmat kaupalliset ohjelmistotalot toimittavat tietokoneohjelmansa ainoastaan objekti- eli binäärimuotoisina. Binäärimuotoisen ohjelman sisällön muuttaminen on ihmiselle erittäin hankalaa [Lerner and Tirole, 2002]. Tavallisesti näiden nk. yksinoikeusohjelmien lisenssiehdot myös kieltävät ohjelman kaikenlaisen muuntelun, jolloin ohjelman toiminnan muuttaminen olisi myös lisenssiehtojen vastaista ja useimmiten laitonta.

Mikäli ohjelmiston lähdekoodia ja dokumentaatiota ei ole saatavilla, niin tietoa ohjelmiston rakenteesta ja ominaisuuksista voi saada *takaisinmallinnuksella* (reverse engineering). Takaisinmallinnus on laaja käsite, joka sisältää yhtenä osanaan konekielisen ohjelman muuntamisen luettavampaan muotoon [Korpela, 2005]. Takaisinmallinnuksen avulla binäärimuotoinen ohjelma saatetaan kyetä muuttamaan lähdekoodimuotoon tai vastaavaan ihmiselle helpommin ymmärrettävään muotoon. Laki voi kuitenkin asettaa omat rajoituksensa ohjelman koodin muodon kääntämiselle. Suomen tekijänoikeuslain 25 k §:n [HE, 2005] mukaan koodin muodon kääntäminen on sallittua vain, jos se on välttämätöntä ohjelman saamiseksi toimimaan muiden ohjelmien kanssa.

## 2.2. Avoimen lähdekoodin historia

Joukkotiedotusvälineitä seuraamalla on voinut saada kuvan, että avoimen lähdekoodin ohjelmistojen historia ei ole kovin pitkä. Tiedotusvälineet ovat löytäneet avoimen lähdekoodin sangen myöhäisessä vaiheessa, kun Internet on räjähdysmäisesti levitessään kasvattanut avoimen lähdekoodin projektien määrää ja suosiota, ja loppukäyttäjille on kyetty tarjoamaan viimeisteltäviä ja helppokäyttöisiä tuotteita. Myös yritysten lisääntynyt mielenkiinto avointa lähdekoodia kohtaan on kasvattanut median kiinnostusta.

Todellisuudessa avoimen lähdekoodin ohjelmistojen historia ulottuu aina 1960-luvun alkuun saakka. Lerner ja Tirole [2002] jakavat avoimen lähdekoodin ohjelmistojen historian kolmeen aikakauteen: ensimmäinen aikakausi ulottuu 1960-luvun alusta 1980-luvun alkuun, toinen 1980-luvun alusta 1990-luvun alkuun ja kolmas 1990-luvun alusta 2000-luvulle. Muita hyviä historiallisia katsauksia vapaaseen ohjelmistoon ja avoimeen lähdekoodiin ovat kirjat *Free as in Freedom: Richard Stallman's Crusade for Free Software* [Williams, 2002], *The Cathedral & The Bazaar* [Raymond, 2000] ja *Open Sources: Voices from the Open Source Revolution* [DiBona et al., 1999].

1960- ja 1970-luvuilla kehitettiin monet käyttöjärjestelmien ja Internetin keskeisimmät piirteet. Niiden kehityspaikkoina toimivat erityisesti Kalifornian yliopisto, Massachusettsin teknillinen korkeakoulu (MIT) ja eräiden yritysten tutkimuskeskukset, joissa tutkijat saivat hyvin vapaasti päättää omista tekemisistään. Näinä aikoina eri organisaatioiden välillä oli arkipäiväistä, että tietokoneohjelmien perustoimintakoodia jaettiin tutkijoiden kesken. [Lerner and Tirole, 2002]

1980-luvun alussa yhdysvaltalainen johtava telekommunikaatioyritys AT&T ryhtyi vaatimaan oikeuksiaan Unix-käyttöjärjestelmän suhteen, joka oli saanut alkunsa AT&T:n Bell Laboratories -tutkimuskeskuksessa. Oikeudenkäyntien uhka synnytti ensimmäiset ponnistelut vakiinnuttaa yhteistyönä tapahtuvan ohjelmistokehityksen perussäännöt. Samalla alkoi avoimen lähdekoodin ohjelmistojen toinen aikakausi, jonka tärkeimpänä instituutiona toimi Richard Stallmanin perustama Free Software Foundation. Free Software Foundationin suurimpana innovaationa voidaan pitää muodollisen lisensointimenettelyn kehittämistä avoimen lähdekoodin ohjelmistoille, jonka seurauksena syntyivät tämän hetken suosituimmat avoimen lähdekoodin lisenssit GNU General Public License (GPL) ja GNU Lesser General Public License. [Lerner and Tirole, 2002]

1990-luvulla Internetin kasvanut levinneisyys kiihdytti voimakkaasti avoimen lähdekoodin yhteisön toimintaa. Tuolloin käynnistettiin lukemattomia uusia avoimen lähdekoodin projekteja, joista merkittävimpanä Linus Torvaldsin aloittama Linux. Linuxista tuli ohjelmisto, jonka myötä GPL löi lopullisesti läpi [Välimäki, 2005a]. Yritykset havaitsivat avoimen lähdekoodin potentiaalin, ja yritysten ja avoimen lähdekoodin yhteisön yhteistoiminnasta tuli arkipäivää. 1990-luvulla kehitettiin Free Software Foundationin vapaan ohjelmiston määritelmää joustavampia ohjesääntöjä, joista merkittävimmäksi nousi Open Source Initiativen ylläpitämä, Debianin vapaan ohjelmiston ohjeistoon pohjautuva avoimen lähdekoodin määritelmä. [Lerner and Tirole, 2002; OSI, 2005b]

2000-luvun ensimmäisellä vuosikymmenellä avoimen lähdekoodin suosion kasvu niin yrityksissä kuin yhteisöissä on ollut edelleen voimakasta, mutta samalla avoimen lähdekoodin liike on kohdannut uusia haasteita [Lerner and Tirole, 2002]. Tapaus, jossa yhdysvaltalainen SCO haastoi IBM:n oikeuteen patenttinsa loukkauksesta [SCO v. IBM, 2005], on saanut eniten huomiota. SCO väitti, että IBM on kopioinut SCO:n omistamaa lähdekoodia GNU/Linuxiin ja vaatii GNU/Linux-käyttäjiä ostamaan lisenssin SCO:lta. Myöhemmin kävi selväksi, että SCO:n patenttirikkomisväitteillä ei ollut tosiasiallista pohjaa [Välimäki, 2005a].

Avoin lähdekoodi on perinteisesti toiminut parhaiten tehokäyttäjien ja alan asiantuntijoiden parissa sekä alemman tason ohjelmakirjastoissa ja työkaluissa [Lerner ja Tirole. 2002]. Saavuttaakseen suosiota myös tavallisten loppukäyttäjien keskuudessa avoimen lähdekoodin yhteisön on panostettava enemmän käyttöliittymäsuunnitteluun ja dokumentointiin, keskityttävä uusien toimintojen toteuttamisen sijasta ohjelmien pitämiseen vakaina, pidettävä jokaisen ohjelman kohdalla kohderyhmä mielessä ja otettava oppia kaupallisten ratkaisujen saavutuksista [Levesque, 2004]. Viitteitä positiivisesta kehityksestä onkin saatu esimerkiksi Mozilla- ja OpenOffice.org-projektien myötä. Avoimen lähdekoodin yhteisö tarvitsee yritysten apua kehittyäkseen ja vallatakseen alueita, jotka ovat perinteisesti kuuluneet yksinoikeusohjelmille. Avoimen kehitysprosessin ylivertaisuuden osoittaminen yritysmaailmalle onkin eräs Open Source Initiatiiven perustavoitteista [OSI, 2005a].

### 2.3. Vapaa ohjelmisto

Richard Stallman ryhtyi 1980-luvun alussa toteuttamaan vapaata käyttöjärjestelmää, joka voitaisiin siirtää konetyypistä toiseen. Unix oli ainoa Stallmanin tietämä tällainen järjestelmä, ja hän halusi tehdä omasta käyttöjärjestelmästäan yhteensopivan Unixin kanssa, jotta siirtyminen Unixista siihen olisi helppoa ja Unixille luodut ohjelmat toimisivat myös uudessa järjestelmässä [Moody, 2001; Stallman, 1999b]. Hakkereille tyypillisesti Stallman antoi hankkeelle rekursiivisen lyhenteen *GNU*, joka tulee sanoista *GNU's Not UNIX* [FSF, 2005f].

GNU-projekti käynnistyi virallisesti tammikuussa 1984, kun Stallman ryhtyi kirjoittamaan työkaluja käyttöjärjestelmän toteuttamiseksi. Hän lopetti työsuhteensa MIT:n tekoälylaboratoriossa, jotta yliopisto ei voisi sekaantua GNU-ohjelmistojen vapaaseen levitykseen [Stallman, 1999b]. Stallmanin työn tuloksena syntyivät mm. GNU Emacs ja GNU C Compiler (GCC), jotka ovat edelleen arvostettuja työkaluja ohjelmistokehittäjien keskuudessa.

GNU-projektin alkuperäisenä tavoitteena oli vapauden antaminen käyttäjille – ei siis ainoastaan suosion saavuttaminen [Stallman, 1999b]. Jos projekti olisi

halunnut vain suosiota ja ohjelmistojen mahdollisimman laajaa levinneisyyttä, niin sen olisivat mahdollistaneet jo olemassa olevat kaiken sallivat lisenssit tai ohjelmistojen antaminen yleiseen omistukseen (public domain). Koska Stallman halusi pitää kiinni luomiensa ohjelmien vapaudesta, hän julkaisi jakeluehdot, jotka kielsivät ohjelmiston ehtojen muuttamisen. Tätä metodia kutsutaan *copy-leftiksi*, käyttäjänoikeudeksi.

Copyleft tarjoaa keinot ohjelmiston säilyttämiseksi vapaana. Keskeinen copyleftin idea on antaa kaikille lupa käyttää, kopioida ja muunnella ohjelmaa ja levittää muunneltuja versioita, mutta evätä lupa omien rajoitusten lisäämiseen [Stallman, 1999b]. Ratkaisevat vapaudet, jotka määrittävät vapaan ohjelmiston, ovat siten taatut kaikille, joilla on ohjelmiston kopio. Niistä tulee luovuttamattomia oikeuksia. Vapaa ohjelmisto (free software) on Richard Stallmanin keksimä termi kuvaamaan ohjelmistoa, joka sallii nämä vapaudet.

GNU-projektin kasvaessa yhä isommaksi perustettiin vuonna 1985 Free Software Foundation (FSF) rahoittamaan GNU-projektia [Stallman, 1999b]. Nykyään FSF on suurin yksittäinen organisaatio, joka tukee käyttäjien oikeuksia käyttää, opiskella, kopioida, muunnella ja levittää tietokoneohjelmia – kaikki alkuperäisiä GNU-projektin periaatteita. Verrattuna avoimen lähdekoodin vastaavaan järjestöön, Open Source Initiativeen, Free Software Foundation on huomattavasti herkempi ottamaan kantaa tietokoneohjelmien vapauden ideologisiin kysymyksiin.

FSF tunnetaan parhaiten GPL:sta ja LGPL:sta. GNU-projektin alkuvaiheessa syntyneet ohjelmat olivat kukin lisensoitu omalla ohjelmakohtaisella lisenssilään, mikä vaikeutti asioita ja aiheutti yhteensopivuusongelmia jopa GNU-ohjelmien välillä [Moody, 2001]. Stallman kirjoitti GPL:n vapaiden ohjelmistojen yleiseksi lisenssiksi, jolla voitaisiin lisensoida kaikki vapaat ohjelmistot. GNU-lisensseistä on sittemmin tulleet tämän hetken suosituimmat avoimen lähdekoodin lisenssit.

### **2.3.1. Vapaan ohjelmiston määritelmä**

Vapaan ohjelmiston idea tulee parhaiten esille tarkastelemalla käsitteen neljäkohtaista määritelmää [FSF, 2004] (käännös kirjoittajan):

1. Vapaus käyttää ohjelmaa mihin tarkoitukseen tahansa.
2. Vapaus tutkia kuinka ohjelma toimii ja muunnella sitä omiin tarpeisiin sopivaksi. Pääsy lähdekoodiin on tämän ennakkoehto.
3. Vapaus jakaa kopioita ohjelmasta lähimmäisen auttamiseksi.
4. Vapaus parantaa ohjelmaa ja julkaista tehdyt parannukset yleisölle siten, että koko yhteisö hyötyy. Pääsy lähdekoodiin on tämän ennakkoehto.

Mikäli ohjelmiston lisenssi antaa lisenssin saajalle kaikki vapaan ohjelmiston määritelmän sisältämät vapaudet, niin ohjelmaa voidaan kutsua vapaaksi ohjelmistoksi.

Vapaan ohjelmiston määritelmässä korostuu mahdollisuus päästä käsiksi ohjelman lähdekoodiin: tähän viitataan määritelmän toisessa ja neljännessä kohdassa. Ohjelman käytölle ei saa asettaa rajoitteita esimerkiksi kieltämällä kaupallinen käyttö, vaan kaikenlainen käyttö on sallittava. Myöskään ohjelman levitystä ei saa rajoittaa. Vapaata ohjelmistoa saa siis vapaasti jakaa muutoksitta tai muutoksien kanssa, ilmaiseksi tai maksua vastaan, kenelle tahansa, missä tahansa. Mikäli ohjelmistoon tekee muutoksia, ja ohjelmisto on vain yksityisessä omassa käytössä, so. sitä ei levitetä, niin muutoksia ei tarvitse julkistaa tai ilmoittaa niistä kenellekään. [FSF, 2004]

Määritelmässä ei erikseen sanota mitään vapaan ohjelmiston kaupallisesta käytöstä, mutta kuten määritelmän ensimmäisessä ehdossa todetaan, vapaata ohjelmistoa saa käyttää mihin tarkoitukseen tahansa – myös kaupallisiin tarkoituksiin. Määritelmässä ei myöskään vaadita vapaiden ohjelmistojen olevan ilmaisia. Päinvastoin Free Software Foundation jopa rohkaisee ihmisiä veloittamaan vapaista ohjelmista niin paljon kuin he pystyvät [FSF, 2001a].

Vapaan ohjelmiston alkuperäisen termin, *free software*, suureksi haittapuoleksi voidaan todeta se, että englannin kielen sana *free* voi tarkoittaa vapaan lisäksi ilmaista. Stallman on usein joutunut korostamaan, että puhuttaessa vapaasta ohjelmistosta tarkoitetaan vapautta (*freedom*), ei hintaa (*price*) [FSF, 1991]. Myös GPL- ja LGPL-lisenssit sisältävät tämän huomautuksen johdannossaan [OSI\_GPL, 1991; OSI\_LGPL, 1999] (käännös Välimäen [2001]):

Kun tässä Lisenssissä puhutaan vapaasta ohjelmasta, silloin ei tarkoiteta hintaa.

Rosenin [2004] mukaan termin *vapaus* epäselvyys oli suurin syy *avoimen lähdekoodin* luomiseen.

## 2.4. Avoin lähdekoodi

Avoimen lähdekoodin (Open Source) käsite syntyi helmikuussa 1998, kun joukko alan johtavia hakkereita Eric Raymondin johdolla kokoontui pohtimaan vapaan ohjelmiston tilannetta ja tulevaisuutta. Kokoontuminen oli reagointi Netscape Communicationsin viikkoa aikaisemmin julkaisemaan suunnitelmaan [Netscape, 1998] avata uuden Netscape Communicatorin lähdekoodi ilmaiseksi kaikille [Williams, 2002]. Mukana olleet hakkerit olivat huomanneet Netscapen ilmoituksen myötä syntyneen mahdollisuuden yritysmailman kiinnostuksen herättämiseksi avoimesta kehitysprosessista. He halusivat puhdistaa pöydältä negatiiviset asenteet, jotka liittyivät vapaaseen ohjelmistoon, ja keskittyä myymään samaa ideaa liiketoiminnan pohjalta. Uusi liike tarvitsi uuden nimen, ja parhaaksi ehdotukseksi valittiin *Open Source*, avoin lähdekoodi. Samalla sai alkunsa voittoa tavoittelematon Open Source Initiative (OSI), joka hallinnoi ja markkinoi avoimen lähdekoodin määritelmää erityisesti OSI-aitoustodistusohjelmansa avulla. [OSI, 2005a]

### 2.4.1. Avoimen lähdekoodin määritelmä

Open Source Initiativen hallinnoima avoimen lähdekoodin määritelmä on yksityiskohtaisempi kuin vapaan ohjelmiston määritelmä. Sen alkuperäisversion on kirjoittanut Bruce Perens, ja määritelmä pohjautuu Debianin vapaiden ohjelmistojen ohjeistoon [Debian, 2004; OSI, 2005b]. Määritelmän tarkoituksena on tehdä yhteenveto avoimen lähdekoodin periaatteista ja tarjota ohje avoimen lähdekoodin lisensseille [Rosen, 2004].

Open Source Initiativella on tarkoin määritelty menettelytapa lisenssin hyväksymiseksi avoimen lähdekoodin lisenssiksi [OSI, 2005d]. Tärkein hyväksymiseen vaikuttava tekijä on lisenssin yhteensopivuus avoimen lähdekoodin määritelmän kanssa: virallinen hyväksyntä avoimen lähdekoodin lisenssiksi edellyttää, että lisenssi täyttää kaikki avoimen lähdekoodin määritelmän ehdot. Open Source Initiativen hyväksymien avoimen lähdekoodin lisenssien alaiset ohjelmat voidaan varustaa *OSI Certified* -merkinnällä, joka takaa ohjelmiston lisenssin olevan avoimen lähdekoodin määritelmän mukainen [OSI, 2005d]. Kuvassa 1 on merkin graafinen versio, jota voidaan käyttää tekstimerkin sijasta.



**Kuva 1. OSI Certified -merkki, jolla Open Source Initiativen hyväksymällä lisenssillä lisensoitu teos voidaan varustaa [OSI, 2005d].**

Rosen [2004] esittää kritiikkiä avoimen lähdekoodin määritelmän epämääräisyydestä. On kuitenkin todettava, että avoimen lähdekoodin määritelmä on huomattavasti tarkempi kuin vapaan ohjelmiston määritelmä, jossa kaikki perustuu ohjelmistoteollisuudessa harvoin käytettyyn vapauden käsitteeseen. Avoimen lähdekoodin määritelmän epämääräisyyden vuoksi Rosen on kirjoittanut viisi avoimen lähdekoodin periaatetta, jotka kuvailevat avoimen lähdekoodin ohjelmiston. Rosenin periaatteet ovat yhdenmukaisia virallisen avoimen lähdekoodin määritelmän ja vapaan ohjelmiston määritelmän kanssa. Myös Välimäki [2005a] soveltaa Rosenin avoimen lähdekoodin tiivistelmää. Seuraavaksi perehdymme avoimen lähdekoodin määritelmään Rosenin periaatteiden avulla (käännös kirjoittajan).

Avoimen lähdekoodin ohjelmiston lisenssin tulee sallia:

1. ohjelmiston vapaa käyttö mihin tarkoitukseen tahansa.
2. ohjelmiston kopioiminen ja levittäminen ilman maksuvelvoitetta.
3. ohjelmistoon perustuvien teosten luominen ja levittäminen ilman maksuvelvoitetta.
4. pääsy ohjelmiston lähdekoodiin ja lähdekoodin käyttö.
5. ohjelmiston yhdistäminen toisten ohjelmistojen kanssa.

Vapaa käyttö tarkoittaa, ettei ohjelmiston käyttöä saa rajoittaa mitenkään eikä mitään käyttäjäryhmää saa syrjiä. Näin esimerkiksi kaupallista käyttöä tai käyttäjien lukumäärää ei saa rajoittaa [Välimäki, 2005a]. Lisenssinhaltijan ei tarvitse raportoida lisensoijalle ohjelman käytöstä tai kertoa ohjelman käyttöta-voista tai -tarkoituksesta [Rosen, 2004].

Kopioiminen ja levitys ilman maksuvelvoitetta rajaa lisenssimaksut pois toteuttamiskelpoisista liiketoimintamalleista [Välimäki, 2005a]. Tämä ei tarkoita, etteikö lisenssinhaltija voisi myydä lisenssiä avoimen lähdekoodin ohjelmistoon, sillä se on täysin luvallista. Ehto tarkoittaa sitä, että lisenssinhaltija voi valmistaa ohjelmistosta kopioita ja jakaa niitä vapaasti muille ilman, että hänen täytyy suorittaa ohjelmiston omistajalle erillinen maksu [Rosen, 2004]. Tämä väistä-



mättä johtaa avoimen lähdekoodin ohjelmistojen hintojen painumiseen tuotanto- ja levityskustannusten tasolle [Rosen, 2004].

Lähdekoodin muuntelu on sallittava ilman maksuvelvoitetta ja johdannaisia teoksia on voitava jakaa ohjelmiston alkuperäisellä lisenssillä. Lisenssiä voi myös muuttaa, mikäli alkuperäinen lisenssi tämän sallii. Näin aikaisemman ohjelmiston päälle tai sitä muuten hyödyntäen voidaan kehittää uusia ohjelmistoja. Lisenssi ei saa vaatia maksua oikeudesta tehdä ja levittää johdannaisia teoksia tai asettaa rajoituksia johdannaisen teoksen tyyppille tai luonteelle [Rosen, 2004]. Lisenssi voi kuitenkin sisältää muita muunteluun liittyviä ehtoja, kuten vaatimuksen julkaista kaikki tehdyt muutokset, tai että johdannaisten teosten tulee käyttää alkuperäisestä ohjelmasta poikkeavaa nimeä tai versionumeroa [Välimäki, 2005a; OSI, 2005b].

Ohjelmiston lähdekoodiin on taattava halukkaille vapaa ja helppo pääsy, mikä on käytännön edellytys muuntelujen tekemiseksi ohjelmistoon. Tämä on ennakkoehto Rosenin [2004] esittämälle kolmannelle periaatteelle eli johdannaisten teosten luomiselle. Lähdekoodin on oltava siistiä ja tahallisesti vaikeaselkoiseksi tehdyn koodin käyttö ei ole sallittua. Myöskään välimuodot, kuten esisuorittimen tai kielen tulkin tulos, eivät ole sallittuja, vaan lähdekoodin on oltava alkuperäisessä muodossaan [OSI, 2005b]. Mikäli lähdekoodia ei ole sisällytetty ohjelman jakeluun, niin jakelun mukana on oltava selvät ohjeet kuinka lähdekoodin voi saada kohtuullisin kopiokustannuksin tai mieluiten ilman kuluja, esimerkiksi lataamalla Internetistä [OSI, 2005b].

Lisenssi ei saa asettaa rajoituksia muille ohjelmistoille, jotka jaetaan lisensoidun ohjelman mukana [OSI, 2005b]. Näin ollen on kiellettyä asettaa ehtoja muille samalle tallennusmedialle tai tietokoneen muistiin tallennetuille ohjelmille [Rosen, 2004]. Lisenssi ei esimerkiksi saa vaatia, että kaikki muut samalla medially levitetyt ohjelmat ovat avointa lähdekoodia [OSI, 2005b].

Avoimen lähdekoodin lisenssit antavat käyttäjille paljon vapauksia yksinoikeuslisensointiin tottuneen näkökulmasta. Perinteinen ajattelu ohjelmistoteollisuuden lisensointikäytännöistä ja aineettoman omaisuuden oikeuksista järkkyy, kun avoimen lähdekoodin lisenssit antavat käyttäjälle oikeuden käyttää, kopioida, muunnella ja levittää ohjelmaa [Välimäki, 2005a]. Avoimen lähdekoodin maailmassa yrityksen ansaitsemismalliksi lisenssimaksut sopivat huonosti monilisensointia lukuun ottamatta, mutta avoin lähdekoodi mahdollistaa useita muita liiketoimintamuotoja palveluista ohjelmistojen räätälöintiin ja avoimen lähdekoodin hyödyntämiseen omissa tuotteissa. Tässä tutkimuksessa ei paneuduta syvällisesti avoimen lähdekoodin mahdollistamiin ansaitsemis- ja liiketoimintamalleihin, vaan niitä käytetään tarvittaessa vain esimerkkeinä.

FSF:n mukaan termit vapaa ohjelmisto ja avoin lähdekoodi kuvaavat samaa ohjelmistokategoriaa, mutta kertovat eri asioita ohjelmistoista ja niihin liittyvistä arvoista. Näin ollen FSF jatkaakin vapaa ohjelmisto -termin käyttöä tuodakseen esiin, että myös vapaus, teknologian ohella, on tärkeää [Stallman, 2002].

## **2.5. Avoimen lähdekoodin hyödyt ja riskit hyödyntäjille**

Yrityksen ei pidä rynnätä avoimen lähdekoodin maailmaan punnitsematta ensin avoimen lähdekoodin hyötyjä ja riskejä omalle toiminnalle. Yrityksen liikeideasta ja toimintamallista riippuu kuinka paljon avoin lähdekoodi voi parantaa yrityksen liiketoimintaa, ja miten keskeiseksi tekijäksi se kannattaa ottaa mukaan yrityksen toimintastrategiassa. Voidaan kuitenkin todeta, että avointa lähdekoodia pystytään hyödyntämään useimmissa yrityksissä, joissa on käytössä tietoteknisiä ratkaisuja.

Seuraavaksi esitellään avoimen lähdekoodin hyötyjä ja riskejä. Tarkoitus ei ole tarjota kaikenkattavaa listaa vaan tuoda esiin eräitä olennaisia seikkoja, joita kannattaa pohtia mietittäessä yrityksen strategiaa avoimen lähdekoodin käytössä.

### **2.5.1. Avoimen lähdekoodin hyödyt**

Avoimen lähdekoodin ohjelmistojen ei tarvitse olla ilmaisia, mutta yleensä ne ovat sitä. Internetissä on valtavasti avoimen lähdekoodin ohjelmia, jotka ovat kaikkien saatavilla ilman maksua. Vaikka avoimen lähdekoodin ohjelmisto alun perin maksaisi jotakin, niin siitä saa tehdä kopioita ja käyttää niin monessa tietokoneessa kuin haluaa.

Ohjelmien ilmaisuus voi madaltaa yrityksen perustamiskynnystä ja vähentää taloudellista riskiä, kun yrityksen pääomaa ei tarvitse sitoa ohjelmistolisensseihin. Lisenssien osuus yrityksen menoerissä voi olla pitkällä aikavälillä pieni, mutta varsinkin yrityksen alkutaipaleella avoimen lähdekoodin ohjelmistojen tuoma säästö voi olla merkittävä. Avoimen lähdekoodin lisensseillä ei myöskään ole ylläpito- tai versiopäivitysmaksuja. Yksinoikeusohjelmaan päätyvä yritys ei voi olla tietoinen kaikista tulevaisuudessa koittavista välttämättömistä päivitysoperaatioista ja niiden kustannuksista.

Avoin lähdekoodi takaa riippumattomuuden yhdestä toimittajasta. Yksinoikeudella lisensoidun ohjelman hyviin puoliin yleensä lasketaan parempi tuki, jonka toimittaja tai kolmas osapuoli tarjoaa. Tämä pitää usein paikkansa ja tukipalvelut ovatkin monen yksinoikeusohjelmia tarjoavan yrityksen keskeistä liiketoimintaa, johon panostetaan merkittävästi. Sitoutuminen yhden toimittajan suljettuun ratkaisuun on kuitenkin suuri riski. Yksinoikeusohjelman tuki saattaa loppua yllättäen esimerkiksi toimittajayrityksen lopettamiseen, yritys-

ostoon, yritysfuusioon tai yrityksen päätökseen olla tukematta enää kyseistä tuotetta. Tällöin ohjelma on tullut evoluutionsa päätepisteeseen, jossa sitä ei enää kehitetä tai soviteta uusiin käyttötarkoituksiin. Asiakas on neuvoton tilanteessa, jossa tukea ei saa toimittajalta eikä kukaan muukaan voi auttaa, koska ohjelman lähdekoodi ei ole saatavilla, ja ohjelman muuttaminen voi olla lisenssiehtojen vastaista.

Avoimen lähdekoodin ohjelmiston rakenne, tiedostomuodot ja rajapinnat ovat avoimia. Tämän vuoksi ohjelmien integraatio ja laajennettavuus sekä tietojen siirrettävyys toisiin ohjelmiin on mahdollista. Yksinoikeusohjelmissa on usein valittu suljetut ratkaisut kilpailuedun saavuttamiseksi. Tämä lisää riippuvuutta yhdestä toimittajasta.

## **2.5.2. Avoimen lähdekoodin riskit**

### **2.5.2.1 Omat immateriaalioikeudet**

Perinteisesti yrityksen tuottamaa lähdekoodia suojellaan huolellisesti ulkopuolisilta ja sitä pidetään yrityksen tärkeänä kilpailuvalttina. Eräät avoimen lähdekoodin lisenssit tekevät kaikkensa, jotta kaikki lähdekoodi olisi vapaata ja kaikkien saatavilla. Tällaisen lisenssin alaisen lähdekoodin käyttäminen omissa sovelluksissa voi aiheuttaa sen, että yritys ei voi sulkea edes itse toteuttamaansa lähdekoodia kilpailijoiltaan. Tällaista mahdollisuutta pidetään monissa yrityksissä erittäin suurena riskinä, ja varmuuden vuoksi myös sallivammilla avoimen lähdekoodin lisensseillä varustettuja ohjelmistoja ja komponentteja välteään – yleensä täysin turhaan.

### **2.5.2.2 Tekninen laatu**

On helppo olettaa, että yritysten ammattitaitoisen osaamisen ja laadunvalvonnan piirissä ohjelmista tulee luotettavampia ja parempia kuin avoimen lähdekoodin vastineista, tai että yritysten tulosvastuu ja kilpailu takaavat sen, että yksinoikeusohjelmia kehitetään jatkuvasti paremmiksi. Käytännössä asia ei ole näin mustavalkoinen. Virheellisesti toimivia ohjelmia on niin yksinoikeusohjelmissa kuin avoimen lähdekoodin ohjelmissa.

Eric Raymond [2000] on nimennyt Linus Torvaldsin esittämän lauseen *given enough eyeballs, all bugs are shallow* Linuksen laiksi. Linuksen lausahdus kuvaa sitä, että kun tarpeeksi moni henkilö tarkastelee ohjelmaa ja sen lähdekoodia, niin lopulta kaikki virheet löydetään ja korjataan. Tässä on yksi avoimen lähdekoodin suurista vahvuuksista: ohjelman kehitysprosessiin voi osallistua kuka tahansa. Vapaaehtoisista muodostuu parhaassa tapauksessa resurssi, johon millään yrityksellä ei olisi varaa. Esimerkiksi yritystason käyttöjärjestelmän val-

mistaminen ja ylläpito tulisi erittäin kalliiksi [Sutor, 2005], mutta GNU/Linux on tehty isoksi osaksi harrastajien voimin ja ilmaiseksi kaikkien käyttöön, ja sen ylläpito ja kehitys jatkuu koko ajan.

Nykyään monet yritykset, kuten IBM ja Google, osallistuvat avoimen lähdekoodin ohjelmistojen kehitykseen, joten myös avoimen lähdekoodin projektit pääsevät osalliseksi yritysten laaduntarkkailusta ja ammattiosaamisesta. Lisäksi on muistettava, että avoimen lähdekoodin ohjelmistojen kehittäjissä on mukana paljon myös ammatikseen ohjelmistoja toteuttavia.

### 2.5.2.3 Vastuut ja takuut

Avoimen lähdekoodin lisenssit eivät yleensä anna mitään takuita ohjelmiston toimivuudesta. Tämä onkin ymmärrettävää, sillä kuka tahansa voi levittää ohjelmiston muunneltuja versioita samoilla lisenssiehdoilla, eikä ohjelmiston alkuperäinen kehittäjä voi millään varmistua ohjelmiston kaikkien versioiden toimivuudesta tai laillisuudesta. Avoimen lähdekoodin ohjelmistot ovat usein ilmaisia, joten vastuu- ja takuurajoitukset ovat tämänkin vuoksi ymmärrettäviä. GPL ilmaisee lisenssin saajan vastuun ohjelmiston laadusta seuraavasti [OSI\_GPL, 1991] (käännös Välimäen [2001]):

... Lisenssin saajalla on kaikki riski Ohjelman laadusta ja suorituskyvystä.  
Jos ohjelma osoittautuu virheelliseksi, Lisenssin saajan vastuulla ovat kaikki huolto- ja korjauskustannukset.

Mikäli GPL-lisenssoitu ohjelmisto ei täytä luvattuja ominaisuuksia tai toimii väärin, niin lisenssin saaja on omien neuvojen varassa, ellei ohjelmiston toimittaja ole antanut ohjelmistoon erillistä takuuta.

Vastuu asiakkaalle toimitetusta avoimen lähdekoodin ohjelmistosta on lähdekohtaisesti toimittajalla eikä ohjelmiston alkuperäisellä tekijällä, joten ohjelmiston toimivuus on hyvä testata huolella ennen sen levittämistä omille asiakkaille. Vaikka avoimen lähdekoodin ohjelmiston lisenssi irtisanoutuu vastuista ja takuista, niin saman ohjelmiston levittäjä saa halutessaan tarjota ohjelmistolle takuun ja myös periä siitä maksun.

### 2.5.2.4 Patentti-, tekijänoikeus- ja yhteensopimattomuusriskit

Monien avoimen lähdekoodin lisenssien ehdot ovat epäselvät ja tulkinnanvaraiset. Avoimen lähdekoodin lisenssien ehtojen pätevyydestä ja tulkinnasta ei ole juurikaan olemassa ennakkotapauksia, joten lisenssien käyttäminen voi aiheuttaa oikeudellisen riskin.

Avoimen lähdekoodin komponenttiin tai sen osaan voi kohdistua tuntemattoman kolmannen osapuolen patentti, nk. piilevä patentti. Osa patenteista on hyvin tunnettuja, esimerkiksi eräät tietoturva-algoritmit ja multimedian pakkausalgoritmit, kuten MP3, ja ne pystytään jossain määrin välttämään. Tuntemattomat, yllättäen esille tulevat patentit ovat kaikkein hankalimpia. [Välimäki, 2005d]

Avoimen lähdekoodin ohjelmiston omistusoikeuksissa voi olla epäselvyyksiä. Tällainen tilanne voi syntyä esimerkiksi kehitettäessä avoimen lähdekoodin ohjelmistoa työajalla. Avoimen lähdekoodin lisenssi saattaa loukata työnantajan tekijänoikeutta, jos työntekijällä ei ole oikeutta lisensoida kirjoittamaansa lähdekoodia ilman työnantajan suostumusta [Välimäki, 2002].

Teosta, jonka tekijää ei voida varmuudella selvittää, ei voida käyttää sellaisenaan ilman oikeudellisia riskejä [Nykänen, 2004]. Vaikka tekijä tiedettäisiinkin, niin aina ei voida olla varmoja kaikkien ohjelman osien lisensseistä. Ohjelma voi sisältää lähdekooditiedostoja, joissa ei ole mitään merkintää koodien käyttö-, kopiointi-, levitys- ja muunteluehdoista. Lähdekoodi on voitu liittää ohjelmaan luvatta, tai lisenssiehdot on mahdollisesti unohdettu kirjoittaa tiedostoon. Joka tapauksessa tällaiset tilanteet muodostavat riskejä, jotka on tiedostettava.

Avoimen lähdekoodin lisenssejä on useita kymmeniä erilaisia. Kaikki lisenssit eivät ole yhteensopivia keskenään, ja yhteensopimattomuuksien takia kaikkia ohjelmistoja ei voida yhdistää keskenään synnyttämättä lisenssiristiriitoja. Isoissa ohjelmistoprojekteissa saatetaan käyttää kymmenien eri lisenssien alaisia komponentteja, ja näiden kaikkien yhteensovittaminen voi olla suuri haaste niiden toisistaan poikkeavien ehtojen takia. Lisenssirikkomukset ovat samalla tavalla tekijänoikeuden loukkauksia avoimen lähdekoodin ohjelmistojen kuin yksinoikeusohjelmienkin tapauksessa, ja niitä on pyrittävä välttämään kaikissa tilanteissa.

#### **2.5.2.5 Käyttöönotto**

Avoin lähdekoodi tuo yrityksen hallinnolle uuden haasteen: kuinka tiedetään mitä kaikkea avointa lähdekoodia käytetään ja missä yrityksen tuotteissa ja työkaluissa. Yritysten onkin tehtävä selvät pelisäännöt avoimen lähdekoodin käytölle ja valvottava sääntöjen noudattamista.

Tiettyyn tarkoitukseen sopivien ja luotettavien avoimen lähdekoodin komponenttien ja ohjelmien löytäminen ja valitseminen kymmenien tai jopa satojen vaihtoehtojen joukosta on hidasta ja kallista. Harvalla avoimen lähdekoodin projektilla on varaa markkinoida tuotteitaan ja sopivien ohjelmien etsiminen jää käyttäjän tai erityisten asiantuntijoiden vastuulle. Yksinoikeusohjelmien tun-

nettuus on monissa tapauksissa paljon parempi ja tuntemattomamman avoimen lähdekoodin vaihtoehdon valitseminen voi arveluttaa. Yrityksen voikin olla järkevää pyytää konsultointiapua haluttujen ohjelmistojen löytämiseksi, ellei tarvittavaa tietoa löydy yrityksen sisältä.

Hyödynnettävän ohjelmiston tai komponentin käyttötarkoitus ja -tapa pitää tuntea ennen kuin sitä voidaan hyödyntää täysipainoisesti. Tätä edesauttaa ohjelmiston hyvä toteutus ja dokumentointi. Avoimen lähdekoodin ohjelmistokomponenttien dokumentoinnissa voi olla puutteita, jolloin komponentin täysipainoinen hyödyntäminen vaatii ylimääräistä työtä. Mikäli komponentti on suosittu, niin suurella todennäköisyydellä sen käyttöön saa apua ja mahdollisesti jopa ilmaiseksi.

Siirtyminen vanhoista ja tutuista ohjelmistoista uusiin tuo mukanaan haasteita. Jos yrityksen yksinoikeusohjelmat tai osa ohjelmista korvataan avoimen lähdekoodin vaihtoehdoilla, niin henkilöstöä joudutaan todennäköisesti kouluttamaan. Henkilöstö voi olla muutosta vastaan, mikäli se ei vaikuta tuovan selviä parannuksia heidän työhön. Koulutuksella ja avoimella tiedottamisella muutosvastarintaa kyetään lieventämään.

#### **2.5.2.6 Ylläpito ja muutoksenhallinta**

Aktiivisten avoimen lähdekoodin projektien tahti voi olla liian nopea ja muutokset ohjelmistoihin suuria. Kehitys voi tapahtua käytettävyyden ja vakauden kustannuksella, kun ohjelmistoon toteutetaan nopealla tahdilla uusia toimintoja. Versiopäivitysten kanssa tuleekin olla huolellinen ja varovainen ja tarvittaessa käyttää vain kokemuksen perusteella vakaiksi ja toimiviksi todettuja versioita.

Yksinoikeusohjelman tapaan myös avoimen lähdekoodin ohjelma voi vaihua hiljaiseloon, alkuperäisen projektin toiminta voi kuihtua olemattomaksi tai koko projekti päättyä. Avoimessa kehitysprosessissa on kuitenkin mahdollisuus, että joku ulkopuolinen elvyttää ohjelman henkiin, muuntelee sitä tai kirjoittaa koko ohjelman uudelleen – mahdollisesti sovellettavaksi aivan uuteen käyttötarkoitukseen. Asiakkaalla on myös mahdollisuus itse kehittää ohjelmaa tai palkata ulkopuolinen tekemään haluttuja muutoksia. Yksinoikeusohjelman asiakkaalla ei yleensä ole näitä vaihtoehtoja.

#### **2.5.2.7 Kustannukset**

Avoimen lähdekoodin käyttöä perustellaan usein kustannusten leikkaamisella. Onkin totta, että avoimen lähdekoodin käytöllä voidaan saavuttaa merkittäviä säästöjä, mutta mikään itseisarvo säästöjen saavuttaminen ei ole. Avoimen läh-

dekodin käyttö voi tulla ennakoitua kalliimmaksi, jos kaikkia kuluja ei osata ennakoida.

Avoimen lähdekoodin ohjelmiston muuntelu voi tulla kalliiksi, mikäli muutoksia ryhdytään tekemään ilman ohjelmiston rakenteen tarkempaa tuntemusta, mutta jo muutosmahdollisuuden olemassaolo voi rauhoittaa yrityksiä ja olla kriittisessä tilanteessa korvaamaton etu. Yleensä huomattavasti kalliimmaksi tulee vaihtoehto, jossa ohjelmalle ei voida tehdä minkäänlaisia muutoksia ja joudutaan hankkimaan täysin uusi tuote ja kouluttamaan henkilöstö uudelleen.

## 3. Ohjelmistolisenssit

Tietokoneohjelmistoa ei yleensä myydä, vaan se lisensoidaan. Ohjelmisto varustetaan ohjelmistolisenssillä, jossa määritellään lisenssin saajan oikeudet ohjelmistoon. Sama pätee myös ilmaiseksi jaettaviin ohjelmistoihin, eikä ohjelmiston hinnalla olekaan lain näkökulmasta vaikutusta ohjelmistoon myönnettäviin oikeuksiin.

Tässä luvussa kerrotaan aluksi tekijänoikeudesta ja johdannaisesta teoksesta. Tämän jälkeen esitellään lyhyesti yksinoikeuslisenssin periaate ja perehdytään tarkemmin avoimen lähdekoodin lisensseihin, avoimen lähdekoodin lisenssien erityispiirteisiin sekä vaihtoehtoisiin ohjelmistojen levitysmalleihin.

### 3.1. Tietokoneohjelmien tekijänoikeus

Tietokoneohjelmia suojataan Suomessa tekijänoikeuslain 27 §:n 2 momentin mukaan kirjallisina teoksina [HE, 2005]. Lisäksi Euroopan Unionin direktiivissä tietokoneohjelmien oikeudellisesta suojasta säädetään tietokoneohjelmistojen tekijänoikeudesta [ETY, 1991]. Suomen tekijänoikeuslain mukaan tekijänoikeudellinen suoja ulottuu ohjelmakoodiin sekä lähdekoodimuodossa että objektiokoodimuodossa. Ohjelman muut osat, kuten käyttöliittymän kuvat ja äänet, voivat saada suojaa muun tyyppisinä teoksina [Nykänen, 2004].

Tekijänoikeuslain mukaisesti tekijänoikeus tietokoneohjelmaan syntyy sille, joka on kirjoittanut ohjelman [HE, 2005]. Poikkeuksena tekijänoikeuslain 40 b §:ssä on säädetty, että työ- ja virkasuhteessa luotujen tietokoneohjelmien ja niihin välittömästi liittyvien teoksien tekijänoikeus siirtyy työnantajalle, mikäli tietokoneohjelma on luotu täytettäessä työsuhteesta johtuvia tehtäviä [HE, 2005; Nykänen, 2004].

Jos ohjelmiston tuotantoprosessi on hajautettu, myös tekijänoikeuden omistus hajautuu [Välimäki, 2002]. Avoimen lähdekoodin ohjelmistoprojekteissa yhdellä ohjelmistolla voi olla kymmeniä, satoja tai jopa tuhansia omistajia, sillä



projekteihin voi teoriassa osallistua kuka tahansa, missä päin maailmaa tahansa.

Tekijänoikeuslaki rajoittaa ohjelmistojen kopiointia ja muuntelua. Lähtökohtana on, että tietokoneohjelmasta saa ottaa vain välttämättömät varmuuskopiot, ja muu kopioiminen on yleensä kielletty [Nykänen, 2004]. Tietyn tyyppistä ohjelman muuntelua, kuten takaisinmallinnusta, ei voida pätevästi kieltää, jos se on välttämätöntä ohjelman saamiseksi toimimaan muiden ohjelmien kanssa [Nykänen, 2004; HE, 2005].

Tietokoneohjelmien lisensointi perustuu tekijänoikeuslain 27 §:n 2 momenttiin [HE, 2005], jonka mukaan teoskappaleen luovutus ei sisällä tekijänoikeuden luovutusta, ellei tästä ole erikseen nimenomaisesti sovittu [Välimäki, 2002]. Tekijänoikeuden haltija siis antaa lisenssisopimuksella ohjelmaan rajoitetun käyttöoikeuden, mutta ei luovu tekijänoikeudesta ohjelmaan [Välimäki, 2002].

### 3.2. Johdannainen teos

Kaikki avoimen lähdekoodin lisenssit sallivat tekijänoikeuslain mukaisten johdannaisten, ts. jälkiperäisten, teosten luomisen, mutta monissa niistä on tarkat ehdot johdannaisten teosten levitykselle. Aina ei ole itsestään selvää, onko tietokoneohjelma lain mukaan toisen ohjelman johdannainen teos. Lisätyötä tulkintatilanteisiin aiheuttavat Yhdysvaltojen ja EU:n tekijänoikeuslakien eroavaisuudet johdannaisten teoksen määrittelyssä. [Välimäki, 2005a]

Yhdysvalloissa tuomioistuimet ovat määritelleet, että ollakseen johdannainen teos tietokoneohjelman täytyy olla oleellisesti *samankaltainen* ja jossain muodossa *sisältää osa* tekijänoikeuden alaisesta työstä. EU:n tekijänoikeuslain määritelmä johdannaisten teoksesta on Yhdysvaltojen lakia laajempi ja tiukempi. EU:ssa *mikä tahansa muutos* olemassa olevaan teokseen luo johdannaisten teoksen. Yhdysvalloissa johdannaisten teoksen täytyy *perustua* olemassa olevaan työhön. [Välimäki, 2005a]

Esimerkiksi GPL tulkitsee tekijänoikeuslakia siten, että ohjelma, joka sisältää edes osan toisesta teoksesta, on automaattisesti kyseisen teoksen johdannainen teos. GPL:n mukaan sillä ei ole väliä onko hyödynnetty teos tai sen osa otettu muuttamattomana tai muutettuna mukaan toiseen teokseen. Myöskään siihen GPL ei ota kantaa, että täytyykö luodun teoksen olla samankaltainen alkuperäisen teoksen kanssa ollakseen johdannainen teos. GPL:n tulkinta johdannaisten teoksesta on näin ollen samankaltainen eurooppalaisen määritelmän kanssa [Välimäki, 2005a].

### 3.3. Yksinoikeuslisenssit

Yksinoikeusohjelma on lisensoitu jollakin yksinoikeuslisenssillä, joka antaa lisenssin saajalle yleensä tiukasti rajatun käyttöoikeuden ohjelmaan. Tavallisesti yksinoikeuslisenssi sallii ohjelman kopioinnin vain käytön yhteydessä ja rajoittaa ohjelman asennusten määrää. Yksinoikeusohjelma sekoitetaan usein kaupallisen ohjelman kanssa, vaikka termit voivat tarkoittaa eri asioita. Kaupallista ohjelmaa levitetään myymällä ja rahan ansaitsemiseksi ohjelman avulla, mutta yksinoikeusohjelmaa voidaan levittää myös ilmaiseksi [FSF, 2005c]. Useimmat kaupalliset ohjelmat ovat yksinoikeusohjelmia, mutta on olemassa kaupallisia vapaita ohjelmia ja ei-kaupallisia yksinoikeusohjelmia [FSF, 2005c]. Esimerkki kaupallisesta vapaasta ohjelmasta on MySQL-tietokantajärjestelmä ja ei-kaupallisesta yksinoikeusohjelmasta freeware-ohjelmat. Kaupallisia yksinoikeusohjelmia hinnoitellaan mm. käyttäjien, käyttökopioiden lukumäärän tai käyttöajan mukaan [Välimäki, 2002].

Yksinoikeusohjelmia toteuttavat ohjelmistoyritykset voivat vielä nykyäänkin olla hyvin varautuneita avoimen lähdekoodin suhteen. Varautuneisuutta lisäävät epätietoisuus, väärät tulkinnat ja julkisuudessa esitetyt virheelliset väitteet [Greene, 2001].

### 3.4. Avoimen lähdekoodin lisenssit

Avoimen lähdekoodin lisenssillä lisensoitu tietokoneohjelma täyttää Open Source Initiativen avoimen lähdekoodin määritelmän ehdot, ja sitä saa kopioida, muunnella, levittää ja käyttää rajoituksetta. Avoimen lähdekoodin lisenssit sanelevat niillä lisensoitujen ohjelmien ehdot. Avoimen lähdekoodin ohjelmien levittäminen täysin ilmaiseksi ei vaikuta mitenkään lisenssiehtojen noudattamiseen. Avoimen lähdekoodin lisenssit on otettava yhtä vakavasti kuin yksinoikeuslisenssit, vaikka avoimen lähdekoodin ohjelmistot ovat yleensä saatavilla ilmaiseksi: ohjelman hinta ei mitenkään vaikuta velvollisuuteen noudattaa lisenssiehtoja.

#### 3.4.1. Avoimen lähdekoodin lisenssin pysyvyys, tarttuvuus ja sallivuus

Eräillä avoimen lähdekoodin lisensseillä on erityispiirteitä, joita ei tapaa yksinoikeuslisensseissä ja jotka voivat tulla yllätyksenä ainoastaan perinteisiin yksinoikeuslisensseihin tottuneille. GPL toi ohjelmistolisenssien maailmaan kaksi uutta ominaisuutta – lisenssin *pysyvyys* (persistence) ja *tarttuvuus* (inheritance). Ne on luotu säilyttämään ohjelmiston keskeiset vapaudet myös muuntelun ja uudelleenlevityksen jälkeen. Yksinoikeuslisensseissä tällaisia ominaisuuksia ei

ole, koska niiden lähdekoodi ei tavallisesti ole vapaata, eikä ohjelmien muunte-  
lua yleensä sallita.

Lisenssin pysyvyys, ts. *tavanomainen vastavuoroisuusehto* (standard reciprocity obligation), tarkoittaa, että kaikki teoksen muunnelmien on lisensoitava kokonaisuudessaan teoksen alkuperäisellä lisenssillä [Välimäki, 2005a]. Avoimen lähdekoodin maailmassa tämä tarkoittaa sitä, että teos pysyy avoimena, vaikka sen lähdekoodia muutettaisiin; näin kukaan ei voi asettaa toiselle enemmän rajoituksia kuin itsellä on [FSF, 2005b]. Pysyvyys takaa sen, että alkuperäiseen ohjelmaan tai sen muunnoksiin tehdyt muutokset palautuvat takaisin yhteisölle, jos ohjelmaa levitetään.

Lisenssin tarttuvuus, ts. *voimakas vastavuoroisuusehto* (strong reciprocity obligation), on lisenssin pysyvyyden laajennus, joka säilyttää lisenssiehdot jo pelkästään yhdistettäessä lähdekoodia toisen lisenssin alaiseen lähdekoodiin. Toisin sanoen, jos tarttuvan lisenssin alaisen ohjelmiston tai ohjelmiston osan yhdistää toisen lisenssin alaiseen ohjelmistoon, niin tarttuva lisenssi syrjäyttää toisen lisenssin, ja syntynyt teos on lisensoitava yksinomaan kyseisellä tarttuvalla lisenssillä. Näin siis teoriassa, sillä tarttuvuuden vaikutus on tulkinnanvarainen, eikä ole aina selvää, miten laaja tarttuvan lisenssin vaikutusalue on, ja miten pienen lähdekoodimäärän kopiointi aiheuttaa lisenssin tarttumisen kokonaisuuteen. Lisenssin tarttuvuutta kutsutaan joissain yhteyksissä, erityisesti negatiivisessa sävyssä puhuttaessa, *virusvaikutukseksi* (viral effect).

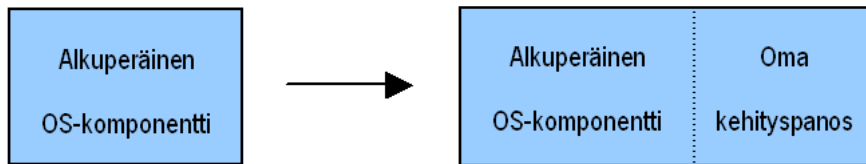
Pysyvät ja tarttavat lisenssit asettavat paljon velvoitteita muunneltujen ohjelmistojen levittäjille. *Sallivat* (permissive) lisenssit antavat lisenssin saajalle paljon enemmän vapauksia eivätkä aseta juuri mitään velvoitteita. Ne eivät vaadi lähdekoodin julkaisua, kun ohjelmistoa levitetään, ja voivat jopa sallia lisenssiehtojen muuttamisen alkuperäisen lähdekoodin muunnelmissa [Välimäki, 2005a].

Kuvissa 2-4 on havainnollistettu avoimen lähdekoodin lisenssien toiminnallisuuden eroja, kun tarttuvalla, pysyvällä ja sallivalla lisenssillä lisensoitua komponenttia muunnellaan tai hyödynnetään toisen ohjelmiston osana. Kun komponentin lähdekoodia muuttaa, niin tuloksena on alkuperäisen teoksen johdannainen teos. Yhdistetty tai uusi teos voi syntyä esimerkiksi käyttämällä komponenttia toisesta ohjelmasta komponentin tarjoaman rajapinnan kautta tai linkittämällä komponentti dynaamisesti toiseen ohjelmaan.

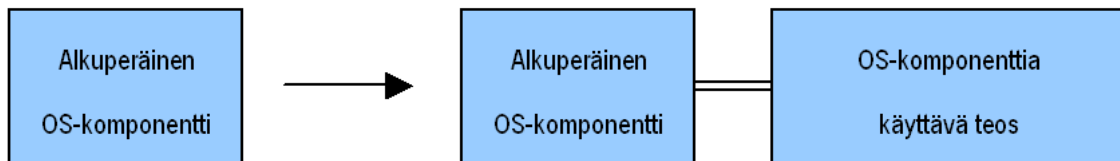
---

## Tarttuva lisenssi

### a) Kehitystyön tuloksena on johdannainen teos



### b) Kehitystyön tuloksena on uusi tai yhdistetty teos



 Lähdekoodi on julkaistava, jos teosta levitetään

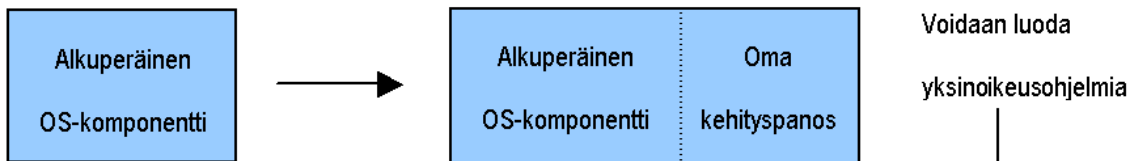
---

**Kuva 2. Tarttuvan lisenssin toiminnallisuus hyödyntämistilanteessa.**

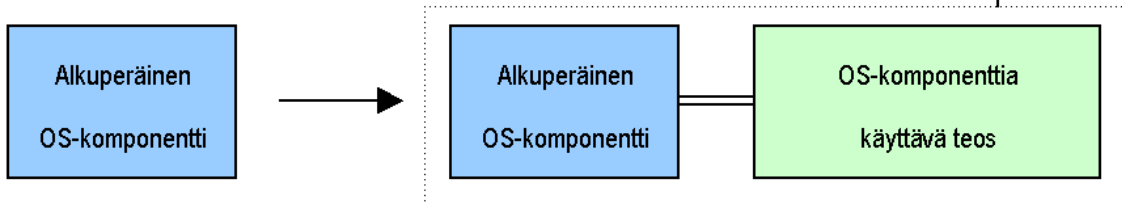
Tarttuvalla lisenssillä (kuva 2) lisensoidun komponentin lisenssi periytyy niin komponentin johdannaiseen kuin komponenttia hyödyntävään teokseen [Välimäki, 2005a]. Pelkkä tarttuvalla lisenssillä lisensoidun lähdekoodin yhdistäminen muuhun lähdekoodiin saattaa edellyttää koko kokonaisuuden lähdekoodin julkistamista, mikäli teosta levitetään. Yksinoikeusohjelmien tekeminen ei ole mahdollista lisenssin tarttuvuusominaisuuden vuoksi.

## Pysyvä lisenssi

### a) Kehitystyön tuloksena on johdannainen teos



### b) Kehitystyön tuloksena on uusi tai yhdistetty teos



Lähdekoodi voidaan pitää suljettuna ja lisenssistä päättää itse

Lähdekoodi on julkaistava, jos teosta levitetään

Kuva 3. Pysyvän lisenssin toiminnallisuus hyödyntämistilanteessa.

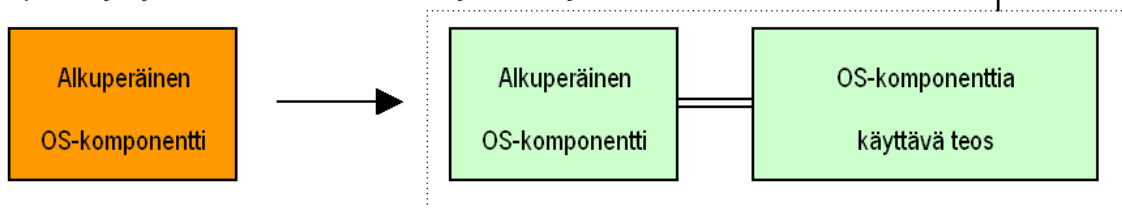
Pysyvällä lisenssillä (kuva 3) lisensoituun komponenttiin tehdyt muutokset täytyy lisensoida alkuperäisen teoksen lisenssiehdoilla. Jos pysyvällä lisenssillä varustetun komponentin yhdistää isompaan kokonaisuuteen, niin uusi teos voidaan lisensoida muilla ehdoilla [Välimäki, 2004], mutta tässäkin tapauksessa pysyvän lisenssin komponentti säilyttää alkuperäiset lisenssiehtonsa. Koska pysyvä lisenssi ei uudessa teoksessa tartu pysyvän komponentin ulkopuolisiin osiin, niin se ei estä yksinoikeusohjelmien luomista, kuten tarttuva lisenssi tekee.

## Salliva lisenssi

### a) Kehitystyön tuloksena on johdannainen teos



### b) Kehitystyön tuloksena on uusi tai yhdistetty teos



- Lähdekoodi voidaan pitää suljettuna ja lisenssistä päättää itse
- Lähdekoodi voidaan pitää suljettuna

Kuva 4. Sallivan lisenssin toiminnallisuus hyödyntämistilanteessa.

Salliva lisenssi (kuva 4) sallii yksinoikeusohjelmien luomisen niin johdannaisista kuin yhdistetyistä teoksista eikä edellytä lähdekoodin julkaisua missään tilanteessa. Mikäli kehitystyön seurauksena syntyy uusi teos, niin koko teoksen saa lisensoida haluamallaan lisenssillä. Sallivalla lisenssillä lisensoitu ohjelma tai komponentti soveltuukin erinomaisesti hyödynnettäväksi yksinoikeusohjelmissa.

### 3.4.2. Avoimen lähdekoodin lisenssien luokittelu

Open Source Initiative oli toukokuussa 2006 hyväksynyt 58 avoimen lähdekoodin lisenssiä [OSI, 2006]. Tämä tarkoittaa, että nämä 58 lisenssiä täyttävät OSI:n hallinnoiman avoimen lähdekoodin määritelmän ehdot, ja niillä lisensoidut ohjelmistot voidaan varustaa OSI Certified -merkinnällä. Koska lisenssejä on niin suuri määrä, niin niiden kaikkien tunteminen ja ymmärtäminen olisi kohtuuton urakka. Lisenssit voidaan onneksi jakaa kategorioihin, jolloin pelkkä kategorian tyyppi kertoo lisenssistä monta olennaista asiaa. Toisaalta vain harva avoimen lähdekoodin lisenssi on yleisessä käytössä, joten kaikkien lisenssien tunteminen ei ole tänäkään vuoksi välttämätöntä.

Välimäki [2005a] käyttää kahta luokittelutapaa lisenssien kategorisoimiseksi. Toiminnallisesta näkökulmasta avoimen lähdekoodin lisenssit voidaan luokitella sen perusteella, miten ne suhtautuvat lähdekoodin muunteluun ja tuloksena syntyviin johdannaisiin teoksiin. Historiallisen alkuperän mukaan luoki-

teltaessa ei keskitytä lisenssien toiminnallisuuteen vaan lisenssien luettavuuteen, laajuuteen, velvoitteisiin ja lähdekoodin tekijänoikeuksiin. Lisenssien toiminnallisuuteen perustuen Välimäki [2005a] jakaa lisenssit tarttuviin, pysyviin ja salliviin lisensseihin ja, historialliseen alkuperään perustuen, GNU-lisensseihin, akateemisiin lisensseihin, yhteisölisensseihin ja yrityslisensseihin.

GNU-lisenssit GPL ja LGPL ylläpitävät ja viestittävät Free Software Foundationin vapaan ohjelmiston ideologiaa. Ne on kirjoitettu ohjelmistosuunnittelijoille ja sisältävät juridisesti katsoen valitettavan paljon epämääräisyyksiä ja aiheuttavat siten tulkintatilanteita [Välimäki, 2005a]. GPL on tarttuva ja LGPL pysyvä lisenssi.

Akateemisten lisenssien alkuperä on akateemisissa instituutioissa, jotka levittivät luomiaan ohjelmistoja vapaasti muille sallien ohjelmistojen käytön ja hyödyntämisen mihin tarkoitukseen tahansa ilman ylimääräisiä velvoitteita [Rosen, 2004]. Akateemiset lisenssit ovat luonteeltaan erittäin sallivia, ja niillä lisensoituja ohjelmistoja voi hyödyntää käytännössä miten tahansa. Jopa ohjelmiston lisenssin muuttaminen on mahdollista, mikäli teosta muutetaan riittävästi [Välimäki, 2002]. BSD-lisenssi on akateemisten lisenssien tyypiesimerkki.

Yhteisölisenssit ovat tyypillisesti peräisin jostain suuresta vapaan ohjelmiston projektista, joka on saavuttanut laajaa suosiota Internetin välityksellä. Yhteisölisenssin taustalla voi olla voittoa tavoittelematon järjestö, kuten Apache Software Foundation tai PHP Group, tai ohjelmistoprojekti, kuten Perl. Yhteisölisenssit voivat poiketa sisällöltään huomattavasti toisistaan: Artistic-lisenssi sisältää paljon moniselitteistä terminologiaa ja useita tulkinnanvaraisia kohtia, Apache 2.0 -lisenssi on juridisesti paljon pätevämpi ja PHP-lisenssi muistuttaa tiiviydellään akateemisia lisenssejä. Yhteisölisenssit ovat yleensä sallivia.

Avoimen lähdekoodin parannettua tunnettuuttaan liike-elämän sektorilla myös yritykset ryhtyivät esittelemään omia avoimen lähdekoodin lisenssejään. Lähtölaukauksen antoi Netscape luomalla ensimmäisen avoimen lähdekoodin yrityslisenssin ja lisensoimalla WWW-selaimensa lähdekoodin sillä. Sittenkin mm. IBM, SUN, Apple ja Nokia ovat kirjoittaneet omat avoimen lähdekoodin lisenssinsä, joille Open Source Initiative on antanut hyväksyntänsä [OSI, 2006]. Yrityslisenssit ovat yleensä ammattilaisjuristien kirjoittamia, erittäin yksityiskohtaisia ja sisältävät vastavuoroisuusehdon.

### 3.4.3. Avoimen lähdekoodin lisenssien yhteensopimattomuus

Kaikki avoimen lähdekoodin lisenssit eivät ole yhteensopivia keskenään, vaan lisenssien ehtojen välillä voi olla ristiriitoja. Kaksi lisenssiä ovat yhteensopivat, jos molemmat lisenssit sallivat niillä lisensoitujen lähdekoodien yhdistämisen ilman, että kummankaan lisenssin ehtoja loukataan. Mikäli molempien lisenssi-

en ehtoja ei pystytä täyttämään, lisenssit ovat yhteensopimattomia keskenään. Lisenssien yhteensopimattomuus on yksi avoimen lähdekoodin riskeistä, johon on suhtauduttava vakavasti. Lisenssien yhteensopimattomuudet on tunnettava ja lisenssiristiriitoja vältettävä, jotta ei syyllistyä lisenssirikkomuksiin. Lisenssiristiriitojen korjaaminen voi aiheuttaa suuria muutostöitä, mikä voi tulla kalliiksi, jos joudutaan korvaamaan ohjelmistoon liitettyjä komponentteja toisilla tai toteuttamaan itse tarvittavat muutokset.

Se, että kaksi lisenssiä täyttävät Open Source Initiativen avoimen lähdekoodin määritelmän ehdot, ei takaa, että lisenssit olisivat yhteensopivia keskenään. Avoimen lähdekoodin määritelmä ei olekaan standardoitu määritelmä, jonka piiriin kuuluvat lisenssit edustaisivat välttämättä samaa ideologiaa, sisältäisivät keskenään samantapaiset ehdot ja toimisivat toisten OSI-aitoustodistuksen saaneiden lisenssien kanssa sulassa sovussa. Lisenssien yhteensopimattomuus hidastaa avoimen lähdekoodin yleistymistä, vaikeuttaa sen hyödyntämistä ja on ylimääräinen IPR-riski käyttäjille [Välimäki, 2004]. GNU-lisenssien suuren suosion myötä yleiseksi puheenaiheeksi onkin noussut GNU-lisenssien yhteensopivuusongelmat muiden lisenssien kanssa, jotka johtuvat suurelta osin Richard Stallmanin tinkimättömästä vapauden ideologiasta [Välimäki, 2005a].

Lisenssien yhteensopimattomuus tarkoittaa käytännössä sitä, että toisilleen yhteensopimattomilla lisensseillä lisensoituja teoksia on lähes mahdotonta yhdistää keskenään. Yhdistettäessä yhteensopimattomien lisenssien lähdekoodia lisenssien ehtojen välille syntyy ristiriitoja, eivätkä ne suostu toimimaan yhteistyössä. Lisenssiehdoiltaan yhteensopimattomien lähdekoodien yhdistäminen loukkaa lähdekoodien omistajien oikeuksia ja on lisenssirikkomus. Esimerkiksi GPL-lisensoitua lähdekoodia ei voi yhdistää Mozilla Public Licensella lisensoituun koodiin [FSF, 2005a].

Lisenssien yhteensopivuus GPL:n kanssa vaatii erityistarkastelun GPL:n laajan suosion ja tarttuvuusominaisuuden takia. Kaikkien lisenssien yhteensopivuusvertailu ristiin ei ole tarpeen, sillä useimmat lisenssit – erityisesti sallivat lisenssit – eivät rajoita yhteensopivuutta mitenkään.

Yhteensopivuus GPL:n kanssa saadaan selville tutkimalla salliiko toinen lisenssi, että GPL syrjäyttää sen: jos lisenssi sallii syrjäyttämisen, niin lisenssit ovat yhteensopivat. Muussa tapauksessa yhteensopivuusongelmien takia lähdekoodeja ei voida yhdistää ilman erillistä lupaa lähdekoodin omistajalta [FSF, 2005d]. Taulukossa 1 on esitetty FSF:n tulkinta tutkimuksessa käsiteltyjen lisenssien yhteensopivuudesta GPL:n kanssa.



**Taulukko 1. FSF:n tulkinnat lisenssien yhteensopivuudesta GPL:n kanssa [FSF, 2005b].**

Lisenssi	Yhteensopiva	Syy yhteensopimattomuuteen
GPL	Kyllä	-
LGPL	Kyllä	-
Uusi BSD	Kyllä	-
MIT	Kyllä	-
Apache	Ei <sup>1</sup>	Sisältää liian laajan patenttiehdon.
PHP	Ei	Sisältää mainostamisehdon.
Artistic	Ei <sup>2</sup>	Epämääräisyytensä takia ei kuulu vapaan ohjelmiston lisensseihin ja on täten automaattisesti yhteensopimaton GPL:n kanssa.
MPL	Ei	Sisältää eräitä monimutkaisia rajoituksia, jotka tekevät lisenssistä yhteensopimattoman. MPL 1.1:ssä on kuitenkin ehto, joka sallii ohjelman (tai ohjelman osan) tarjota myös toista lisenssiä, jolloin ohjelma (tai ohjelman osa) voi olla GPL-yhteensopiva.

<sup>1</sup> ASF:n tulkinnan mukaan lisenssi on yhteensopiva GPL:n kanssa [ASF, 2005a].

<sup>2</sup> Perlin lisenssi (License of Perl) on yhteensopiva GPL:n kanssa [FSF, 2005b].

Monet suuret avoimen lähdekoodin projektit, kuten Mozilla ja Apache, ovat jälkikäteen muuttaneet lisensointipolitiikkaansa GPL-yhteensopivampaan suuntaan [Välimäki, 2005a]. Mozilla on päättänyt lisensoimaan kaikki tuotteen- sa MPL/GPL/LGPL -lisenssiyhdistelmällä taatakseen yhteensopivuuden GNU-lisenssien kanssa [Mozilla, 2005b]. Mozilla-projektin lähdekoodien lisenssien muuttaminen alkoi syyskuussa 2001 ja on vieläkin kesken [Mozilla, 2005b]. Lisenssien yhteensopimattomuudet voivatkin aiheuttaa jälkikäteen suuren määrän ylimääräistä työtä, mikäli lisensointimallia joudutaan muuttamaan ja ohjelmistoa ovat kehittäneet useat ulkopuoliset tahot. Esimerkiksi Linuxin uudelleenlisensointi on käytännössä mahdotonta projektin suuren osallistujamäärän vuoksi. Lähdekoodin lisenssin muuttamiseksi täytyy saada lupa kyseisen koodin tekijältä, ja Linuxin kohdalla kaikkien tekijöiden löytäminen ja luvan saaminen kaikilta heiltä on erittäin epätodennäköistä. Myös MySQL AB on joutunut reagoimaan lisenssien yhteensopimattomuuteen lisäämällä MySQL-tietokantajärjestelmänsä GPL-versioon poikkeuksen [MySQL, 2005] salliakseen PHP-ohjelmien kehityksen MySQL:n päälle ilman yhteensopivuusongelmia GPL:n kanssa [Välimäki, 2005a].

### 3.5. Monilisensointi

Ohjelman omistaja tai omistajat voivat päättää toteuttamansa ohjelmiston lisensoinnista vapaasti. Yleisintä on, että ohjelmisto varustetaan yhdellä lisenssillä, mutta joskus useamman lisenssin käyttö on perusteltua. Saman tuotteen lisensointia useammalla lisenssillä kutsutaan *monilisensoinniksi*.

Monilisensointia käytetään sekä ratkaisemaan lisenssien yhteensopivuusongelmia, kuten mm. Mozilla on tehnyt, että erillisenä liiketoimintamalli-

na [Välimäki, 2003]. Yleisin monilisensoinnin soveltamiskeino on saman ohjelmiston lisensoiminen kahdella eri lisenssillä, joista toinen on avoimen lähdekoodin lisenssi ja toinen yksinoikeuslisenssi. Tutummin tällaista menettelyä kutsutaan *kaksoislisensoinniksi* (dual licensing).

Kaksoislisensointia harjoittava yritys yleensä levittää ilmaiseksi ohjelmistonsa avoimen lähdekoodin lisenssillä varustettua versiota ja myy erillistä maksullista lisenssiä niille, jotka haluavat sisällyttää koodin osaksi omaa yksinoikeusohjelmaansa. Avoimen lähdekoodin lisenssiksi on yleensä valittu vastavuoroisuusehdon sisältävä lisenssi, kuten GPL, jotta muiden kehitystyön tulokset tulevat julkisiksi, ja kaikki – myös ohjelman alkuperäinen omistaja – hyötyisivät tehdyistä parannuksista. Käyttämällä tarttuvaa lisenssiä varmistetaan, että kolmas osapuoli ei voi muuttaa ohjelman lisenssiä ja ryhtyä esimerkiksi myymään sitä omilla ehdoillaan.

Useimpiin kaupallisiin tarkoituksiin tarttuvan lisenssin ehdot ovat liian tiukat. Kaupallisten ohjelmien tekijät haluavat päättää levittämiensä ohjelmien lisensoinnista itse ja välttää niiden lähdekoodien julkaisun. Tätä varten kaksoislisensointimallia käyttävä yritys tarjoaa yksinoikeuslisenssiä, joka voidaan tarvittaessa räätälöidä asiakaskohtaisesti. Kun asiakas hankkii ohjelmistoon yksinoikeuslisenssin, se voi käyttää ohjelmistoa omissa teoksissaan ja levittää teoksia edelleen omille asiakkailleen itse määrittelemillään ehdoilla.

Kaksoislisensointimallisissa ohjelman avoimen lähdekoodin versio on vapaasti saatavilla, mikä parhaassa tapauksessa kasvattaa ohjelman tunnettuutta ja käyttöä merkittävästi. Sen lisäksi, että vapaaehtoiset kehittäjät pääsevät tutkimaan ohjelman lähdekoodia ja toteuttamaan siihen korjauksia ja parannuksia, ilmainen versio toimii ikään kuin kaupallisen ohjelmaversioiden mainoksena [Yeates, 2005]. Norjalainen ohjelmistotalo Trolltech oli pitkään epäilevällä kannalla GPL:n suhteen eikä uskonut sen mahdollistavan yksinoikeuslisenssien myymistä. Kun yritys lopulta valitsi GPL:n Qt-työkalupakkinsa kolmanneksi lisenssiksi, he saivat lisää mediajulkisuutta ja poliittisen hyväksynnän avoimen lähdekoodin yhteisöltä [Välimäki, 2003]. Avoimen lähdekoodin yhteisöt ovat valtava voimavara, joiden tuen saaminen voi auttaa yritystä kehittämään tuotetta entistä paremmaksi ja löytämään sille uusia markkinoita. Onkin tärkeää tuntea ideologiat yhteisöjen toimintojen taustalla ja ottaa yhteisöjen mielipiteet huomioon, jotta saavutetaan paras mahdollinen hyöty yrityksen kannalta ja samalla pidetään yhteisöt tyytyväisinä.

Jotta yksi taho voi päättää ohjelmiston lisensoinnista vapaasti, on sillä oltava kaikki tekijänoikeudet ohjelmistoon. Tämän merkitys korostuu avoimessa kehitysprosessissa, jossa kehitystyötä tekee ennalta arvaamaton määrä ihmisiä hajautetusti. Monilisensointi edellyttää, että kaikkia ohjelmiston käyttö- ja levi-

tysoikeuksia hallitsee yksi taho [Välimäki, 2003]. Muussa tapauksessa ohjelmiston omistajuus hajaantuu, eikä mikään tietty taho voi enää päättää yksin ohjelmiston lisensoinnista. Esimerkiksi MySQL AB hallinnoi MySQL-tietokantajärjestelmänsä kehitystä ja tarkistaa ja kirjoittaa uudelleen kaikki ohjelmistoon toteutetut muiden tekemät muutokset säilyttääkseen jatkossakin täydet tekijänoikeudet tuotteeseen [Välimäki, 2003]. Vaihtoehtoisesti ohjelmaa hallinnoiva taho voi vaatia kaikilta kehitykseen osallistuvilta oikeudet kaikkeen luovutettuun koodiin.

Joskus on tarpeen lisensoida ohjelmiston eri osat eri lisensseillä. Mozillan Firefox-WWW-selaimen käyttöliittymäelementit on lisensoitu toisella lisenssillä kuin itse ohjelma, jotta Firefoxin tavaramerkkiä ei voida käyttää ohjelman epävirallisissa julkaisuissa [Yeates, 2005]. Firefoxin ohjelmakomponentit ovat vapaasti muunneltavissa ja levitettävissä toisin kuin ohjelman käyttöliittymän ulkoasuun liittyvät komponentit.

Monilisensointia voidaan käyttää myös tapauksessa, jossa ohjelma ja ohjelman dokumentaatio lisensoidaan eri lisensseillä – esimerkiksi ohjelma GPL:lla ja dokumentaatio GNU Free Documentation -lisenssillä [Yeates, 2005].

### **3.6. Muita lisensointi- ja levitysmalleja**

1980-luvulla shareware-malli saavutti suuren suosion pienten ja hieman suurempienkin ohjelmien levitys- ja myyntimallina. Freeware- ja public domain-mallit ovat saavuttaneet suosiota tietokoneohjelmien ilmaisjakelumuotoina. Shareware, freeware ja public domain sekoitetaan toisinaan avoimen lähdekoodin kanssa, vaikka ne poikkeavat suuresti avoimen lähdekoodin ajatusmallista. Seuraavaksi käydään läpi mitä nämä kolme käsitettä tarkoittavat. Lisäksi esitellään lyhyesti Creative Commons -sisällönlisensointimalli.

#### **3.6.1. Freeware**

Freeware (ilmaisohjelma) -termillä ei ole yhtä virallista määritelmää, mutta sitä käytetään nykyään yleisesti ohjelmasta, jota levitetään binäärimuodossa ja jonka käytöstä ei peritä maksua. Termin freeware keksi Andrew Fluegelman vuonna 1982 [Ford, 2000]. Fluegelman myös rekisteröi termin tavaramerkiksi [Ford, 2000], mutta tavaramerkki on sittemmin mitätöity [Wikipedia, 2005a]. Freeware-termin käyttökielto kuitenkin ennätti aikoinaan johtaa shareware-termin keksimiseen vuonna 1985 [Ford, 2000]. Alun perin Fluegelmanin freeware tarkoittikin enemmän shareware-mallia kuin freewarea, sillä Fluegelman pyysi freewarena jakamansa PC-Talk-ohjelman käyttäjiä lähettämään rahaa pystyäkseen jatkamaan ohjelman kehitystä [Ford, 2000].

Freewarea ei pidä sekoittaa vapaaseen ohjelmistoon (free software), sillä näiden kahden käsitteen välillä on huomattavia eroja. Freeware-ohjelma on saatavilla ilmaiseksi, mutta sen käyttö, levitys, muuntelu tai tutkiminen ei ole välttämättä vapaata. Ohjelma toimitetaan yleensä ainoastaan binäärimuotoisena ja sen käyttö voi olla ilmaista vain ei-kaupallisiin ja yleishyödyllisiin tarkoituksiin. Kaupallista käyttöä varten on tällöin hankittava erillinen maksullinen versio. Freewaren asettamat rajoitukset käyttäjän vapaudelle erottavatkin sen selvästi vapaan ohjelmiston ideologiasta, eivätkä vapaan ohjelmiston määritelmän ehdot täyty freewaren kohdalla miltään osin.

Kaupallisesta näkökulmasta katsoen freeware-malli toimii tilanteissa, joissa ohjelmistolle halutaan saada mahdollisimman suuri levikki ja ohjelmiston ilmaisuus tietyille käyttäjäryhmälle, esimerkiksi kotikäyttäjille, ei juuri näy yrityksen liikevaihdossa. Onnistuessaan yritys saa ilmaisversion avulla paljon edullista mainosta, joka tuo myös uusia yritysasiakkaita. Esimerkiksi monia virustorjunta- [Grisoft, 2004] ja palomuuriohjelmistoja [Sygate, 2004] levitetään freewarena kotikäyttöön. Muuhun käyttöön on tarjolla erilliset maksulliset versiot, joihin voi sisältyä ilmaisversiota parempi tukipalvelu ja enemmän ominaisuuksia. Riskinä freeware-mallissa on, että käyttäjät käyttävät ohjelman ilmaisversiota myös kiellettyihin käyttötarkoituksiin.

### 3.6.2. Shareware

Shareware (osuusohjelma) tarkoittaa tietokoneohjelman markkinointimallia, jossa käyttäjille tarjotaan tietokoneohjelma rajoitetuksi ajaksi käyttöön ilmaiseksi, ja vasta ohjelman pysyvästä käytöstä vaaditaan maksu. Ohjelmaa saa siis kokeilla ilmaiseksi, esimerkiksi 30 päivän ajan, ennen varsinaista ostopäätöstä. Shareware-mallin ydinajatus on ohjelman vapaa jakelu, johon käyttäjiä kehoitetaan. Shareware olikin 1980-luvulla edullinen keino levittää ja markkinoida ohjelmaa ja parhaassa tapauksessa saavuttaa laaja käyttäjäkunta ja hyvä tuotto. Shareware-ohjelma toimitetaan freeware-ohjelman tavoin binäärimuodossa, eivätkä muutokset ohjelmaan ole sallittuja. Shareware muistuttaakin hyvin paljon perinteistä kaupallista ohjelmaa muutoin kuin jakelutavaltaan ja kokeilumahdollisuudeltaan.

Shareware-mallia voi harkita, mikäli haluaa käyttää mahdollisimman vähän kuluja ohjelman levitykseen ja tarjota kohderyhmälle mahdollisuuden kokeilla ohjelmaa ennen ostopäätöstä. Riskinä on se, että käyttäjät jatkavat ohjelman käyttöä ilmaiseksi rajoitetun ilmaiskäyttöajan umpeuduttuakin. Tämän vuoksi ohjelman ominaisuuksia esittelevän demo-version julkaisu voi olla sharewaremenetelmää järkevämpi vaihtoehto, mikäli käyttäjät saavat demon avulla selkeän kuvan ohjelman toiminnallisuuksista.

Sharewaren suosio oli suurimmillaan 1980-luvulla, ja se on hiipunut sitten 1990-luvun alun [Välimäki, 2005a]. Internet on tarjonnut ohjelmien kehittäjille erinomaisen kanavan ohjelmien markkinoimiseksi ja levittämiseksi. Nykyään on harvinaista, että käyttäjät jakaisivat ohjelmia keskenään tai lehdet sisältäisivät ohjelmaromppuja. Ohjelmien levittämisen sijaan jaetaan tietoa ohjelmien WWW-sivuista, joilta halukkaat käyvät itse hakemassa ohjelmat. Kaupallisista ohjelmista on usein tarjolla demo-versioita tai ominaisuuksiltaan rajoittuneita ilmaisversioita, joiden avulla kiinnostuneet voivat tutustua ohjelmiin ennen ostopäätöstä.

### 3.6.3. Public domain

Public domainiin (julkisohjelma) kuuluva teos tai keksintö on yleistä omaisuutta: vapaasti kaikkien saatavissa, käytettävissä ja jopa myytävissä edelleen. Esimerkiksi Bachin sonaatit, Shakespearen näytelmät, Leonardo da Vincin piirustukset ja Eiffelin tornin suunnitelmat ovat public domainia. [Rosen, 2002]

Tekijänoikeuden alaisesta teoksesta tulee public domainia, kun teos vanhenee riittävästi ja tekijänoikeus umpeutuu. Umpeutumisaika vaihtelee maittain ja teoksen tyyppin mukaan, mutta yleisimmin se on 50-70 vuotta tekijän kuolemasta. Suomen tekijänoikeuslain 73 §:n mukaan tekijänoikeus on voimassa, kunnes 70 vuotta on kulunut tekijän, tai usean tekijän ollessa kyseessä viimeksi kuolleen tekijän, kuolinvuodesta [HE, 2005]. Myös Yhdysvaltojen tekijänoikeuslaissa tekijänoikeus raukeaa 70 vuoden kuluttua tekijän kuolinvuodesta [Rosen, 2002]. Koska tekijänoikeuden umpeutumisaika on ohjelmistojen näkökulmasta sangen pitkä, niin ohjelmistokehitykseen laki ei ole vaikuttanut.

Ohjelmistomaailmassa public domain -ohjelmaksi on totuttu kutsumaan ohjelmaa, jonka tekijä on omasta tahdostaan luopunut oikeuksistaan ohjelmaa kohtaan [Korpela, 2005]. Public domain -ohjelman mukana tulee yleensä joko pelkkä *public domain*- tai *PD*-merkintä tai jonkinlainen luovutuslisenssi, jossa tekijä ilmoittaa luopuvansa oikeuksistaan ohjelmaa kohtaan. Lain näkökulmasta teoksesta ei tällaisen merkinnän tai luovutuslisenssin myötä tule public domainia, ellei tekijän kuolemasta ole kulunut laissa määriteltyä aikaa [Rosen, 2002]. Rosen [2002] suhtautuu public domain -ohjelmiin varauksella, koska niillä ei välttämättä ole lain tukea takanaan, ja suosittelee ennemmin jonkin avoimen lähdekoodin lisenssin käyttämistä vaihtoehtona public domain -julkaisulle. Tällöin koko yhteisö hyötyy saadessaan myös ohjelman lähdekoodin käyttöönsä.

Melko yleinen harhaluulo on, että avoimen lähdekoodin ohjelmistot ovat public domainia. Tämä voi johtua siitä, että vapaan ohjelmiston ja avoimen lähdekoodin periaatteet ovat liian hämmentäviä monille ihmisille, ja he erheel-

lisesti sijoittavat avoimen lähdekoodin tuotteet public domain -ohjelmiin, koska se on lähin käsite, jonka he ymmärtävät. Vapaan ohjelmiston ja avoimen lähdekoodin ohjelmat ovat kuitenkin selvästi tekijänoikeuden alaisia teoksia, joilla on omistaja. Avoimen lähdekoodin lisenssi vain antaa käyttäjälle enemmän vapauksia, kuin mihin useimmat ovat tottuneet. [Perens, 1999]

#### 3.6.4. Creative Commons

Sisältölisenssit antavat taiteellisten töiden, kuten musiikki-, taide-, elokuva- ja kirjallisuusteosten, tekijöille mahdollisuuden levittää teoksiaan eri asteisilla vapauksilla. Internet on tuonut sisällönjakeluun uusia mahdollisuuksia mahdollistamalla teosten levittämisen ilman ulkopuolisia jakelukanavia suoraan yksittäisille ihmisille, ja tämä on avannut mahdollisuuden myös uudelleenlisensointitavalle.

Kaikkein suosituin sisältölisenssien tarjoaja on tällä hetkellä vuonna 2001 käynnistetty voittoa tavoittelematon Creative Commons -projekti [Välimäki, 2005a; CC, 2005a]. Creative Commons -lisensseillä oikeudenhaltija voi luopua hallitusti joukosta oikeuksia, jotka tekijänoikeus lähtökohtaisesti antaa [CC Suomi, 2005]. Esimerkiksi kirjailija voi julkaista kirjansa Internetissä Creative Commons -lisenssiyhdistelmällä, joka antaa lukijoille oikeuden lukea, esittää, kopioida ja levittää kirjaa mutta epää oikeuden käyttää kirjaa kaupallisiin tarkoituksiin tai tehdä johdannaisia teoksia. Oikeudenhaltija saa itse päättää mistä oikeuksista hän haluaa luopua kokoamalla Creative Commonsin 11 lisenssin valikoimasta haluamansa lisenssit. Kaikki 11 lisenssiä on suomennettu ja sovitettu Suomen lainsäädäntöön [CC Suomi, 2005].

Creative Commonsilla on oma lisenssinsä myös public domain -levitykseen [CC, 2005c], ja sen käyttö onkin paljon suositteluvampi vaihtoehto kuin käyttää pelkkää public domain -merkintää tai omatekoista luovutuslisenssiä [CC, 2005b]. Creative Commonsin mukaan jälkimmäinen vaihtoehto ei edes toimi [Wilcox-O'Hearn, 2005].

## 4. Avoimen lähdekoodin lisenssien analyysi

Tässä luvussa tarkastellaan kahdeksaa merkittävää avoimen lähdekoodin lisenssiä. Lisenssien joukossa on sekä sallivia, pysyviä että tarttuvaa lisenssejä. Jokaisen lisenssin osalta esitellään lisenssin alkuperä ja keskeisimmät ehdot. Lisenssiehdot on käännetty suomeksi kirjoittaja, ellei toisin ole mainittu. Käännökset ovat epävirallisia ja lisenssien virallisiin versioihin on viittaukset lähteissä.

### 4.1. Lisenssien valintaperusteet ja yleisyys

Tutkimukseen valittiin tarkempaa käsittelyä varten kahdeksan Open Source Initiativen hyväksymää avoimen lähdekoodin lisenssiä. Nämä lisenssit on lueteltu taulukossa 2. Mikäli lisenssillä on yksilöivä versiotieto, niin myös se on esitetty taulukossa.

Taulukko 2. Tutkimuksessa tarkasteltavat lisenssit.

Lisenssin nimi	Versio
Apache-lisenssi	2.0
Artistic-lisenssi	-
BSD-lisenssi (Berkeley Software Distribution)	uusi
GPL (GNU General Public License)	2.0
LGPL (GNU Lesser General Public License)	2.1
MIT-lisenssi (Massachusetts Institute of Technology)	-
MPL (Mozilla Public License)	1.1
PHP-lisenssi	3.01

Tutkimuksen lisenssit on valittu niiden merkittävyyden ja yleisyyden perusteella. PHP-lisenssi poikkeaa joukosta, koska sillä lisensoitujen ohjelmistojen määrä on hyvin pieni. PHP-lisenssistä tekee erityisen mielenkiintoisen se, että sillä on lisensoitu yhden WWW:n suosituimman [Netcraft, 2002] skriptikielen, PHP:n, käsittelijä [PHP, 2005]. PHP:tä käytetään erittäin laajalti HTTP-

palvelinohjelmisto Apache HTTP Serverin ja MySQL-tietokantajärjestelmän kanssa.

GPL on selvästi yleisin avoimen lähdekoodin lisenssi yli 65 % osuudella kaikista avoimen lähdekoodin projekteista [FLOSSmole, 2006]. Tarkan osuuden määrittäminen on hankalaa, koska avoimen lähdekoodin projekteista ei ole olemassa kaikenkattavaa tietokantaa. Arvioita voidaan kuitenkin tehdä Internetissä olevien avoimen lähdekoodin projektisivustojen, kuten SourceForge.netin [SourceForge, 2006a] ja freshmeat.netin, [freshmeat, 2006a] avulla. Taulukossa 3 on listattu tässä tutkimuksessa käsiteltävien avoimen lähdekoodin lisenssien yleisyys SourceForge.netin ja freshmeat.netin projekteissa.

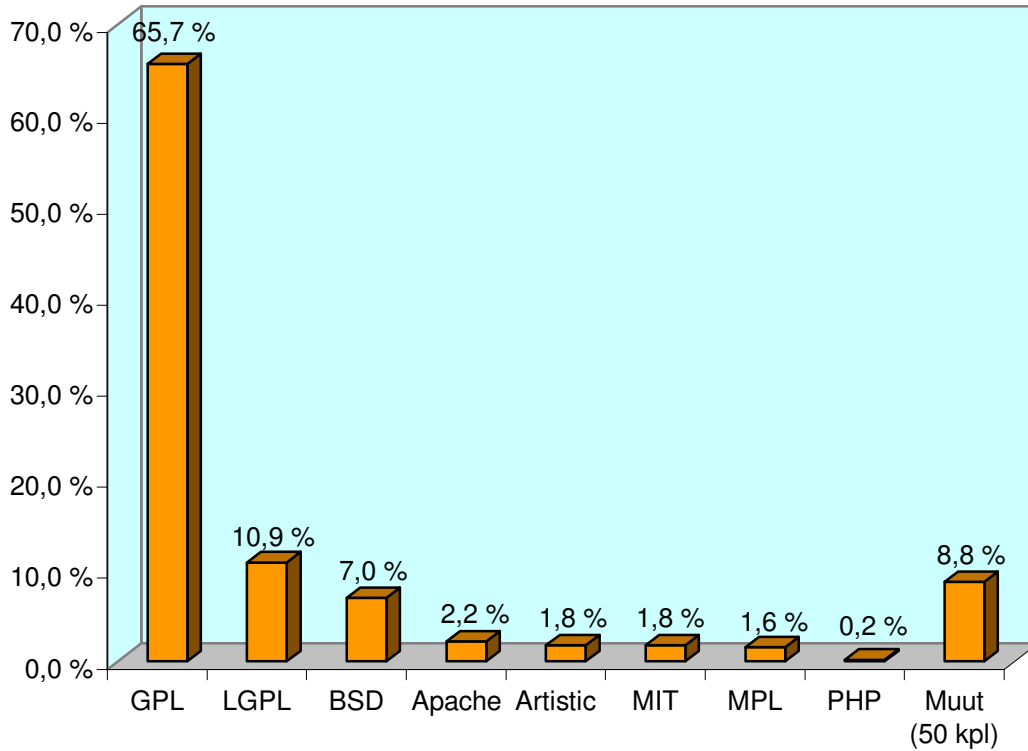
**Taulukko 3. Avoimen lähdekoodin lisenssien yleisyys [FLOSSmole, 2006].**

Lisenssi	SourceForge.net 12/2005 (73 760 projektia)	freshmeat.net 1/2006 (42 522 projektia)
GPL	65,7 %	67,0 %
LGPL	10,9 %	6,2 %
BSD	7,0 %	5,6 %
Apache	2,2 %	1,2 %
Artistic	1,8 %	1,7 %
MIT	1,8 %	1,4 %
MPL	1,6 %	0,6 %
PHP	0,2 %	0,1 %
Muut	8,76 % (50 lisenssiä)	16,20 % (47 lisenssiä)

Taulukosta 3 nähdään, että GNU-lisenssit GPL ja LGPL hallitsevat avoimen lähdekoodin projekteja n. 75 % osuudella kaikista lisensseistä. Myös BSD-lisenssin voidaan todeta olevan suosittu 6-7 % osuudellaan. Näiden kolmen lisenssin takana erot ovat hyvin pieniä, mutta kun huomioidaan, että useimpien lisenssien suosio ei ole edes yhtä prosenttiyksikköä, niin Apache-, Artistic-, ja MIT-lisenssin sekä MPL:n yleisyys on huomionarvoinen.

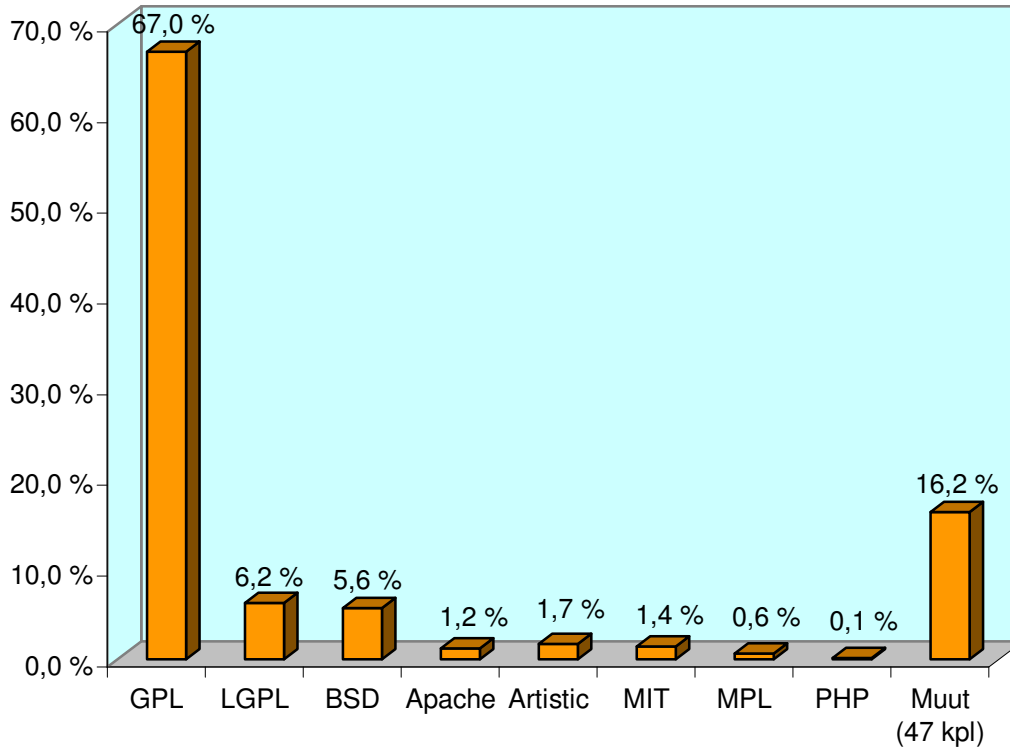
Kuvat 5 ja 6 havainnollistavat GPL:n asemaa ylivoimaisesti suosituimpana avoimen lähdekoodin lisenssinä. Ne osoittavat myös kuinka moneen lisenssiin avoimen lähdekoodin ohjelmat ovat hajaantuneet kolmen suosituimman lisenssin ulkopuolella. On huomioitava, että kuvissa on esitetty vain tässä tutkimuksessa käsiteltävien lisenssien yleisyys, eivätkä ne osoita kaikkien avoimen lähdekoodin lisenssien suosiojärjestystä. Esimerkiksi PHP-lisenssiä yleisempiä lisenssejä on useita, mutta koska tämä tutkimus sisältää PHP-lisenssin tarkastelun, niin sen suosio on esitetty kuvissa erikseen.





Kuva 5. Lisenssien yleisyys SourceForge.netissä [FLOSSmole, 2006].

SourceForge.net on maailman suosituin avoimen lähdekoodin projektien ylläpitösivusto [SourceForge, 2006b]. Kuvasta 5 nähdään, että n. 84 % SourceForge.netin projekteista käyttää GPL:a, LGPL:a tai BSD-lisenssiä. GPL:n asema suosituimpana lisenssinä on kiistaton. Apache-, Artistic- ja MIT-lisenssi sekä MPL ovat vakiinnuttaneet asemansa merkittävinä lisensseinä, mutta PHP-lisenssin suosio on vaatimaton.



Kuva 6. Lisenssien yleisyys freshmeat.netissä [FLOSSmole, 2006].

Freshmeat.net on Internetin suosituin Unix- ja alustariippumattomien ohjelmien ylläpitosivusto [freshmeat, 2006b]. Tätä taustaa vasten voisi olettaa, että perinteiset avoimen lähdekoodin lisenssit ja vapaiden ohjelmistojen yhteisöissä suosittu lisenssi ovat suosiossa myös freshmeat.netissä. Kuvasta 6 nähdään, että näin onkin, sillä GPL:n suosio on jopa suurempi kuin SourceForge.netissä ja toisaalta yrityslisenssiksi luokiteltavan MPL:n yleisyys on hyvin pieni. Kuvasta käy lisäksi ilmi, että freshmeat.netissä lisenssien suosio on hajaantunut selvemmin muihin kuin tässä tutkimuksessa käsiteltäviin lisensseihin.

On huomattava, että edellä esitetyt lukemat kertovat ainoastaan lisenssien yleisyyden kaikissa SourceForge.netissä ja freshmeat.netissä olevissa projekteissa. Yleisyyden lisäksi tärkeitä muuttujia ovat mm. ohjelman latausmäärä, kypsyyden ja merkittävyys sekä projektin aktiivisuus. Krzysztof Klinecicz [2005] on tutkinut SourceForge.netin ylläpitämien projektien innovatiivisuuksia ja todennut, että vain 19 % projekteista on aktiivisia. Lisäksi monet suosittu avoimen lähdekoodin projektit ylläpitävät ohjelmistojaan itse: näin on mm. useimpien GNU/Linux-jakeluiden, Linux-kernelin, Mozillan, Apachen, OpenOffice.org:n ja MySQL:n kohdalla.

Seuraavaksi sovelletaan edellä (luku 3.4.2) esitettyä lisenssien luokittelutapaa ja jaetaan tutkimukseen valitut lisenssit lisenssien toiminnallisuuden ja historiallisen alkuperän perusteella. Tulokset on esitetty taulukossa 4.

Taulukko 4. Tutkimuksessa tarkasteltavien lisenssien luokittelu [Välimäki, 2005a].

Lisenssi	Toiminnallisuus	Alkuperä
GPL	Tarttuva	GNU
LGPL	Pysyvä	GNU
MPL	Pysyvä	Yritys
BSD	Salliva	Akateeminen
MIT	Salliva	Akateeminen
Apache	Salliva	Yhteisö
PHP	Salliva	Yhteisö
Artistic	Salliva	Yhteisö

Toiminnallisuuden mukaan luokiteltaessa tarttuviin lisensseihin kuuluu GPL ja pysyviin lisensseihin kuuluvat LGPL ja MPL. Sallivia lisenssejä ovat BSD-, MIT-, PHP-, Apache- ja Artistic-lisenssi. Historiallisen alkuperän mukaan luokiteltaessa GNU-lisensseihin kuuluvat GPL ja LGPL. Akateemisesta maailmasta ovat kotoisin BSD- ja MIT-lisenssi. Yhteisöllisiin lisensseihin kuuluvat Apache-, Artistic- ja PHP-lisenssi. Yrityspohjan omaa Netscapen julkaisema MPL.

Taulukossa 5 on kuvattu tekijänoikeuden toiminnallisuus eri lisenssityypeissä ja ohjelmistojen levitystavoissa. Taulukosta näkee hyvin, kuinka käyttäjän oikeuksien määrä kasvaa siirryttäessä yksinoikeuslisenssistä vaihtoehtoisten jakelutapojen kautta avoimen lähdekoodin lisensseihin.

Taulukko 5. Tekijänoikeuden toiminnallisuus eri lisenssityypeissä  
(taulukko on mukaelma Välimäen [2005a] taulukosta).

Lisenssi tai levitystapa	Vapaa levitys	Vapaa käyttö	Vapaa muuntelu	Avoin koodi	Pysyvä	Tarttuva
Yksinoikeuslisenssi	-	-	-	-	-	-
Shareware	X	-	-	-	-	-
Freeware	X	-/X <sup>1</sup>	-	-	-	-
Public domain	X	X	X <sup>2</sup>	-	-	-
BSD, MIT, Apache, PHP, Artistic	X	X	X	X	-	-
LGPL, MPL	X	X	X	X	X	-
GPL	X	X	X	X	X	X

<sup>1</sup> Freeware-ohjelma voi olla vapaa kaikkeen käyttöön, mutta se voi sisältää myös ehtoja, jotka esimerkiksi rajoittavat ohjelman käytön vain ei-kaupalliseen toimintaan.

<sup>2</sup> Public domain -ohjelman muuntelu vaikeutuu suuresti, jos ohjelman lähdekoodi ei ole saatavilla.

## 4.2. Velvoittavat lisenssit

### 4.2.1. GPL

GNU General Public License on tunnetuin ja käytetyin avoimen lähdekoodin lisenssi. Lisenssin tunnettuutta on parantanut merkittävimmin GNU/Linux-käyttöjärjestelmä, mutta myös lukuisia muita suosittuja ohjelmistoja on lisen-

soitu GPL:lla: SourceForge.netin ylläpitämistä projekteista yli 46 000 kpl käyttää GPL:a [FLOSSmole, 2006].

GPL syntyi Richard Stallmanin tarpeesta varmistaa, että GNU-projektissa syntyneet ohjelmat säilyvät vapaina siirtyessään käyttäjältä toiselle. GPL kehitettiin suojelemaan käyttäjän oikeuksia eikä rajoittamaan niitä, kuten ohjelmistolisenssit olivat perinteisesti tehneet [Moody, 2001]. Lisenssin ensimmäinen versio julkistettiin vuonna 1989 [FSF, 1989]. Tällä hetkellä GPL on versiossa 2, joka näki päivänvalon vuonna 1991 [FSF, 1991]. Lisenssin kolmannen version on määrä ilmestyä viimeistään vuoden 2007 alkupuolella [FSF, 2005e].

Kuten tässä tutkimuksessa on edellä todettu, GPL tulkitsee johdannaisen teoksen laajempänä käsitteenä kuin Yhdysvaltojen tekijänoikeuslaki tulkinnan ollessa lähempänä eurooppalaista mallia. GPL käyttää lisenssin alaisen ohjelman muunnelmista ja johdannaisista teoksista nimitystä *ohjelmaan perustuva teos*. Lisenssin [OSI\_GPL, 1991] ehdossa 0 määritellään tämä käsite (käännös Välimäen [2001]):

“Ohjelmaan perustuva teos” tarkoittaa joko Ohjelmaa tai mitä tahansa tekijänoikeuslain mukaista jälkiperäistä teosta: toisin sanoen teosta, joka sisältää Ohjelman tai osan siitä, kirjaimellisesti tai muutettuna, tai toiselle kielelle käännettynä.

Ohjelmaan perustuva teos tarkoittaa siis joko itse ohjelmaa tai mitä tahansa sen johdannaista teosta. Johdannaisen teoksen GPL tulkitsee teoksena, joka sisältää ohjelman tai pienenkin osan siitä.

GPL asettaa ehdot ohjelman kopioimiselle, levittämiseksi ja muuttamiselle, mutta ei ohjelman käytölle. Tämä tarkoittaa, että GPL:n alaista ohjelmaa saa käyttää vapaasti, vaikkei hyväksyisikään lisenssin ehtoja. Jotta käyttäjä saisi oikeuden kopioida, muunnella tai levittää ohjelmaa tai sen johdannaista teoksia, niin hänen on hyväksyttävä lisenssin ehdot ja noudatettava niitä. Teosta muuttamalla, kopioimalla tai levittämällä käyttäjä ilmaisee hyväksyvänsä lisenssin ehdot [OSI\_GPL, 1991].

GPL:n kohdassa 1 on ehdot ohjelman lähdekoodin sanatarkkojen kopioiden tekemisestä ja levittämisestä [OSI\_GPL, 1991] (käännös Välimäen [2001]):

1. Lisenssin saajalla on oikeus kopioida ja levittää sanatarkkoja kopioita Ohjelman lähdekoodista sellaisena kuin se on saatu, millä tahansa laitteella. Ehtona on, että asianmukaisesti jokaisesta kopiosta ilmenee kennellä on siihen tekijänoikeus ja että Ohjelmaan ei ole takuuta; edelleen, kaikki viittaukset tähän Lisenssiin ja ilmoitukseen takuun puuttumis-

ta on pidettävä koskemattomana; ja vielä, jokaiselle Ohjelman vastaanottajalle on annettava tämä Lisenssi ohjelman mukana.

Lisenssin saaja voi pyytää maksun Ohjelman kopioimisesta ja voi halutessaan myydä Ohjelmaan takuun.

Yritys voi levittää alkuperäistä ohjelmaa ja periä siitä maksun. Tämä mahdollistaa kenen tahansa aloittaa kaupallinen liiketoiminta, jossa hankitaan GPL-lisenssoitu ohjelmisto ja markkinoidaan ja myydään sitä joko sellaisenaan tai esimerkiksi paketoituna ja itse tehdyillä käyttöohjeilla varustettuna. Maksun perimiseksi ei ole välttämätöntä lisätä ohjelmistoon mitään, mutta niin voi halutessaan tehdä. Toinen ehdon 1 salliva liiketoimintamalli on takuun myyminen GPL-lisenssoituun ohjelmistoon. Kuka tahansa voi myydä takuun, jossa luvataan, että GPL-ohjelmisto toteuttaa tietyt toiminnallisuudet ja tarvittaessa muunnella ohjelmistoa niin, että ohjelmisto todella toimii sovitulla tavalla.

GPL:n tärkeimmät ja eniten julkisuutta saaneet ehdot sisältyvät lisenssin kohtaan 2, joka tekee GPL:sta sekä pysyvän että tarttuvan. Kohta 2 käsittelee GPL:n alaisen ohjelman muuntelua ja muunnellun version levittämistä [OSI\_GPL, 1991] (käännös Välimäen [2001]):

2. Ohjelmakopion tai -kopioiden tai minkä tahansa osan muuttaminen on sallittu. Kun ohjelmaa muutetaan, muodostuu Ohjelmaan perustuva teos. Lisenssin saajalla on lupa kopioida ja levittää näitä muutoksia ja Ohjelmaan perustuvaa teosta ehdolla, että ensinnäkin Kohdan 1 edellytykset täytetään ja lisäksi vielä seuraavat:
  - a) Muuteltujen tiedostojen on sisällettävä selkeä merkintä, josta ilmenee, kuka tiedostoja on muuttanut ja päiväys, jolloin muutokset on tehty.
  - b) Jokainen teos, jonka Lisenssin saaja julkaisee tai levittää edelleen, ja joka kokonaan tai osittain perustuu tai sisältää osia Ohjelmasta, on lisensoitava kokonaisuudessaan ilman maksua kaikille kolmansille osapuolille tämän Lisenssin ehtojen mukaisesti.
  - c) Jos muuteltu ohjelma lukee ajettaessa interaktiivisesti komentoja, Lisenssin saajan on ohjelman käynnistyessä normaaliin interaktiiviseen käyttöön saatettava näytölle tai tulostettavaksi ilmoitus, josta selviää asianmukaisesti ohjelman tekijänoikeus ja ilmoitus, ettei Ohjelmalla ole takuuta (tai vaihtoehtoisesti, että Lisenssin saaja myöntää Ohjelmalle takuun) ja että käyttäjät voivat levittää ohjelmaa

edelleen näiden ehtojen mukaisesti sekä annettava käyttäjille ohjeet, miten he voivat nähdä kopion tästä Lisenssistä. (Poikkeus: jos Ohjelma itsessään on interaktiivinen, mutta ei normaalisti tulosta tällaista ilmoitusta, myöskään Lisenssin saajan tekemän Ohjelmaan perustuvan teoksen ei tarvitse tulostaa ilmoitusta.)

Avoimen lähdekoodin lisensseissä on yleistä, että muutettuihin tiedostoihin vaaditaan selkeä merkintä muuttajasta ja muutospäivämäärästä. Myös GPL vaatii tämän ehdossa 2a). Ehdon tarkoitus on antaa käyttäjälle mahdollisuus tarkistaa, onko hän tekemisissä alkuperäisen vai johdannaisen teoksen kanssa, ja selvittää, mitä lähdekoodeissa on muutettu. Näin voidaan suojella käyttäjiä ja alkuperäisten tekijöiden mainetta virheellisesti toimivien johdannaisten teosten varalta. Muutostietojen avulla voi myös olla mahdollista myös jäljittää muutoksista vastuussa olevat henkilöt. [Laurent, 2004]

Ehto 2b) on GPL:n ehdoista tunnetuin ja mm. Microsoftin pääjohtaja Steve Ballmer on viitannut ehtoon kritisoidessaan GPL:a ja sen niin kutsuttua virusvaikutusta [Greene, 2001; OSI\_GPL, 1991]. Ehdon mukaan johdannaiset teokset on lisensoitava GPL:lla kokonaisuudessaan. Erityisen tärkeä on lause *kokonaan tai osittain perustuu tai sisältää osia*. GPL saattaa tarttua koko teokseen, vaikka siihen liitettävä GPL-lisenssi osa olisi kuinka pieni tahansa.

Yksinoikeusohjelmia toteuttavalle yritykselle GPL:n 2b) on äärimmäisen tiukka ehto, sillä se karsii pois kaikki GPL-lisensoidut ohjelmistot ja ohjelmistojen osat hyödynnettävien ohjelmistojen ja komponenttien vaihtoehdoista. Vähäisenkin GPL-lisensoidun koodin lisääminen yksinoikeusohjelmistoon voi estää tekijää asettamasta mitään vaatimuksia toteuttamansa ohjelmiston suhteen. Näin kuka tahansa voisi kopioida, muokata, levittää ja myydä kyseistä ohjelmistoa välittämättä oikeuksista, jotka muussa tapauksessa annettaisiin tekijälle tekijänoikeuslakien mukaisesti. Yritys ei myöskään voisi sulkea edes itse luomaansa lähdekoodia. [Laurent, 2004]

Monista muista avoimen lähdekoodin lisensseistä poiketen GPL asettaa tiukan rajoituksen johdannaisten teosten hinnoittelulle. Ehdon 2b) mukaan johdannaiset teokset on lisensoitava kokonaisuudessaan ilman maksua kolmansille osapuolille [OSI\_GPL, 1991]. Monet muut pysyvät tai tarttuvat avoimen lähdekoodin lisenssit vaativat ainoastaan johdannaisten teosten lähdekoodin julkistamisen ja teosten lisensoinnin samalla lisenssillä, ja hintakysymys on jätetty markkinavoimien päätettäväksi [Rosen, 2004]. GPL:n kohta 1 sisältää kuitenkin hinnoittelurajoituksen kiertomahdollisuuden. Kohdassa 1 todetaan, että ohjelman kopioimisesta voi pyytää maksun [OSI\_GPL, 1991]. Lisenssin ehdoissa ei ole rajoitettu kopio- ja jakelukustannusten määrää eikä tarkemmin eritelty, mitä

kaikkea kopioimiseen ja kopion lähettämiseen kuuluu. Yritysmaailmassa on helppo allokoita kuluja kopion fyysiseen toimittamiseen, johon voi sisältyä tallennusmedian ja lähetyskulujen lisäksi mm. kopion saattaminen lähetyskuntoon [Rosen, 2004]. Näin kopio- ja jakelukustannuksia saadaan kohotettua. Joissakin maissa voi myös olla kiellettyä rajoittaa hinnoittelua GPL:n tekemällä tavalla ja yrityksille on taattu vapaus päättää itse tuotteidensa hinnoittelusta [Rosen, 2004]. Tällöin GPL:n asettamat rajoitukset ohjelman hinnoittelulle eivät välttämättä ole päteviä.

FSF:n suhtautuminen GNU-lisensoitujen ohjelmien myyntiin on positiivinen ja kannustava [FSF, 2001a], joten järjestöllä ei liene mitään syytä puuttua ohjelmien hinnoitteluun. Tärkeä poikkeus tähän on tilanne, jossa GPL-ohjelmaa levitetään binäärimuodossa ilman lähdekoodia. Tällöin ohjelman levittäjän on pyynnöstä toimitettava lähdekoodi enintään lähdekoodin jakelukustannusten hinnalla (GPL:n kohta 3b) [OSI\_GPL, 1999].

Ehdon 2c) tarkoitus on varmistaa, että lisenssin saajat tietävät GPL:n heille suomista vapauksista ja siitä, että ohjelmistolla ei ole takuuta, ellei ohjelmiston toimittaja ole erikseen myöntänyt sellaista [OSI\_GPL, 1991].

GPL:n kohdan 2 lopuksi tarkennetaan tilannetta, jossa ohjelma sisältää itsenäisiä, ohjelmasta erillisiä teoksia [OSI\_GPL, 1991] (käännös Välimäen [2001]):

Nämä ehdot koskevat muuteltua teosta kokonaisuudessaan. Jos yksilöivät osat tästä teoksesta eivät ole johdettuja Ohjelmasta ja ne voidaan perustellusti katsoa itsenäisiksi ja erillisiksi teoksiksi, silloin tämä Lisenssi ja sen ehdot eivät koske näitä osia, kun niitä levitetään erillisinä teoksina. Mutta jos samoja osia levitetään osana kokonaisuutta, joka on Ohjelmaan perustuva teos, tämän kokonaisuuden levittäminen on tapahduttava tämän Lisenssin ehtojen mukaan, jolloin tämän lisenssin ehdot laajenevat kokonaisuuteen ja täten sen jokaiseen osaan riippumatta siitä, kuka ne on tehnyt ja millä lisenssiehdoilla.

GPL-ohjelmasta erillään levitettyinä itsenäisten teosten ehdoista voidaan päättää vapaasti, mutta kun samat teokset yhdistetään GPL-lisensoituun ohjelmaan, niin ne ovat GPL:n ehtojen alaisia. Näin samaa ohjelmaa voidaan lisensoida itsenäisenä esimerkiksi yksinoikeuslisenssillä ja GPL-ohjelman yhteydessä GPL:lla.

Pelkkä toisen erillisen ohjelman liittäminen samalle medialle GPL-lisensoidun ohjelman kanssa ei vaikuta ohjelman lisenssiin mitenkään, mikäli ohjelman ei tulkita olevan GPL-lisensoidun ohjelman johdannainen teos [OSI\_GPL, 1991] (käännös Välimäen [2001]):

Lisäksi pelkkä toisen teoksen, joka ei perustu Ohjelmaan, liittäminen Ohjelman (tai Ohjelmaan perustuvan teoksen) kanssa samalle tallennus- tai jakeluvälineelle ei merkitse sitä, että toinen teos tulisi tämän Lisenssin sitomaksi.

GPL sallii ohjelman tai sen johdannaisen teoksen kopioimisen ja levittämisen binäärimuodossa edellyttäen, että ohjelman vastaanottajalla on mahdollisuus halutessaan saada myös ohjelman lähdekoodi haltuunsa. GPL määrittelee lähdekoodin seuraavasti [OSI\_GPL, 1991] (käännös Välimäen [2001]):

Teoksen lähdekoodi tarkoittaa sen suositeltavaa muotoa muutosten tekemistä varten. Ajettavan teoksen täydellinen lähdekoodi tarkoittaa kaikkea lähdekoodia kaikkiin teoksen sisältämiin moduuleihin ja lisäksi kaikkiin sen mukana seuraaviin käyttöliittymätiedostoihin sekä skripteihin, joilla hallitaan ajettavan teoksen asennusta ja kääntämistä. Kuitenkin erityisenä poikkeuksena levitetyn lähdekoodin ei tarvitse sisältää mitään sellaista, mikä yleensä levitetään (joko lähdekoodi- tai binäärimuodossa) käyttöjärjestelmän pääkomponenttien (kääntäjä, kernel, jne.) mukana, joiden päällä teosta ajetaan, ellei tällainen komponentti tule ajettavan teoksen mukana.

GPL käsittää lähdekoodiksi myös käännöskriptit ja muut tiedostot, jotka ovat tarpeellisia ohjelman toiminnan muuttamiseksi. Lisenssissä todetaan erikseen, että käyttöjärjestelmän mukana tulleiden komponenttien lähdekoodia ei tarvitse toimittaa käyttöjärjestelmän päällä ajettavan ohjelman mukana, vaikka ohjelma hyödyntäisikin niitä toiminnassaan. Muussa tapauksessa yksinoikeuslissenssillä varustettuihin käyttöjärjestelmiin, kuten Microsoft Windowsiin, ei välttämättä olisi mahdollista toteuttaa käyttöjärjestelmän kirjastoja hyödyntäviä GPL-ohjelmia, ja toisaalta lähdekoodin koko paisuisi hyvin suureksi.

GPL sallii useita eri tapoja saattaa lähdekoodi ohjelman vastaanottajan saataville. Lisenssin kohdassa 3 määritellään ehdot GPL-ohjelman binääriverion kopioimisesta ja levittämisestä [OSI\_GPL, 1991] (käännös Välimäen [2001]):

3. Lisenssin saajalla on oikeus kopioida ja levittää Ohjelmaa (tai siihen perustuvaa teosta, Kohdan 2 mukaisesti) objektikoodina tai ajettavassa muodossa yllä esitettyjen Kohtien 1 ja 2 mukaisesti edellyttäen lisäksi, että yksi seuraavista ehdoista on täytetty:



- a) Ohjelman mukaan liitetään täydellinen koneella luettava lähdekoodi, joka on levitettävä yllä mainittujen Kohtien 1 ja 2 ehtojen mukaisesti välineellä, jota käytetään yleisesti ohjelmistojen jakeluun; tai
- b) Ohjelman mukaan liitetään vähintään kolme vuotta voimassa oleva kirjallinen tarjous luovuttaa kelle tahansa kolmannelle osapuolelle, enintään lähdekoodin fyysisen levittämisen hinnalla, täydellinen koneella luettava lähdekoodi, joka on levitettävä yllä mainittujen Kohtien 1 ja 2 ehtojen mukaisesti välineellä, jota käytetään yleisesti ohjelmistojen jakeluun; tai
- c) Ohjelman mukaan liitetään Lisenssin saajan tieto tarjouksesta, joka koskee lähdekoodin levittämistä. (Tämä vaihtoehto on sallittu vain ei-kaupalliseen levittämiseen ja sillä ehdolla, että ohjelma on saatu objektiivisena tai ajettavassa muodossa yllä mainitun alakohdan b mukaisesti)

Kohdan 3 vaihtoehtoista 3a) on suosituin ja kätevin tapa toimittaa lähdekoodi, eikä se ole sidottu aikaan. Avoimen lähdekoodin projekteissa on nykyään suosituinta tarjota ohjelman binääri- ja lähdekoodiversioita WWW:ssä ohjelman kotisivuilla. Tällöin lähdekoodin ei tarvitse olla samassa paketissa binääriin kanssa vaan riittää, että lähdekoodiin on pääsy samasta paikasta kuin binääriin. Internet-jakelussa on kuitenkin muistettava, että lähdekoodeihin on taattava pääsy vähintään kolmen vuoden ajan ohjelman julkaisusta.

Edellä tässä tutkimuksessa on käsitelty avoimen lähdekoodin lisenssien yhteensopivuusongelmia. GPL:n kohdassa 6 oleva lause aiheuttaa lukemattomat yhteensopimattomuudet muiden lisenssien kanssa [OSI\_GPL, 1991] (käännös Välimäen [2001]):

- 6. ... Vastaanottajalle ei saa asettaa mitään lisärajoitteita tässä annettujen oikeuksien käytöstä. ...

Esimerkiksi jos toinen lisenssi vaatii mainostamaan ohjelman alkuperäistä tekijää kaikissa ohjelman tai sen johdannaisten teosten levitysversioneissa, niin GPL on yhteensopimaton kyseisen lisenssin kanssa. Kohta 6 toimii samalla ehdolla 2b) pelikuvana, sillä kun ehto 2b) vaatii, että johdannaiset teokset levitetään lisenssin asettamilla rajoituksilla, niin kohta 6 kieltää ylimääräisten rajoitusten asettamisen [Laurent, 2004].

Jos lisenssin saaja rikkoo GPL-lisensoidun ohjelman ehtoja, niin hänen lisenssinsä ohjelmaan lakkaa välittömästi. Jos hän on levittänyt ohjelmaa tai sen

johdannaisia teoksia, niin näiden teosten saajat eivät menetä oikeuksiaan teokseen eivätkä ole vaarassa joutua syytetyiksi toimittajan lisenssirikkomuksen vuoksi. Lisenssiehtojen rikkoja sen sijaan voi joutua vastuuseen teoistaan, sillä lisenssin päättymisen myötä henkilöä koskevat normaalit tekijänoikeus- ja patenttisäädökset, jolloin ohjelman kopioimisen, levittämisen tai muuttamisen jatkaminen rikkoo ohjelman tekijöiden ja patenttien haltijoiden oikeuksia. Ohjelmiston käyttöä ei kuitenkaan voida rajoittaa, koska GPL:n ehdot eivät ulotu pelkkään käyttöön. [Laurent, 2004]

Free Software Foundation pidättää itsellään oikeuden julkaista uusia versioita GPL:sta. GPL:n kohdassa 9 todetaan, että uusien versioiden henki on yhtenevä nykyisen version kanssa, mutta ne saattavat erota yksityiskohdissa esimerkiksi ottamalla huomioon uusia ongelmia ja huolenaiheita [OSI\_GPL, 1991]. Samassa kohdassa kerrotaan, miten eri lisenssiversioiden kanssa voidaan toimia [OSI\_GPL, 1991] (käännös Välimäen [2001]):

9. ... Jokaiselle versiolle annetaan ne muista erottava versionumero. Jos Ohjelma käyttää tämän Lisenssin tiettyä versiota tai "mitä tahansa myöhempää versiota", Lisenssin saaja saa valita, käyttääkö sitä tai jokin Free Software Foundationin julkaisemaa myöhempää versiota Lisenssistä. Jos Ohjelma ei mainitse mitä versiota tästä Lisenssistä se käyttää, on sallittua valita mikä tahansa versio, jonka Free Software Foundation on koskaan julkaissut.

Julkaisemalla uusia lisenssejä FSF ei voi asettaa uusia ehtoja olemassa oleviin ohjelmistoihin, vaan lisenssin saajat saavat itse päättää, mitä lisenssiversiota he haluavat käyttää.

GPL 2.0 on jo lähes 15 vuoden ikäinen lisenssi, ja vuosien saatossa on nousut esiin paljon uusia, huomioonotettavia asioita. Esimerkiksi WWW:ä ei ollut olemassakaan lisenssin 2.0-version kirjoitushetkellä, ja patenttien vaikutuksista ohjelmistokehitykseen ei ollut paljon kokemusta. GPL:n uudesta versiosta GPL 3.0:sta, on tarkoitus tehdä aidosti globaali lisenssi, joka ottaa yksityiskohdissa paremmin huomioon maailmanlaajuiset tekijänoikeuslait eikä keskity pelkäämään Yhdysvaltojen lakiin [FSF, 2005e]. Yksistään tämä on iso haaste ja kunnianhimoinen tavoite GPL 3.0:n kehitysryhmälle.

FSF ei ole ulottanut copyleftin ideaa GPL:n lisenssiehtoihin, sillä GPL:sta ei saa tehdä johdannaisia versioita. GPL:sta saa tehdä sanatarkkoja kopioita ja levittää niitä vapaasti, mutta lisenssin muuntelu ei ole sallittua ilman FSF:lta saatua erillistä lupaa.

Kuten edellä on useasti todettu, GPL ei salli sillä lisensoidun ohjelmiston yhdistämistä yksinoikeusohjelman kanssa. GPL:n [OSI\_GPL, 1991] lopussa varsinaisten lisenssiehtojen jälkeen todetaan, että mikäli ohjelmakirjastoa halutaan käyttää yksinoikeusohjelmien kanssa, niin se kannattaa lisensoida GPL:n sijasta LGPL:lla [OSI\_GPL, 1991] (käännös Välimäen [2001]):

Tämä GPL-lisenssi ei salli ohjelman ottamista osaksi yksinoikeudella omistettuja ohjelmia. Jos ohjelma on aliohjelmakirjasto, voi olla käytännöllisempää, että yksinoikeudella omistettut ohjelmat saavat linkittää kirjastoon. Jos tämä halutaan sallia, silloin tulee käyttää GNU kirjastolisenssiä (LGPL) tämän lisenssin sijasta.

#### 4.2.2. LGPL

GNU Lesser General Public Licensen juuret ovat vuodessa 1991, jolloin Free Software Foundation julkaisi lisenssin nimellä GNU Library General Public License [FSF, 2001b]. Kuten lisenssin alkuperäinen nimi kertoo, lisenssi suunniteltiin ohjelmakirjastoja varten, ja alun perin FSF suositteli sitä kaikkien ohjelmakirjastojen lisenssiksi. FSF päätti muuttaa lisenssin nimen vuonna 1999 GNU Lesser General Public Licenseksi ja käyttää erilaista strategiaa lisenssin markkinoinnissa. FSF ei enää suosittele LGPL:a kaikkiin kirjastoihin vaan tavoittelee enemmän GPL:lla lisensoituja ohjelmakirjastoja [Stallman, 1999a]. FSF:n tavoitteena on saada enemmän ainoastaan vapaan yhteisön ulottuvissa olevia kirjastoja, ja kirjastojen ja ohjelmistojen lisensoiminen GPL:lla edesauttaa tätä tavoitetta.

LGPL:n nimessä oleva *lesser* tarkoittaa, että lisenssi on vähemmän käyttäjän vapautta suojeleva kuin GPL [OSI\_LGPL, 1999]. Kaupallisesti ajatellen se tarjoaa enemmän mahdollisuuksia, sillä LGPL ei ole tarttuva lisenssi ja sallii tietyin ehdoin sillä lisensoidun ohjelmakirjaston yhdistämisen yksinoikeusohjelmaan. Lisenssi on laajassa käytössä ja toiseksi yleisin avoimen lähdekoodin lisenssi.

LGPL on pysyvä lisenssi: suorat muutokset LGPL-ohjelmistoon itseensä on levitettävä LGPL:n (tai GPL:n) ehdoilla, mutta LGPL-ohjelmiston ja toisen ohjelmiston yhdistelmät voi lisensoida jopa yksinoikeuslisenssillä tietyin varauksin. LGPL ei siis ole GPL:n tavoin tarttuva lisenssi, mikä onkin suurin ero LGPL:n ja GPL:n välillä. Muilta ominaisuuksiltaan lisenssit ovat hyvin samankaltaiset ja suureksi osaksi täysin yhtenevät. Koska edellä on jo käsitelty GPL:n ominaisuudet, niin tässä yhteydessä ei ole tarpeen käydä samoja asioita läpi. Seuraavaksi keskitytäänkin lähinnä LGPL:n eroavaisuuksiin GPL:sta.

GPL:n tapaan myös LGPL sisältää ehdot ainoastaan teoksen kopioimiseen, levittämiseen ja muuttamiseen. Lisenssin saajan tehdessä jotain näistä kolmesta

toimenpiteestä hän osoittaa hyväksyvänsä lisenssin ehdot ja sitoutuu noudattamaan niitä. LGPL-kirjaston käyttäminen on sallittua, vaikka lisenssiehtoja ei hyväksyisikään. Mikäli lisenssin saaja rikkoo LGPL:n ehtoja, hänen lisenssinsä päättyy välittömästi.

LGPL on suunniteltu ohjelmakirjastojen lisensointiin, mutta sitä voi käyttää myös kokonaisten ohjelmistojen lisensointiin, kuten lisenssin kohdan 0 aluksi todetaan [OSI\_LGPL, 1999]:

0. Tämä Lisenssisopimus pätee mihin tahansa kirjastoon tai muuhun ohjelmaan...

Merkille pantavaa on, että vaikka LGPL soveltuu käytettäväksi myös muihin kuin ohjelmakirjastoihin, niin lisenssiehdoissa puhutaan jatkuvasti kirjastosta eikä ohjelmasta, kuten GPL:ssa. Tämä johtunee lisenssin alkuperästä kirjastolisenssinä. Yksi tunnetuimmista LGPL:lla lisensoiduista ohjelmistoista on toimisto-ohjelmisto OpenOffice.org [OpenOffice.org, 2006].

LGPL määrittelee kirjaston seuraavasti [OSI\_LGPL, 1999]:

”Kirjasto” tarkoittaa ohjelmafunktioiden kokoelmaa ja/ tai tietoja, jotka on suunniteltu linkittymään vaivattomasti ohjelmistoihin (jotka käyttävät joi-tain näistä funktioista ja tiedoista) muodostaakseen suoritettavia tiedostoja.

Kirjasto, pienellä alkukirjaimella, on LGPL:n mukaan toteutettu suunnitel-mallisesti siten, että se on vaivattomasti linkitettävissä toisiin ohjelmiin. Rosen [2004] muotoilee tämän määritelmän tarkoittavan tekijänoikeuslain termein itsenäistä alkuperäistä teosta, joka on tarkoitettu liitettäväksi toisiin teoksiin jonkinlaisen linkityksen avulla. Seuraavaksi lisenssissä määritellään kirjasto isolla alkukirjaimella [OSI\_LGPL, 1999] (käännös Välimäen [2001] ja kirjoitta-jan):

Alla ”Kirjasto” viittaa mihin tahansa ohjelmistokirjastoon tai teokseen, jota on levitetty näiden ehtojen alaisena. ”Kirjastoon perustuva teos” tarkoittaa joko Kirjastoa tai mitä tahansa tekijänoikeuslain mukaista johdannaista te-osta: toisin sanoen teosta, joka sisältää Kirjaston tai osan siitä, kirjaimellises-ti tai muutettuna, tai toiselle kielelle käännettynä. ...

Isolla alkukirjaimella kirjoitettuna kirjasto kattaa LGPL:ssa minkä tahansa ohjelmakirjaston tai teoksen, jota on levitetty LGPL:n ehtojen alaisena. Se ei siis

ole rajattu pelkästään vaivattomasti toisiin ohjelmiin linkittyväksi ohjelmakirjastoksi, vaan myös muut teokset, kuten itsenäiset ohjelmistot, kuuluvat määritelmän alle.

GPL:sta poiketen LGPL määrittelee lähdekoodin erikseen muiden käsitteiden joukossa lisenssin kohdassa 0. Määritelmä on samankaltainen GPL:n määritelmän kanssa, mutta sisältää kirjastoon liittyvän lisäyksen [OSI\_LGPL, 1999]:

... Kirjaston täydellinen lähdekoodi kattaa kaikkien kirjaston sisältämien moduulien lähdekoodin, rajapintojen määrittelytiedostot ja kirjaston asennusta ja käännöstä hallinnoivat skriptit.

LGPL:n kohdassa 2 asetetaan ehdot kirjaston muuttamiselle, kopioimiselle ja levittämiseksi [OSI\_LGPL, 1999]. GPL:n kolmen ehdon sijaan LGPL:ssä on täytettävä neljä ehtoa, jotka käydään seuraavaksi yksitellen läpi [OSI\_LGPL, 1999] (käännös Välimäen [2001] ja kirjoittajan):

2. Kirjastokopion tai -kopioiden tai minkä tahansa osan muuttaminen on sallittu. Kun ohjelmaa muutetaan, muodostuu Kirjastoon perustuva teos. Lisenssin saajalla on lupa kopioida ja levittää näitä muutoksia ja Kirjastoon perustuvaa teosta sillä ehdolla, että ensinnäkin Kohdan 1 edellytykset täytetään ja lisäksi vielä seuraavat:
  - a) Muutellun teoksen pitää itse olla ohjelmistokirjasto.

LGPL:n kohta 2 on muuten identtinen GPL:n kohdan 2 kanssa, mutta LGPL:ssä puhutaan jälleen kirjastosta ohjelman sijaan. Ehto 2a) herättää kysymyksiä: LGPL:n kohdassa 0 todetaan lisenssin pätevän sekä kirjastoihin että muihin ohjelmiin, mutta ehdon 2a) mukaan LGPL rajoittaa johdannaisten teosten tekemisen tapauksiin, joissa lopputuloksena on kirjasto. Tämä on huomattava eroavaisuus GPL:iin, joka ei rajoita johdannaisten teosten tyyppiä mitenkään. On huomattava, että LGPL:n ehdossa 2a) kirjasto on kirjoitettu pienellä alkukirjaimella, mikä viittaa siihen, että kyseisellä merkinnällä tarkoitetaan nimenomaan ohjelmistokirjastoa ja vain kirjastoa. Laurentin [2004] mukaan tämä monimutkaistaa tai mahdollisesti estää kokonaan johdannaisten teosten tekemisen ohjelmista, jotka on lisensoitu LGPL:lla, mutta eivät ole ohjelmistokirjastoja. Käytännössä tämä asia pitää ratkaista tapauskohtaisesti pohtien sopijapuolten intentioita ja olosuhteita. Esimerkiksi OpenOffice.org:n tapauksessa on ilmeistä, että lisenssin antaja ei ole tarkoittanut ehtoa 2a) luettavaksi kirjaimellisenä muotovaatimuksena vaan pikemminkin joustavana määritelmänä siitä,

mitä lisensoidaan [Välimäki, 2005b]. On myös otettava huomioon, että lisenssin antaja ei ole OpenOffice.org:n tapauksessa itse kirjoittanut lisenssiä eikä ole täten välttämättä osannut huomioida tai tulkita oikein lisenssin epämääräisiä kohtia.

Ehto 2b) on täysin sama kuin GPL:n ehto 2a) ja vaatii muutosten tekijän ja muutospäivämäärän merkitsemisen muutettuihin tiedostoihin [OSI\_LGPL, 1999] (käännös Välimäen [2001]):

- b) Muuteltujen tiedostojen on sisällettävä selkeä merkintä, josta ilmenee, kuka tiedostoja on muuttanut ja päiväys, jolloin muutokset on tehty.

Ehto 2c) on tiivistetty versio GPL:n ehdosta 2b) ja tekee LGPL:sta pysyvän lisenssin. Se edellyttää, että kaikki johdannaiset teokset lisensoidaan ilman maksua LGPL:n ehtojen mukaisesti [OSI\_LGPL, 1999]:

- c) Teos on lisensoitava kokonaisuudessaan ilman maksua kaikille kolmansille osapuolille tämän Lisenssin ehtojen mukaisesti.

Ehto 2d) pyrkii asettamaan rajoituksia siihen, kuinka riippuvainen kirjasto saa olla ohjelmasta, johon kirjasto yhdistetään [OSI\_LGPL, 1999]:

- d) Jos muokatun kirjaston ominaisuus viittaa metodiin tai tietotaulukkoon, joka toimitetaan sellaisen teoksen mukana, joka käyttää ominaisuutta muuten kuin argumenttina ominaisuutta käynnistettäessä, niin lisenssin saajan täytyy varmistaa, että tapauksessa, jossa ohjelma ei toimita metodia tai taulukkoa ominaisuus silti toimii ja suoriutuu siitä tarkoituksensa osasta, joka edelleen säilyy mielekkäänä.

Ehto 2d) sisältää useita teknisiä ilmaisuja, kuten *metodi*, *tietotaulukko* ja *argumentti*, jotka eivät ole suotavia yleiskäyttöisissä lisensseissä varsinkaan, jos termejä ei ole määritelty lisenssin alussa. Rosen [2004] kritisoi voimakkaasti kyseistä ehtoa ja sanoo, aivan oikeutetusti, sen aiheuttavan varmasti päänvaihavan lisenssin saajalle. Ehdon 2d) jälkeen esitetty esimerkki onneksi selventää ehtoa hieman [OSI\_LGPL, 1999]:

(Esimerkiksi kirjastossa olevalla neliöjuuria laskevalla metodilla on tarkoitus, joka on kokonaisuudessaan määritelty itsenäiseksi ohjelmasta. Sen

vuoksi, alikohta 2d vaatii, että mikä tahansa tämän metodin käyttämä ohjelman metodi tai taulukko täytyy olla valinnainen: jos ohjelma ei tarjoa kyseistä metodia tai taulukkoa, niin neliöjuurimetodin täytyy silti laskea neliöjuuria.)

Ehdon 2d) tarkoitus on kannustaa kehittäjiä tekemään kirjastoistaan mahdollisimman siirrettäviä [Laurent, 2004]. Parhaassa tapauksessa kirjasto toimii täysin itsenäisesti, ja sen voi yhdistää kaikkiin ohjelmiin, jotka tarvitsevat kirjaston tarjoamia palveluita.

LGPL:n kohdassa 3 todetaan, että mikä tahansa LGPL-lisensoidun kirjaston lisenssin voi vaihtaa GPL:ksi, kun muuttaa kaikki LGPL:iin viittaavat kohdat viittaamaan GPL:iin. Laurent [2004] arvelee tämän vaihtoehdon olevan olemassa, jotta LGPL-lisensoidusta teoksesta voitaisiin luoda johdannaisia teoksia, jotka eivät ole kirjastoja. Tämä todetaan myös LGPL:ssa [OSI\_LGPL, 1999]:

3. ... Tämä vaihtoehto on hyödyllinen, kun Lisenssin Saaja haluaa kopioida osan Kirjaston koodista ohjelmaan, joka ei ole kirjasto.

Tämä lisenssin kohta tukee sitä tulkintaa, että LGPL-lisensoidusta teoksesta ei saisi tehdä johdannaista teosta, joka ei ole kirjasto. Mikäli ohjelmakopion lisenssin vaihtaa LGPL:sta GPL:ksi, niin sitä ei voi jälkikäteen vaihtaa takaisin LGPL:ksi.

LGPL sallii sillä lisensoidun kirjaston levittämisen objektikoodina tai suoritettavassa muodossa edellyttäen, että mukana toimitetaan kirjaston täydellinen lähdekoodi. Jos kirjaston objektikoodiversiota levittää esimerkiksi WWW-sivujen kautta, niin samassa paikassa on oltava ladattavissa kirjaston lähdekoodit. GPL:sta poiketen LGPL vaatii siis aina, että kirjaston binääriverision mukana toimitetaan myös lähdekoodiversio tai vähintään annetaan mahdollisuus sen kopioimiseksi samasta paikasta binääriverision kanssa.

Suurin ero LGPL:n ja GPL:n välillä on, että LGPL sallii sillä lisensoidun lähdekoodin toiminnan muiden lisenssien kanssa. Tästä mainitaan LGPL:n kohdassa 5, joka sisältää tärkeän määritelmän kirjastoa käyttävästä teoksesta [OSI\_LGPL, 1999]:

5. Ohjelmaa, joka ei sisällä Kirjaston minkään osan johdannaista teosta, mutta joka on suunniteltu toimimaan Kirjaston kanssa olemalla käännettävissä tai linkitettävissä sen kanssa, kutsutaan "Kirjastoa käyttäväk-

si teokseksi”. Sellainen teos yksinään ei ole Kirjaston johdannainen teos eikä sen vuoksi kuulu tämän Lisenssin ehtojen alaisuuteen.

LGPL käsittää kirjastoa käyttäväksi teokseksi ohjelman, joka hyödyntää LGPL-lisensoidun kirjaston toiminnallisuutta, mutta ei sisällä kirjastoa tai osia siitä. Tällainen teos ei yksinään tai muun kuin LGPL-kirjaston yhteydessä käytettynä ole kirjaston johdannainen teos eikä siten kuulu LGPL:n ehtojen alaisuuteen. Suurin huomio edellisessä lainauksessa kannattaa kiinnittää sanaan *yksinään*, sillä kirjastoa käyttävä teos välttää johdannaisen teoksen määritelmän vain, kun se on erillään kirjastosta. Yhdistettynä LGPL-kirjastoon ohjelma on johdannainen teos, kuten kohdan 5 seuraava kappale osoittaa [OSI\_LGPL, 1999]:

Kuitenkin, ”Kirjastoa käyttävän teoksen” linkittäminen Kirjaston kanssa luo suoritettavan, joka on Kirjaston johdannainen teos (koska se sisältää osia Kirjastosta) eikä ”Kirjastoa käyttävä teos”. Suoritettava on sen vuoksi tämän lisenssin alainen...

Kun ohjelmaan yhdistetään LGPL-kirjasto ja luodaan suoritettava tiedosto, niin kokonaisuus on lisensoitava LGPL:lla, koska kyseessä on tällöin LGPL-kirjaston johdannainen teos. Ohjelmaa saa kuitenkin levittää kirjastosta erillään haluamallaan ehdoilla, kuten edellä todettiin. LGPL:n ehdot eivät siis tartu kaikkeen koodiin, johon yhdistetään LGPL-koodia.

Poikkeuksena edelliseen LGPL sallii LGPL-kirjaston ja toisen ohjelmiston yhdistelmän, nk. yhdistetyn teoksen, lisensoinnin millä tahansa lisenssillä, jopa yksinoikeuslisenssillä, mikäli lisenssin kohdassa 6 olevat levitysehdot täytetään. LGPL-kirjasto säilyttää tässäkin tapauksessa lisenssiehtonsa. LGPL:n kohta 6 määrittelee ehdot kirjastoa käyttävän teoksen ja kirjaston yhdistelmän levittämiselle [OSI\_LGPL, 1999]:

6. Poikkeuksena yllä oleviin Kohtiin, Lisenssin saaja voi myös yhdistää tai linkittää ”Kirjastoa käyttävän teoksen” Kirjaston kanssa luodakseen teoksen, joka sisältää osia Kirjastosta, ja levittää teosta haluamallaan ehdoilla edellyttäen, että ehdot sallivat teoksen muuttamisen asiakkaan omaa käyttöä varten ja takaisinmallintamisen muutoksissa olevien virheiden korjaamiseksi.

Lisenssin saajan täytyy liittää jokaiseen teoksen kopioon silmiinpistävä ilmoitus, jossa kerrotaan, että Kirjastoa on käytetty teoksessa, ja että



Kirjasto ja sen käyttö kuuluu tämän Lisenssin ehtojen alle. Teoksen kopiaan on liitettävä kopio tästä Lisenssistä. Jos teos näyttää suorituksen aikana tekijänoikeustiedot, niin Kirjaston tekijänoikeustiedot on liitettävä niiden joukkoon, samoin kuin myös viittaus, joka ohjaa käyttäjän tämän Lisenssin kopiaan.

LGPL siis tarjoaa mahdollisuuden levittää kirjaston ja kirjastoa käyttävän teoksen yhdistelmän itse valituilla ehdoilla, mutta tietyin rajoituksin. Asiakkaalle on annettava lupa teoksen muuttamiseen asiakkaan omaa käyttöä varten ja takaisinmallintamiseen tehtyjen muutoksien sisältämien virheiden korjaamiseksi. Lisäksi jokainen teoksen kopio on varustettava LGPL:lla ja teoksessa on mainittava, että siinä on hyödynnetty LGPL-lisenssoitua kirjastoa, ja että kirjaston käyttö kuuluu LGPL:n ehtojen alle. Näiden lisäksi lisenssin saajan on täytettävä yksi seuraavista viidestä ehdosta [OSI\_LGPL, 1999]:

- a) Varusta teos Kirjaston täydellisellä koneella luettavalla lähdekoodilla, joka sisältää kaikki teokseen tehdyt muutokset (jotka on levitettävä yllä olevien Kohtien 1 ja 2 mukaisesti); ja, jos teos on suoritettava, joka on linkitetty Kirjaston kanssa, "Kirjastoa käyttävän teoksen" täydellisellä koneella luettavalla objekti- ja/ tai lähdekoodilla, jotta käyttäjä voi muuttaa Kirjastoa ja sitten linkittää sen uudelleen tuottaakseen muutetun suoritettavan, joka sisältää muutetun Kirjaston. (On ymmärrettävää, että käyttäjä, joka muuttaa Kirjaston määrittämissä tiedostojen sisältöä ei välttämättä ole kyvykäs kääntämään ohjelmaa uudelleen käyttääkseen muutettuja määrittämiä.)
- b) Käytä sopivaa jaetun kirjaston tekniikkaa Kirjaston linkittämiseen. Sopiva tekniikka on sellainen, joka (1) käyttää ajoaikana käyttäjän tietokonejärjestelmässä jo olevaa kirjaston kopiota sen sijaan, että kopioisi kirjaston metodeja suoritettavaan tiedostoon, ja (2) toimii oikein muutetun kirjaston kanssa, jos käyttäjä asentaa sellaisen, niin pitkään kuin muutettu versio on rajapinnoiltaan yhteensopiva sen version kanssa, jonka kanssa teos tehtiin.
- c) Varusta teos vähintään kolme vuotta voimassa olevalla kirjallisella tarjouksella, jossa lupaat toimittaa kyseiselle käyttäjälle yllä olevassa alakohdassa 6a määritellyt materiaalit enintään niiden jakelukustannusten hinnalla.

- d) Jos teoksen levitys toteutetaan tarjoamalla pääsy kopioida se tietystä paikasta, niin Lisenssin saajan on tarjottava vastaava pääsy kopioida yllä määritellyt materiaalit samasta paikasta.
- e) Lisenssin saajan on varmistettava, että käyttäjä on jo saanut näiden materiaalien kopion, tai että käyttäjälle on jo lähetetty kopio.

Kohdan 6 alakohdista kaikki paitsi 6b) vaativat ohjelman levittäjältä toimia LGPL-lisensoidun kirjaston lähdekoodin julkistamiseksi. Kohta 6a) vaatii, että LGPL-kirjaston lähdekoodi mahdollisine muutoksineen toimitetaan teoksen kopion mukana. Kirjastoa käyttävän teoksen lähdekoodia sen sijaan ei tarvitse liittää mukaan jakeluun. Kirjaston lähdekoodin avulla käyttäjät voivat muuttaa kirjaston toimintaa ja käyttää muutettua kirjastoa teoksen kanssa.

Kohta 6b) tulee kysymykseen lähinnä tilanteessa, jossa asiakkaan käyttöjärjestelmään on jo asennettu, joko käyttöjärjestelmän mukana tai jälkikäteen, teoksen käyttämä kirjasto. Tällöin teoksen mukaan ei ole tarpeen liittää kirjaston objekti- tai lähdekoodiversiota. Kohta 6c) on hyvin samankaltainen GPL:n kohdan 3b) kanssa. GPL:sta poiketen LGPL:n ehto koskee kuitenkin vain sitä nimenomaista asiakasta, jolle teos on toimitettu. Lähdekoodia ei siis tarvitse toimittaa edes pyynnöstä muille kuin teoksen vastaanottajille.

Kohta 6d) mahdollistaa GPL:n tapaan teosten jakelun tietoverkkojen avulla. Kohta 6e) voi tulla kyseeseen tilanteessa, jossa levitetään muutettua versiota kirjastoa käyttävästä teoksesta käyttäjälle, joka on jo saanut kirjaston aikaisemman jakeluversion mukana.

On huomattava, että mikäli teos kokonaisuudessaan sisältää sellaisia ohjelman osia, joihin teoksen tekijällä ei ole oikeuksia, teos rikkoo LGPL:n ehtoja. Tätä ei lieventäisi sekään, että asiakkaalla olisi omassa laitteistossaan tarvittava kirjasto, johon ohjelman toimittajalla ei ole oikeuksia. Mikäli teoksen toimittaja ei saa levittää kaikkia yhdistetyn teoksen komponentteja, niin se ei saa levittää mitään teoksen osaa [Laurent, 2004].

LGPL:a saa kopioida ja levittää vapaasti, mutta siitä ei saa tehdä johdannaisia teoksia kuten ei GPL:stakaan. Qwt – Qt Widgets for Technical Applications -ohjelmisto on lisensoitu Qwt-lisenssillä, joka sisältää LGPL:n kokonaisuudessaan ja muuttamattomana ja lisäksi ylimääräisiä kohtia, jotka lieventävät muutamia LGPL:n ehtoja ja rajoituksia. Esimerkiksi Qwt-lisensoidusta ohjelmasta periytetty ohjelma ei Qwt-lisenssin mukaan ole alkuperäisen teoksen johdannainen.

Qwt-lisenssi vaikuttaa olevan LGPL:n johdannainen teos, ja tämä herättää kysymyksen lisenssin laillisuudesta. Mikäli LGPL:n ehtoja tulkitaan kirjaimellisesti ja tinkimättä, niin Qwt-lisenssin kaltaisia lisenssejä ei saisi tehdä ilman

erillistä lupaa FSF:lta. Jos Qwt-lisenssin tulkitaan olevan laillinen ilman FSF:n lupaa, niin GNU-lisenssejä voi soveltaa omien lisenssien pohjana ainakin siten, että lisenssien peruseriaatteet säilytetään ja tarjotaan lievennyksiä joihinkin lisenssien ehtoihin.

#### 4.2.3. MPL

Vuonna 1998 Netscape Communications päätti julkaista Communicator-WWW-selaimensa binääri- ja lähdekooditiedostot ilmaiseksi kaikkien saataville [Laurent, 2004]. Yrityksessä pohdittiin aluksi jonkin olemassa olevan avoimen lähdekoodin lisenssin käyttämistä selaimen lähdekoodin lisensoinnissa. BSD-lisenssi hylättiin, koska se ei takaa, että kehittäjät palauttavat toteuttamansa muutokset takaisin yhteisölle [Mozilla, 2005d]. GPL:ssa Netscapea hiersivät lisenssin yhteensopimattomuus joidenkin selaimen kolmansien osapuolten valmistamien komponenttien kanssa, epävarmuus lisenssin mahdollisista ongelmista Yhdysvaltojen salaustekniikoita käsittelevän lain kanssa, GPL:n tarttuvuus myös muihin Netscapen tuotteisiin, jotka hyödyntävät Communicatorin koodia ja huoli siitä, että muut yritykset voivat epäröidä käyttää selaimen koodia, jos se on lisensoitu GPL:lla [Mozilla, 2005d]. Koska Netscape ei löytänyt tarkoituksiinsa sopivaa lisenssiä, niin se päätyi kirjoittamaan itse kaksi uutta lisenssiä: kirjoitusprosessin tuloksena olivat Netscape Public License (NPL) ja Mozilla Public License (MPL). NPL oli pääpiirteiltään yhtenevä MPL:n kanssa, mutta antoi ylimääräisiä oikeuksia Netscapelle, joista tärkeimpänä oikeus muuttaa kolmannen osapuolen kehittämän Netscape Communicatorista polveutuvan koodin lisenssiä [Laurent, 2004]. MPL ei sisällä vastaavia vaatimuksia oikeuksien luovuttamisesta Netscapelle [Laurent, 2004].

MPL:n kehitysvastuu on nykyään voittoa tavoittelemattomalla Mozillasäätiöllä, ja vain se voi julkaista MPL:sta uusia versioita [Mozilla, 2005e]. Mozillasäätiö on perustettu vuonna 2003 hankkimaan organisatorista, juridista ja taloudellista tukea avoimen lähdekoodin Mozilla-projektille, jonka tunnetuimpia tuotteita ovat Mozilla Suite -internet-ohjelmisto, Firefox-WWW-selain ja Thunderbird-sähköpostiohjelma [Mozilla, 2005c; Mozilla, 2005a].

MPL:a voidaan sanoa ensimmäiseksi avoimen lähdekoodin yrityslisenssiksi [Välimäki, 2005a]. Sen rakenne on hyvä, ja lisenssi on juridisesti hyvin kirjoitettu [Rosen, 2004]. Lisenssi on ollut mallina useille yritysten julkaisemille avoimen lähdekoodin lisensseille: mm. Nokia, Apple ja Sun ovat käyttäneet MPL:a pohjana omille OSI-aitoustopidustuksen saaneille avoimen lähdekoodin lisensseille [Laurent, 2004]. MPL:sta saakin tehdä johdannaisia teoksia, toisin kuin esimerkiksi GPL:sta ja LGPL:sta.

MPL sallii sillä lisensoidun lähdekoodin muuntelun, mutta vaatii, että muutokset on lisensoitava MPL:lla: MPL on siis pysyvä lisenssi. Tämä käy ilmi lisenssin ehdoissa 3.1 ja 3.2 [OSI\_MPL, 2005]:

### 3.1 Lisenssin käyttö

Muutokset, jotka Lisenssin Saaja luo tai joihin Lisenssin Saaja osallistuu, ovat tämän Lisenssin ehtojen alaisia...

### 3.2. Lähdekoodin saatavuus

Kaikki muutokset, jotka Lisenssin Saaja luo tai joihin Lisenssin Saaja osallistuu, täytyy laittaa saataville lähdekoodimuotoisina tämän Lisenssin ehtojen alaisina...

MPL-lisensoituun ohjelmaan tehdyt muutokset täytyy tarjota lähdekoodimuodossa joko samalla medially kuin ohjelman suoritettava versio tai jonkin sähköisen levitystavan avulla, joka voi tarkoittaa esimerkiksi HTTP- tai FTP-palvelinta. Muutokset täytyy olla saatavilla vähintään 12 kuukauden ajan teoksen julkaisusta tai kuuden kuukauden ajan uuden version toimittamisen jälkeen. Muutettuihin tiedostoihin pitää laittaa maininta muuttajasta ja muutospäivämäärästä. Lisäksi on kirjoitettava tiedoksi, jossa kerrotaan, että muutos perustuu alkuperäisen tekijän lähdekoodiin, ja liittää alkuperäisen tekijän nimi koodiin, ohjelmiston suoritettavaan versioon tai ohjelmiston dokumenttiin [OSI\_MPL, 2005].

MPL myöntää lisenssin saajalle erilaiset oikeudet ohjelmiston koodiin riippuen siitä, levitetäänkö koodia osana laajempaa teosta vai alkuperäisessä muodossaan. Laajemman teoksen MPL määrittelee teokseksi, joka yhdistää MPL:n alaista koodia lisenssin ulkopuolisen koodin kanssa [OSI\_MPL, 2005]:

### 3.7 Laajemmat Teokset

Lisenssin Saaja voi luoda Laajemman Teoksen yhdistämällä Sisällytettyä Koodia tämän Lisenssin ulkopuolisen koodin kanssa ja levittää Laajempaa Teosta yksittäisenä tuotteena. Tällaisessa tapauksessa Lisenssin Saajan täytyy varmistaa, että Lisenssin vaatimukset täyttyvät Sisällytetyn Koodin osalta.

Sisällytetty koodi tarkoittaa alkuperäistä koodia, muutoksia tai näiden yhdistelmää [OSI\_MPL, 2005].

MPL ei ole tarttuva lisenssi, ja se antaa yksinoikeusohjelmien tekijöille mahdollisuuden hyödyntää MPL-koodia ohjelmissaan. Laajempaa teosta suunnitte-

levan kannattaa varmistaa, onko kyseinen teos tulkittavissa alkuperäisen teoksen muokkaukseksi, joka pitää lisensoida MPL:lla, vai omaksi tuotteekseen, jolloin oman sovellusosuuden lisensoinnista voidaan päättää itse.

Laajemmassa teoksessa olevaan MPL-lisensoituun koodiin lisenssin saaja saa vain tekijänoikeuden alaiset oikeudet, jotka eivät sisällä patentti- ja tuotemerkkioikeuksia. Tämä on erittäin tärkeä erityispiirre, joka on huomioitava, mikäli aikoo levittää MPL-lisensoitua koodia osana laajempaa teosta [Laurent, 2004]. MPL-lisensoituun ohjelmistoon myönnetyt patenttioikeudet ovat yksiselitteisesti voimassa vain ja ainoastaan ohjelmiston muokkaamattomissa versioissa. Tämän vuoksi tilanteessa, jossa lisenssin saaja haluaa käyttää MPL-lisensoitua koodia osana laajempaa teosta, lisenssin saajan tulisi ensin selvittää MPL-koodiin mahdollisesti sisältyvät patentit ja tarvittaessa tämän jälkeen ottaa yhteyttä ohjelmiston alkuperäiseen kehittäjään saadakseen selville, voisiko tarvittavat oikeudet hankkia lisenssin ulkopuolelta [Laurent, 2004]. On myös huomattava, että lisenssi ei myönnä oikeuksia tuotemerkkeihin missään tapauksessa.

MPL sallii ohjelmiston suoritettavan version levityksen minkä tahansa lisenssin alaisena edellyttäen, että levittäjä suostuu MPL:n ehtoihin ja suoritettavan version lisenssi ei pyri rajoittamaan tai muuttamaan vastaanottajan oikeuksia ohjelmiston lähdekoodiversioon [OSI\_MPL, 2005]. Tämä mahdollistaa suoritettavan koodin julkaisun esimerkiksi yksinoikeuslisenssillä. Yrityksen kannalta asian mielekkyyttä heikentää se, että MPL:n ehtojen mukaisesti suoritettavan koodin lähdekoodi on julkaistava [Laurent, 2004]. Näin kenellä tahansa on mahdollisuus hakea lähdekoodi ja kääntää se suoritettavaksi koodiksi. Levitetäessä suoritettavaa koodia toisen lisenssin ehtojen alaisena lisenssin saajalle on tehtävä täysin selväksi, että kaikki ehdot, jotka poikkeavat MPL:sta ovat toimitajan tarjoamia, ja toimittaja on yksin vastuussa niistä [OSI\_MPL, 2005].

Taulukossa 6 on esitetty tutkimuksessa käsiteltyjen kolmen velvoittavan lisenssin keskeisimmät ominaisuudet.

**Taulukko 6. Yhteenveto velvoittavien lisenssien keskeisimmistä ominaisuuksista.**

Ominaisuus	GPL	LGPL	MPL
Vapaa käyttö-, kopiointi-, muuntelu- ja levitysoikeus.	X	X	X
Pysyvä.	X	X	X
Tarttuva.	X	-	-
Yhteensopiva GPL:n kanssa.	X	X	- <sup>1</sup>
Sallii hyödyntämisen yksinoikeusohjelmissa.	-	X <sup>2</sup>	X <sup>2</sup>
Lähdekoodi on vapautettava levitettäessä teosta.	X	X	X
Binäärikoodi on vapautettava levitettäessä teosta.	-	-	-
Johdannaisen teoksen lähdekoodi tai osa siitä voidaan sulkea.	-	-	-
Yhdistetyn tai uuden teoksen lähdekoodi tai osa siitä voidaan sulkea.	-	X <sup>3</sup>	X <sup>3</sup>
Muutetuista lähdekooditiedostoista täytyy käydä ilmi muuttajan nimi.	X <sup>4</sup>	X <sup>4</sup>	X <sup>4</sup>
Johdannaisen tai uuden teoksen voi hinnoitella vapaasti.	- <sup>5</sup>	- <sup>5</sup>	X
Teoksen lisenssiehdot, vastuunrajoitus ja tekijänoikeustiedot on säilytettävä johdannaississa teoksissa.	X	X	X
Sallii lisenssiehtojen muuttamisen, mikäli luotu teos on tulkittavissa uudeksi teokseksi.	-	-	-

<sup>1</sup> Sallii ohjelman tai sen osan tarjota eri lisenssiä. Jos ohjelman osa sallii GPL:n vaihtoehtoisena lisenssinä, niin silloin kyseisellä ohjelman osalla on GPL-yhteensopiva lisenssi.

<sup>2</sup> Yksinoikeusohjelmia voi luoda vain yhdistetyistä ja uusista teoksista, ei johdannaissista.

<sup>3</sup> Johdannaississa teoksissa oma koodi on julkaistava. Yhdistetyissä tai uusissa teoksissa oman koodin lisensoinnista, ja siten myös julkaisusta, voi päättää itse.

<sup>4</sup> Lisäksi on mainittava muutospäivämäärä.

<sup>5</sup> Lisenssin mukaan johdannaissiset teokset on lisensoitava kokonaisuudessaan ilman maksua kolmansille osapuolille. Käytännössä kuluja voi kuitenkin allokoida kopion fyysiseen toimittamiseen, johon voi sisältyä tallennusmedian ja lähetykskulujen lisäksi mm. kopion saattaminen lähetykskuuntoon.

## 4.3. Sallivat lisenssit

### 4.3.1. BSD-lisenssi

BSD-lisenssin juuret ovat Kalifornian yliopistossa, Berkeleyssä, ja se edustaa perinteisiä akateemisen vapauden periaatteita. BSD-lisenssi on vanhin avoimen lähdekoodin lisenssi ja tunnetuin salliva lisenssi [Välimäki, 2005; Rosen, 2004]. Se sisältää hyvin vähän rajoittavia ehtoja ja mm. sallii binäärikoodin levittämisen ilman, että ohjelman lähdekoodi pitää julkaista. BSD-lisenssi on kolmanneksi käytetyin avoimen lähdekoodin lisenssi heti GNU-lisenssien jälkeen.

BSD-lisenssistä on olemassa kaksi virallista versiota: alkuperäinen BSD-lisenssi ja uusi BSD-lisenssi. Alkuperäinen BSD-lisenssi on yhteensopimaton GPL:n kanssa [FSF, 2003], koska se sisältää kolmannessa kappaleessaan ehdon, jonka mukaan BSD-lisensoitua teosta tai sen osaa käyttävän ohjelman kaiken mainosmateriaalin täytyy sisältää seuraava huomautus [Hoskins, 1999]:

This product includes software developed by the University of California, Berkeley and its contributors.

Mainosvaatimus sai aikoinaan paljon kritiikkiä, kun ihmiset eivät sulattaneet sitä, että johdannaisten teosten kaikkiin mainosmateriaaleihin oli liitettävä maininta yliopistosta [Rosen, 2004]. Vielä suurempaa huolta aiheutti pelko siitä, että BSD-lisenssin muuntelijat liittävät omiin lisenssiversioihinsa vielä hämmentävämpiä mainosehtoja [Rosen, 2004]. Vastaavat mainosvaatimukset eivät ole enää sallittuja avoimen lähdekoodin lisensseissä, koska ne häiritsevät johdannaisten teosten luomisen vapautta [Rosen, 2004]. Vuonna 1999 alkuperäisestä BSD-lisenssistä poistettiin mainosehto [Hoskins, 1999] ja syntyi nk. uusi BSD-lisenssi (New BSD License, Modified BSD License tai Revised BSD License). Uusi BSD-lisenssi on yhteensopiva GPL:n kanssa [FSF\_2005a]. Jatkossa puhuttaessa BSD-lisenssistä tarkoitetaan lisenssin uutta versiota.

BSD-lisenssi on salliva lisenssi. Se on pituudeltaan erittäin lyhyt, mikä takaa sen, että lisenssi tulee usein luetuksi. Samalla lyhyys kuitenkin lisää tulkintatilanteita, sillä lisenssissä ei ole eksplisiittisesti kuvattu monia oleellisia asioita. BSD-lisenssin ensimmäisessä kappaleessa tekijänoikeustietojen jälkeen mainitaan lisenssin saajan oikeudet [OSI\_BSD, 2005] (käännös Välimäen [2005c] ja kirjoittajan):

Tämän ohjelmiston levittäminen ja käyttö lähdekoodina ja binäärinä, muutettuna tai muuttamattomana, on sallittu...

BSD-lisenssissä käsitellään ainoastaan ohjelmiston levittämistä ja käyttöä, eikä erikseen eritellä kaikkia lisenssin saajan vapauksia kuten ohjelmiston kopiointia, myyntiä ja esitystä sekä johdannaisten teoksen luontia. Tämä ei kuitenkaan tarkoita, että ne olisivat kiellettyjä [Rosen, 2004]. Lisenssin on tulkittu antavan vapauden käyttää, muunnella, kopioida ja myydä teosta sekä levittää sitä muutettuna tai muuttamattomana edellyttäen, että täytetään lisenssissä olevat kolme ehtoa [OSI\_BSD, 2005] (käännös Välimäen [2005c] ja kirjoittajan):

1. Lähdekoodikopioissa on säilytettävä yllä oleva maininta tekijänoikeudesta, nämä ehdot ja seuraava vastuunrajoitus.
2. Binäärikoodikopioiden on toistettava yllä oleva maininta tekijänoikeudesta, nämä ehdot ja seuraava vastuunrajoitus dokumentaatiossa tai muussa ohjelmiston mukana tulevassa aineistossa.
3. <ORGANISAATION> tai muiden ohjelmiston kehittäjien nimeä ei saa käyttää tästä ohjelmistosta kehitettyjen tuotteiden mainostamiseen tai kehumiseen ilman erillistä kirjallista lupaa.

Ensimmäinen BSD-lisenssin ehdoista edellyttää, että ohjelmiston tekijänoikeustiedot, lisenssiehdot ja vastuunrajoitukset toimitetaan lähdekoodikopion mukana. Ehto täyttyy yksinkertaisesti siten, että lähdekoodikopion varustaa BSD-lisenssillä, joka sisältää myös tekijänoikeustiedot. *Yllä oleva maininta tekijänoikeudesta* tarkoittaa BSD-lisenssin alussa olevaa tietoa ohjelmiston omistajasta. *Nämä ehdot* sisältävät lisenssin kolme ehtoa. *Seuraava vastuunrajoitus* tarkoittaa lisenssin lopussa olevaa takuu- ja vastuunrajoitusta, jossa avoimen lähdekoodin lisensseille tyypillisesti irtisanoudutaan kaikista ohjelmiston aiheuttamista vahingoista ja ohjelmiston laatuun liittyvistä takuista.

Toinen BSD-lisenssin ehto käsittelee binäärikopioiden levitystä. Kuten aikaisemmin tässä tutkimuksessa on mainittu, BSD-lisenssoitua ohjelmistoa voi levittää suoritettavassa muodossa myös ilman lähdekoodia. Tämän vuoksi lisenssiehdoissa on erillinen ehto tekijänoikeustietojen, lisenssiehtojen ja vastuunrajoitusten liittämiseksi ohjelman binäärimuotoiseen versioon. Tämä voidaan tehdä esimerkiksi liittämällä seuraava merkintä ja lisenssiehdot samaan paikkaan kuin ohjelman muutkin tekijänoikeusmerkinnät [Välimäki, 2005d]:

Portions of this program are copyright (c) <vuosi> <omistajat>

Merkinnässä oleviin tekijänoikeustietoihin on luonnollisesti täytettävä oikeat tiedot ja kulmasulkeet poistettava.

Kolmas BSD-lisenssin ehto pyrkii suojelemaan ohjelman kehittäjiä kieltämällä heidän nimien käytön ohjelmaa käyttävien tuotteiden markkinointitarkoituksissa ilman lupaa. Tämän asian tärkeys korostuu avoimen lähdekoodin ohjelmistoissa, joita voi muunnella kuka tahansa, eikä alkuperäinen tekijä tai muut ohjelmiston kehittäjät pääse välttämättä valitsemaan ja hyväksymään ohjelmistoon tehtyjä muutoksia. Ehto suojelee tekijöiden mainetta, joka voisi muuten tahriintua kolmannen osapuolen tekemistä heikkotasoisista muutoksista ohjelmistoon. On epäselvää, saisiko kehittäjien nimiä käyttää ilman tätä ehtoakaan, mutta eksplisiittinen maininta poistaa mahdolliset epäilyt [Fogel, 2005].

BSD-lisenssi ei ota kantaa siihen, mitä lähdekoodiin kuuluu, joten lisenssi ei välttämättä laske ohjelmiston dokumentaatiota, ohjelman käännohjeita ja käännoškriptejä lähdekoodiksi [OSI\_BSD, 2005].

BSD-lisenssi sallii teoksen lisenssiehtojen muuttamisen sillä edellytyksellä, että teosta muunnellaan sen verran, että siitä tulee uusi tai johdannainen teos, johon muuntelijalla on tekijänoikeus [Välimäki, 2002]. Tällaisessa tapauksessa BSD-ohjelman johdannaisen teoksen tai BSD-lisenssoitua komponenttia hyödyntävän ohjelman voi vapaasti lisensoida millä lisenssillä tahansa. BSD-



lisensoidut ohjelmat ovatkin yksinoikeusohjelmia kehittävän yrityksen kannalta paljon houkuttelevampia vaihtoehtoja kuin GNU-lisensseillä lisensoidut ohjelmat.

BSD-lisenssiä voi käyttää uusien lisenssien pohjana. BSD-lisenssistä onkin tehty useita variaatioita, joista tunnetuimpia ovat FreeBSD-, OpenBSD- ja NetBSD-lisenssit.

#### 4.3.2. MIT-lisenssi

MIT-lisenssi on peräisin yhdysvaltalaisesta Massachusettsin teknillisestä korkeakoulusta ja on BSD-lisenssin tapaan hyvin vanha avoimen lähdekoodin lisenssi. MIT-lisenssi kehitettiin alkuperäisen BSD-lisenssin pohjalta pyrkimyksenä tehdä lisenssistä helpommin luettava ja ymmärrettävä. Pohjimmiltaan se on hyvin samankaltainen BSD-lisenssin kanssa: suurin toiminnallinen ero lisenssien välillä on se, että MIT-lisenssistä puuttuu BSD-lisenssin ehto, joka kieltää ohjelmiston tekijöiden nimien käytön markkinoinnissa.

MIT-lisenssissä luetellaan erikseen teoksen sallitut hyödyntämismuutokset, mikä vähentää tulkintatilanteita BSD-lisenssiin verrattuna [Rosen, 2004]. Lisenssin saajalle annetaan rajoitukseton oikeus käyttää, kopioida, muunnella, yhdistää, julkaista, levittää ja myydä ohjelmiston kopioita sekä jopa muuttaa ohjelmiston lisenssiä [OSI\_MIT, 2005]:

Kaikille tämän ohjelmiston kopion ja siihen liittyvän dokumentaation ("Ohjelmisto") saaneille myönnetään vastikkeetta lupa Ohjelmiston vapaaseen käyttöön, sisältäen rajoituksettoman oikeuden käyttää, kopioida, muunnella, yhdistää, julkaista, levittää, muuttaa lisenssiä ja/tai myydä Ohjelmiston kopioita ja myöntää sama lupa henkilöille, joille Ohjelmisto toimitetaan...

BSD-lisenssistä poiketen MIT-lisenssi määrittelee ohjelmistoon kuuluvaksi myös ohjelmistoon liittyvät dokumentit. Lisenssin ehdot koskevat siis kaikkia ohjelmiston dokumentteja käyttöohjeista lähdekoodin käännöskripteihin. Ilmaus *vastikkeetta* tarkoittaa, että lisenssin sallimista käyttövaihtoehdoista ei peritä mitään erillistä palkkiota tai lisenssimaksua. Tämä ei kuitenkaan mitenkään rajoita mahdollisuutta veloittaa omilta asiakkailta maksua ohjelmiston toimitamisesta.

MIT-lisenssi sisältää vain yhden ehdon ja avoimen lähdekoodin lisensseille tyypillisen takuu- ja vastuunrajoituksen. Ehto vaatii käytännössä sen, että lisenssi sisällytetään kaikkiin ohjelmiston sanastarkkoihin kopioihin ja merkittävään ohjelmiston osiin [OSI\_MIT, 2005]:

1. Yllä oleva maininta tekijänoikeuksista ja tämä lupamerkintä täytyy liittää kaikkiin Ohjelmiston kopioihin tai huomattaviin Ohjelmiston osiin.

Ehdossa 1 olevalla *tekijänoikeustiedoilla* tarkoitetaan lisenssin alussa olevia ohjelmiston omistajan tietoja ja *lupamerkinnällä* tarkoitetaan lisenssiehtoja.

Levitettävän ohjelmiston lähdekoodia ei ole pakko liittää ohjelmiston jakeeluun tai luovuttaa edes pyynnöstä. Myös binäärijulkaisun täytyy kuitenkin täyttää lisenssin ehto tekijänoikeustietojen ja lisenssin lupamerkintöjen näkyvyydestä. Edellä BSD-lisenssin käsittelyn yhteydessä on esitetty yksi tapa hoitaa tämä.

BSD-lisenssin tapaan MIT-lisenssi on GPL-yhteensopiva [FSF, 2005a] ja sallii lisenssiehtojen muuttamisen tai korvaamisen kokonaan uudella lisenssillä uusiksi teoksiksi tulkittavissa yhdistetyissä ja johdannaisissa teoksissa [Välimäki, 2005a].

MIT-lisenssistä itsestään voi tehdä muuteltuja versioita. Tällä tavoin on syntynyt esimerkiksi X11-lisenssi [X11, 2005], jota kutsutaan joissakin yhteyksissä MIT X Window -lisenssiksi. X11-lisenssi sisältää vain pieniä lisäyksiä verrattuna MIT-lisenssiin, ja peruseriaatteet ovat molemmissa samat. Ehkä tunnetuin MIT-lisenssipohjainen ohjelmisto on X Window System, joka tunnetaan yleisemmin nimellä X11 tai X. Se on työkalu ja protokolla graafisen käyttöliittymän toteuttamiseksi Unixiin ja Unixin kaltaisiin käyttöjärjestelmiin kuten Linuxiin. [Wikipedia, 2005b]

### 4.3.3. Apache-lisenssi

Apache-lisenssi on yhdysvaltalaisen voittoa tavoittelemattoman Apache Software Foundation (ASF) -säätiön hallinnoima salliva avoimen lähdekoodin lisenssi. ASF on perustettu tukemaan avointa, yhteistyöhön perustuvaa ohjelmistokehitystä, vastaanottamaan lahjoituksia toimintansa rahoittamiseksi, tarjoamaan projekteihinsa osallistuville suojaa projekteja vastaan nostetuista oikeusjutuista ja suojelemaan Apache-tuotemerkkiä [ASF, 2005d]. ASF:n tuotteet ovat suosittuja erityisesti ohjelmistokehittäjien ja WWW-palveluja tarjoavien keskuudessa.

Vuonna 2004 julkaistu Apache-lisenssin versio 2.0 on säätiön uusin lisenssi, jota käytetään lähes poikkeuksetta kaikissa Apache Software Foundationin projekteissa. Myös monet säätiön ulkopuoliset projektit käyttävät lisenssiä. Apache-lisenssistä on julkaistu aikaisemmin versiot 1.0 ja 1.1, jolloin lisenssiä kutsuttiin Apache Software -lisenssiksi. Kun tässä tutkimuksessa puhutaan Apache-lisenssistä, niin sillä tarkoitetaan lisenssin versiota 2.0.

Apache-lisenssi on yksi huolellisimmin ja yksityiskohtaisimmin kirjoitetuista sallivista avoimen lähdekoodin lisensseistä [Välimäki, 2005a]. Se sisältää hyvin paljon samankaltaisuuksia BSD-lisenssin kanssa mutta lisäksi myös uusia ehtoja ja tarkennuksia. Lisenssi on paljon pitempi kuin BSD- ja MIT-lisenssit, ja se ottaa eksplisiittisesti kantaa moniin asioihin, joihin BSD- ja MIT-lisenssit viittaavat vain implisiittisesti. Tunnetuin Apache-lisenssillä lisensoitu ohjelma on maailman suosituin HTTP-palvelinohjelmisto Apache HTTP Server [ASF, 2005c], joka oli Netcraftin syyskuussa 2005 julkaistun tutkimuksen [Netcraft, 2005] mukaan käytössä lähes 70 %:ssa HTTP-palvelimista.

Apache-lisenssi sisältää ehdot teoksen käytölle, kopioinnille ja levittämiselle. Lisenssin alussa on määritelty lisenssissä käytetyt termit, joista yksi tärkeimmistä on lähdekoodimuoto. Apache-lisenssi käsittää lähdekoodiin kuuluvaksi ohjelmiston lähdekoodin lisäksi dokumentit ja alustustiedostot eli yleensä ottaen kaikki materiaalit, jotka ovat hyödyllisiä muutosten ja johdannaisten teosten tekemisessä [Laurent, 2004; OSI\_Apache, 2004].

Johdannainen teos määritellään Apache-lisenssissä seuraavasti [OSI\_Apache, 2004]:

“Johdannaiset Teokset” tarkoittavat lähdekoodi- tai objektikoodimuotoisia teoksia, jotka perustuvat Teokseen, ja joiden muutokset edustavat kokonaisuudessaan teoksen alkuperää. Johdannaiset Teokset eivät sisällä teoksia, jotka pysyvät Teoksesta tai Johdannaisista Teoksista erillisinä tai ainoastaan linkittyvät niiden rajapintoihin.

Määritelmän mukaan teokset, jotka säilyvät erillisinä ja itsenäisinä tai ainoastaan linkittyvät Apache-lisenssin alaisen teoksen rajapintoihin eivät ole johdannaisia teoksia. Tämä on tärkeä lisäys, joka vähentää johdannaisten teosten tulkintatilanteita.

Levitysoikeuksista kertova lause lisenssin [OSI\_Apache, 2004] kohdassa 4 antaa lisenssin saajalle luvan levittää ohjelmistoa tai sen johdannaista teosta myös ilman lähdekoodia [OSI\_Apache, 2004]:

#### 4. Jälleenlevitys

Lisenssin saaja voi kopioida Teosta ja Johdannaista Teosta ja levittää kopioita millä tahansa tallenteella, muutettuna tai muuttamattomana, Lähdekoodi- tai Objektikoodimuodossa...

Saadakseen nämä oikeudet lisenssin saajan on täytettävä neljä ehtoa, jotka muistuttavat hyvin paljon BSD-lisenssin ehtoja [OSI\_Apache, 2004]:

1. Kaikille Teoksen tai Johdannaisen Teoksen vastaanottajille täytyy antaa kopio lisenssistä.
2. Kaikkien muunneltujen tiedostojen pitää sisältää näkyvä merkintä muuttajasta.
3. Kaikki tekijänoikeus-, patenti-, tuotemerkki- ja osallistumismerkinnät täytyy säilyttää kaikissa levitettävissä Johdannaisten Teosten lähdekoodiversioissa pois lukien sellaiset merkinnät, jotka eivät kuulu mihinkään Johdannaisten Teosten osaan.
4. Jos Teos sisältää "NOTICE"-tekstitiedoston osana jakelua, niin jokaisen levitettävän Johdannaisen Teoksen on sisällettävä luettava kopio osallistumismerkinnöistä, jotka kyseinen NOTICE-tiedosto sisältää pois lukien ne NOTICE-tiedostot, jotka eivät kosketa mitään Johdannaisen Teoksen osaa, ainakin yhdessä seuraavista paikoissa: Johdannaisen Teoksen mukana levitettävässä NOTICE-tekstitiedostossa; Lähdekoodimuodossa tai dokumentaatiossa, mikäli ne ovat levitysversion mukana; tai Johdannaisen Teoksen tulostamana näytöllä, jos vastaavia kolmannen osapuolen ilmoituksia tavallisestikin esitetään näytöllä. NOTICE-tiedoston sisältö on ainoastaan informointia varten ja ei muuta Lisenssiä.

Ensimmäinen ehto vaatii, että Apache-lisenssi liitetään ohjelmiston tai sen johdannaisen teoksen jakeluun. Tämä on tehtävä riippumatta siitä, sisältyykö jakeluun ohjelmiston lähdekoodi. Apache-lisenssin lopussa on esimerkki lisenssin liittämistä omaan teokseen. Sen mukaan asia hoituu parhaiten lisäämällä teoksen muiden tekijänoikeustietojen kanssa samaan paikkaan seuraava teksti, jossa hakasulkeissa olevat tiedot korvataan tekijänoikeuden haltijan tiedoilla ja hakasulkeet jätetään pois [OSI\_Apache, 2004]:

Copyright [vuosi] [omistajan nimi]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

On huomattava, että kyseinen esimerkki on vain malli, eikä se kuulu lisenssin varsinaisiin ehtoihin.

Lisenssin toinen ehto vaatii, että muunnellut tiedostot varustetaan näkyväällä merkinnällä, josta käy ilmi muuttajan nimi. Kolmas ehto vaatii säilyttämään tekijänoikeus-, patenti-, tuotemerkki ja osallistumismerkinnät kaikissa johdannaisten teosten lähdekoodiversioissa, jos ne liittyvät johonkin johdannaisten teoksen osaan. Neljännessä ehdossa vaaditaan ns. NOTICE-tiedostojen liittämistä ohjelmistojakeluun.

Apache-lisenssi ei myönnä lisenssin saajalle oikeuksia lisensoijan tuotenimiin ja -merkkeihin. Lisenssin saaja ei myöskään saa liittää lisensoijan nimeä johdannaisiin teoksiinsa tai alkuperäisen teoksen jakeluihinsa, paitsi tilanteissa, joissa se on välttämätöntä teoksen alkuperän ilmoittamiseksi [Laurent, 2004]. Tästä on kerrottu lisenssin kohdassa 6 [OSI\_Apache, 2004]:

Tämä Lisenssi ei anna lupaa käyttää Lisensoijan tuotenimiä, tuotemerkkejä tai palvelumerkkejä, lukuun ottamatta kohtuullista ja tavanomaista käyttöä Teoksen alkuperän kuvaamisessa ja NOTICE-tiedoston sisällön toistamisessa.

Ohjelmiston muuttaja voi lisätä omat tekijänoikeusmerkintänsä muuttamiinsa kohtiin. Lisäksi on mahdollista tarjota ylimääräisiä tai erilaisia lisenssiehtoja omille muutoksille tai kokonaisten johdannaisten teosten käytölle, kopiointille ja levitykselle. Johdannaisten teokset voi siis lisensoida millä tahansa lisenssillä, esimerkiksi yksinoikeuslisenssillä. Tästä on erikseen mainittu lisenssin kohdassa 4 [OSI\_Apache, 2004]:

Voit lisätä omat tekijänoikeuslauseesi tekemiisi muokkauksiin ja voit tarjota ylimääräisiä tai erilaisia lisenssiehtoja muokkaustesi tai minkä tahansa Johdannaisen Teoksen käyttöön, kopiointiin ja levitykseen, edellyttäen, että oma käyttösi, kopiointisi ja levityksesi täyttävät muutoin tämän Lisenssin ehdot.

BSD- ja MIT-lisensseissä ei erikseen mainita tätä mahdollisuutta, mutta niiden on tulkittu sallivan samanlaisen menettelyn [Laurent, 2004].

Apache-lisenssin GPL-yhteensopivuus on aiheuttanut keskustelua ja erilaisia tulkintoja. Apache Software Foundationin tulkinnan mukaan lisenssit ovat yhteensopivat, mutta Free Software Foundationin mielestä Apache-lisenssin patenttien vastainen ehto on liian laaja ja lisenssi siten yhteensopimaton GPL:n kanssa [FSF, 2005a]. Patenttiehto sijaitsee lisenssin kohdassa 3 [OSI\_Apache, 2004]:

... Jos nostat patentinloukkauksen mitä tahansa tahoa vastaan väittäen, että Teos tai Teokseen sulautettu Osuus muodostaa suoran tai osittaisen patenttirikkomuksen, niin kaikki tämän Lisenssin alaiset patenttilisenssit, jotka sinulle on myönnetty, päättyvät riita-asian käynnistämispäivänä.

Apache-lisenssin patenttien vastaisen ehdon mukaan se, joka nostaa patentt loukkauksen tuomioistuimessa Apache-lisenssillä levitettävän ohjelman kehittäjää tai käyttäjää vastaan menettää oikeutensa kaikkiin ohjelmaan käytettyihin patentteihin. Alkuperäinen kanteen nostaja ei täten saa enää hyödyntää kyseistä ohjelmaa mitenkään, tai hän syyllistyy patenttilain rikkomiseen, ja muut patenttien haltijat voivat esittää omat vaatimuksensa häntä vastaan. Vaikka FSF ei sinänsä pidä patenttien lakkauttamista luonnostaan huonona ideana, niin GPL ei salli tällaisia lisäehtoja [FSF, 2005a]. FSF:n mukaan Apache-lisenssin kaikki versiot ovat yhteensopimattomia GPL:n kanssa, ja Apache-lisenssoidun koodin yhdistäminen kiinteästi GPL-lisensoituun koodiin rikkoo GPL:n lisenssiehtoja. ASF on asiasta eri mieltä eikä näe Apache- ja GPL-lisenssien lähdekoodien yhdistämisessä ongelmaa [ASF, 2005a]. On mahdollista, että GPL:n ja Apache-lisenssin seuraavissa versioissa löydetään yhteinen sävel patenttiasioissa, ja yhteensopimattomuudet poistuvat [ASF, 2005a; FSF, 2005a].

Apache Software Foundation pohtii jatkuvasti tarpeita lisenssiensä päivittämiseksi ja kerää käyttäjien mielipiteitä. Tulevaisuudessa erityisesti patentteihin ja avustuksiin liittyvät yksityiskohdat voivat tuoda muutoksia Apache-lisenssiin. [ASF, 2005b]

#### 4.3.4. PHP-lisenssi

PHP on yleistarkoitukseen sopiva palvelinpään ohjelmointikieli, joka on suunniteltu erityisesti WWW-ohjelmistojen toteuttamiseksi. PHP on noussut yhdeksi suosituimmista WWW:ssä käytetyistä skriptikielistä [Netcraft, 2002] eikä vähiten PHP:n, Apache HTTP Server -palvelimen ja MySQL-tietokantajärjestelmän toimivan symbioosin ansiosta.

PHP:n esiversion loi Rasmus Lerdorf vuonna 1995, jolloin kieli koostui yksinkertaisesta kokoelmasta Perl-skriptejä. Tuolloin kieli oli nimeltään PHP/FI ja lyhenne PHP muodostui sanoista *Personal Home Page Tools*. Nykyaikaisen PHP:n voidaan katsoa syntyneen vuonna 1997, kun kielen kolmas versio kehitettiin. Samalla kieli nimettiin pelkäksi PHP:ksi, joka on rekursiivinen lyhenne sanoista *PHP: Hypertext Preprocessor*. [PHP, 2005]

PHP-ohjelmisto on lisensoitu PHP-lisenssillä, joka on salliva BSD-tyylinen avoimen lähdekoodin lisenssi. Lisenssi on tällä hetkellä versiossa 3.01 ja sitä ylläpitää voittoa tavoittelematon järjestö nimeltä The PHP Group.

PHP-lisenssillä lisensoidun ohjelmiston käyttö ja kopiointi on vapaata, ja kopioiden levittäminen on sallittua lähdekoodi- tai binäärimuodossa, muutettuna tai muuttamattomana edellyttäen, että kuusi ehtoa täytetään. PHP-lisenssin ehdoista ensimmäinen ja toinen ovat identtisiä BSD-lisenssin kanssa, ja ne on käsitelty jo BSD-lisenssin yhteydessä [OSI\_PHP, 2005] (käännös Välimäen [2005c] ja kirjoittajan):

1. Lähdekoodikopioissa on säilytettävä yllä oleva maininta tekijänoikeudesta, nämä ehdot ja seuraava vastuunrajoitus.
2. Binäärikoodikopioiden on toistettava yllä oleva maininta tekijänoikeudesta, nämä ehdot ja seuraava vastuunrajoitus dokumentaatiossa tai muussa ohjelmiston mukana tulevassa aineistossa.

Kaksi seuraavaa ehtoa keskittyvät suojelemaan PHP:n mainetta ja nimeä johdannaisissa teoksissa. Kolmannessa ehdossa kielletään *PHP*-nimen käyttö mainostamistarkoituksissa, ja neljännessä ehdossa rajoitetaan sen käyttöä johdannaisissa teoksissa [OSI\_PHP, 2005]:

3. Nimeä "PHP" ei saa käyttää mainostamaan tuotteita, jotka ovat tämän ohjelmiston johdannaissia teoksia ilman aikaisemmin saatua kirjallista lupaa. Kirjallista lupaa voi tiedustella osoitteesta [group@php.net](mailto:group@php.net).
4. Tämän ohjelmiston johdannaisille teoksille ei saa antaa nimeä "PHP", eikä sana "PHP" saa esiintyä niiden nimessä ilman aikaisempaa kirjal-

lista lupaa osoitteesta group@php.net. Voit viitata siihen, että ohjelmistosi toimii yhteydessä PHP:hen antamalla ohjelmistolle nimeksi "Foo for PHP" sen sijaan, että annat sille nimeksi "PHP Foo" tai "phpfoo".

Lisenssin ehdossa 5 on huomioitu se, että tarpeita lisenssiehtojen muuttamiselle voi tulla tulevaisuudessa [OSI\_PHP, 2005]:

5. PHP Group voi aika ajoin julkaista muutettuja versioita lisenssistä. Jokaiselle versiolle annetaan erottava versionumero...

Lisenssin saaja voi aina valita joko ohjelmiston mukana tulleen PHP-lisenssin tai minkä tahansa uudemman version lisenssistä. PHP Group pidättää itsellään oikeuden muuttaa PHP-lisenssin ehtoja.

Kuudes ja viimeinen ehto vaatii, että ohjelmiston levitysversiot mainostavat PHP:tä ja sen WWW-sivuja [OSI\_PHP, 2005]:

6. Kaikkien levitysversioiden, levitysmuodosta riippumatta, on säilytettävä seuraava tunnustus:

"This product includes PHP software, freely available from  
<<http://www.php.net/software/>>".

Mainostamiseen liittyvän ehdon takia PHP-lisenssi on Free Software Foundationin tulkinnan mukaan yhteensopimaton GPL:n kanssa, koska GPL ei salli tällaisia lisäehtoja [FSF, 2005a]. Tämä tarkoittaa, että jos PHP-lisenssin alaista lähdekoodia yhdistetään kiinteästi GPL:lla lisensoituun lähdekoodiin, rikotaan FSF:n mielestä PHP:n mainostamisehtoa. Kuitenkaan GPL-lisensoidun komponentin käyttäminen PHP-lisensoidusta ohjelmasta ei ole ongelmallista, sillä tekijänoikeus ei koske käyttöä [Välimäki, 2005d]. Linkitettäessä voidaan kuitenkin joutua tulkintatilanteisiin siitä, onko pääohjelmisto riippuvainen linkitetystä komponentista ja komponentti siten osa pääohjelmaa, jolloin GPL:n voidaan mahdollisesti tulkita tarttuvan koko ohjelmaan. Jos linkitettyä kokonaisuutta ajetaan esimerkiksi GNU/Linux-ympäristössä, ja tämä linkitetty kokonaisuus lisensoidaan GPL:lla ja mainitaan, että osissa kokonaisuutta on PHP-lisenssin ehtoja, ei GPL-komponentin kehittäjällä pitäisi olla mitään pätevää juridista vaadetta komponentin käyttäjää kohtaan [Välimäki, 2005d].



#### 4.3.5. Artistic-lisenssi

Larry Wall esitteli Artistic-lisenssin osana Perl 4.0 -ohjelmistoa vuonna 1991. Syynä lisenssin kirjoittamiseen GPL:n käyttämisen sijaan oli GPL:n tiukat rajoitukset yksinoikeusohjelmien suhteen [Välimäki, 2005a]. Artistic-lisenssin nimi kuvastaa lisenssin tavoitetta säilyttää taiteellinen kontrolli ohjelmaan ja sen johdannaisiin teoksiin ohjelman alkuperäisellä kehittäjällä. Lisenssin tarkoituksena oli säilyttää Wallilla ja hänen ryhmällään Perl-projektin hallinta, mutta samalla rohkaista sekä liittymistä projektiin että osallistumista kehitystyöhön projektin ulkopuolelta [Laurent, 2004]. Lisenssi onkin suunniteltu hallinnoltaan keskitettyjä projekteja silmällä pitäen. Se pyrkii pitämään huolta alkuperäisen tekijän oikeuksista eikä siten ole kaikkein omimmillaan avoimessa ohjelmistokehityksessä [Laurent, 2004]. Artistic-lisenssiä ei ole siis suunniteltu yleiseksi lisenssiksi, kuten muut tässä tutkimuksessa esitellyt lisenssit, joskaan mikään ei estä sen käyttämistä muun kuin Perlin lisensointiin [Laurent, 2004]. Lisenssiä voidaan myös käyttää pohjana kirjoitettaessa uutta lisenssiä.

Artistic-lisenssi on hakkereiden tekemä ohjelmistolisenssi hakkereille, joka luottaa suuresti lisenssin saajien moraaliin tietokoneyhteisöä kohtaan. Wall [Wall, 1994] on sanonut olevansa tyytyväinen, jos Artistic-lisenssi välittää hänen aikeensa kunniallisille ihmisille ja samalla antaa yritysten juristeille lämpimän tunteen, että he voisivat keplotella itsensä vapaiksi siitä, jos he todella tahtoisivat.

Rosen [2004] arvostelee Artistic-lisenssin kieltä ja kutsuu lisenssiä amatöörlisenssiksi. Hänen mielestään Artistic-lisenssiä on juristin vaikea selittää ja tuomarin ehkä mahdoton ymmärtää [Rosen, 2004]. Lisenssissä käytetään useissa kohdissa juridisesti epäselviä termejä kuten *toivoo*, *ajatella*, *oikeuttaa* ja *kohtuullinen*. Esimerkki epämääräisestä kielestä on lisenssin kohdassa 5 mainittu mahdollisuus periä *kohtuullinen maksu* ohjelmiston levittämisestä [OSI\_Artistic, 2005]. Käytännössä markkinat määräävät sen, mikä on kohtuullinen hinta ohjelmistolle: jos tuote menee kaupaksi, niin hinta tuskin on kohtuuton.

Artistic-lisenssillä lisensoitua ohjelmistoa saa käyttää vapaasti, ja sen sanatarkkoja kopioita saa levittää edellyttäen, että levitysversion varustaa ohjelmiston lisenssillä. Jos ohjelmistoa muuntelee ohjelmiston tekijänoikeuden haltijan tarjoamalla tai yleisessä omistuksessa (public domain) olevilla muutoksilla, niin ohjelmistoa koskevat alkuperäisen version ehdot – sitä ei siis tällöin tulkita muunnelluksi versioksi.

Mikäli ohjelmistoon tekee omia muutoksia, niin Artistic-lisenssi asettaa ylimääräisiä ehtoja. Jokaiseen muutettuun tiedostoon täytyy liittää näkyvä maininta tehdystä muutoksesta, muuttajasta ja ajankohdasta. Tämän lisäksi on täytettävä yksi seuraavista lisenssin kohdan 3 ehdoista [OSI\_Artistic, 2005]:

- a) Muutoksista tehdään yleistä omaisuutta (public domain), tai ne asetetaan muutoin vapaasti ja ilmaiseksi saataville, jotta tekijänoikeuden haltija voi liittää muutokset ohjelmiston standardiversioon.
- b) Muunneltua ohjelmistoa käytetään vain yrityksen sisällä.
- c) Muunnellut ajettavat tiedostot nimetään alkuperäisistä poikkeaviksi, jotta muunnokset ja alkuperäiset versiot voidaan erottaa toisistaan. Lisäksi jokainen muunneltu tiedosto pitää varustaa ohjeella, jossa kerrotaan selkeästi kuinka se poikkeaa alkuperäisestä.
- d) Sovitaan muista levitysjärjestelyistä tekijänoikeuden haltijan kanssa.

Ehdossa 3a) mainittu ohjelmiston standardiversio tarkoittaa ohjelmiston muuttamatonta versiota tai sellaista versiota, jota on muutettu tekijänoikeuden haltijan toiveiden mukaisesti.

Artistic-lisenssillä lisensoitua ohjelmistoa tai sen osaa voi levittää binäärimuodossa muutettuna tai muuttamattomana. Tämän edellytyksenä on, että lisenssin saaja täyttää yhden seuraavista lisenssin kohdan 4 ehdoista [OSI\_Artistic, 2005]:

- a) Jakeluun liitetään standardiversioon ajettavat tiedostot ja kirjastot sekä ohjeet (ohjesivulla tai vastaavassa) standardiversioon saamiseksi.
- b) Jakeluun liitetään ohjelmiston ja tehtyjen muutosten lähdekoodit.
- c) Jakeluun liitetään ajettavien tiedostojen standardiversiot ja muunnellut ajettavat tiedostot nimetään alkuperäisistä poikkeaviksi, jotta muunnokset ja alkuperäiset versiot voidaan erottaa toisistaan. Lisäksi versioiden väliset muutokset ja tieto standardiversioon saamiseksi dokumentoidaan selvästi ohjesivuilla (tai vastaavassa).
- d) Sovitaan muista levitysjärjestelyistä tekijänoikeuden haltijan kanssa.

Yksinoikeusohjelmia toteuttavan tahon kannalta käytännöllisin kohdan 4 ehdoista on ehto 4a), joka edellyttää, että levitysversion sisällytetään suoritettavien tiedostojen alkuperäiset versiot ja ohjeet siitä, mistä saa alkuperäisen ohjelmiston lähdekoodit. Tämä täyttää lisenssin vaatimukset, vaikkakin lisenssissä lienee tarkoitettu, että muokattujen versioiden levittämisessä toimitaan ehdon 4c) mukaisesti [Laurent, 2004].

Vaikuttaa siltä, että Artistic-lisenssillä varustettua komponenttia saa käyttää hyvin vapaasti omassa ohjelmistossa, kun vain liittää jakeluun muunneltujen

tiedostojen alkuperäisversiot, ohjeet alkuperäisen ohjelmiston lähdekoodien hankkimiseksi ja mainitsee, että teoksen levittäjä on vastuussa tekemistään muutoksista. Rosen [2004] suosittelee suhtautumaan Artistic-lisenssiin kuin sallivaan lisenssiin, joka antaa laajan vapauden käyttää, kopioida, muunnella, levittää ja myydä alkuperäistä ohjelmaa. Johdannaisen teoksen levityksessä on tärkeää painottaa, että ohjelman toimittaja – ei alkuperäinen tekijä – on vastuussa toimitetusta ohjelmistosta. Lisäksi on oltava huolellinen, ettei mainosta ohjelmaa omana tuotteenaan eikä käytä alkuperäisten tekijöiden nimiä johdannaisen teosten mainostamistarkoituksiin. Mikäli nämä ohjeet pitää mielessä, niin Artistic-lisenssiin voi suhtautua sallivana lisenssinä, eikä ongelmia pitäisi tulla [Rosen, 2004].

Artistic-lisenssistä on olemassa erilaisia versioita. Open Source Initiative on hyväksynyt sekä alkuperäisen Artistic-lisenssin että lisätyn 8. lauseen sisältävän version, jota käytetään Perlin Artistic-lisenssissä [OSI\_Artistic, 2005]:

8. Tämän Paketin liittäminen kaupalliselle jakelulle on aina sallittua, edellyttäen, että Paketin käyttö on sulautettua; tämä tarkoittaa, että ei ole tehty ilmeistä yritystä saattaa tämän Paketin rajapinnat näkyviksi kaupallisen jakelun loppukäyttäjälle. Tällaista käyttöä ei tulkita tämän Paketin levittämiseksi.

Wall lisäsi kyseisen kappaleen lisenssiin selventääkseen ajatustaan, että sulautettu käyttö ei ole levitystä ja täten sallittua myös kaupallisissa jakeluissa [Wall, 1994].

Artistic-lisenssi on FSF:n mielestä liian epämääräinen ollakseen vapaan ohjelmiston lisenssi [FSF, 2005a]. Samalla lisenssi on automaattisesti yhteensopimaton GPL:n kanssa. Lisenssin ongelmat ovat FSF:n mukaan enemmänkin käytetyissä sanamuodoissa kuin sisällössä, ja lisenssistä onkin tehty selvennetty versio, Clarified Artistic License, joka on yhteensopiva GPL:n kanssa [FSF, 2005a; Vedova, 2005].

Artistic-lisenssin uudesta versiosta on tehty esityksiä, ja Perl 6.0:n lisenssiksi on kaavailtu lisenssin versiota 2.0 [Perl, 2000]. Se on kuitenkin vasta ehdotusasteella, eikä ole varmaa julkaistaanko sitä virallisesti koskaan. FSF on joka tapauksessa ilmoittanut sen olevan vapaan ohjelmiston lisenssi ja yhteensopiva GPL:n kanssa [FSF, 2005a].

Taulukossa 7 on esitetty tutkimuksessa käsiteltyjen viiden sallivan lisenssin keskeisimmät ominaisuudet.

Taulukko 7. Yhteenveto sallivien lisenssien keskeisimmistä ominaisuuksista.

Ominaisuus	BSD	MIT	Apache	PHP	Artistic
Vapaa käyttö-, kopiointi-, muuntelu- ja levitysoikeus.	X	X	X	X	X
Pysyvä.	-	-	-	-	-
Tarttuva.	-	-	-	-	-
Yhteensopiva GPL:n kanssa.	X	X	- <sup>1</sup>	-	-
Sallii hyödyntämisen yksinoikeusohjelmissa.	X	X	X	X	X
Lähdekoodi on vapautettava levitettäessä teosta.	-	-	-	-	-
Binäärikoodi on vapautettava levitettäessä teosta.	-	-	-	-	-
Johdannaisen teoksen lähdekoodi tai osa siitä voidaan sulkea.	X	X	X	X	X
Yhdistetyn tai uuden teoksen lähdekoodi tai osa siitä voidaan sulkea.	X	X	X	X	X
Muutetuista lähdekooditiedostoista täytyy käydä ilmi muuttajan nimi.	-	-	X	-	X <sup>2</sup>
Johdannaisen tai uuden teoksen voi hinnoitella vapaasti.	X	X	X	X	X
Teoksen lisenssiehdot, vastuunrajoitus ja tekijänoikeustiedot on säilytettävä johdannaisissa teoksissa.	X	X	X <sup>3</sup>	X	X
Sallii lisenssiehtojen muuttamisen, mikäli luotu teos on tulkittavissa uudeksi teokseksi.	X	X	X	X	X <sup>4</sup>

<sup>1</sup> ASF:n tulkinnan mukaan lisenssi on yhteensopiva GPL:n kanssa, mutta FSF:n mielestä ei.

<sup>2</sup> Lisäksi on mainittava muutospäivämäärä ja tehty muutos.

<sup>3</sup> Lisäksi patenti-, tuotemerkki- ja muut huomautukset täytyy säilyttää kaikissa johdannaisten teosten lähdekoodiverzioissa pois lukien sellaiset huomautukset, jotka eivät kuulu mihinkään johdannaisen teoksen osaan. Ns. NOTICE-tekstitiedostot on säilytettävä jakelussa, mikäli ne liittyvät johonkin johdannaisen teoksen osaan.

<sup>4</sup> Lisenssin epämääräisyyden takia tämä ei ole selvää, mutta lisenssin on tulkittu käyttäytyvän muiden sallivien lisenssien kaltaisesti.

## 5. Avoimen lähdekoodin lisenssit hyödyntämistilanteissa

Kaupallista ohjelmistokehitystä harjoittavan yrityksen mielenkiinto avoimen lähdekoodin lisensseissä kohdistuu ehtoihin teoksen levittämisestä, myymisestä, muuttamisesta ja yhdistämisestä omiin teoksiin. Perinteiset ohjelmistoyritykset näkevät tuottamansa lähdekoodin kilpailuvaltinaan, eivätkä halua paljastaa sitä kilpailijoilleen. Suljetun lähdekoodin avulla yrityksellä on myös mahdollisuus sitoa asiakas jatkossakin yrityksen asiakkaaksi esimerkiksi ylläpitopalveluilla ja ohjelmien uusilla versioilla. Mikäli lähdekoodi olisi vapaasti saatavilla, niin asiakkaalla olisi mahdollisuus hankkia vastaavia palveluita kolmansilta osapuolilta, tai kyetessään toteuttaa itse tarvittavat muutokset. Jotkut yritykset ovatkin kääntäneet avoimuuden kilpailuvaltiksi ja uskovat asiakkaiden arvostavan avoimia rajapintoja ja standardeja sekä lähdekoodin vapautta.

Avoimen lähdekoodin lisenssit sisältävät paljon muutakin kuin ehtoja lähdekoodin julkaisulle. Lähtökohtana niissä on, että teoksen vapaa käyttö, kopiointi, levitys ja muuntelu on sallittua. Jos avoimen lähdekoodin ohjelmistoa pelkästään käytetään työkaluna, niin lisensseistä ei tarvitse tietää juuri mitään. Tilanne muuttuu, jos teosta kehitetään edelleen tai yhdistetään se tai osia siitä toiseen teokseen ja levitetään muunneltua versiota muille.

### 5.1. Avoimen lähdekoodin hyödyntäminen

Avoimella lähdekoodilla on monia käyttökohteita ohjelmistoyrityksen toiminnassa. Valmiita avoimen lähdekoodin ohjelmistoja voidaan hyödyntää työkaluina, ohjelmistokomponentteja ja -kirjastoja voidaan liittää osaksi omaa tuotetta ja lähdekoodista voidaan ottaa oppia ja kopioida sitä suoraan omaan tuotteesseen. Tässä luvussa pohditaan, miten luvussa 4 esitellyt lisenssit suhtautuvat

erilaisiin avoimen lähdekoodin hyödyntämistapoihin ja mitä rajoituksia – jos mitään – ne asettavat esimerkiksi lähdekoodin sulkemiselle, johdannaisten teosten myynnille ja lisenssiehtojen muuttamiselle.

### 5.1.1. Käyttö työkaluna

Avoimen lähdekoodin lisenssit eivät rajoita niillä lisensoitujen ohjelmistojen käyttöä. Vapaa käyttö on ehtona sekä vapaan ohjelmiston että avoimen lähdekoodin määritelmässä, ja kaikki tässä tutkimuksessa käsitellyt lisenssit täyttävät avoimen lähdekoodin määritelmän. Avoimen lähdekoodin lisenssit sallivat ohjelmistojen käytön niin kaupallisiin kuin muihinkin tarkoituksiin, ja ne ovat näin myös yritysten käytettävissä ilman lisäehtoja [OSI, 2005b]. Lisäksi tekijänoikeuslaki ei lähtökohtaisesti koske käyttöä vaan ainoastaan kopiointia, levitystä ja muuntelua [HE, 2005].

Erikoistapaus on ohjelmistojen tuottama tuloste, jonka osalta voidaan joissakin tapauksissa pohtia, syntyykö käyttäjän antaman syötteen ja ohjelmiston suorittaman laskemisen tuloksena johdannainen teos, jota koskevat johdannaiseen teokseen liittyvät lisenssiehdot. Paras esimerkki tällaisesta tapauksesta on kääntäjäohjelma, joka saa syötteenä lähdekoodin ja tuottaa sen perusteella binäärikoodin. Mikäli kääntäjäohjelma ei sisällytä binäärikoodiin omaa koodiaan ja ilmaisujaan, niin kyseessä ei missään tapauksessa ole kääntäjäohjelman johdannainen teos [Välimäki, 2005a]. Jos kääntäjä lisää tuottamaansa binääriin omaa koodiaan, esimerkiksi kirjastojen muodossa, niin asiaa joudutaan tulkitsemaan tapauskohtaisesti.

Osa avoimen lähdekoodin lisensseistä huomioi ohjelmistojen tulosteen ehdoissaan, osa ei ota asiaan kantaa. GPL:ssa ohjelman tulosteesta mainitaan ympärilyöreästi lisenssin kohdassa 0 [OSI\_GPL, 1991] (käännös Välimäen [2001]):

... Ohjelman tuloste on tämän Lisenssin alainen vain silloin, kun se muodostaa Ohjelmaan perustuvan teoksen (riippumatta siitä ajetaanko Ohjelmaa vai ei). Milloin tuloste on Lisenssin alainen riippuu siitä, mitä Ohjelma tekee.

GPL:n kannanotto jättää hyvin paljon tulkinnanvaraa eikä anna yleispätevää vastausta. GPL-lisensoituun GCC-kääntäjään (GNU C Compiler) on lisätty erillinen vapautus GPL:n ehtoihin, jonka mukaan GCC:n tuottamat binäärit eivät kuulu GPL:n alle, vaikka ne sisältäisivät GCC-ohjelmiston kirjastoja. Ilman tätä poikkeusta ja tulkittaessa johdannaisten teosten syntyä tinkimättömästi GCC:llä olisi mahdollista muodostaa ainoastaan GPL-lisensoituja ohjelmia [Välimäki, 2005a].

Jätettäessä ohjelmistojen tulosteisiin liittyvät seikat huomioimatta, voidaan todeta avoimen lähdekoodin ohjelmistojen käytön alkuperäisessä muodossaan olevan vapaata niin kaupallisessa kuin muussakin toiminnassa.

#### **5.1.1.1 Työkaluna käytettävän ohjelmiston levitys**

Ohjelmistojen alkuperäisversioiden levitys on sallittua aina edellyttäen, että säilyttää kaikki alkuperäiset tiedostot säilytetään levitysversion mukana. Sallivat lisenssit BSD, MIT, PHP, Apache ja Artistic sallivat ohjelmistojen jakelun myös ilman lähdekoodia, kun jakeluversion mukaan liittää lisenssiehdot, tekijänoikeustiedot ja vastuurajoitusilmoituksen. GPL, LGPL ja MPL vaativat lähdekoodin vapaan saatavuuden.

#### **5.1.1.2 Työkaluna käytettävän ohjelmiston myynti**

Alkuperäisten ohjelmaversioiden myynnille ei yleensä ole rajoitteita, mutta ohjelmien levityksessä on luonnollisesti aina noudatettava kunkin lisenssin levitysehtoja. GNU-lisenssit eivät rajoita myytävän ohjelmiston hintaa. Myöskään sallivat lisenssit Apache, BSD, MIT ja PHP sekä pysyvä MPL eivät aseta ehtoja myyntihinnalle.

Artistic-lisenssissä todetaan, että sillä lisensoidun ohjelmiston levittämisestä saa periä kohtuullisen maksun. On hyvin vaikeaa määrittää kohtuullinen hinta ohjelmistotuotteelle, eikä ehto todennäköisesti rajoita ohjelmiston hinnoittelua käytännössä mitenkään [Laurent, 2004].

Voidaankin todeta, että tutkimuksen lisensseillä lisensoituja ohjelmistoja on sallittua myydä eteenpäin, ja myyntihinnan voi määrittää vapaasti itse. Tämä mahdollistaa liiketoiminnan perustamisen valmiiden ohjelmistojen myynnille, mutta toiminnan järkevyyttä luonnollisesti rajoittaa se tosiseikka, että ohjelmat ovat yleensä ilmaiseksi saatavilla Internetissä, ja kuka tahansa voi ruveta harjoittamaan samaa liiketoimintaa missä tahansa.

#### **5.1.1.3 Työkaluna käytettävän ohjelmiston lisenssin muuttaminen**

LGPL sallii sillä lisensoidun kirjaston levittämisen LGPL:n lisäksi velvoittavammalla GPL:lla. Muutoin mikään käsitellyistä lisensseistä ei salli ohjelmiston lisenssin muuttamista, mikäli ohjelmistoa jakaa alkuperäisessä muodossaan. Sallivan lisenssin ehtojen muuttaminen ei tavallisesti ole tarpeenkaan, sillä ne eivät juuri aseta rajoituksia niillä lisensoitujen teosten hyödyntämiselle. Esimerkiksi lähdekoodin julkaisua ei edellytä mikään tutkimuksen sallivista lisensseistä.

## 5.1.2. Rajapintakäyttö ja linkitys

### 5.1.2.1 Rajapintakäyttö

Ohjelmisto voi tarjota ulkoisia rajapintoja, joiden kautta sitä voidaan käyttää toisista ohjelmista. Hyödynnettäessä ohjelmistoa rajapintakutsujen avulla siitä ei tule kutsuvan ohjelmiston kiinteää osaa. Koska tekijänoikeuslaki ei koske käyttöä, niin komponentin ajaminen rajapinnan kautta muuntelematta millään tavoin komponentin lähdekoodia ei tavallisesti yhdistä kahta teosta ja muodosta johdannaista teosta. Näin ollen, jos ohjelmisto käyttää rajapinnan kautta avoimen lähdekoodin komponenttia tai ohjelmistoa, niin avoimen lähdekoodin ohjelmiston lisenssi ei voi asettaa ehtoja kutsuvalle ohjelmistolle.

Tietokoneeseen liitettävien laitteiden ajuriohjelmistot ovat esimerkki rajapintoja tarjoavista ohjelmista. Nykyään tietokoneohjelmat eivät laitteiden palveluita tarvitessaan kutsu suoraan laitteita vaan laitevalmistajien tai kolmansien osapuolten tarjoamia ajuriohjelmistoja, jotka välittävät kutsut laitteille. Jos kehitetään ohjelmisto, joka käyttää jotain tietokoneeseen liitettyä laitetta ajuriohjelmiston välityksellä, niin kuulostaisi omituiselta, että kehitetystä ohjelmistosta tulisi ajuriohjelmiston johdannainen teos. Näin ei tapahdukaan, vaan ajuriohjelmistoa ainoastaan käytetään toisesta ohjelmistosta, eikä tekijänoikeus näin ollen ulotu kutsuvaan ohjelmistoon.

Jos ohjelma käyttää hyväkseen käyttöjärjestelmän tarjoamia palveluja ja niiden avulla käynnistää toisen ohjelman, voidaan varsin selvästi todeta, että ohjelmat ovat toisistaan erillisiä ja itsenäisiä kokonaisuuksia. Jos taas rajapinnan kautta käynnistettävä ohjelma, esimerkiksi erillinen lisäohjelma (plug-in), on erityisesti tehty noudattamaan käynnistävän ohjelman lähettämiä käskyjä, niin ohjelmien yhteistoiminta on varsin tiivistä. Yleensä tällaisessa tapauksessa lisäohjelmaa ei voida lainkaan käyttää ilman kutsuvaa ohjelmaa ja voidaan kohtalaisen perustellusti olla sitä mieltä, että nämä kaksi ohjelmaa eivät ole erillisiä. [Iivonen ja Kähkönen, 2001]

Toteutettujen teknisten ratkaisujen lisäksi on syytä kiinnittää huomiota ohjelmien välillä siirrettävän tiedon määrään, merkittävyyteen ja sisältöön. Jos ohjelmien välinen tiedonvälitys käyttää korkealla tasolla määriteltyjä rajapintoja, jotka perustuvat edes jollain tavalla yleiseen standardiin, voidaan ohjelmia pitää erillisinä. Rajapinnan ollessa määrittelyltään tiukasti vain näihin kahteen ohjelmaan sidonnainen ja keskittynyt ohjelmien sisäisiin tietorakenteisiin voidaan ohjelmia pitää niin läheisinä, että ne muodostavat yhtenäisen kokonaisuuden. [Iivonen ja Kähkönen, 2001]

Tietokantoja käytetään yleensä erillisen tietokantarajapinnan kautta. GPL-lisenssoitu MySQL-tietokantajärjestelmä on herättänyt laajaa keskustelua siitä,



tarttuuko GPL tietokantaa rajapinnan välityksellä käyttävään ohjelmaan. Tekijänoikeuslain mukaan lisenssi ei tartu pelkän käytön seurauksena. MySQL Ab:n tulkinta asiasta poikkeaa tästä, mutta todennäköistä on, että oikeudessa asia päätettäisiin MySQL Ab:n tappioksi [Välimäki, 2005d]. On todennäköistä, että virallinen päätös riippuu siitä, miten tiukasti ohjelma on sidoksissa nimenomaan MySQL-tietokantaan. Jos ohjelma ei millään muotoa toimi ilman GPL:n alaista tietokantajärjestelmää, niin voidaanko tällöin vetää johtopäätös, että ohjelma on riippuvainen tietokantajärjestelmästä, ja siten vasta ohjelma ja tietokantajärjestelmä yhdessä muodostavat kokonaisen ohjelman? Tällaisessa tapauksessa voi olla mahdollista, että GPL tarttuu myös tietokantaa käyttävään ohjelmaan.

Asiakas-palvelin-rakenne on esimerkki tapauksesta, jossa kaksi ohjelmaa voidaan käynnistää erillään, mutta ohjelmien keskinäinen toiminta ja vuorovaikutus on vilkasta. Ohjelmat eivät välttämättä itsenäisinä tarjoa mitään hyötyä käyttäjälle, vaan varsinainen hyöty saadaan vain käytettäessä kumpaakin ohjelmaa yhdessä [Iivonen ja Kähkönen, 2001]. Perinteisesti asiakas-palvelin-rakenne tarkoittaa tilannetta, jossa asiakas ja palvelin ovat omissa prosessoriyksiköissään, jolloin ohjelmien välinen kommunikointi hoidetaan yleensä jonkin tietoliikenneverkon välityksellä, mutta rajapinnan yli tapahtuva käyttö voi tapahtua myös yhdessä prosessoriyksikössä. Vaikka asiakas- ja palvelinohjelmat eivät välttämättä itsenäisinä tarjoa hyötyä, niin ei voida silti perustellusti sanoa, että ne muodostaisivat keskenään yhtenäisen itsenäisen ohjelman varsinkaan, mikäli ohjelmien välinen yhteys on toteutettu jonkin kummastakin ohjelmasta erillisen yhteyskäytännön eli protokollan välityksellä [Iivonen ja Kähkönen, 2001]. Vaikka protokollan määrittely liittyisikin läheisesti toiseen ohjelmaan, ei tämä vielä tee ohjelmista yhtenäistä kokonaisuutta varsinkin, mikäli kumpikin ohjelma voisi toimia yhteydessä myös jonkin toisen ohjelman kanssa tai yksinään [Iivonen ja Kähkönen, 2001].

Apache-lisenssi poikkeaa muista tutkimuksen lisensseistä siten, että sen ehtoissa [OSI\_Apache, 2004] mainitaan yksiselitteisesti, että teoksesta tai johdannaisesta teoksesta erillisinä pysyvät tai ainoastaan niiden rajapintoihin linkittyvät teokset eivät ole johdannaisia teoksia.

Yhteenvetona voidaan todeta, että tapauksissa joissa ohjelma käynnistää toisen ohjelman ja ohjelmien välillä välitetään paljon tietoa käyttäen kovin läheisiä tiedonsiirtotapoja voidaan joutua tulkintatilanteeseen siitä, muodostavatko ohjelmat yhden yhtenäisen ohjelman. Niin kauan kuin ohjelmien välinen tiedonsiirto on vähäistä tai suorastaan olematonta, on turvallista sanoa niiden olevan erillisiä [Iivonen ja Kähkönen, 2001].

### 5.1.2.2 Staattinen ja dynaaminen linkitys

Avoimen lähdekoodin ohjelmistoja ja niiden osia voidaan hyödyntää linkittämällä ne mukaan suoritettavaan tiedostoon. Linkitys voidaan tehdä staattisesti tai dynaamisesti, ja linkitettävä teos on yleensä ohjelmakirjasto.

Staattisessa linkityksessä kaikki ohjelmakoodissa viitatus moduulit ja kirjastorutiinit on linkitetty ennen suoritusta, ja linkitettävät kirjastot ovat kiinteä osa ohjelmistoa. Linkitettävä koodi siis upotetaan ohjelmatiedostoon, ja symboli- viitteet selvitetään käynnöksen aikana. Staattisesti linkitettyssä ohjelmassa eri osien yhdistämisen tekee ohjelmoija kirjoittaessaan komennot, jotka johtavat erillisten osien yhdistämisen. Vaikka todellinen yhdistäminen tapahtuukin vasta linkitettäessä objektitiedostoissa ohjelmakirjastoa, ei linkitysohjelman käyttäjällä kuitenkaan ole muita vaihtoehtoja kuin yhdistää mainitut osat. Kiertääkseen tämän käyttäjän tulisi muuttaa ohjelman lähdekoodia. [Iivonen ja Kähkönen, 2001]

Staattisessa linkityksessä komponentin ja pääohjelman voidaan selvästi todeta muodostavan yhden ohjelman, sillä koodit on yhdistetty yhdeksi kokonaisuudeksi jo käynnöksen aikana. Staattinen linkitys tekeekin selvästi koko teoksesta osiensa johdannaisen teoksen [Välimäki, 2005a].

Dynaamisessa linkityksessä jätetään kutsukohdat moduuleihin auki ja linkitettävä kirjasto voidaan vaihtaa toiseen ilman ohjelmakoodin uutta kääntämistä. Siinä kirjastotiedostojen koodi ja data kuvataan ajonaikaisesti ohjelmaprosessin muistiavaruuteen ja viitteet selvitetään juuri ennen ohjelmakoodin suorituksen aloittamista [Kyrö, 2004]. Käytännössä on siis mahdollista, että käyttäjä voi vaihtaa ohjelman käyttämän dynaamisesti linkitetyn kirjaston ennen ohjelman suoritusta, ja ohjelma toimii myös tämän muutoksen jälkeen. Jos kirjasto olisi linkitetty ohjelmistoon staattisesti, niin sama muutos vaatisi käyttäjältä ohjelmiston koodin muokkaamista ja uudelleenkäntämistä. Dynaamista linkitystä käytettäessä ajoympäristön asetuksilla voidaan monesti ohjata sitä mistä osista ohjelma koostuu. Näin ollen ohjelman yhdistämisen tekee viime kädessä ohjelman lataava ja ajava henkilö, eli käyttäjä. [Iivonen ja Kähkönen, 2001]

Aikojen saatossa on usein kiistelty siitä, aiheuttaako kirjaston linkittäminen ohjelmaan johdannaisen teoksen syntymisen. Joitain aikoja sitten monet vapaan ohjelmiston yhteisön jäsenet väittivät, että ajonaikainen eli dynaaminen linkitys ei muodosta johdannaista teosta [Välimäki, 2005a]. Nykyään vapaan ohjelmiston kehittäjien keskuudessa yhteinen ymmärrys vaikuttaa olevan, että molempien linkitystapojen tulkitaan muodostavan johdannaisen teoksen [Välimäki, 2005a]. Jos avoimen lähdekoodin alainen funktio ladataan tietokoneen muistiin samaan aikaan funktiota kutsuvan pääohjelman kanssa ja linkitetään pääohjel-

maan, jolloin syntyy käytännössä yksi suoritettava tiedosto, niin koko teoksen voidaan tulkita olevan osiensa johdannainen teos [Välimäki, 2005a].

Erillisenä poikkeuksena käyttöjärjestelmän mukana tulevien komponenttien linkittämisen ei katsota synnyttävän johdannaista teosta. Tämä mahdollistaa avoimen lähdekoodin ohjelmien kehittämisen yksinoikeuslisenssillä varustettuihin käyttöjärjestelmiin kuten Microsoft Windowsiin. GPL:ssa tämä on erikseen mainittu lähdekoodin määritelmässä [OSI\_GPL, 1991].

FSF:n tulkinnan mukaan johdannaisten teosten syntyminen linkityksessä riippuu sekä ohjelmien keskustelutavasta että ohjelmien välillä vaihdetusta tiedosta [FSF, 2005d]. Jos ohjelmien osat on sisällytetty samaan suoritettavaan tiedostoon, niin ne ehdottomasti muodostavat yksittäisen ohjelman. Jos osat on suunniteltu ajettaviksi linkitettyinä jaetussa muistiavaruudessa, niin niiden voidaan suurella varmuudella todeta muodostavan yksittäisen ohjelman. Toisaalta, jos osat käyttävät tavallisia kommunikointitapoja keskinäiseen tietojen vaihtamiseen, niin osat voivat hyvinkin olla erillisiä ohjelmia. Jos kuitenkin kommunikoinnin merkitys on suuri ja siinä vaihdetaan monimutkaisia, sisäisiä tietorakenteita, niin voidaan jälleen pohtia, että muodostavatko ne yhden laajemman ohjelman.

Komponentin dynaaminen linkittäminen voi siis aiheuttaa vaikeita tulkintatilanteita, joihin pahimmassa tapauksessa joudutaan hakemaan lopullista vastausta oikeusistuimesta asti. Dynaamisessa linkittämisessä joudutaan tarkastelemaan sitä, miten tärkeä linkitetty komponentti on isäntäohjelmalle, toisin sanoen kuinka riippuvainen isäntäohjelma on linkitetystä komponentista. Jos isäntäohjelma vain käyttää linkitetyn komponentin palveluja, niin johdannaista teosta ei välttämättä synny. Samoin mikäli linkitetty komponentti sisältää vain joitain rutiinitoimintoja, jotka eivät ole elintärkeitä isäntäohjelmalle ja komponentti tulee valmiina käyttöjärjestelmän mukana, niin isäntäohjelman tuskin tulkitaan olevan komponentin johdannainen teos. Jos isäntäohjelma ei käynnisty tai suorita sille keskeisiä toimintoja ilman linkitettävää komponenttia, niin tilanne muuttuu epäselvemmäksi: jotta ohjelmaa voidaan käyttää, myös komponentti on kopioitava ja levitettävä ohjelman mukana. [Välimäki, 2005a]

Myös lisenssit voivat asettaa poikkeuksia sille, milloin ohjelmista syntyy johdannaista teoksia. Jos Apache-lisenssin alainen ohjelmistokirjasto linkitetään toiseen täysin erilliseen teokseen, niin tällöin ei Apache-lisenssin mukaan luoda johdannaista teosta, ja syntynyt teos on täten Apache-lisenssin ehtojen ulkopuolella [Välimäki, 2005d]. Muut sallivat lisenssit eivät ota kantaa johdannaisten teosten syntyamiseen linkittämisen yhteydessä.

GPL tarttuu koko teokseen linkitystavasta riippumatta. LGPL sallii sekä dynaamisen että staattisen linkittämisen ilman, että lisenssi kattaa koko synty-

vän teoksen. Tällöin on kuitenkin yhdistettyä teosta levitettäessä täytettävä LGPL:n kohdan 6 ehdot. Mikäli LGPL-lisensioitu ohjelmakirjasto on linkitetty teokseen staattisesti ja kirjastoa käyttävän teoksen lähdekoodi halutaan pitää suljettuna, niin LGPL:n ehtojen täyttäminen voi olla kohtuuttoman hankalaa. Jotta käyttäjä voisi LGPL:n ehdon 6 mukaisesti muuttaa kirjastoa ja käyttää muutettua versiota pääohjelman kanssa, olisi hänen päästävä käsiksi kirjaston lisäksi myös pääohjelman lähdekoodiin. Myös linkitettäessä LGPL-lisensioitua kirjastoa dynaamisesti on jo ohjelmaa suunniteltaessa hyvä ottaa huomioon LGPL:n ehdot, mikäli oman ohjelmaosuuden lähdekoodi halutaan pitää suljettuna.

MPL sallii molemmat linkitystavat ilman, että MPL-komponentin lisensiehtot välttämättä siirtyvät koko teokseen. Oleellista MPL:n tapauksessa on se, onko lopputuloksena MPL-komponentin johdannainen teos, jolloin koko teos on lisensoitava MPL:lla, vai laajempi teos, jossa MPL-komponentti muodostaa yhden osan suurempaa kokonaisuutta. Jälkimmäisessä tapauksessa MPL-koodin ulkopuolisen osan lisenssistä saa päättää vapaasti.

### 5.1.3. Lähdekoodin kopiointi

Tietokoneohjelmiston tekijänoikeus pätee vain ohjelmiston ilmaisuihin, ei ideoihin [Välimäki, 2005a]. Tämän vuoksi jokaisella on tekijänoikeuslain puitteissa mahdollisuus kehittää uusi käyttöjärjestelmä tai pakkausalgoritmi, vaikka vastaavia ideoita omaavia sovelluksia on jo aikaisemmin toteutettu. Olemassa olevia ohjelmia voi tutkia ja analysoida uuden teoksen pohjana, ja ainoastaan kirjaimellinen lähde- tai objektikoodin kopioiminen on kielletty, ellei tekijänoikeuden haltija ole sitä erikseen sallinut.

Avointa lähdekoodia voidaan hyödyntää muutenkin kuin valmiina komponentteina tai ohjelmistoina. Avoimen lähdekoodin ohjelmistojen lähdekoodista saa ottaa mallia ja ideoita, niitä saa tutkia, ja saatuja tuloksia saa hyödyntää. Nämä toimenpiteet eivät aseta vaateita hyödyntäjälle. Kopioitaessa lähdekoodia omaan ohjelmaan on kuitenkin muistettava, että tällöin kopioidaan toisen tekijänoikeudenalaista materiaalia, ja lähdekoodin lisenssin ehdot pätevät kyseiseen lähdekoodin osaan.

Voidaan kuvitella tilanne, jossa osia avoimen lähdekoodin ohjelmiston tai kirjaston funktioista käytetään suoraan mukana jo käänöksessä, eli kopioitu koodi muodostaa kiinteän osan uuden ohjelmiston lähdekoodia ja käännettyssä muodossa osan suoritettavaa koodia. Tämä on hyvin samankaltainen tilanne kuin komponentin staattinen linkittäminen, ja myös seuraamukset koodin liittämisestä ovat samanlaiset. Lähdekoodin kopioimisessa on kuitenkin se poikkeava piirre valmiin kirjaston linkittämiseen, että edellisessä ei välttämättä ko-

pioida mitään valmista toimivaa kokonaisuutta, vaan kopiointi voi koskea vain yhtä funktiota tai funktion osaa. Tämä voi aiheuttaa lisätulkintoja siitä, kuinka laajalti kopioitavan koodin lisenssi vaikuttaa isäntäohjelmaan, ja muodostuuko pienenkin koodimäärän kopioinnin seurauksena johdannainen teos. Tällaisissa tilanteissa täytyy tapauskohtaisesti pohtia kopioidun koodin merkitystä koko teokselle ja miettiä vaihtoehtoisia toimintatapoja kopioinnille kuten koodin uudelleenkirjoittamista itse.

## **5.2. Lisenssien vaikutus muuntelutilanteissa**

### **5.2.1. Muunnellun ohjelmiston sisäinen käyttö**

Avoimen lähdekoodin ohjelmiston muuntelu yrityksen sisäiseen käyttöön ei aseta yritykselle lisävelvoitteita. Toisin sanoen, jos muunneltua ohjelmistoa käytetään vain organisaation sisällä, niin levityksen ehdot eivät koske kyseistä ohjelmistoa – ohjelmaahan ei tällöin levitetä. Tämä kattaa myös nk. sovellusvuokrauspalvelut ja verkkosovellukset, joissa asiakas ajaa ohjelmistoa verkon yli toimittajan tietokoneella. Verkkokäytössä ja sovellusvuokrauksessa asiakkaalle ei toimiteta fyysistä kopiota ohjelmistosta, vaan ohjelmiston kopio sijaitsee koko ajan toimittajan palvelimella. Verkossa toimiva sovellusvuokraaminen on eräs keino välttää lisenssien levittämistä koskevat ehdot.

Kun ohjelmistoa ei levitä, niin mikään tässä tutkimuksessa käsitelty lisenssi ei vaadi lähdekoodin julkistamista. Esimerkiksi GPL:n alaisen WWW-ohjelmiston laittaminen suoritettavaksi julkiselle HTTP-palvelimelle ei ole ohjelmiston levittämistä, eikä esimerkiksi ohjelmiston lähdekoodia täydy tällöin julkaista. Tarkasteltujen lisenssien näkökulmasta on siis eri asia laittaa WWW-ohjelmisto suoritettavaksi toimittajan tietokonepalvelimelle kuin asentaa se asiakkaan tietokoneelle, vaikka kummassakin tapauksessa asiakas käyttäisi ohjelmistoa WWW-selaimellaan.

### **5.2.2. Muunnellun ohjelmiston levitys**

Kun ohjelmistoa jaetaan itsenäisinä tiedostoina jollakin medially tai Internetistä ladattavana pakettina, niin kyseessä on ohjelmiston levitys, joka on hoidettava ohjelmiston lisenssin asettamilla ehdoilla. Muunnellun ohjelmiston kohdalla merkityksellistä avoimen lähdekoodin lisenssien näkökulmasta on se, että tulkitanko syntynyt teos alkuperäisen teoksen johdannaiseksi teokseksi vai uudeksi, itsenäiseksi teokseksi.

Mikäli sallivan lisenssin alaista ohjelmistoa muuttaa tarpeeksi, niin että tuloksena syntyy uusi ohjelmisto, niin uusi ohjelmisto voidaan lisensoida itse va-

lituilla ehdoilla. Muussa tapauksessa kyseessä on alkuperäisen teoksen johdannainen teos, jonka alkuperäisen teoksen osat säilyttävät vanhan lisenssin ja uudet osat voi lisensoida haluamallaan ehdoilla. Sallivien lisenssien tapauksessa lähdekoodia ei ole pakko julkaista missään tilanteessa.

GPL:lla tai MPL:lla varustettujen ohjelmistojen tai komponenttien muunnellut versiot on lisensoitava alkuperäisen ohjelmistoversion lisenssillä. Tällöin koko teoksen lähdekoodi on julkaistava, mikäli teosta levittää. GPL sallii kirjastoa käyttävän teoksen vapaan lisensoinnin, kun teosta käytetään kirjastosta erillään. Jos kirjasto liitetään mukaan kirjastoa käyttävään teokseen ja luodaan yhdistetty teos, niin lähtökohtaisesti koko teos on lisensoitava GPL:lla. Tästä voidaan kuitenkin poiketa, ja lisensoida kirjastoa käyttävä teos vapaasti sillä edellytyksellä, että täytetään GPL:n kohdan 6 levitysehdot, jotka vaativat mm. lähdekoodin luovuttamista ja teoksen muuntelun sallimista asiakkaalle.

MPL:n tapauksessa muunnellun ohjelmiston levittämisessä on huomioitava, että alkuperäiseen ohjelmistoon liitetyt patentti- ja tuotemerkkioikeudet raukeavat ohjelmiston muuttamisen yhteydessä. Ohjelmiston muuttamisen lisäksi MPL sallii MPL-koodin yhdistämisen lisenssin ulkopuolisen koodin kanssa, jolloin ulkopuolisen koodin lisenssiehdot saa määrittää vapaasti ja ulkopuolisen koodin pitää salassa myös levitettäessä syntynyttä yhdistelmäteosta. Erikoisuutena MPL sallii ohjelmiston suoritettavan version levityksen minkä tahansa lisenssin alaisena, mutta suoritettavan lähdekoodi on kuitenkin julkaistava ja MPL:n ehdot täytettävä muiltakin osin.

GPL:n alaisia osia sisältävät ohjelmistot on levitettävä kokonaisuudessaan GPL:n alla ja ohjelmiston lähdekoodi julkaistava GPL:n ehtojen mukaisesti.

### 5.2.3. Muunnellun ohjelmiston hinnoittelu

Avoimen lähdekoodin lisenssit voivat asettaa rajoituksia ohjelmistojen ja niiden johdannaisten teosten hinnoittelulle tai kieltää maksun perimisen itse ohjelmistosta kokonaan, jolloin tulot on kerättävä esimerkiksi tuotteen ympärille koostuista palveluista.

GPL vaatii, että lisenssin alaiset johdannaiset teokset on lisensoitava ilman maksua kolmansille osapuolille, mutta toteaa myös, että ohjelman kopioimisesta voi pyytää maksun, jonka summaa ei ole lisenssissä erikseen rajattu. GPL noudattaa GPL:n linjaa alkuperäisen ja johdannaisten teoksen osalta, mutta mikäli GPL-kirjaston yhdistää laajempaan teokseen ja teosta levittää GPL:n kohdan 6 mukaisesti, niin teoksen saa hinnoitella vapaasti. Käytännössä kumpikaan lisenssi ei todennäköisesti aseta rajoituksia hinnoittelulle.

Artistic-lisenssi sallii sen alaista koodia sisältävien ohjelmistojen myymisen kohtuulliseen hintaan, joka tarkoittanee markkinoiden kestämää hintaa. Juridi-

sesti tällainen ehto ei tuo lisenssiin mitään lisää [Laurent, 2004] eikä käytännössä rajoita ohjelmiston hinnoittelua mitenkään.

BSD-, MIT-, Apache- ja PHP-lisenssi eivät rajoita niillä lisensoitujen ohjelmistojen, ohjelmistojen johdannaisten ja ohjelmistoja sisältävien uusien teosten myyntiä millään tavoin. Myöskään MPL ei aseta rajoituksia maksun perimiselle.

Ohjelmistomaksujen lisäksi yritys voi hinnoitella esimerkiksi tekemänsä työn, paketoinnin, ohjeistuksen, tuotetuen, takuun ja ylläpidon erikseen. Näihin lisenssien mahdolliset rajoitukset eivät yllä, ja niiden hinnoittelu on toimitajan vapaasti päätettävissä.

#### 5.2.4. Muunnellun ohjelmiston lisenssin muuttaminen

GPL on sekä pysyvä että tarttuva lisenssi, ja se on ehdoton lisenssiehtojen muuttamisen suhteen: jos ohjelmisto sisältää GPL-lisensoidun ohjelmiston tai ohjelmiston osan, niin kokonaisuus on lisensoitava GPL:lla.

LGPL on pysyvä lisenssi, joka säilyy kaikissa LGPL-ohjelman muunnelmissa. LGPL sallii minkä tahansa lisenssin käytön LGPL-kirjastoa käyttävässä teoksessa, jos teosta levittää erillään LGPL-kirjastosta. Myös kirjastoa käyttävä teos on lisensoitava LGPL:lla, mikäli jakeluun liittyy kyseisen LGPL-kirjaston eikä ole valmis täyttämään LGPL:n kohdan 6 ehtoja. Näiden lisäksi LGPL sallii lisenssin vaihtamisen velvoittavampaan GPL:iin, mutta tämä vaihtoehto tuo harvoin lisäarvoa yritykselle.

BSD-, MIT-, PHP- ja Apache-lisenssit sallivat lisenssiehtojen muuttamisen, mikäli teosta muunnellaan sen verran, että siitä tulee uusi tai johdannainen teos, johon muuntelijalla on tekijänoikeus [Välimäki, 2002]. Lisenssin muuttamisoikeuteen ei riitä se, että valmista ohjelmaa muuttaa hieman esimerkiksi korjaamalla virheitä tai lisäämällä muutaman uuden ominaisuuden, vaan ohjelmaan on tehtävä perusteellisempia uudistuksia, sen toimintatarkoitus on muutettava tai ohjelmisto on liitettävä uuteen itsenäiseen ohjelmistoon, joka käyttää liitettyä ohjelmistoa jollakin tavalla.

Artistic-lisenssi suhtautuu lisenssin muuttamiseen mahdollisesti samalla tavalla kuin sallivat lisenssit. Lisenssin epäselvyyden takia tämä ei ole selvää, mutta mikäli Artistic-lisenssillä varustettua ohjelmistoa muuntelee niin paljon, että syntyy uusi teos, niin teoksen saa mitä ilmeisimmin lisensoida haluamallaan lisenssillä. Varmuuden vuoksi voi olla kuitenkin järkevää mainita teoksessa käytettävän Artistic-lisenssin alaisia osia, ja että teoksen toimittaja on vastuussa kyseisten osien toimivuudesta.

MPL on pysyvä lisenssi ja siksi kaikki MPL-lisensoidun ohjelmiston johdannaisten teokset on lisensoitava MPL:lla. Jos MPL-lisensoitua koodia ja lisens-

sin ulkopuolista koodia yhdistämällä luodaan uusi teos, niin MPL:n ulkopuolisen osan lisensoinnista saa päättää itse edellyttäen, että valittu lisenssi sallii useamman lisenssin käytön samassa teoksessa. Hyödynnetty MPL-koodi on tässäkin tapauksessa lisensoitava MPL:lla.



## 6. Yhteenveto

Tässä tutkimuksessa on käsitelty avointa lähdekoodia sekä avoimen lähdekoodin lisensointia ja paneuduttu perusteellisesti kahdeksaan avoimen lähdekoodin lisenssiin. Erityisen tarkastelun aiheena on ollut avoimen lähdekoodin lisenssien soveltuvuus yrityksen kaupalliseen ohjelmistokehitykseen ja lisenssien ehdot erilaisissa hyödyntämistilanteissa.

Tutkimuksessa on käynyt selväksi, että avoimen lähdekoodin kaupallinen hyödyntäminen on hyvinkin mahdollista, ja avoin lähdekoodi voi tarjota liiketoiminnalle merkittäviä etuja. Onnistunut avoimen lähdekoodin hyödyntäminen edellyttää avoimen lähdekoodin käsitteen ymmärtämistä ja yhteisten pelisääntöjen tuntemista. Avoimen lähdekoodin lisenssit asettavat säännöt niillä lisensoitujen teosten hyödyntämiselle, joten lisenssiehtojen ymmärtäminen ja noudattaminen ovat ensimmäiset edellytykset, kun avointa lähdekoodia ryhdytään käyttämään omassa toiminnassa. Sallivat lisenssit antavat lähes täysin vapaat kädet niillä lisensoitujen teosten käyttöön, kun taas velvoittavat lisenssit, tunnetuimpana GPL, saattavat sisältää ehtoja, jotka rajoittavat niillä lisensoitujen teosten hyödyntämistä kaupallisesti. Ei voida kuitenkaan sanoa, että jokin tietty lisenssityyppi ei sovellu missään tapauksessa yritysten käyttöön, sillä avoimen lähdekoodin hyödyntämistavat ovat hyvin monimuotoiset, ja lisenssin sopivuutta on tarkasteltava aina tapauskohtaisesti.

Kaksi suosituinta avoimen lähdekoodin lisenssiä, GNU-lisenssit GPL ja LGPL, sisältävät tiukkoja ehtoja niillä lisensoitujen lähdekoodien levittämiseksi. Myös MPL luokitellaan velvoittavaksi lisenssiksi, ja se sisältää LGPL:n tapaan lisenssin pysyvyysominaisuuden. Yritysten onkin oltava erityisen tarkkoja GPL:n, LGPL:n ja MPL:n suhteen. Tutkimuksessa tarkastellut sallivat lisenssit, BSD-, MIT-, PHP-, Apache- ja Artistic-lisenssi, antavat paljon suuremmat vapaudet hyödyntää ohjelmistoja ja ohjelmakomponentteja, mutta myös niillä lisensoitujen teosten kohdalla on muistettava, että kyseessä ovat tekijänoikeus-

lain alaiset teokset. Sallivilla lisensseillä varustettuja teoksia on tarjolla valtava määrä, joten mikäli yritys ei halua tai pysty hyödyntämään velvoittavilla lisensseillä lisensoituja teoksia, niin avointa lähdekoodia ei tarvitse silti hylätä.

GPL, LGPL ja MPL edellyttävät niillä lisensoitujen teosten ja johdannaisten lähdekoodin vapauttamista, mikäli teoksia levitetään. Lisäksi ne estävät yksinoikeusohjelmien luomisen joko kokonaan tai osittain. Nämä ovat merkittäviä rajoittavia tekijöitä, kun pohditaan lisensseillä varustettujen komponenttien hyödynnettävyyttä kaupallisesti. LGPL ja MPL kuitenkin tarjoavat GPL:aa enemmän vapauksia yhdistetyissä teoksissa, joissa oma sovellusosa käyttää avoimen lähdekoodin komponenttia. Tällöin yksinoikeusohjelman luominen ja oman sovellusosan lähdekoodin sulkeminen on mahdollista, joskin kaikkien LGPL:n ehtojen täyttäminen voi vaatia erityisiä sovellusarkkitehtuuritason ratkaisuja, jotka voivat käydä kalliiksi ja vaatia ikäviä kompromisseja toteutuksessa.

Tutkimuksessa käsitellyt sallivat lisenssit eivät edellytä avoimen lähdekoodin komponentin tai oman sovellusosan lähdekoodin julkaisua missään tilanteessa. Ne sallivat yksinoikeusohjelmien luomisen ja soveltuvat hyödynnettäväksi kaupallisessa maailmassa erinomaisesti.

Erityisenä yksityiskohtana on huomattava, ettei mikään tutkituista lisensseistä tulkitse verkon yli tapahtuvaa ohjelmiston käyttöä ohjelmiston levittämiseksi. Tämän vuoksi esimerkiksi GPL-lisensoidun WWW:ssä toimivan ohjelmiston ajaminen omalla HTTP-palvelimella ei edellytä kyseisen ohjelmiston lähdekoodin julkaisua. Tämä mahdollistaa myös velvoittavilla avoimen lähdekoodin lisensseillä lisensoitujen komponenttien hyödyntämisen omassa yksinoikeudellisissa verkko-ohjelmistoissa ja ohjelmistovuokrauspalveluissa. Samoin hyödynnettäessä avoimen lähdekoodin ohjelmistoja ainoastaan organisaation sisällä ei täytetä ohjelmiston levityksen määritelmää, eivätkä lisenssien levitystä koskevat ehdot ole voimassa. Joissakin tilanteissa voi kuitenkin olla epäselvää, mikä tulkitaan organisaation sisällä tapahtuvaksi käytöksi ja milloin ylitetään tämä raja.

On tiedostettava, että vaikka lisenssi ei ehdoissaan eksplisiittisesti rajoittaisi jotain yksittäistä toimenpidettä, niin vallitseva laki saattaa näin tehdä. Toisaalta lisenssit voivat sisältää ehtoja, joita pitäisi noudattaa, vaikka niitä ei olisi erikseen kirjoitettu lisenssiin. Yleensä tulkintatilanteita ja epäselvyyksiä vähentää, jos lisenssissä on erikseen määritelty ehdot erilaisiin tilanteisiin.

Tutkimuksessa on käsitelty yksi tarttuva, kaksi pysyvää ja kuusi sallivaa lisenssiä. Varsinkin sallivat lisenssit ovat keskenään hyvin samankaltaisia, ja on perusteltua todeta, että tutkimuksessa olisi voinut käsitellä enemmän toisistaan

poikkeavia lisenssejä. Lähtökohtana oli valita tutkittavaksi suosituimmat lisenssit, mutta epäonneksi suuri osa niistä on keskenään hyvin samankaltaisia.

Avoimen lähdekoodin lisenssien tulkinnanvaraisuuksien vuoksi tämä tutkimus ei pysty antamaan vastauksia kaikkien tutkittujen lisenssien vaikutuksista erilaisissa hyödyntämistilanteissa. Esimerkiksi rajanpintakäytön ja dynaamisen linkityksen tapauksessa liikutaan nk. harmaalla alueella, jossa saatetaan joutua hakemaan lopullista ratkaisua oikeusistuimesta asti. Oikeuden päätöstäkin on vaikea ennustaa, koska ennakkotapauksia ei juuri ole.

Avoimen lähdekoodin lisenssien suuri määrä, lisenssien yhteensopimattomuudet, ehtojen epämääräisyydet ja epävarmuus ehtojen kestävydestä oikeudessa ovat esimerkkejä avoimen lähdekoodin lisenssien heikkouksista ja riskeistä, jotka on tiedostettava, mikäli avoin lähdekoodi otetaan mukaan omaan toimintaan. Avoimen lähdekoodin käyttöön on erikseen valmistauduttava ja laadittava organisaatiossa yleiset ohjeet ja pelisäännöt avoimen lähdekoodin käytölle sekä valvottava sääntöjen noudattamista. Lisäksi avoimen lähdekoodin käytöstä on sovittava aina projektikohtaisesti erityisesti velvoittavien lisenssien osalta. Sallivilla lisensseillä lisensoitujen komponenttien hyödyntämisessä voidaan usein olla vapaamielisempiä, mutta myös niiden osalta on muistettava täyttää lisenssien ehdot kuten tekijänoikeustietojen, lisenssiehtojen ja vastuunrajoitusten liittäminen komponentteja hyödyntäviin teoksiin.

Näyttää siltä, että sekä avoimen lähdekoodin ohjelmistojen määrä että avoimen lähdekoodin hyödyntäminen ovat jatkuvassa kasvussa [Lacy, 2005]. Erityistä kasvua avoimen lähdekoodin käytössä on odotettavissa yrityksissä ja julkisella sektorilla. On tärkeää tehdä lisätutkimusta avoimen lähdekoodin mahdollisuuksista eri toimialoilla, mutta myös avoimen lähdekoodin yhteisöjen ja kaupallisen sektorin vuorovaikutuksesta, ja tämän suhteen kehittämisestä. Lisensseihin liittyen olisi mielenkiintoista tietää, millä avoimen lähdekoodin lisensseillä lisensoituja ohjelmia ja komponentteja hyödynnetään eniten yritysmaailmassa. Todennäköisesti GNU-lisensoidut teokset eivät ole tiukkojen ehtojensa vuoksi yrityksissä niin yleisiä kuin avoimen lähdekoodin yhteisöjen käytössä.

GPL tulee säilyttämään asemansa merkittävimpänä avoimen lähdekoodin lisenssinä lähitulevaisuudessa, ja lisenssin kolmannen version sisältö ja vaikutukset ohjelmistoteollisuuteen kiinnostavat monia tahoja. Onkin todennäköistä, että monia GPL 3.0:a käsitteleviä tutkimuksia tullaan näkemään lähiaikoina.

Yritysten kannattaa tutustua avoimeen lähdekoodin avoimin mielin ja tutkia sen mahdollisuuksia omalle liiketoiminnalle. On ilmeistä, että avoin lähdekoodi tulee olemaan jatkossa vielä nykyistä merkittävämpi tekijä, joka tukee nykyisiä toimintamalleja ja luo kokonaan uusia liiketoimintamuotoja. Jo nyt

avoimen lähdekoodin lisensseillä on lisensoitu kaikkien vapaaseen käyttöön valtava määrä ohjelmistoja ja ohjelmakomponentteja, jotka vain odottavat hyödyntämistä ja jatkokehittämistä.

## Lähdeluettelo

- [ASF, 2005a] The Apache Software Foundation, Apache License v2.0 and GPL Compatibility. <http://www.apache.org/licenses/GPL-compatibility.html> (8.9.2005).
- [ASF, 2005b] The Apache Software Foundation, Proposed Licenses. <http://www.apache.org/licenses/proposed/> (8.9.2005).
- [ASF, 2005c] The Apache Software Foundation, The Apache HTTP Server Project. <http://httpd.apache.org/> (9.9.2005).
- [ASF, 2005d] The Apache Software Foundation, How the ASF Works. <http://www.apache.de/foundation/how-it-works.html> (5.9.2005).
- [CC, 2005a] Creative Commons. <http://creativecommons.org/> (29.9.2005).
- [CC, 2005b] Creative Commons, Frequently Asked Questions. <http://creativecommons.org/faq> (1.12.2005).
- [CC, 2005c] Creative Commons, Public Domain Dedication. <http://creativecommons.org/licenses/publicdomain/> (1.12.2005).
- [CC Suomi, 2005] Creative Commons Suomi. <http://creativecommons.fi/> (29.9.2005).
- [Debian, 2004] Debian Project, Debian Social Contract, version 1.1. [http://www.debian.org/social\\_contract](http://www.debian.org/social_contract) (23.9.2005).
- [DiBona *et al.*, 1999] Chris DiBona, Sam Ockman and Mark Stone, *Open Sources: Voices from the Open Source Revolution*. O'Reilly, 1999. Also available as <http://www.oreilly.com/catalog/opensources/book/toc.html> (16.5.2006).
- [ETY, 1991] Euroopan unionin neuvoston direktiivi tietokoneohjelmien oikeudellisesta suojasta (91/250/ETY). Saatavilla myös [http://europa.eu.int/smartapi/cgi/sga\\_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=FI&numdoc=31991L0250&model=guichett](http://europa.eu.int/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=FI&numdoc=31991L0250&model=guichett) (27.11.2005).
- [FLOSSmole, 2006] FLOSSmole Project, Collaborative collection and analysis of free/libre/open source project data. <http://ossmole.sourceforge.net/> (28.1.2006).
- [Fogel, 2005] Karl Fogel, *Producing Open Source Software: How to Run a Successful Free Software Project*. O'Reilly, 2005. Also available as <http://www.oreilly.com/catalog/producingoss/> (13.5.2006).
- [Ford, 2000] Nelson Ford, The History of Shareware & PsL. <http://www.asp-shareware.org/users/history-of-shareware.asp> (26.9.2005).
- [freshmeat, 2006a] freshmeat. <http://freshmeat.net/> (16.5.2006).
- [freshmeat, 2006b] freshmeat, About. <http://freshmeat.net/about/> (16.5.2006).

- [FSF, 1989] Free Software Foundation, GNU General Public License Version 1.  
<http://www.gnu.org/copyleft/copying-1.0.html> (18.7.2005).
- [FSF, 1991] Free Software Foundation, GNU General Public License Version 2.  
<http://www.fsf.org/licensing/licenses/gpl.html> (5.9.2005).
- [FSF, 2001a] Free Software Foundation, Selling Free Software.  
<http://www.fsf.org/licensing/essays/selling.html> (5.9.2005).
- [FSF, 2001b] Free Software Foundation, GNU Library General Public License.  
<http://www.gnu.org/copyleft/library.html> (7.5.2006).
- [FSF, 2003] Free Software Foundation, The BSD License Problem.  
<http://www.fsf.org/licensing/essays/bsd.html> (5.9.2005).
- [FSF, 2004] Free Software Foundation, The Free Software Definition.  
<http://www.fsf.org/licensing/essays/free-sw.html> (5.9.2005).
- [FSF, 2005a] Free Software Foundation, Licenses.  
<http://www.fsf.org/licensing/licenses/> (5.9.2005).
- [FSF, 2005b] Free Software Foundation, The Free Software Definition.  
<http://www.fsf.org/licensing/essays/free-sw.html> (1.8.2005).
- [FSF, 2005c] Free Software Foundation, Categories of Free and Non-Free Software. <http://www.fsf.org/licensing/essays/categories.html> (20.9.2005).
- [FSF, 2005d] Free Software Foundation, Frequently Asked Questions about the GNU GPL. <http://www.fsf.org/licensing/licenses/gpl-faq.html> (20.1.2006).
- [FSF, 2005e] Free Software Foundation, GPLv3 Development and Publicity Project. <http://gplv3.fsf.org/process-definition> (4.12.2005).
- [FSF, 2005f] Free Software Foundation, The GNU Operating System.  
<http://www.gnu.org/> (29.6.2005).
- [Greene, 2001] Thomas Greene, Ballmer: "Linux is a cancer".  
[http://www.theregister.co.uk/2001/06/02/ballmer\\_linux\\_is\\_a\\_cancer/](http://www.theregister.co.uk/2001/06/02/ballmer_linux_is_a_cancer/) (20.9.2005).
- [Grisoft, 2004] Grisoft s.r.o., AVG Anti-Virus Free Edition.  
<http://free.grisoft.com/freeweb.php> (26.9.2005).
- [HE, 2005] Suomen laki, tekijänoikeuslaki 8.7.1961/404. Saatavilla myös  
<http://www.finlex.fi/fi/laki/ajantasa/1961/19610404> (15.9.2005).
- [Hoskins, 1999] William Hoskins, BSD license change.  
<ftp://ftp.cs.berkeley.edu/pub/4bsd/README.Impt.License.Change> (15.10.2005).
- [Iivonen ja Kähkönen, 2001] Joonas Iivonen ja Sari Kähkönen, GPL-lisenssistä. Teknologiaoikeuden seminaari, Teknillinen korkeakoulu.
- [KTM, 2005] Kauppa- ja teollisuusministeriö, Teollisoikeudet.  
<http://www.ktm.fi/index.phtml?s=862> (21.4.2006).

- [Klincewicz, 2005] Krzysztof Klincewicz, Innovativeness of open source software projects. Tokyo Institute of Technology, School of Innovation Management, Report **W9-108**, August 2005. Also available as <http://opensource.mit.edu/papers/klincewicz.pdf> (3.12.2005).
- [Korpela, 2005] Jukka Korpela, Tekijänoikeus: vastauksia usein esitettyihin kysymyksiin. <http://www.cs.tut.fi/~jkorpela/tekoik/all.html> (24.5.2006).
- [Kyrö, 2004] Jarkko Kyrö, Verkkokäyttö, prosessien välinen viestintä ja prosessin ajo. Linux kernel –seminaari, Helsingin yliopisto.
- [Lacy, 2005] Sarah Lacy, A watershed for open source. Available as [http://www.businessweek.com/technology/content/dec2005/tc20051228\\_262746.htm](http://www.businessweek.com/technology/content/dec2005/tc20051228_262746.htm) (5.3.2006).
- [LG München, 2004] Landgericht München I, 19.5.2004. Also available as [http://www.jbb.de/urteil\\_lg\\_muenchen\\_gpl.pdf](http://www.jbb.de/urteil_lg_muenchen_gpl.pdf) (7.1.2006). An unofficial translation available as [http://www.jbb.de/judgment\\_dc\\_munich\\_gpl.pdf](http://www.jbb.de/judgment_dc_munich_gpl.pdf) (7.1.2006).
- [Laurent, 2004] Andrew M. St. Laurent, *Understanding Open Source and Free Software Licensing*. O'Reilly, 2004. Also available as <http://www.oreilly.com/catalog/osfreesoft/book/> (9.9.2005).
- [Lerner and Tirole, 2002] Josh Lerner and Jean Tirole, Some simple economics of open source. *Journal of Industrial Economics* **L**, 2 (2002), 197-234. Also available as <http://turingmachine.org/opensource/papers/lerner2002.pdf> (22.9.2005).
- [Levesque, 2004] Michelle Levesque, Fundamental issues with open source software development. *First Monday* **9**, 4 (2004). Also available as [http://firstmonday.org/issues/issue9\\_4/levesque/](http://firstmonday.org/issues/issue9_4/levesque/) (6.10.2005).
- [Moody, 2001] Glyn Moody, *Kapinakoodi – Linus Torvalds ja vapaan ohjelmoinnin vallankumous*. Gummerus Kirjapaino Oy, Jyväskylä, 2001.
- [Mozilla, 2005a] Mozilla Foundation, Mozilla Products. <http://www.mozilla.org/products/> (18.9.2005).
- [Mozilla, 2005b] Mozilla Foundation, Mozilla Relicensing FAQ, version 1.1. <http://www.mozilla.org/MPL/relicensing-faq.html> (18.9.2005).
- [Mozilla, 2005c] Mozilla Foundation, About the Mozilla Foundation. <http://www.mozilla.org/foundation/> (18.9.2005).
- [Mozilla, 2005d] Mozilla Foundation, Netscape Public License FAQ. <http://www.mozilla.org/MPL/FAQ.html> (18.9.2005).
- [Mozilla, 2005e] Mozilla Foundation, Annotated Mozilla Public License, version 1.1. <http://www.mozilla.org/MPL/MPL-1.1-annotated.html> (18.9.2005).

- [MySQL, 2005] MySQL AB, FLOSS License Exception v0.4.  
<http://www.mysql.com/company/legal/licensing/foss-exception.html>  
(20.9.2005).
- [Netcraft, 2002] Netcraft, Netcraft Web Server Survey.  
<http://survey.netcraft.com/index-200204.html> (9.9.2005).
- [Netcraft, 2005] Netcraft, September 2005 Web Server Survey.  
[http://news.netcraft.com/archives/2005/09/05/september\\_2005\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2005/09/05/september_2005_web_server_survey.html) (9.9.2005).
- [Netscape, 1998] Netscape Communications Corporation, Netscape Announces Plans to Make Next-Generation Communicator Source Code Available Free on the Net.  
<http://wp.netscape.com/newsref/pr/newsrelease558.html> (2.8.2005).
- [Nykänen, 2004] Pirkko Nykänen, Esitys aiheesta *Tietojärjestelmä- ja ohjelmistoprojekteihin liittyvistä sopimuksista, sopimuskäytännöistä ja oikeuksista*, 2004.
- [OpenOffice.org, 2006] OpenOffice.org. <http://www.openoffice.org/>  
(25.3.2006).
- [OSI, 2005a] Open Source Initiative, History of the OSI.  
<http://www.opensource.org/docs/history.php> (2.8.2005).
- [OSI, 2005b] Open Source Initiative, The Open Source Definition.  
<http://www.opensource.org/docs/definition.php> (2.8.2005).
- [OSI, 2005d] Open Source Initiative, Certification Mark and Process.  
[http://www.opensource.org/docs/certification\\_mark.php](http://www.opensource.org/docs/certification_mark.php) (26.9.2005).
- [OSI, 2006] Open Source Initiative, Licensing.  
<http://www.opensource.org/licenses/> (8.5.2006).
- [OSI\_Apache, 2004] Open Source Initiative, Apache License, Version 2.  
<http://www.opensource.org/licenses/apache2.0.php> (20.7.2005).
- [OSI\_Artistic, 2005] Open Source Initiative, The Artistic License.  
<http://www.opensource.org/licenses/artistic-license.php> (13.9.2005).
- [OSI\_BSD, 2005] Open Source Initiative, The BSD License.  
<http://www.opensource.org/licenses/bsd-license.php> (20.7.2005).
- [OSI\_GPL, 1991] Open Source Initiative, The GNU General Public License (GPL), Version 2, June 1991. <http://www.opensource.org/licenses/gpl-license.php> (20.4.2006).
- [OSI\_LGPL, 1999] Open Source Initiative, GNU Lesser General Public License.  
<http://www.opensource.org/licenses/lgpl-license.php> (20.7.2005).
- [OSI\_MIT, 2005] Open Source Initiative, The MIT License.  
<http://www.opensource.org/licenses/mit-license.php> (20.7.2005).
- [OSI\_MPL, 2005] Open Source Initiative, Mozilla Public License 1.1 (MPL 1.1).  
<http://www.opensource.org/licenses/mozilla1.1.php> (21.7.2005).



- [OSI\_PHP, 2005] Open Source Initiative, The PHP License, version 3.0. <http://www.opensource.org/licenses/php.php> (21.7.2005).
- [Oxford, 1989] Oxford University, *Oxford English Dictionary* (second edition, 1989). Oxford University Press, 2005.
- [Perens, 1999] Bruce Perens, The Open Source Definition. In: Chris DiBona, Sam Ockman and Mark Stone (eds.), *Open Sources: Voices from the Open Source Revolution*. O'Reilly, 1999. Also available as <http://www.oreilly.com/catalog/opensources/book/toc.html> (16.5.2006).
- [Perl, 2000] The Perl Foundation, Perl6's License Should be (GPL | Artistic-2.0). <http://dev.perl.org/perl6/rfc/346.html> (13.9.2005).
- [PHP, 2005] PHP, History of PHP and related projects. <http://fi.php.net/manual/en/history.php> (9.9.2005).
- [Raymond, 2000] Eric Raymond, The Cathedral and the Bazaar (version 3.0). <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/> (8.9.2005).
- [Rosen, 2002] Lawrence Rosen, Geek law: Why the public domain isn't a license. *Linux Journal* **2002**, 102 (Oct. 2002).
- [Rosen, 2004] Lawrence Rosen, *Open Source Licensing: Software Freedom and Intellectual Property Law*, Prentice Hall PTR, 2004. Also available as <http://www.rosenlaw.com/oslbook.htm> (13.9.2005).
- [SCO v. IBM, 2005] Groklaw, SCO Group v. International Business Machines. <http://www.groklaw.net/> (23.9.2005).
- [Seppä, 2006] Arto Seppä, Open source in Finnish software companies. The Research Institute of the Finnish Economy in its series Discussion Papers with number 1002, 2006. Also available as <http://ideas.repec.org/p/rif/dpaper/1002.html> (6.5.2006).
- [SourceForge, 2006a] SourceForge.net. <http://sourceforge.net/> (15.5.2006).
- [SourceForge, 2006b] SourceForge.net, What is SourceForge.net? <http://sourceforge.net/docs/about> (15.5.2006).
- [Stallman, 1999a] Richard Stallman, Why you shouldn't use the Library GPL for your next library. <http://www.gnu.org/philosophy/why-not-lgpl.html> (1.9.2005).
- [Stallman, 1999b] Richard Stallman, The GNU Operating System and the Free Software Movement. In: Chris DiBona, Sam Ockman and Mark Stone (eds.), *Open Sources: Voices from the Open Source Revolution*. O'Reilly, 1999. Also available as <http://www.oreilly.com/catalog/opensources/book/toc.html> (29.9.2005).

- [Stallman, 2002] Richard Stallman, *About the GNU project*.  
<http://www.gnu.org/gnu/thegnuproject.html> (21.9.2005).
- [Stallman, 2005] Richard M. Stallman, *Why "Free Software" is better than "Open Source"*.  
<http://www.fsf.org/licensing/essays/free-software-for-freedom.html> (1.8.2005).
- [Sutor, 2005] Bob Sutor, *Open Standards, Open Source and Communities – Backbones for Business and Innovation*. Presentation in Open Mind 2005 seminar in Tampere, 2005. Also available as  
<http://www.coss.fi/attachment/2863c6a7ce2660a29280308725fb9134/ca4dfe3db846f74df45cb17cbf146ec8/Sutor+-+OpenMind+-+Tampere.pdf>  
 (26.11.2005).
- [Sygate, 2004] Sygate Technologies, Inc., Sygate Personal Firewall,  
[http://smb.sygate.com/products/spf\\_standard.htm](http://smb.sygate.com/products/spf_standard.htm) (26.9.2005).
- [Vedova, 2005] Gianluca Della Vedova, *The Clarified Artistic License*.  
<http://www.statistica.unimib.it/utenti/dellavedova/software/artistic2.html>  
 (24.10.2005).
- [Välimäki, 2001] Mikko Välimäki, GNU yleinen lisenssi (GPL lisenssi).  
[http://www.turre.com/licenses/gpl\\_fi.php](http://www.turre.com/licenses/gpl_fi.php) (18.9.2005).
- [Välimäki, 2002] Mikko Välimäki, Avoimen lähdekoodin ohjelmistolisensseistä. *Defensor Legis* **82**, 5 (2002), 851-861. Saatavilla myös  
[http://www.hiit.fi/de/valimaki\\_dl\\_0502.pdf](http://www.hiit.fi/de/valimaki_dl_0502.pdf) (29.8.2005).
- [Välimäki, 2003] Mikko Välimäki, Dual licensing in open source software industry. *Systemes d'Information et Management* **8**, 1 (2003), 63-75. Also available as  
[http://www.valimaki.com/org/dual\\_licensing.pdf](http://www.valimaki.com/org/dual_licensing.pdf) (29.7.2005).
- [Välimäki, 2004] Mikko Välimäki, Open source licensing.  
[http://www.valimaki.org/org/open\\_source\\_tre\\_2004.pdf](http://www.valimaki.org/org/open_source_tre_2004.pdf) (29.8.2005).
- [Välimäki, 2005a] Mikko Välimäki, *The Rise of Open Source Licensing - A Challenge to the Use of Intellectual Property in the Software Industry*. Turre Publishing, Helsinki, 2005. Also available as  
[http://pub.turre.com/openbook\\_valimaki.pdf](http://pub.turre.com/openbook_valimaki.pdf) (20.4.2006).
- [Välimäki, 2005b] Mikko Välimäki, henkilökohtainen tiedonanto, lokakuu 2005.
- [Välimäki, 2005c] Mikko Välimäki, BSD-lisenssi.  
[http://www.turre.com/licenses/bsd\\_fi.php](http://www.turre.com/licenses/bsd_fi.php) (16.10.2005).
- [Välimäki, 2005d] Mikko Välimäki, henkilökohtainen tiedonanto, toukokuu 2005.
- [Wall, 1994] Larry Wall's message to gnu.misc.discuss 6.4.1994. Also available as  
<http://groups.google.fi/group/gnu.misc.discuss/msg/aa15573107070561?dmode=source> (21.4.2006).

- [Walli *et al.*, 2005] Stephen Walli, Dave Gynn and Bruno Von Rotz, The Growth of Open Source Software in Organizations. [http://www.optaros.com/publications\\_wpapers.shtml](http://www.optaros.com/publications_wpapers.shtml) (8.3.2006).
- [Wikipedia, 2005a] Wikipedia, the free encyclopedia, Freeware. <http://en.wikipedia.org/wiki/Freeware> (26.9.2005).
- [Wikipedia, 2005b] Wikipedia, the free encyclopedia, MIT License. [http://en.wikipedia.org/wiki/Mit\\_license](http://en.wikipedia.org/wiki/Mit_license) (20.7.2005).
- [Wikipedia, 2006] Wikopedia, Vapaa tietosanakirja, Immateriaalioikeus. <http://fi.wikipedia.org/wiki/Immateriaalioikeus> (21.4.2006).
- [Wilcox-O'Hearn, 2005] Bryce Wilcox-O'Hearn, Quick Reference For Choosing a Free Software License. [http://www.zooko.com/license\\_quick\\_ref.html](http://www.zooko.com/license_quick_ref.html) (1.12.2005).
- [Williams, 2002] Sam Williams, *Free as in Freedom: Richard Stallman's Crusade for Free Software*. O'Reilly, 2002. Also available as <http://www.oreilly.com/openbook/freedom/> (8.9.2005).
- [X11, 2005] X11 License, Terms and Conditions. [http://www.x.org/Downloads\\_terms.html](http://www.x.org/Downloads_terms.html) (20.7.2005).
- [Yeates, 2005] Stuart Yeates, Dual Licensing: A threat to Open Source Software? <http://www.oss-watch.ac.uk/resources/duallicence.xml> (30.7.2005).



Muuntelu	Apache 2.0	Artistic	BSD (uusi)	GPL 2.0	LGPL 2.1	MIT	MPL 1.1	PHP 3.01
Johdannaisten teosten nimeämiselle on rajoituksia. <sup>1</sup> Teoksen nimessä saa käyttää fraaseja, kuten "based on Apache", "powered by Apache" tai "based on Apache technology", mutta esim. "Apache Esimerkkiohjelma" on kielletty. <sup>2</sup> Joissakin levitystilanteissa edellytetään levitysversioiden nimeämistä alkuperäisistä poikkeaviksi. <sup>3</sup> Johdannaisille ei saa antaa nimeä "PHP", eikä sana "PHP" saa esiintyä niiden nimessä ilman kirjallista lupaa. Esim. "Foo for PHP" on sallittu, mutta "phpfoo" ei ole.	Kyllä <sup>1</sup>	Kyllä <sup>2</sup>	Ei	Ei	Ei	Ei	Ei	Kyllä <sup>3</sup>
Lisensointi, levitys ja myynti	Apache 2.0	Artistic	BSD (uusi)	GPL 2.0	LGPL 2.1	MIT	MPL 1.1	PHP 3.01
Teoksen sanatarakan kopion myynti on sallittua.	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
Johdannaisten tai uuden teoksen voi hinnoitella vapaasti. <sup>1</sup> Lisenssin mukaan johdannaisten teokset on lisensoitava kokonaisuudessaan ilman maksua kolmansille osapuolille. Käytännössä kuluja voi kuitenkin allokoida kopion fyysiseen toimittamiseen, johon voi sisältyä tallennusmedian ja lähetyksien lisäksi mm. kopion saattaminen lähetyksuntoon.	Kyllä	Kyllä	Kyllä	Ei <sup>1</sup>	Ei <sup>1</sup>	Kyllä	Kyllä	Kyllä
Teoksen verkkokäyttö ja sovellusvuokraus tulkitaan teoksen levittämiseksi.	Ei	Ei	Ei	Ei	Ei	Ei	Ei	Ei
Ohjelman mainosmateriaalin mainittava alkuperäinen tekijä.	Ei	Ei	Ei	Ei	Ei	Ei	Ei	Ei
Lähdekoodi on vapautettava levitettäessä teosta.* * Lähdekoodia ei ole aina pakko sisällyttää samalle jakelumedialelle binääriversion kanssa.	Ei	Ei	Ei	Kyllä	Kyllä	Ei	Kyllä	Ei
Binäärikoodi on vapautettava levitettäessä teosta.	Ei	Ei	Ei	Ei	Ei	Ei	Ei	Ei
Vastaanottajalle on aina annettava oikeus levittää johdannaista teosta eteenpäin.	Ei	Ei	Ei	Kyllä	Kyllä	Ei	Kyllä	Ei
Ohjelman tekijöiden ja kehittäjien nimien käyttö markkinoinnissa vaatii kirjallisen luvan.* * Laki voi estää nimien käytön ilman erillistä lisenssiehtoa. <sup>1</sup> Nimien käytön lisäksi johdannaisten mainostamisessa ei saa käyttää nimeä "PHP" ilman aikaisemmin saatua kirjallista lupaa.	Ei	Kyllä	Kyllä	Ei	Ei	Ei	Ei	Kyllä <sup>1</sup>
Vaatii mainoslauseen liittämisen teokseen. <sup>1</sup> Kaikkien levitysversioiden, levitysmuodosta riippumatta, on säilytettävä seuraava tunnus: "This product includes PHP software, freely available from <http://www.php.net/software/>".	Ei	Ei	Ei	Ei	Ei	Ei	Ei	Kyllä <sup>1</sup>
Johdannaisten tai yhdistetyn teoksen osa voidaan julkaista toisen lisenssin alaisena.	Kyllä	Kyllä	Kyllä	Ei	Kyllä	Kyllä	Kyllä	Kyllä
Sallii lisenssiehtojen muuttamisen, mikäli luotu teos on tulkittavissa uudeksi teokseksi. <sup>1</sup> Lisenssin epämääräisyyden takia tämä ei ole selvää, mutta lisenssin on tulkittu käyttäytyvän muiden sallivien lisenssien kaltaisesti.	Kyllä	Kyllä <sup>1</sup>	Kyllä	Ei	Ei	Kyllä	Ei	Kyllä

# Liite 2. The Open Source Definition

Version 1.9

## Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

### 1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

### 2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

### 3. Derived Works

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

### 4. Integrity of The Author's Source Code

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The licen-

se may require derived works to carry a different name or version number from the original software.

### **5. No Discrimination Against Persons or Groups**

The license must not discriminate against any person or group of persons.

### **6. No Discrimination Against Fields of Endeavor**

The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

### **7. Distribution of License**

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

### **8. License Must Not Be Specific to a Product**

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

### **9. License Must Not Restrict Other Software**

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

### **10. License Must Be Technology-Neutral**

No provision of the license may be predicated on any individual technology or style of interface.