

TESTIT & TEKNIikka

Vertailussa avoimen
lähdekoodin
lisenssit

- Apache 2.0
- European Union Public Licence 1.2
- GNU AGPLv3
- GNU GPL 3.0
- Microsoft Public License (MS-PL) 2.0
- MIT

VAPPAUS, VELJEYS JA TEKIJÄNOIKEUS

Avoimen lähdekoodin lisenssillä myönnetään ehdollinen lupa ohjelman käyttöön. Se on yksipuolinen sopimus, joka tarjoaa oikeuksia ja velvollisuuksia.

Jyrki Oraskari

Vapaa ohjelmisto ja avoin koodi. Ensikuulemalta voi tuntua, että kyse on yhdestä ja samasta. Kyse on kuitenkin painotuseroista ja yksityiskohdista.

Ensimmäisessä painotetaan vapaiden ohjelmistojen aktivistin **Richard Stallmanin** hengessä vapautta käyttää koodia, tutkia miten se on luotu, muokata ja jakaa versioita ohjelmasta. Vapaan ohjelmiston rinnalla avoin koodi on merkitykseltään lähes identtinen, mutta paino on sillä, miten ohjelmia kehitetään ja lähestymistapa on yritysmuotoisempi.

Näkemyserot näkyvät lisensseissä. Vapautta säilyttävien lisenssien tavoitteena on ohjata vastavuoroisuuteen. Toivotaan, etteivät ulkopuoliset yrittäjät muokkaa ja sulje koodia kaupalliseksi tuotteeksi. Paljon ollut rajanvetoa sille, mihin asetetaan lisenssin kattaman koodin rajat. Viime aikoina polemiikka on ollut siitä, saako vapaata koodia tarjota pilvipalvelun osana.

Toisena äärimmäisyytenä ovat yliopistoista lähtöisin olevat lisenssit. Niissä paino on tiedon vapaudella. Sallivat lisenssit pyrkivät asetettujen ehtojen minimointiin. Esimerkkejä näistä ovat BSD, MIT ja Apache 2.0.

AVOIMET LISENSSIT

Lisenssi	Apache 2.0	EUPL 1.2	GNU GPL 3.0	GNU AGPL 3.0	Microsoft MS-PL	MIT
Laatija	Apache Software Foundation	Euroopan komissio	Free Software Foundation	Free Software Foundation	Microsoft	Massachusetts Institute of Technology
Julkaisuvuosi	2004	2017	2007	2007	2001	1988

KÄYTTÖ

Jakelu	Salliva	CopyLeft	CopyLeft	CopyLeft	Salliva	Salliva
Muokkaus	Salliva	CopyLeft	CopyLeft	CopyLeft	Salliva	Salliva
Yksityiskäyttö	Kyllä	Kyllä	Kyllä	Kyllä	Salliva	Kyllä
Kaupallinen käyttö	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
Verkkokäytössä julkaistava lähdekoodi	Ei	Kyllä	Ei	Kyllä	Ei	Ei
Muutokset koodiin dokumentoitava	Kyllä	Kyllä	Kyllä	Kyllä	Ei	Ei

HYVÄKSYNTÄ

Free Software Foundation hyväksyntä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
OSI hyväksyntä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä
Debian hyväksyntä	Kyllä	Kyllä	Kyllä	Kyllä	Ei	Kyllä
Fedora hyväksyntä	Kyllä	-	Kyllä	Kyllä	Kyllä	Kyllä

YHTEENSOPIVUUS

GPLv3 yhteensopivuus	Kyllä	Kyllä	Kyllä	Kyllä	Ei	Kyllä
----------------------	-------	-------	-------	-------	----	-------

MUUT

Lisenssin pysyvyys	Ei	Samanlainen	Kyllä	Kyllä	Ei	Ei
Lähdekoodin julkaisu	Ei pakollinen	Aina	Aina	Aina	Ei pakollinen	Ei pakollinen
Patenttisuojat	Kyllä	Kyllä	Kyllä	Kyllä	Kyllä	Ei sisälly
Vastuunrajoite	Kyllä	Kyllä	Kyllä	Kyllä	Ei	Kyllä

Miksi sitten tarvitaan lisenssejä? Ilman lisenssiä ei ole lupaa. Juuret ovat vuoden 1973 Yhdysvaltain uuden teknologian tekijänoikeuslautakunnan lausunnossa, jossa ensi kertaa myönnettiin, että myös ohjelmakoodi voi olla tekijänoikeuden alainen. Vuodesta 1991 alkaen Suomen laissa on ollut yhtenevä maininta. Lisenssillä tekijä myöntää mahdollisia oikeuksia koodiinsa. Esimerkiksi ohjelmakoodin asettaminen näyttille ilman lisenssiä ei merkitse, että se olisi vapaata riistaa.

AVOIMEN LÄHDEKOODIN lisenssejä on arviolta kolmatta sataa, joista Open Source Initiative (OSI) on vahvistanut reilun kahdeksankymmenen noudattavan avoimen lähdekoodin periaatteita. Oikean valintaan on useita oppaita.

Avoimen lähdekoodin alustan GitHubin nyrkkisääntö on hyvin suoraviivainen. Jos projekti liittyy johonkin koodiyhteisöön, kannattaa käyttää samaa lisenssiä kuin muutkin kyseisessä yhteisössä. Silloin välttää yllätykset. Jos haluaa mahdollisimman yksinkertaisen ja sallivan lisenssin, kannattaa valita MIT. Se sallii jotakuinkin kaiken, mitä ohjelman käyttäjät voivat toivoa. Toisaalta jos henkilölle on tärkeää, että ohjelmaan tehtävät parannukset tulevat yleiseen jakoon, GNU GPLv3 voi olla hyvä valinta.

Lisenssin valintaan voi vaikuttaa myös se, miten kehitettyä koodia toivotaan käytettävän ja mihin käyttäjäkuntaan halutaan vedota. Jos toivoo, että kehitettyä ohjelmakirjastoa käytetään laajasti, kannattaa valita MIT. Jos taas toivoo paljon yrityskäyttäjää, Apache 2.0 on hyvä valinta.

GNU AGPLv3 on paras, jos ei halua ohjelman päätyvän osaksi kaupallista palvelua. Aukoton se ei kuitenkaan ole. Esimerkiksi MongoDB-tietokannan kehittäjiltä meni usko sen suojaan, kun aasialaiset saas-palvelun tarjoajat koettelivat GNU AGPLv3 -lisenssin rajoja.

Toisaalta jos tavoitteena on vain oman koodin kaupallistaminen siten, että avoin lähdekoodi tuo sille käyttäjäkuntaa, Linux-käyttäjärjestelmän Red Hat -jakeluversion tapainen kaksoislisensointi voi olla tehokasta. Silloin kehitetyn koodin kaupallinen käyttö on maksullista. GNU-lisenssillä voi estää, ettei kilpaileva yritys kopioi ohjelmaa ja paketoit sitä osaksi ohjelmatuotettaan.

Lisenssin valinnalla on myös kauaskantoisia seurauksia sille, minkä kirjastojen varaan ohjelma voidaan luoda. Sallivan lisenssin käyttö estää käyttämästä kirjastoja, joiden lisenssit vaativat koodin säilyttämistä vapaana. Vastaavasti kun käyttää ohjelmakirjastoja, kannattaa suosia Apache 2.0 -lisenssiä käyttäviä kirjastoja. Lisenssi suojaa käyttäjää patenttikanteilta.

Oraclen lisenssivalinta on kiinnostava. Sen JavaFX-käyttöliittymäkirjasto on suojattu GNU GPLv2+CE -lisenssillä. Kirjasto on vahvasti suojattu GNU-lisenssillä, mutta lisälause antaa erityislupaa linkittää ohjelmaan.

VALITSIMME LISENSSIT tähän vertailuun käyttäen neljää kriteeriä. Ensimmäisenä oli lisenssien yleisyys, jonka ensisijaisena lähteenä käytimme White-Sourcen ja GitHubin julkaisemia tilastoja.

HELSINKI SUOSII MIT-LISENSSIÄ

Kahdenkymmen Yhdysvaltain kaupungin yhteisen Code4America -hankkeen perusajatus oli tuoda lahjakkaita kehittäjiä julkisen sektorin avuksi tavoitteena luoda konkreettisia ja inspiroivia esimerkkejä myös ohjelman ulkopuolisille kaupungeille.

Helsingin kaupunki on ollut mukana siitä versionneesta eurooppalaisessa rinnakkaisohjelmassa vuodesta 2012 lähtien. Kaupungin tietohallintojohtaja **Markku Raitio** pitää sitä lähtölaukauksena kaupungin avoimen koodin laaja-alaiselle hyödyntämiselle.

Helsingin kaupunginhallitus hyväksyi 2015 tietotekniikkaohjelman, jonka mukaan ohjelmakoodi, joka kehitetään kaupungin toimeksiannosta, tulisi julkaista avoimena lähdekoodina, ellei ole erityistä syytä tehdä toisin. Helsingin kaupunginvaltuuston puheenjohtaja **Otso Kivekäs** oli yksi sen nimekkäimpiä puolestapuhujia.

”Ei enää toimittajaloukkua, vaan sen sijaan järjestelmien laatu parantunee ja hinta laskee pikkuhiljaa”, hän hehkutti tuolloin.

Lähtökohtana oli Euroopan unionin avoimen lähdekoodin lisenssi EUPL. Periaate oli, että kun käytetään julkisia varoja, on luontevaa julkaista koodi yhteiseksi hyväksi.

Helsingin kaupunki käyttää nykyisissä avoimissa ohjelmaprojekteissa usein MIT-lisenssiä ja joskus voidaan valita BSD-lisenssi. Päätymisen näihin oli lopputreilla luontevaa. GNU GPLv2:n vahva tarttuvuus olisi EUPL:n kanssa tuonut lisenssiiviidakon ja vaatimus vastavuoroisuudesta turhia esteitä koodin jatkokäytölle.

SEKÄ MIT ETTÄ BSD ovat hyvin sallivia lisenssejä. Helsinki ei halua rajata ohjelmakoodin jatkokäyttöä, vaan se haluaa tehdä yleiskäyttöisiä komponentteja, jotka hyödyttäisivät koko kaupunkia.

Kaupunginlaajuisesti järjestelmähankkeita on paljon. Osaa tehdään perinteisellä mallilla toimittajien kanssa tai voidaan ostaa jonkun järjestelmätoimittajan tuote. Pyrkimyksenä on kuitenkin mahdollisimman suuri avoimen lähdekoodin osuus.

”Avoimen lähdekoodin käyttö helpottaa kaupunkien välistä yhteistyötä”,



MUILLEKIN HYÖTYÄ. Helsingin kaupungin Annukka Varteve ja Juha Suvanto arvioivat, että avoimen lähdekoodin hyödyntäminen helpottaa kaupunkien välistä yhteistyötä.

kommentoi Helsingin kaupungin ict-kehityspäällikkö **Annukka Varteve**.

Esimerkiksi Helsingin kaupungin kehittämää Varaamo-palvelua on kehitetty kaupunkien kesken erityisesti 6Aika-ohjelman yhteydessä. Palveluun kirjautuneet käyttäjät voivat varata kaupungin tiloja tai tarvikkeita, ja ohjelmakoodista on oma versionsa useilla kaupungeilla. Muitakin versioita on.

Ohjelmasta on yhteensä 18 erillistä kopiota, joita kehitetään erillisinä projekteina. Salliva lisensointi mahdollistaa tämän.

Tällä hetkellä Helsingillä on seitsemän muun kaupungin ja Kuntaliiton kanssa meneillään yhteistyöhanke, jossa kehitetään yhteistyömallia avoimella koodilla tehtävän kehityksen koordinoitiin.

Koordinoinnilla pyritään siihen, että eri tahojen tekemät parannukset saadaan yleiseen jakoon ja että koodi ei haaraudu liikaa. Turku on yksi muokkaajista.

”Teemme lain vaatimat saavutettavuusmuutokset koodiin ja ne tulevat kaikkien käyttöön”, tiivistää Turun kaupungin erityisasiantuntija **Sanna Moisala**.

WhiteSourcen tilastoimat vuoden 2018 kolme yleisintä lisenssiä olivat järjestyksessä: MIT (26 %), Apache (22 %) ja GPL 3.0 (16 %). GitHubin lista on vuodelta 2015. Sen järjestys oli MIT (45 %), GPLv2 13 %, Apache (11 %) ja GPLv3 (9 %). Yleisten lisenssien suosimiseen on kaksi hyvää syytä. Ensiksi itsestäänselvyys: On todennäköistä, että käyttäjät jossain vaiheessa ohjelmaa, joka on lisensoitu jollain näistä. Toiseksi, yleisen lisenssin käyttö auttaa käyttäjiäsi ymmärtämään lisenssiehdot ilman lakimiesten apua.

Kattavuus oli yksi kriteereistä. Mukana oli hyvä olla sekä avoimuuteen pakottavia että laajasti sallivia lisenssejä. Oli myös kiinnostava tuoda mukaan eurooppalainen EUPL.

Kaksi vertailuun valituista lisensseistä heijastelee vuoden 2019 trendejä: Microsoft Public License (MS-PL) ja GNU AGPLv3. MongoDB:n uusi Server Side Public -lisenksi SSPL rajattiin pois, koska OSI pitää sitä kaupallisena lisenssinä.

YLEISEN LISENSSIN KÄYTTÖ HELPOTTA LISENSSI- EHTOJEN YMMÄRTÄ- MISTÄ.

TESTAAJAN KOMMENTTI



AVOIMUUS JATKAA KOODIN ELÄMÄÄ

Peilaan avoimen lähdekoodin kehitystä omiin kokemuksiini. Rakennusmallien uusilla tietomuo-doilla pyritään tiedon parempaan saatavuuteen. Olin itse mukana IFCToLBD -konvertterin kehityksessä vuonna 2017. Konvertteri sai alkunsa joulun alla Ghentissä vietetystä paripäiväisestä tutkijataapaamisesta.

Pienessä neuvotteluhuoneessa ideoimme ja testasimme tuotoksia. Me neljä. Minä sormet punahekkuisina työstin ajatuksen tynkiä koodiksi. Jo toisen päivän iltana saattoi todeta, että intuitio oli ollut oikea. Rakennusmallista saattoi tulostaa koneen ymmärtämän tiivistelmän rakennuksen elementeistä ja niiden suhteista.

Alusta alkaen oli selvää, että tuon hackathonin tulokset jaetaan tiedeyhteisölle rajoituksetta. MIT-lisenksi on sallivuutensa vuoksi ollut oletus, mutta nyt sitä vaihtoehtoa ei ollut. Koska käytin Ghentin yliopiston koodia pohjana, vaihtoehtoksi rajautui GNU Affero General Public License v3.0.

Se ei ehkä ole onnistunein valinta houkuttelemaan yrityksiä. Lisenssi ei kuitenkaan rajoittanut kenenkään omaa käyttöä edes kaupallisesti. Eräs suunnittelutoimisto halusi prosessoida kokonaisen sairaalan rakennusmallia nätiimpään muotoon. Se mahtui juuri ja juuri muistiin.

Parin vuoden aikana yksi jos toinen on osallistunut ja jatkanut koodia. Tekijänoikeuksien lisäksi kiitos kuuluu kaikille.

Jyrki Oraskari

ANSAINNAMALLI RATKAISEE MENESTYKSEN

Ohjelmistotalo Codenton pääosakkaan **Petri Aukian** mukaan suljetut lisenssit ovat pienille yrittäjille helpoin ansaintamalli.

"Avointa lähdekoodia käyttäessä on muistettava, että vain 2–3 prosenttia käyttäjistä on valmiita maksamaan avoimesta koodista", Aukia sanoo.

Yrittäjällä pitää olla selvä suunnitelma sille, millä tehdään rahaa. Hyvää myyjää tarvitaan löytämään oikeat asiakaskunnat ja mallit, joista he ovat valmiita maksamaan.

Yllätyksiä voi tulla. Esimerkiksi räätälöintiin perustuva liiketoimintamalli saattaa toimia odottamattomalla tavalla. Monesti asiakas on ensin yrittänyt itse ja kehittäjältä pyydetään ratkaisua vasta, kun päätös on isketty seinään hyvän tovin.

"Kun lopulta käännetään kehittäjän puoleen, työn painopiste voi olla jotain muuta, mikä alkujaan on inspiroinut koodaria", Aukia varoittaa.

Usein pyyntönä on sovittaa ohjelma toimimaan uudessa ympäristössä. Esimerkiksi tiedostojen muuntamisessa käytetylle Ghostscript-ohjelmalle voitaisiin pyytää sen kehittämistä toimimaan uuden kirjoittimen kanssa.

VIIME AIKONA on havahduttu siihen, että rajoittavan GPL-lisenssin alla kirjoitettuja ohjelmia tarjotaan osana pilvipalveluina. Tästä toimivat esimerkkinä Amazonin tarjotamat palvelut, jotka voivat loppukäyttäjälle näyttäytyä houkuttelevamalta kuin niitä alkujaan kehittäneen pikkuriikkisen firman



HAASTE. Petri Aukia muistuttaa, että vain kahdesta kolmeen prosenttiin käyttäjistä on valmiita maksamaan avoimesta koodista.

tarjonta. Ei siis ihme, että yhä useammin lisenssiehdoissa tilkitään mahdollisuus tarjota avoimen lähdekoodin tuotosta verkon yli.

Haasteita on monenlaisia. Kun käytetään sallivaa MIT-lisenssiä, asiakkaan lisäksi myös kilpailijat näkevät ohjelmatoimituksen yksityiskohdat. Mikään ei estä heitä parantelemasta ohjelmaa ja tarjota samasta koodista viilattua versiota. Kilpailu on siis kovaa.

"Julkishallinnolle kilpailusta on kuitenkin etua", Aukia painottaa. Voidaan ostaa paras, mutta avoimuus tarjoaa myös sen edun, että toimittajien tarjonta on yhteensopivaa käytössä olevan koodin kanssa.

Kaksoislisensointi on ollut yksi toimivista konsepteista. Tästä Linux-jakelija Red Hat on hyvä esimerkki, mutta sen jäljissä myös Ubuntu on onnistunut toistamaan menestystarinaa. Yrityskäyttäjät ovat olleet valmiit maksamaan siitä, että laitteisto toimii katta-



APACHE 2.0

- + patenttisuoja
- + selkeä kieli
- + kaupallinen käyttö sallitaan
- + muutokset koodiin pitää dokumentoida
- ei yhteensopiva GNU GPL 2.0:n kanssa

Apache 2.0 on yksi sallivimpia avoimen lähdekoodin lisenssejä. Se vaatii käyttäjältä lähinnä, että tekijänoikeus- ja lisenssimerkinnät säilytetään. Lisenssi sallii ohjelmiston käytön, muokkauksen ja muunnetun koodin levittämisen kehittäjän valitsemien ehdoin myös ilman lähdekoodia. Oikeudet ovat pysyvät ja maailmanlaajuiset, ja niitä ei voi vetää takaisin. Eli jos lisenssin myöntäjä asuu Intiassa, lisenssi on yhtä lailla voimassa Suomessa.

Muutettua koodia voi levittää käyttäen toista lisenssiä, mutta alkuperäinen lisenssi pitää säilyttää ja muutokset kuhunkin muutettuun tiedostoon dokumentoida. Dokumentointi takaa läpinäkyvyyden sille, kenen kirjoittamaa koodia on. Muuttamattoman koodin lisenssiä ei saa vaihtaa.

Apache 2.0 on patenttien osalta sikäli erityinen, että se myöntää maailmanlaajuiset oikeudet kehittäjien relevantteihin patentteihin. Apache 2.0 on yhteensopiva GNU GPL 3.0 -lisenssin kanssa, mutta ei version 2.0 kanssa. Tällä on merkitystä siinä tapauksessa, että Apache 2.0 -lisenssin koodia halutaan siirtää osaksi ohjelmaa, jolla on GNU GPL 2.0 -lisenssi.

Lisenssi on suosittu. Whitesourcen tilastoissa 22 prosentilla projekteista oli Apachen lisenssi. Google käyttää lisenssiä yleisesti. Myös Swift, Kubernetes ja PDF.js ovat tunnettuja lisenssiä hyödyntäviä projekteja.



EUROPEAN UNION PUBLIC LICENCE 1.2

- + EU:n lainsäädäntö huomioitu
- + yhteensopiva GNU GPL 2.0:n ja 3.0:n sekä AGPL 3.0:n kanssa
- + patenttisuoja
- + 23 kielellä
- vähän käytetty

Eurooppalainen EUPL 1.2 on Euroopan komission luoma ja hyväksymä avoimen koodin ohjelmalisenssi. Lisenssin versio 1.2 on vuodelta 2017. Se on saatavilla Euroopan unionin 23 kielellä, myös suomeksi. Erityistä on, että jokainen käännös on yhtä pätevä juridisesti. Eli suomenkielistä lisenssiä voi lukea ja sen pilkut, pisteet ja sanamuodot ovat se tulkinta, jonka mukaan toimitaan.

Lisenssin valmistelussa on huomioitu Euroopan unionin jäsenmaiden olemassa oleva lainsäädäntö. Käytännössä erot GNU-lisenssiin ovat pieniä. Heikkoutena on kuitenkin pidetty sitä, että yhdistettäessä koodia, jolla on EUPL lisenssi, GNU lisenssin ohjelmakoodin kanssa, EUPL:n on katsottu olevan väistytävä lisenssi. Joustavuudesta on hyötyä, mutta on aiheesta kysytty, miksi ei suoraan siirrytä GNU-lisenssiin, joka on yhtä lainvoimainen.

EUPL:n heikkous on sen harvinaisuus. Vain harva ohjelma käyttää sitä ja useimmat ohjelmakirjastot käyttävät jotain muuta yleisempää lisenssiä.



GNU GPL 3.0

- + Apache 2.0 on yhteensopiva GPLv3:n kanssa
- + patenttisuoja
- + muutokset koodiin pitää dokumentoida
- lähdekoodi täytyy liittää mukaan

GNU GPL 3.0 oli 2018 WhiteSource-listan mukaan suosituin Copyleft- eli käyttäjän-oikeuslisensseistä. Jos ohjelmaa muokataan, lisenssi tulee säilyttää siten, että myös tuleville käyttäjille säilyvät samat oikeudet teokset käyttöön, kopiointiin ja muokkaukseen. Itse tehtyjä muokkauksia ei tarvitse antaa yleiseen jakoon, ellei julkista ohjelmaa. Levitettäessä koodia, myös siihen tehtyjen muutosten pitää olla julkisia.

Periaatteena on tarjota käyttäjälle vapaus käyttää ohjelmaa mihin tahansa tarkoitukseen. Sitä voi muokata vapaasti, jotta se soveltuu käyttäjän tarpeisiin. Ohjelmaa saa myös levittää vapaasti kelle tahansa ja myös itse tekemiä muutoksia saa levittää. Ohjelmia voi käyttää kaupallisissa yhteyksissä, mutta sen lähdekoodin levitystä ei saa estää. Periaatteessa ohjelmasta saa myös pyytää rahaa, mutta sen jakamista ilmaiseksi ei saa kieltää.

Yleinen liiketoimintamalli on tarjota kaksoislisenssiä. Siinä asiakasyritys haluaa ohjelmakoodin osaksi ohjelmaansa ja on valmis maksamaan siitä, että voi sulkea ohjelmakoodin.

TUNNETUIMMAT AVOIMEN KOODIN PROJEKTIT

Avoimen lähdekoodin projekteja on joka lähtöön. Linux on yksi tunnetuimpia. Sen lisenssinä on GNU GPLv2, mikä selittyy pääosin sillä, että merkittävä osa Linux-versioiden ohjelmakoodista on lähtöisin vapaiden ohjelmistojen aktivistin **Richard**

Stallman 1980-luvulla käynnistämästä GNU-käyttöjärjestelmäprojektista.

Muita yleisesti tunnettuja avoimen lähdekoodin projekteja ovat Firefox, Wordpress, LibreOffice, GNU Gimp ja Videolan VLC media player. Näistä Gimp-ohjelman

lisenssi on GPLv3+. VLC on lisensoitu GNU General Public License Version 2:llä.

Uusissa projekteissa huomionarvoista on, että useimmat ovat alkujaan jonkun ison yrityksen kehittämiä. Näitä ovat esimerkiksi Uberin kehittämä Ludwig, joilla



GNU AGPLV3

- + patenttisuoja
- + muutokset koodiin pitää dokumentoida
- + verkkokäyttö katsotaan levitykseksi
- hyvin rajoittava

GNU AGPL 3.0 hyvin samanlainen muiden GNU GPL -lisenssien kanssa, mutta se on yksi rajoittavimpia lisenssejä. Se velvoittaa toimittamaan lähdekoodin, jos ohjelmaa tarjoaa saas-palveluna tai verkon yli osana sovellusvuokrausta. Siten se on ajan hermolla ohjelmakehityksen siirtyessä saas-palveluiden päälle.

Aukoton lisenssi ei ole. Siksi esimerkiksi MongoDB-tietokannan kehittäjät halusivat luoda oman Server Side Public -lisenssin, jossa vaaditaan, että tarjottaessa ohjelmaa osana kaupallista palvelua, koko palvelun lähdekoodi pitää avata.

Ero GNU AGPL-lisenssiin vaikuttaa hiuksenhienolta, mutta jälkimmäinen on avoimen koodin periaatteiden vastainen. Käyttötarkoitusta ei saa rajata. Jos joku haluaa esimerkiksi luoda varmennuspalvelun, jossa tietokannan tiedot varmuuskopioidaan tai palvelun, joka skaalaa tietokantapalvelua tarpeen mukaan, ei tulisi vaatia, että näiden koodi on avointa. GNU AGPL 3.0 on tältä osin joustavampi.



MICROSOFT PUBLIC LICENSE (MS-PL) 2.0

- + lähdekoodia ei tarvitse liittää mukaan
- + patenttisuoja
- ei yhteensopiva GNU GPL:n kanssa

Microsoftin MS-PL -lisenssi käyttäjänoikeus-lisenssi. GNU-lisenssien tapaan se säilyttää käyttäjän oikeudet lähdekoodissa. Eli jos ohjelmakoodia muokataan, lisenssi tulee säilyttää siten, että myös tuleville käyttäjille säilyvät samat oikeudet teokset käyttöön, kopiointiin ja muokkaukseen. Tässä tapauksessa tulee säilyttää MS-PL-lisenssi, mikä käytännössä estää GNU -pohjaisen ohjelmakoodin käytön ohjelman osana. Siltä osin se rajoittaa ohjelmoijan mahdollisuuksia käyttää haluamiaan kirjastoja.

Microsoftin MS-PL -lisenssi erityistä on, että ohjelmakoodi voidaan toimittaa myös käännettynä binäärimuotoon ilman, että vaaditaan että mukana tulisi toimittaa myös ohjelman lähdekoodi. Tältä osin se poikkeaa Microsoft Reciprocal License (MS-RL)-lisenssistä, joka vaatii, että muokatun ohjelma lähdekoodi toimitetaan aina ohjelman mukana.

Kun ohjelma toimitetaan binäärisenä, se voitaisiin WhiteSourceen tulkin mukaan haluttaessa toimittaa eri ehdoin kuin lähdekoodin. Periaatteessa voidaan siis myös rajoittaa käyttäjän oikeuksia. Ainoa ehto on, että ohjelma koostuu osista, jotka kääntyvät Microsoftin MS-PL -lisenssin koodin kanssa.



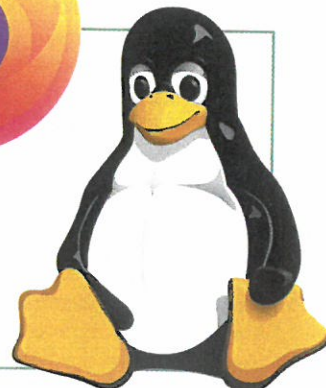
MIT

- + selkeä ja lyhyt lisenssi
- + yksi sallivimmista lisensseistä
- standardimuodon rinnalla variointeja
- ei tarvitse dokumentoida muutoksia koodiin
- ei suoraa patenttisuoja
- tarjoaa vähän suojaa

MIT-lisenssi on alkujaan kehitetty Massachusetts Institute of Technologyssä. Itse asiassa se on lisenssiperhe, mikä on osin sen heikkous. Suositeltavaa on käyttää OSI:n standardoitua muotoa. Variointien vuoksi aina kannattaa liittää mukaan koko lisenssiteksti. Hyvää on, että lisenssi on selkeä ja sen joka lauseesta löytyy kattavat selitykset verkossa.

Se on viime vuosina ollut yksi suosituimmista avoimen lähdekoodin lisensseistä. Esimerkiksi se on WhiteSourceen tilastoissa ykkösenä ja avoimen lähdekoodin alustan GitHubin listaamana se saavutti 45 prosentin osuuden vuonna 2015. Kehittäjälle se tarjoaa varsin vähän suojaa. Käytännössä se tarkoittaa, että kuka tahansa voi tehdä mitä tahansa ohjelmakoodille, kunhan se viittaavat alkuperäiseen tekijään lähdekoodissa.

Tässä heikkoutena on se, ettei muutoksia koodiin tarvitse dokumentoida. Angular.js, jQuery, Rails, Bootstrap ja Facebook React.js ovat suosittuja projekteja, jotka käyttävät MIT-lisenssiä. MIT on luotu ennen ohjelmistopatentteja. Siksi se ei ota kantaa niihin.



voi koodaamatta testaila syväoppivia neuroverkoja.

Microsoftin avoimen lähdekoodin ohjelmaeditori Visual Studio Code on viime aikoina kerännyt suosiota. TensorFlow puolestaan on Googlen vuonna 2015 avaama

avoimen koodin syväoppimisalusta.

Kaikkien näiden lähdekoodi löytyy GitHub-koodinjakopalvelusta. Uusille ohjelmille tyypillisesti Ludwig ja TensorFlow ovat lisensoitu Apache 2.0-lisenssillä. Visual Studio on MIT -lisensin alla.