

## Avoimen lähdekoodin hankintaopas

**Versio:**

**Julkaistu:** 7.10.2008

**Voimassaoloaika:** toistaiseksi

---

### Sisällys

1	Johdanto	2
2	Soveltamisala	2
3	Termit ja määritelmät	2
4	Johdanto avoimeen lähdekoodiin	5
4.1	Avoimen lähdekoodin evoluutio	5
4.2	Avoimen lähdekoodin ohjelman määritelmä	5
4.3	Tietokoneohjelman lähdekoodi	6
4.4	Avoimen lähdekoodin ohjelma on myös kaupallinen ohjelma	7
4.5	Avoimet rajapinnat ja standardit	7
4.6	Yhteenveto avoimen lähdekoodin hyödyistä ja haasteista	8
4.7	Avoimen ja suljetun lähdekoodin vertailutaulukko	12
4.8	Avoimen lähdekoodin liiketoiminta	12
4.9	EU:n linjaukset suhteessa avoimiin ohjelmistoihin ja standardeihin	13
4.10	Avoin lähdekoodi muissa suosituksissa	14
5	Avoin lähdekoodi ja hankintaprosessi	15
5.1	Toimintamalleja esiselvitysvaiheeseen	15
5.2	Seitsemän askelta avoimen lähdekoodin ohjelmiston mahdollisuuden arviointiin	15
5.3	Avoimen ohjelmiston ja suljetun ohjelmiston hankinnan erot	16
5.4	Esiselvitysvaiheessa avoimiin ohjelmistoihin liittyvät erityispiirteet	16
5.5	Avoimiin ohjelmistoihin liittyvät erityiskysymykset tarjouspyyntöä laadittaessa ja arvioitaessa	19
5.6	Kelpoisuuskriteereihin kuuluvat laatutekijät	21
5.7	Tarjousten arviointiin kuuluvat laatutekijät	22
5.8	Ohjelmiston hankkiminen avoimen lähdekoodin lisenssillä	23
6	Opas juridisiin aiheisiin	23
6.1	Tekijänoikeus ja avoimen lähdekoodin ohjelmistolisenssit	23
6.2	Avoimen lähdekoodin riskit ja niiden hallinta	26
6.3	JIT 2007 ja avoin lähdekoodi	33
7	Suosituksset	35
8	Opastavat tiedot	35
8.1	Suosituksen ylläpito	35
8.2	Liitteet	35
9	Liite 1: Linkejä tietovarantoihin ja ohjelmistovarastoihin	36

## 1 Johdanto

Tämän suosituksen tarkoitus on opastaa julkisen hallinnon organisaatioita avoimen lähdekoodin ohjelmistojen hankinnassa ja käyttöönotossa. Suositus keskittyy julkisen hallinnon toimijoihin ohjelmistojen hankkijana ja käyttäjänä, eikä se ota kantaa esimerkiksi avoimeen lähdekoodiin liittyviin elinkeino- tai innovaatiopoliittisiin kysymyksiin. Valitusta näkökulmasta tarkasteltuna oleellisia avoimeen lähdekoodiin liittyviä kysymyksiä ovat mm. seuraavat:

- Voidaanko avoimen lähdekoodin ohjelmistoilla alentaa ohjelmistohankintojen kokonaiskustannuksia?
- Kuinka avoimen lähdekoodin ohjelmistoilla voidaan parantaa tietojärjestelmien yhteentoimivuutta ja joustavuutta?
- Voidaanko avoimen lähdekoodin avulla vähentää tietojärjestelmien hankintaan liittyviä riskejä?
- Kuinka julkisen sektorin hankintoja koskevaa lainsäädäntöä ja ohjeistusta tulee soveltaa avoimen lähdekoodin hankintaan?
- Millaisilla hankintakäytännöillä turvataan erilaisten lisensointimallien ja niihin perustuvien liiketoimintamallien tasapuolinen kohtelu ja vapaa kilpailu?

Opas on jaettu kolmeen päälukuun:

- **4. Johdanto avoimeen lähdekoodiin** tarjoaa perustiedot avoimesta lähdekoodista ja kertoo EU:n linjauksista avoimen lähdekoodin käyttöön.
- **5. Avoin lähdekoodi ja hankintaprosessi** käsittelee erityispiirteitä avoimen lähdekoodin ohjelmiston hankinnassa.
- **6. Opas juridisiin aiheisiin** sisältää tietoa avoimen lähdekoodin lisensseistä, riskeistä ja niiden hallinnasta. Luvussa tarkastellaan myös *Julkisen hallinnon IT-hankintojen yleisiä sopimusehtoja* (JHS 166, JIT 2007).

## 2 Soveltamisala

Tämä suositus koskee julkisen hallinnon tietohallintoa ja erityisesti tietojärjestelmien ja ohjelmistojen hankintaa, mutta sitä voidaan hyödyntää myös yksityissektorilla.

## 3 Termit ja määritelmät

### Avoimen lähdekoodin lisenssi

Avoimen lähdekoodin lisenssit täyttävät Open Source Initiativen (OSI) avoimen lähdekoodin määritelmän ehdot (ks. avoin lähdekoodi). Ne antavat lisenssinsaaajalle oikeuden käyttää, kopioida, levittää ja muokata ohjelmaa vapaasti. OSI ylläpitää listaa hyväksymistään avoimen lähdekoodin lisensseistä.

<http://www.opensource.org/licenses>

### Avoimen lähdekoodin ohjelma

Ks. avoin ohjelma.

### Avoin lähdekoodi

Avoin lähdekoodi (engl. open source) -termin vuonna 1998 lanseerannut Open Source Initiative on julkaissut kymmenkohtaisen avoimen lähdekoodin määritelmän, joka voidaan tiivistää viiteen oikeuteen, jotka avoimen lähdekoodin ohjelman käyttäjä automaattisesti saa:

1. käyttää ohjelmaa mihin tarkoitukseen tahansa;
2. kopioida ja levittää ohjelmaa;
3. luoda ohjelman muunnelmia ja levittää niitä;
4. saada ohjelman lähdekoodi ja hyödyntää sitä;
5. yhdistää ja levittää ohjelmaa toisten ohjelmien kanssa.

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

Avoim lähdekoodi on käytännössä vaihtokelpoinen termin *vapaa ohjelma* kanssa.

<http://www.opensource.org/>

### Avoim ohjelma

Tietokoneohjelma, jota jaellaan jollakin avoimen lähdekoodin lisenssillä. Tiukasti tulkiten kyse on "avoimen lähdekoodin ohjelmasta", mutta tässä oppaassa käytetään myös termin lyhyempää versiota.

### Avoim standardi

European Interoperability Frameworkin (EIF) mukaan avoimen standardin täytyy täyttää seuraavat ehdot:

- Standardia ylläpitää voittoa tavoittelematon organisaatio ja sen kehitys tapahtuu kaikille sidosryhmille avoimella ja tasapuolisella menettelyllä.
- Standardi on julkaistu ja sen määrittelydokumentti on tarjolla ilmaiseksi tai nimellistä maksua vastaan (myös kopiointi, jakelu ja käyttö).
- Standardi ja sen osat ovat pysyvästi käytettävissä ilman teollisoikeudellisia korvauksia.
- Standardin uudelleenkäyttöä ei ole rajoitettu.

### Escrow

Lähdekoodin escrow-menettelyssä ohjelmiston lähdekoodi talletetaan kolmannen osapuolen, nk. escrow-agentin, haltuun. Ohjelmiston toimittaja ja hankkija tekevät sopimuksen, jossa määritellään ehdot, joiden mukaan escrow-agentti on velvollinen luovuttamaan ohjelmiston lähdekoodin hankintayksikölle. Tällaisia tilanteita voivat olla esimerkiksi toimittajan konkurssi tai ohjelman ylläpidon päättyminen.

### European Union Public Licence (EUPL)

EUPL on Euroopan komission hyväksymä avoimen lähdekoodin lisenssi. Se on saatavilla kaikilla EU:n virallisilla kielillä ja siinä on huomioitu EU:n juridinen viitekehys, joten se on EU-maissa laillisesti pitävä lisenssi. EUPL on yhteensopiva mm. GPL-lisenssin version 2 kanssa.

<http://www.osor.eu/eupl/>

### Free Software Foundation (FSF)

Vuonna 1985 perustettu voittoa tavoittelematon järjestö, jonka tarkoituksena on edistää tietokoneen käyttäjien vapauksia ja puolustaa vapaiden ohjelmien käyttäjien oikeuksia. FSF on luonut vapaan ohjelman määritelmän ja julkaissut avoimen lähdekoodin GPL- ja LGPL-lisenssit.

<http://www.fsf.org/>

### Freeware

Tietokoneohjelman jakelumalli, jossa ohjelmaa saa yleensä käyttää, kopioida ja levittää ilman maksua. Freeware-ohjelman lähdekoodi ei yleensä ole saatavilla, eikä lisenssinsaajalle anneta vastaavia vapauksia kuin avoimen lähdekoodin lisensseissä. Tämä opas luokittelee freeware-ohjelman suljetuksi ohjelmaksi.

### Free software

Ks. vapaa ohjelma.

### GNU General Public License (GPL)

Yleisin avoimen lähdekoodin lisenssi, joka takaa kyseisellä lisenssillä lisensoitujen ohjelmien käyttäjille avoimen lähdekoodin ja vapaan ohjelman määritelmien mukaiset vapaudet. Lisenssistä on olemassa nykyään kaksi laajasti käytössä olevaa versiota (GPL 2.0 ja GPL 3.0).

<http://www.gnu.org/copyleft/gpl.html>

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

### **Hankinta**

Hankinnalla tarkoitetaan tässä oppaassa tavarahan, hyödykkeen tai palvelun vastikkeellista hankkimista julkisia hankintoja koskevan lainsäädännön tarkoittamalla tavalla.

### **IDABC**

IDABC Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens) on Euroopan komission DG DIGIT:n alainen ohjelma, jonka avulla tähdätään tietotekniikan parempaan käyttöön yleiseurooppalaisissa sähköisissä hallinnollisissa prosesseissa (PEGS; Pan-European eGovernment Services). IDABC tekee suosituksia EU:n tavoitteenasettelun näkökulmasta hyödyntäen kulloinkin parasta ammatillista saatavilla olevaa osaamista.

<http://ec.europa.eu/idabc/>

### **Käyttöönotto**

Käyttöönotolla tarkoitetaan tässä oppaassa organisaation tai sen toimittajan suorittamia toimenpiteitä, joilla tietty järjestelmä tai ohjelmisto otetaan käyttöön.

### **Open Source Initiative (OSI)**

Vuonna 1998 perustettu voittoa tavoittelematon organisaatio, jonka tarkoituksena on edistää avoimen lähdekoodin ohjelmistojen käyttöä. OSI on lanseerannut käsitteen open source, suomeksi avoin lähdekoodi, ja ylläpitää avoimen lähdekoodin määritelmää.

<http://www.opensource.org/>

### **OSOR**

OSOR (Open Source Observatory and Repository for European public administrations) on Euroopan Unionin IDABC-ohjelman alainen hanke, joka edistää vapaiden ja avoimien ohjelmistojen käyttöä ja hyviä tietohallintokäytäntöjä EU:n julkisella sektorilla. OSOR sisältää myös julkisen sektorin ohjelmakoodi ja -dokumentaatiovaraston ja mahdollistaa siten avoimen lähdekoodin mallin mukaisen koodin jakamisen, jälleenhyödyntämisen ja kehitystyöhön osallistumisen.

<http://www.osor.eu/>

### **Open source**

Ks. avoin lähdekoodi.

### **Shareware**

Tietokoneohjelman jakelumalli, jossa ohjelmaa saa yleensä kopioida ja levittää vapaasti ja käyttää tietyn aikaa maksutta. Shareware-ohjelman lähdekoodi ei yleensä ole saatavilla, eikä lisenssinsaajalle anneta vastaavia vapauksia kuin avoimen lähdekoodin lisensseissä. Tämä opas luokittelee shareware-ohjelman suljetuksi ohjelmaksi.

### **Suljettu koodi**

Avoimen lähdekoodin vastakohta siinä mielessä, että ohjelmiston lähdekoodi pidetään liikesalaisuutena, eikä lisenssinsaajalle anneta pääsyä ja oikeuksia lähdekoodiin.

### **Suljettu ohjelma**

Tietokoneohjelma, jonka lähdekoodi ei ole käyttäjän saatavilla ohjelman toiminnan tarkastamista tai kehittämistä varten, eivätkä lisenssiehdot salli ohjelman muuttamista ja muutosten julkaisua.

### **Tarttuvuus**

Tarttuvuudella tai vahvalla vastavuoroisuuden velvoitteella tarkoitetaan ominaisuutta, joka edellyttää ohjelman julkaisemista alkuperäisellä lisenssillä niin kopioissa, muunnelmissa ja silloinkin, kun ohjelmaan yhdistetään uusia elementtejä esimerkiksi linkittämällä. Yleisin tarttuva lisenssi on GPL.

### Toimittajariippuvuus

Toimittajariippuvuudella (engl. vendor lock-in) tarkoitetaan tilannetta, jossa asiakas lukkiutuu tiettyyn tuotteeseen ja tuotteen tai palvelun toimittajaan. Monesti tilanteessa syntyy nk. de facto -monopoli, koska vaihtamisesta kilpailevaan tuotteeseen syntyisi liian suuret kustannukset. Ohjelmistojen kohdalla suljetut ohjelmistot voivat muodostaa tällaisen tilanteen, koska yleensä kukaan muu kuin ohjelmiston kehittäjäyhtiö tai sen valtuuttamat myyntikanavat eivät voi tarjota palveluja hankitulle ohjelmistolle.

### Vapaa ohjelma

Vapaa ohjelmaa voi käyttää mihin tarkoitukseen tahansa, muokata vapaasti ja levittää sekä alkuperäisiä että muokattuja versioita vapaasti. Ohjelman toiminnan tutkiminen ja muokkaaminen edellyttävät pääsyä ohjelman lähdekoodiin. Termin ja määritelmän taustalla on Free Software Foundation. Termi on käytännössä vaihtokelpoinen termin *avoin lähdekoodi* kanssa.

## 4 Johdanto avoimeen lähdekoodiin

### 4.1 Avoimen lähdekoodin evoluutio

Avoimen lähdekoodin ohjelma on tietokoneohjelma, jonka lisenssi takaa ohjelman käyttäjälle useita perinteisiä tietokoneohjelman lisenssiosopimuksia laajempia vapauksia. Käsitteenä avoin lähdekoodi on jo yli kaksi vuosikymmentä vanha. Tuona aikana käsite on saanut uusia sävyjä ja sen liiketoiminnallinen merkitys on muuntunut. Avoimen lähdekoodin evoluutioprosessissa voidaan erottaa kolme vaihetta:

- free software - vapaa ohjelma
- open source - avoin lähdekoodi
- commercial open source - kaupallinen avoin lähdekoodi.

Ensimmäisessä vaiheessa luotiin käsite vapaa ohjelma (engl. free software). Se korostaa vuonna 1985 perustetun Free Software Foundationin (FSF) vapaan ohjelmistokehityksen ja ohjelmistojen käyttäjien vapauden ideaalia. FSF:n tavoitteena on muuttaa ohjelmistoliiketoimintaa siten, että se turvaisi määrittelemiensä perusvapauksien toteutumisen. FSF:n luoma GPL-lisenssi on keskeinen työkalu tämän päämäärän saavuttamisessa.

Toisessa vaiheessa luotiin käsite avoin lähdekoodi (engl. open source). Siinä vapauden eettiset periaatteet siirrettiin sivummalle ja yritykset, ohjelmistokehityksen hajautettu prosessi ja kehittäjien välinen avoin yhteistoiminta nousivat keskiöön. Tässä vaiheessa perustettiin avoimen lähdekoodin lisenssejä hyväksyvä ja lisenssilistaa ylläpitävä Open Source Initiative (OSI), jota myös ohjelmistoteollisuus alkoi laajamittaisesti tukea ja jonka ympärille organisoitui kasvava kehittäjäyhteisö.

Kolmannelle vaiheelle, kaupalliselle avoimelle lähdekoodille (engl. commercial open source), on tyypillistä avoimen lähdekoodin hyödyntäminen osana perinteisesti lisenssoituja kaupallisia tuotteita, palveluita ja alan vakiintuneita liiketoimintamalleja. Avoimen lähdekoodin käytön tavoitteena on erityisesti kustannusten alentaminen ja kehitysnopeuden lisääminen. Tyypillisiä ilmiöitä ovat suljetun ja avoimen lähdekoodin tuotteita yhdistävät mixed source -tuotteet, joihin sisältyy sekä avoimia että suljettuja komponentteja, sekä ohjelmistojen tarjoaminen palveluina (SaaS, Software as a Service). Tämä kehitysvaihe alkoi noin vuonna 2005.

### 4.2 Avoimen lähdekoodin ohjelman määritelmä

Avoimen lähdekoodin ohjelmalla ei ole yhtä standardoitua määritelmää, eikä avoin lähdekoodi ole juridinen termi. Yhdysvaltalainen voittoa tavoittelematon Free Software Foundation (FSF) loi 1980-luvulla termin *free software* (vapaa ohjelma) ja niin ikään yhdysvaltalainen Open Source Initiative (OSI) loi vuonna 1998 termin *open source* (avoin lähdekoodi).

Julkishallinnon näkökulmasta käsitteillä ei ole merkittävää sisällöllistä eroa ja ne molemmat sisältävät samat käyttöön, kopiointiin, muokkaamiseen ja levittämiseen liittyvät perusvapaudet. Open Source Initiativen

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

määritelmä on yleisimmin viitattu erityisesti yritysmaailmassa ja esimerkiksi JIT 2007 -ehdoissa viitataan siihen määriteltäessä avointa lähdekoodia.

Open Source Initiativen määritelmän mukaan avoimen lähdekoodin ohjelman tulee täyttää seuraavat vaatimukset (käännöslähde: Wikipedia):

1. Ohjelman täytyy olla vapaasti levitettävissä ja välitettävissä.
2. Lähdekoodin täytyy tulla ohjelman mukana tai olla vapaasti saatavissa.
3. Myös johdettujen teosten luominen ja levitys pitää sallia.
4. Lisenssi voi rajoittaa muokatun lähdekoodin levittämistä vain siinä tapauksessa, että lisenssi sallii korjaustiedostojen ja niiden lähdekoodin levittämisen. Lisenssi voi edellyttää, että johdannaisissa teoksissa käytetään erilaista nimeä tai versionumeroa kuin alkuperäisessä ohjelmassa.
5. Yksilöitä tai ihmisryhmiä ei saa asettaa eriarvoiseen asemaan.
6. Käyttötarkoituksia ei saa rajoittaa.
7. Kaikilla ohjelman käsiinsä saaneilla on samat oikeudet.
8. Lisenssi ei saa olla riippuvainen laajemmasta ohjelmistokokonaisuudesta, jonka osana ohjelmaa levitetään, vaan ohjelmaan liittyvät oikeudet säilyvät, vaikka se irrotettaisiin kokonaisuudesta.
9. Lisenssi ei voi asettaa ehtoja muille ohjelmille. Ohjelmaa saa levittää myös yhdessä sellaisten ohjelmien kanssa, joiden lähdekoodi ei ole avointa.
10. Lisenssin sisällön pitää olla riippumaton teknisestä toteutuksesta. Oikeuksiin ei saa liittää varaumia jakelutavan tai käyttöliittymän varjolla.

### 4.3 Tietokoneohjelman lähdekoodi

Lähdekoodi on tietokoneohjelman rakennepiirustukset. Ohjelman tekijät kirjoittavat lähdekoodiin käskyt ja ohjeet, joiden avulla tietokone saadaan toimimaan halutulla tavalla. Jotta tietokone ymmärtää syötettyjä käskyjä, lähdekoodi on useimmiten käännettävä niin sanottuun suoritettavaan muotoon ennen sen ajamista tietokoneessa.

Ohjelman suoritettava muoto ei ole ihmisen tulkittavissa. Koska suljetuissa ohjelmissa lähdekoodi yleensä pidetään liikesalaisuutena eikä siihen anneta muutos- ja levitysoikeuksia, niin hankintayksiköllä ei ole teknisiä ja juridisia edellytyksiä tutkia ohjelman toteutusta tai tehdä tai teettää muutoksia ohjelmaan. Avoimen lähdekoodin ohjelman toteutus on lähdekoodin avoimuuden ja vapaampien lisenssiehtojen myötä tarkastettavissa ja muutettavissa joko hankintayksikön tai minkä tahansa kolmannen osapuolen toimesta.

Harva käyttäjä ja hankintayksikkö katsoo tarvitsevansa lähdekoodia, hankinnathan koskevat tietokoneohjelmia ja tietojärjestelmiä. Lähdekoodin avoimuudella voi kuitenkin olla hankintayksikölle suurikin merkitys. Tärkein etu on suurempi riippumattomuus toimittajasta. Kun toimituksen kuuluu myös lähdekoodi ja oikeus käyttää sitä ohjelman toiminnan muuttamiseen, voidaan pienentää yhden toimittajan varaan joutumisen riskiä. Kehityksen ja ylläpidon ja muut mahdolliset työt voi tällöin tehdä joku muukin kuin ohjelmiston alkuperäinen kehittäjä.

Riskienhallinnan näkökulmasta tilannetta voisi verrata rakennuksen ostamiseen. Rakennuksen hoito ja huolto edellyttää sitä, että käytettävissä on rakennuksen rakennepiirustukset ja muu tekninen dokumentaatio. Jos rakennus hankitaan leasing-sopimuksella siten, että rakentaja vastaa huollosta, huoltotoimen laatuun on vaikea vaikuttaa. Jos rakennuksen käyttäjällä on piirustukset ja muu tekninen dokumentaatio, on mahdollista kilpailuttaa huoltotoimen toteuttaminen. Sama tilanne on tietokoneohjelmien kohdalla, jos rakennuspiirustuksia vastaava lähdekoodi on saatavilla ja oikeudellisesti hyödynnettävissä.

On hyvä huomata, että esimerkiksi kriittisten tai erityisiä läpinäkyvyysvaatimuksia sisältävien järjestelmien toteutus voidaan haluta tarkastaa ennakolta. Ohjelmistojen kohdalla tämä on tehtävissä ainoastaan käymällä läpi ohjelmiston lähdekoodi.

## 4.4 Avoimen lähdekoodin ohjelma on myös kaupallinen ohjelma

Avoimen lähdekoodin ohjelmia pidetään usein virheellisesti ei-kaupallisina niiden vapaan jakelumallin vuoksi. Avoimen lähdekoodin lisenssit eivät kuitenkaan estä kaupallisen liiketoiminnan harjoittamista. Markkinoilla toimii laaja joukko yrityksiä, joiden ratkaisut ja palvelut perustuvat avoimen lähdekoodin ohjelmistoihin. Lisäksi monet suljetun koodin ratkaisuja tarjoavat yritykset harjoittavat liiketoimintaa myös avoimen lähdekoodin puolella: mallit eivät sulje toisiaan pois. Monet kaupalliset suljetut ohjelmistot sisältävät avoimen lähdekoodin teknologiaa.

Avoimen lähdekoodin ohjelmista ei makseta lisenssimaksuja, mikä ymmärrettävästi luo mielikuvan ei-kaupallisista ilmaisohjelmista. Lisenssimaksujen sijaan useat kaupalliset avoimen lähdekoodin toimijat tarjoavat avoimen lähdekoodin ohjelmien ympärille koottuja palveluja, jolloin liiketoiminta syntyy palveluista eikä lisenssimyynnistä. Kaupallisten ja ei-kaupallisten ohjelmistojen sijaan onkin mielekkäämpää erotella avoimiin ohjelmistoihin ja suljettuihin ohjelmistoihin perustuva liiketoiminta.

Taulukko 1. Eräitä tunnusmerkkejä avoimen ja suljetun ohjelmiston liiketoiminnassa.

	<b>Avoin ohjelmisto</b>	<b>Suljettu ohjelmisto</b>
Toimittajien määrä	Ohjelmisto saatavissa useilta toimittajilta	Ohjelmisto saatavissa vain yhdeltä toimittajalta
Liiketoiminta	Painopiste palveluissa	Painopiste lisensseissä ja niiden jakelutavassa

Perinteisten kaupallisten ohjelmistojen rinnalla käytössä on myös laajasti ns. shareware- ja ilmaisohjelmistoja (freeware), jotka usein virheellisesti yhdistetään avoimeen lähdekoodiin. Näitä ohjelmistoja saa ohjelmien mukana tulevien käyttöehtojen perusteella yleensä kopioida ja levittää vapaasti ja myös käyttää tietyissä rajoissa. Ohjelmien mukana ei kuitenkaan tule lähdekoodia eikä avoimen lähdekoodin perusvapauksia, joten niitä ei voi esimerkiksi kehittää itsenäisesti edelleen. Lisäksi shareware-tuotteissa osa kehittyneemmistä ominaisuuksista on sidottu maksun suorittamiseen, joten ohjelmistoa ei voi käyttää täysipainoisesti ostamatta siihen lisenssiä. Ilmaisohjelmien käyttöoikeus puolestaan voi olla rajattu esimerkiksi vain ei-kaupalliseen toimintaan.

## 4.5 Avoimet rajapinnat ja standardit

Ohjelmiston lähdekoodin avoimuus ei yksinään takaa täydellistä yhteentoimivuutta ja helppoa liitettävyyttä muihin ohjelmistoihin, vaan siihen tarvitaan avoimia ohjelmistorajapintoja ja standardeja. Verrattuna suljettuun ohjelmistoon jo pelkkä lähdekoodin avoimuus mahdollistaa kuitenkin ohjelmiston toiminnallisuuden tutkimisen ja muuttamisen, jolloin yhteentoimivuus on jollakin työmäärällä saavutettavissa.

### 4.5.1 Avoimet rajapinnat

Avoimilla rajapinnoilla tarkoitetaan sellaisia ohjelmistoon toteutettuja vapaasti käytettäviä, huolellisesti dokumentoituja liittymäpintoja, joiden välityksellä toiset ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja rajapinnan toteuttavan ohjelman kanssa. Rajapinnan päätarkoitus on tarjota palveluja yleisimmille toiminnoille, kuten esimerkiksi henkilön tietojen haku tietokannasta henkilötunnuksen perusteella.

Avoin lähdekoodi ei yksinään takaa ohjelman vaivatonta liitettävyyttä toisiin ohjelmiin. Mikäli avoimen lähdekoodin ohjelman rajapinnat ovat heikosti toteutetut ja dokumentoidut, voi ohjelman liitettävyyttä vaatia suuria määriä työtä. Toisaalta suljetut ohjelmistot ovat liitettävissä toisiin ohjelmistoihin, kun avoimien rajapintojen lisäksi käytetään avoimien standardien mukaisia tiedon kuvaustapoja sekä yhteisesti sovittuja ontologioita.

### 4.5.2 Avoimet standardit

Avoimilla standardeilla tarkoitetaan sellaisia vapaasti saatavilla ja toteutettavissa sekä hyödynnettävissä olevia määritelmiä, jotka jokin avoin, toimittajariippumaton foorumi tai organisaatio on esittänyt kuvaukseksi siitä, miten jokin asia tulisi tehdä. Avoin standardi ei voi estää miltään toimijalta sen käyttöä. Esimerkiksi tiedostot ovat muodossa, jonka avaamiseen ja tallentamiseen tarvittavan ohjelman saa jokainen toteuttaa.

Avoimia standardeja käytettäessä asiakkailla on suurempi mahdollisuus valita useamman valmistajan valikoimasta omaan tarkoitukseensa sekä ominaisuuksiltaan että hinnaltaan sopivin ratkaisu. Riskinä muilta toimijoilta joko teknisesti, juridisesti tai kustannuksiltaan suljetun teknologiaratkaisun valitsemisessa on, että teknologian valmistaja sitoo käyttäjät omiin tuotteisiinsa.

Avoimet standardit lisäävät kilpailua ja pienentävät toimittajariippuvuutta. Niillä parannetaan ohjelmiston yhteentoimivuutta muiden ohjelmistojen kanssa, mikä lisää ohjelmiston ja tuotetun sisällön käyttö- ja jälleenyödyntämismahdollisuuksia. Kaikkiin käyttösovelluksiin ei kuitenkaan ole olemassa avointa standardia, jolloin teknologian ja lähdekoodin avoimuuden merkitys kasvaa.

Avoimen lähdekoodin projekteille on tyypillistä avoimien ja yleisten standardien noudattaminen, mikäli se vain on järkevää ja mahdollista.

## 4.6 Yhteenveto avoimen lähdekoodin hyödyistä ja haasteista

Avoimen lähdekoodin ohjelmien käytöllä saavutettavissa olevat hyödyt ja mahdolliset haasteet ja riskit riippuvat avoimen lähdekoodin käyttötavasta ja -laajuudesta. Esimerkiksi käytettäessä ohjelmia työasemissa ja palvelimissa organisaation sisällä monet avoimen lähdekoodin juridiset riskit jäävät kokonaan pois, mutta jos tuotetta muutetaan ja levitetään, niin huomioon otettavia asioita on paljon enemmän. Toisaalta aktiivisella osallistumisella ja omalla kehityspanoksella avoimen lähdekoodin projekteihin voidaan saavuttaa paljon enemmän hyötyjä kuin pelkällä passiivisella hyödyntämisellä.

### 4.6.1 Hyötyjä

#### 4.6.1.1 Ohjelmistokehityksen nopeutuminen

Avoin lähdekoodi voi nopeuttaa joustavien ja innovatiivisten IT-ratkaisujen rakentamista ja alentaa kustannuksia. Tämä on mahdollista käyttämällä jo tehtyä koodia uudelleen ja hyödyntämällä avoimen lähdekoodin yhteisöjen resursseja. Haasteena on löytää käyttötarkoitukseen sopivia ohjelmia ja ohjelmakomponentteja. Myös lisenssien väliset yhteensopimattomuudet voivat estää ohjelmien yhdistämisen.

#### 4.6.1.2 Paikallisten toimijoiden kilpailuedellytysten parantaminen

Avoin lähdekoodi ei sulje mitään toimijaa pois markkinoilta, jolloin myös paikalliset toimijat voivat tuottaa ohjelmakoodia ja palveluja. Tyypillisesti avoin lähdekoodi lisää kilpailua ja yrittäjyyttä ja mahdollistaa uusien toimijoiden pääsemisen markkinoille helpommin. Tältä osin avoin lähdekoodi on erityisen positiivinen aloittaville yrittäjille ja PK-yrityksille.

#### 4.6.1.3 Toimittajariippuvuuden pieneminen ja vapaa kilpailutus

Avoimen lähdekoodin takaamat vapaudet poistavat yhden suljettuihin ohjelmistoihin liittyvän riskin, toimittajariippuvuuden. Tilanne jossa toimittaja poistuu markkinoilta ja vie poistuessaan mukanaan tuen ei ole mahdollinen käytettäessä avointa lähdekoodia. Suljetun lähdekoodin sovelluksissa sovellettavaa koodin rajoitettua luovuttamista escrow-menettelyllä ei avoimen ohjelmistojen osalta tarvita.

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

Kun yksi yritys ei voi kontrolloida ohjelman hyödyntämisessä tarvittavia tietoja ja oikeuksia, kaikki ohjelmistoon liittyvät toimenpiteet ovat kilpailutettavissa. Vapaa kilpailutus on mahdollista tuotteen elinkaaren kaikissa vaiheissa.

Jotta palveluiden kilpailu aidosti toimisi, on markkinoilla oltava useita kyvykkäitä palveluntarjoajia. Vähän käytettyjen ohjelmien kohdalla kilpailu voi olla vähäistä, mutta ohjelman suosion kasvaessa tilanne voi nopeasti muuttua.

### 4.6.1.4 Yhteiskunnallinen riippumattomuus

Kokonaisuudessaan avoin lähdekoodi vähentää yhteiskunnan riippuvuutta ja haavoittuvuutta. Esimerkiksi kauppapoliittisten kriisitilanteidenkin aikana voidaan luottaa siihen, että pakottavassa tapauksessa voidaan itse ylläpitää toiminnallisuutta tai alihankkia ylläpito paikalliselta toimittajalta.

### 4.6.1.5 Kustannussäästöt

Avoimen lähdekoodin ratkaisut voivat alentaa ohjelmistohankintojen kustannuksia. Suurimmat mahdollisuudet säästöihin ovat vapaan kilpailutuksen mahdollistamassa toimittajariippumattomuudessa ja lisenssimaksuttomuudessa. Kokonaisedullisuuden laskeminen voi olla haastavaa, sillä säästöt voivat olla suoria (ei lisenssimaksuja) tai epäsuoria (yhteentoimivuus tulevaisuudessa mahdollisesti hankittavan järjestelmän kanssa).

Kustannussäästöjen saavuttaminen edellyttää avoimen lähdekoodin hallittua hyödyntämistä. Mitään ohjelmistoa ei tosiasiaa voida ottaa ilmaiseksi käyttöön, sillä käyttöönotto ei ole kulutonta edes itse tehtynä. Mikäli ohjelmisto ladataan verkosta, tulee varata riittävästi aikaa ohjelmiston omaehtoiseen opiskeluun, evaluointiin ja käyttöönottoon. Useissa tilanteissa on suositeltavaa käyttää apuna ulkoa hankittavaa osaamista.

Seuraavassa on listattu eräitä avoimen lähdekoodin mahdollisuuksia kustannussäästöihin:

- Ei lisenssimaksuja. Avoimien ohjelmien oikeuksien antamisesta ei saa periä rojalteja tai muita maksuja.
- Kaikkien palveluiden vapaa kilpailutus tuotteen koko elinkaaren ajan, myös järjestelmästä toiseen siirryttäessä. Toimittajaa voidaan vaihtaa tarvittaessa ja eri palveluihin voidaan valita eri toimittaja, mikäli näin halutaan.
- Pilotointia ja kokeilua voidaan harrastaa sängen pienellä riskillä ja lopettaa heti, jos huomataan, että ratkaisu ei toimi tai on liian työläs. Näin riski sitoutua sopimattomiin ja vähälle käytölle jääviin ratkaisuihin pienenee.
- Oman ja avoimen lähdekoodin yhteisön osaamisen hyödyntäminen.
- Yhteistyön tekeminen muiden organisaatioiden kanssa esimerkiksi ohjelmistoja levittämällä.

### 4.6.1.6 Kansalaisten tasavertaisuuden paraneminen

Koska avoimen lähdekoodin tuotteet voidaan luovuttaa vastikkeetta eteenpäin, ei tule tilannetta, jossa yhteiskunnan kansalaisten yhdenvertaisuuden saavuttamisessa olisi tämän suhteen ongelmia. Jokainen kansalainen voi ottaa käyttöön tarvittavat toiminnallisuudet, mikäli käytössä on tietotekniset perusvalmiudet.

Esimerkiksi organisaatio, joka tuottaa jotain sähköistä materiaalia jaettavaksi suurelle ihmismäärälle voi avointen ohjelmistojen avulla tarjota vapaasti materiaalin käyttöön tarvittavia ohjelmistoja. Tilastokeskus on ratkaissut tällä tavalla tilastotietojen levittämisen ongelman: samalle tallennusmedialle on tilastotietojen taulukkoaineiston lisäksi tallennettu avoimen lähdekoodin toimisto-ohjelmisto OpenOffice.org, jolla aineisto voidaan avata muokattavaksi. Tätä ohjelmistoa ei kuitenkaan tarvitse asentaa, jos käyttäjällä on jo valmiina ohjelmisto, joka osaa avata kyseiset taulukkolaskentadokumentit. Tilastokeskuksen ei tarvitse maksaa OpenOffice.org:n sisällyttämisestä tallennusmedialle lisenssi- tai muita lupamaksuja.

### 4.6.1.7 Koodin tarkistettavuus

Avoimen lähdekoodin ratkaisut voivat joissakin turvallisuuskriittisissä tapauksissa olla koodin escrow-menettelyn ohella ainoita vaihtoehtoja toteuttaa ohjelmiston tarkistus esimerkiksi haittaohjelmien ja piilotettujen toiminnallisuuksien osalta. Kaikki suljettujen ohjelmistojen koodin katseluintimahdollisuudet eivät kuitenkaan mahdollista sovelluksen kääntämistä luovutetuista lähdekoodeista. Tällöin lopullista varmuutta ajettavan ohjelman ominaisuuksista ei suljetulla puolella voida saavuttaa.

### 4.6.1.8 Ohjelmiston avoimen jakamisen edut ja versionhallinnan haasteet

Avoimen lähdekoodin kehityksen eteenpäin vievä voima on lähdekoodin julkinen saatavuus. Kuka tahansa voi kehittää ohjelmaa ja jakaa tekemänsä muutokset muille, jolloin muutkin hyötyvät. Mikäli organisaatio päätyy tilanteeseen, jossa valmista ohjelmistoa ei ole kohtuullisin ehdoin tai lainkaan olemassa, vaan sellainen kannattaa tehdä tai teettää omana tuotekehityksenä, voidaan harkita koodin avaamista avoimen lähdekoodin projektiksi.

Jos koodin osana käytetään valmiita komponentteja ja päätetään levittää omaakin koodia, niin kannattaa etukäteen selvittää käytettyjen komponenttien lisenssien yhteensopivuusasiat sekä koodin jakamisesta aiheutuvat velvoitteet. Mikäli ohjelmistokehityksessä käytetään ulkopuolista asiantuntija-apua, on jo tarjouspyyntövaiheessa huomioitava koodin jakelumahdollisuus ja kirjattava tarjouspyyntöön reunaehdot tuotettavan koodin lisenssille.

Julkishallinnossa, jossa toimintaympäristö on jokseenkin vakioitu, samanlaisia hallinnollisia prosesseja on kaikkialla hallinnossa ja kaikkia koskee sama lainsäädäntö, ohjelmiston jakaminen voi olla erittäin edullista. Avoimia ohjelmistoja voi jakaa vapaasti muille ja muut organisaatiot voivat ottaa niitä käyttöön ja saavuttaa näin säästöjä. Koodin avoimella lisenssillä jakeluun luovuttava saavuttaa etuja, mikäli muut organisaatiot tekevät tai teettävät ohjelmistoon parannuksia, jotka voidaan ottaa käyttöön. Ohjelmiston laajempi käyttö voi synnyttää yhteistyötä organisaatioiden välillä esimerkiksi ohjelmiston kehittämisessä, jolloin kustannuksia voidaan jakaa.

Ero suljettujen ohjelmistojen toimintamalliin on merkittävä. Siinä asiakkaan rahoilla tehdyt parannukset jäävät useimmiten ohjelmiston toimittajan omaisuudeksi ja niiden jatkohyödyntäminen edellyttää kaupankäyntiä tämän nimenomaisen yrityksen kanssa.

Julkinen jakelun mallissa on kuitenkin tekninen haaste, joka suljettujen ohjelmien mallissa kuuluu ohjelmistotaloille: ohjelmiston eri versioiden hallinta. Kun ohjelmistoon tehdään muutoksia, pitää olla menettelytapa, jolla varmistetaan että versiot voidaan erottaa toisistaan. Tätä kutsutaan versionhallinnaksi.

Vaikka versionhallinta ei ole pakollista, se on käytännössä välttämätöntä, jotta ohjelmiston kehitys pysyisi hallittavana prosessina. Jos versionhallintaa ei tehdä, ohjelmiston kehitys haarautuu useiksi rinnakkaisiksi versioiksi, jotka saattavat olla keskenään yhteensopimattomia. Tällainen kehitys ei ole toivottavaa.

Oheiseen taulukkoon 2 on tiivistetty ohjelmistojen avoimella lisenssillä jakamisen etuja ja haittoja verrattuna siihen, että ohjelmistoa ei jaeta. On syytä huomata, että oikeanpuoleinen sarake pätee niin suljettujen ohjelmistojen käyttöön kuin siihenkin, että organisaatio ostaa ohjelmiston lähdekoodeineen ja päättää olla jakamatta sitä. Luvussa 5.8 on käsitelty ohjelmiston hankkimista avoimen lähdekoodin lisenssillä.

Taulukko 2. Ohjelmiston jakaminen verrattuna jakamatta jättämiseen.

<b>Ohjelmisto jaellaan julkisesti avoimella lisenssillä</b>	<b>Ohjelmistoa ei jaella julkisesti</b>
Muut voivat käyttää ohjelmistoa ja hyötyä siitä.	Muut voivat käyttää ohjelmistoa maksamalla siitä sen omistajalle.
Muut voivat kehittää ohjelmistoa edelleen ja jakaa muutoksiaan muille. Näin kaikki voivat hyötyä kehityksen tuloksista.	Ainoastaan ohjelmiston omistaja tai sen valtuuttama taho voi kehittää ohjelmistoa.

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

Mielekäs kehitys edellyttää versionhallintaa. Tämä vaatii ostaja- tai tuottajataholta osaamista. Panostuksen vastapainona on toisten tekemästä jatkokehityksestä itselle koitua hyöty, jota ei kuitenkaan voida ennakoita tai taata. Julkaistu koodi ei enää ole omassa hallinnassa.	Ohjelmiston lähdekoodi ja kehityskaari ovat täysin ohjelmiston kehittäjän hallinnassa. Kehitys tapahtuu tämän saneleman aikataulun mukaisesti. Myös päätös siitä, milloin ohjelmisto on kehitetty uudeksi versioksi, josta pitää erikseen maksaa, on yksin ohjelmiston omistajan päätösvallassa. Toisaalta ratkaisu on helppo: ostajan ei tarvitse huolehtia versionhallinnasta.
Ohjelmiston käyttöiän määrittää prosessuaalinen tai tekninen ympäristö. Tarvittaessa se voidaan muuntaa eri käyttöjärjestelmässä tai -ympäristössä toimivaksi. Muutoksia ohjelmistoon voidaan tehdä tai teettää niin pitkään kuin halutaan.	Ohjelmiston valmistaja päättää ohjelmiston käyttöiän esimerkiksi siten, että siihen ei enää tehdä päivityksiä tai sitä ei tueta uudessa käyttöjärjestelmäversiossa.
Jakamiseen liittyvät juridiset riskit on huomioitava.	Jakamiseen liittyvistä juridisista riskeistä ei tarvitse huolehtia.

### 4.6.2 Haasteita

#### 4.6.2.1 Avoimen lähdekoodin ohjelmistojen saatavuus ja sopivuus

Vaikka avoimen lähdekoodin ohjelmistoja on tehty valtava määrä ja uusia projekteja käynnistyy jatkuvasti, niin kaikkiin tarkoituksiin ei ole vielä saatavilla kypsiä avoimen lähdekoodin ratkaisuja. Toisaalta vaikka sopiva ohjelmisto olisikin olemassa, niin siitä ei välttämättä tiedetä eikä näin ollen osata etsiäkään.

#### 4.6.2.2 Kaupallisten toimijoiden löytäminen

Erityisesti vähemmän käytettyjen ohjelmistojen kohdalla ratkaisutoimittajia ja tukipalveluita tarjoavia yrityksiä ei välttämättä ole olemassa tai nämä eivät markkinoi itseään. Kaupallisen ja erityisesti kotimaisen toimijan löytäminen voikin osoittautua haastavaksi erityisesti vähemmän käytettyjen tuotteiden osalta. Vähemmän käytetty tuote saattaa olla varsin uusi ja myös tuotetukea tarjoavat yritykset voivat olla ainakin osittain uusia toimijoita ICT-markkinoilla.

#### 4.6.2.3 Ylläpito- ja muutoksenhallinta

Monissa voimakkaasti kehittyvässä avoimen lähdekoodin projektissa muutostahti on erittäin ripeä. Osa näistä projekteista voi julkaista uuden ohjelmistoversion muutaman päivän tai viikon välein. Joissain tilanteissa tämä liian nopea kehitys ja suuret muutokset ovat aiheuttaneet ohjelmiston luotettavuuden ja vakauden heikkenemistä. Kannattaa huomata, että tämä muutosten nopeus koskee myös osaa suljetuista ohjelmistoista.

Avoimen lähdekoodin ratkaisua valittaessa onkin suositeltavaa varmistua siitä millainen julkaisuaikataulu valittavalla ohjelmistolla on. Jotkut avoimen lähdekoodin projektit ovat siirtyneet selkeään julkaisukäytäntöön, jossa uusi versio tuotteesta tuodaan markkinoille esimerkiksi puolen vuoden välein. Tämän uuden version lisäksi ylläpidetään vanhojen versioiden virhekorjauksia esimerkiksi kolmen vuoden ajan.

Ohjelmistoa käyttävän organisaation on myös punnittava uusien versioiden hyödyllisyys suhteessa omaan käyttöön. Mikäli uusi versio ei tuo mitään oleellista parannusta sen hetkiseen toteutukseen, niin päivitystyö on todennäköisesti turha ajankäyttöä ja riskinotto. Tämä koskee niin suljettuja kuin avoimia ohjelmistoja.

#### 4.6.2.4 Avoimen lähdekoodin käyttöönoton vaatimat omat resurssit

Mikäli hankintayksikkö päätyy hankkimaan avoimen lähdekoodin ohjelman omana käyttöönottona, niin yksikön on syytä varmistua, että käytettävissä on tarvittavissa määrin omia resursseja ja osaamista. Useisiin avoimen lähdekoodin ohjelmiin on vuosien kuluessa toteutettu suuri määrä ominaisuuksia, jotka ovat

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

oletusasennuksessa käytettävissä. Näiden ohjelmien oletusasennus saattaa tuntua tarpeettoman laajalta ja ennen laajempaa käyttöönottoa voi olla tarpeen poistaa tarpeettomia ominaisuuksia. Etuna tässä kannattaa huomata se, että tarpeen myöhemmin ilmetessä näitä ominaisuuksia voidaan ottaa vaivattomasti uudelleen käyttöön.

Omassa käyttöönotossa ei luonnollisesti kukaan ulkopuolinen taho tarjoa oletuksena takuita ohjelmiston toimivuudesta ja vastaa oikeudellisesti esimerkiksi ohjelmistossa mahdollisesti piilevistä tekijänoikeusloukkauksista. Lähes poikkeuksetta vähänkään enemmän käytetyllä ohjelmalla on kuitenkin tukisivusto, josta kysymyksiin voi etsiä vastauksia ja jonka avulla ongelmat mahdollisesti ratkeavat. Tämä vaatii kuitenkin omaa aktiivisuutta ja osaamista.

Omassa käyttöönotossa hankintayksikkö tulee ainakin jossain määrin riippuvaiseksi kehittäjäyhteisöstä, johon voi vapaaehtoisten kehittäjien lisäksi kuulua useita yrityksiä.

### 4.7 Avoimen ja suljetun lähdekoodin vertailutaulukko

Taulukko 3. Avoimen ja suljetun lähdekoodin vertailu.

	<b>Avoin lähdekoodi</b>	<b>Suljettu lähdekoodi</b>
<b>Ohjelmistokehitys</b>	Voidaan hyödyntää avoimen lähdekoodin ohjelmistoja ja komponentteja eli jo valmiiksi tehtyä työtä. Lisenssien yhteensopivuus tarkistettava, mikäli suunniteltua teosta aiotaan levittää.	Avoimen lähdekoodin ohjelmien hyödyntäminen on rajoitetusti mahdollista, mikäli tekijänoikeuksien hajautus ei haittaa. Mikäli halutaan säilyttää tekijänoikeudet itsellä, voidaan käyttää vain niitä komponentteja, joihin on hankittu oikeus tai jotka on toteutettu itse.
<b>Paikalliset toimijat</b>	Voivat tuottaa ohjelmakoodia ja palveluja.	Palvelujen ja erityisesti ohjelmakoodin tuotanto on rajoitettua.
<b>Toimittajariippuvuus</b>	Ei riippuvuutta oikeuksien omistajaan, toimittaja vaihdettavissa.	Ohjelmiston oikeuksien omistaja kontrolloi toimittajakenttää.
<b>Kansalaiset</b>	Tasavertaisessa roolissa. Ohjelmistoja voidaan jakaa kaikille.	Ohjelmistojen käyttölisenssit yleensä maksullisia. Asettaa kansalaiset eriarvoiseen asemaan mm. varallisuuden mukaan.
<b>Koodin tarkistettavuus</b>	Tarkistettavissa. Koodi voidaan vapaasti luovuttaa kenelle tahansa arvioitavaksi ja tarkistettavaksi.	Oletuksena ei tarkistettavissa, vaan vaatii sopimisen ohjelmiston oikeuksien omistajan kanssa.
<b>Koodin laadukkuus</b>	Todennettavissa. Koodi voidaan vapaasti luovuttaa kenelle tahansa arvioitavaksi.	Oletuksena ei todennettavissa, vaan vaatii sopimisen ohjelmiston oikeuksien omistajan kanssa.

### 4.8 Avoimen lähdekoodin liiketoiminta

Avoin lähdekoodi ja kaupallinen liiketoiminta eivät sulje toisiaan pois. Avoimen lähdekoodin teknologioita hyödynnetäänkin liiketoiminnassa laajalti, mutta hyödyntämisen syvyys ja laajuus vaihtelee yrityksissä suuresti.

Yleisintä on avoimen lähdekoodin hyödyntäminen yrityksen sisäisissä operaatioissa työkaluina, työpöytäkäytössä ja palvelimissa. Askel pidemmälle on liittää avointa lähdekoodia omiin myytäviin

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

tuotteisiin ja palveluihin tai rakentaa liiketoiminta olemassa olevien avoimien tuotteiden ympärille. Pisimmälle vietyä perustetaan oma avoimen lähdekoodin projekti, hallinnoidaan ja ylläpidetään sitä aktiivisesti, kehitetään tuotteelle palvelukonsepti ja luodaan yhteisöasemaan voimakkaasti nojautuva liiketoimintamalli.

Koska avoimen lähdekoodin lisenssit estävät perinteisen, käytön rajoituksiin perustuvan lisenssimyynnin, on avoimen lähdekoodin liiketoiminnassa käytettävä muita tulonlähteitä. Seuraavaksi esitellään eräitä avoimen lähdekoodin teknologioihin perustuvia liiketoimintamalleja.

Avoimen lähdekoodin liiketoimintamallit voidaan jakaa muutamaaan pääryhmään sen mukaan mistä ansainta muodostuu. Omaan ryhmään näiden varsinaisten liiketoimintamallien ulkopuolella on puhdas harrastustoiminta, jossa siinäkin useimmiten on nähtävissä jonkinlainen edun tavoittelu esimerkiksi siten, että tekijä hyödyntää avoimen lähdekoodin projektin tuomaa osaamista ja julkisuutta muussa toiminnassaan, kuten työnhaussa.

Taulukko 4. Hankinnan kohde eräissä liiketoimintamalleissa avoimella ja suljetulla puolella.

Liiketoimintamalli	Hankinnan kohde - avoin	Hankinnan kohde - suljettu	Huom.
Palvelupohjainen	Ostetaan palvelu	Ostetaan palvelu	Avoimella puolella tehtävää työtä voi tarjota kuka vain. Suljetulla puolella työn tekijä on ohjelmistovalmistajan kontrolloitavissa.
Käyttäjälle ilmaiset, mainosrahoitteiset palvelut	Ei hankinta, koska vastikkeeton	Ei hankinta, koska vastikkeeton	Ei tyypillisesti käytössä julkishallinnossa.
Laitteeseen sulautettu	Hankitaan laite	Hankitaan laite	Laitteen räätälöinti organisaatiolle sopivammaksi on mahdollista avoimella puolella. Avoin lähdekoodi ei kasvata laitteen hintaa ohjelmistolisenssimaksuilla ja voi madaltaa ohjelmiston kehitys- ja ylläpitokustannuksia.
Lisenssimyynti	Ei hankinta, koska vastikkeeton	Käyttölisenssi	Avoimen lähdekoodin ohjelmissa ei ole perinteisiä lisenssimaksuja. Ns. kaksoislisensointi mahdollinen, jolloin avoimesta ohjelmasta on maksullisesti saatavilla eri lisenssiehdoilla varustettu versio.
Tilauspalvelu	Hankitaan tuote- ja palvelupaketti	Hankitaan tuote- ja palvelupaketti	Avoimella puolella ei välttämättä lisenssimaksuja.
Ohjelmisto palveluna, Software as a Service (SaaS),	Palveluhankinta	Palveluhankinta	Hankinnassa (ja palvelussa) ei ole mitään eroa.

### 4.9 EU:n linjaukset suhteessa avoimiin ohjelmistoihin ja standardeihin

Euroopan unionin eri elimet ovat useissa yhteyksissä kiinnittäneet huomiota toimittajariippuvuuden aiheuttamiin ongelmiin ja tilanteen kilpailua rajoittavaan vaikutukseen. Tavoitteena on aiempaa avoimempi kilpailu, alhaisemmat tietojärjestelmäkustannukset ja parempi tietojen vaihdettavuus eri ohjelmistojen välillä eli yhteentoimivuus (interoperability).

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

Ministerikokouksen julistus Brysselissä marraskuussa 2001 toteaa: *"Ministerit ilmaisivat huolensa yhdestä ICT-toimittajasta tai valmistajasta ja peräänkuuluttivat kilpailun lisäämistä. Ministerit [...] kehottivat komissiota kiihdyttämään avoimen lähdekoodin vaihtoehtojen kehitystä [...] ja lisäksi avoimet standardit ja "teknologianeutraali" sääntely on elintärkeää"*.

Comossa heinäkuussa 2003 *"Ministerit kannustivat hallintoa määrittelemään järjestelmänsä ja prosessinsa uudelleen, jotta hallinnon eri tasojen toimia voidaan koordinoida paremmin avoimia standardeja käyttämällä."*

Muun muassa näiden tavoitteiden toteuttamiseksi EU:n komissio on perustanut IDABC-ohjelman. IDABC:n suositukset ja ohjeet laaja-alaisen valmistelun jälkeen ovat päättyneet suosittamaan avoimen lähdekoodin käytön huomioon ottamista, jopa priorisoimista hankintatoimen yhteydessä hyvänä vaihtoehtona EU:n poliittisten tavoitteiden, kuten yhteismarkkinat, kilpailukyky ja tasa-arvoinen kohtelu hallinnossa, saavuttamiseksi.

IDABC:n alla toimii OSOR-hanke, joka mm. kokoaa tietoa jäsenmaissa tehdyistä avoimen lähdekoodin hankinnoista ja -käyttöönotoista, levittää tietoa eri maista saaduista hyvistä käytännöistä ja hallinnoi julkisen sektorin avoimen lähdekoodin ohjelmien säilytyspaikkaa. Toinen esimerkki EU:n ja IDABC:n avoimeen lähdekoodiin liittyvistä käytännön toimista on EUPL-lisenssi. EUPL on avoimen lähdekoodin lisenssi, joka luotiin vastaamaan Euroopan komission vaatimuksiin kieliriippumattomasta ja terminologialtaan EU:n lain kanssa yhteensopivasta lisenssistä.

European Interoperability Framework (EIF) -dokumentti suosittelee, että julkisen sektorin tulisi tutustua avoimen lähdekoodin yhteisön työskentelytapoihin, jotta avoimen lähdekoodin ohjelmista koituvat hyödyt avautuvat. Tämän lisäksi suositellaan avoimien ohjelmistojen käytön inventointia, koska monet organisaatiot ovat yllättyneet siitä, kuinka paljon niitä on jo käytössä. Näin syvällistä perehtymistä suositellaan siksi, että ohjelmistojen avoimuudesta on selkeitä hyötyjä. Tavoitteena on tilanne, jossa julkisella hallinnolla on oma ohjelmistohakemisto, jossa olevat ohjelmat ovat vapaasti hyödynnettävissä.

EU:n komissio edistää aktiivisesti avoimien ohjelmistojen käyttöä ja pyrkii asettamaan avoimet ohjelmistot etusijalle. Kansallisella tasolla esimerkiksi Iso-Britannia on ottanut kannan, jonka mukaan avoin ohjelmisto on oletusarvoinen valinta. Tanskassa muun kuin avoimen lähdekoodin ohjelmiston valitsevat viranomaiset joutuvat erikseen perustelevaan valintansa Tanskan valtiovarainministeriölle.

### 4.10 Avoin lähdekoodi muissa suosituksissa

Ennen tätä opasta avointa lähdekoodia on käsitelty mm. valtiovarainministeriön dokumentissa *Suositus valtion tietojärjestelmien ja rajapintojen avoimuudesta* (Valtiovarainministeriön työryhmämuistioita 23/2003). Suosituksessa todetaan mm. seuraavaa:

*"Selvityksessä todettiin avoimen lähdekoodin menetelmin rakennetun järjestelmän olevan varteenotettava vaihtoehto erityisesti silloin, kun on kyse useiden hallinnon organisaatioiden tarvitsemasta palvelusta, järjestelmän avoimen lähdekoodin komponentteja on olemassa tai järjestelmän läpinäkyvyydellä on erityistä merkitystä."*

Suosituksessa korostetaan, että *"järjestelmätoimituksissa tilaajan tulisi ehdottomasti saada oikeus järjestelmän lähdekoodin hallussapitoon sekä järjestelmämuutosten tekemiseen ja teettämiseen"*. Valmisohjelmistojen osalta tulisi *"huolehtia ainakin kriittisten järjestelmien koodin saatavuudesta Escrow-sopimuksin"*.

Avoimen lähdekoodin toimintamallin mainitaan sopivan erityisesti projekteihin, joita leimaa monistettavuus ja muunneltavuus. Joissakin julkisissa palveluissa teknologian läpinäkyvyys voi olla ensiarvoisen tärkeää ja lähdekoodin on oltava saatavilla ja myös muiden kuin tilaajan tarkasteltavissa.

Avoimen lähdekoodin lisäksi suosituksessa painotetaan rajapintojen ja standardien avoimuuden tärkeyttä:

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

*"Aina kun mahdollista tulisi järjestelmät rakentaa niin, että ei ajauduta yksittäisen toimittajan suljetun rajapinnan takia riippuvaisiksi kyseisen toimittajan järjestelmistä. Järjestelmäkomponentteja ja kokonaisia järjestelmiä tulisi olla mahdollista tarvittaessa vaihtaa."*

Julkisen hallinnon IT-hankintojen yleisissä sopimusehdoissa (JHS 166, JIT 2007), joita Julkisen hallinnon tietohallinnon neuvottelukunta (JUHTA) suosittelee valtion virastojen, liikelaitoksien ja rahastojen sekä kuntien ja kuntayhtymien käyttävän hankkiessaan IT-tuotteita ja -palveluita, on myös avoin lähdekoodi huomioitu. JIT 2007 -ehtoja on käsitelty luvussa 6.3.

### 5 Avoin lähdekoodi ja hankintaprosessi

Avoimeen lähdekoodiin ja hankintaprosessiin liittyvät erityiskysymykset voidaan jakaa seuraavasti:

1. Avoin lähdekoodin ohjelmistoihin liittyvät erityispiirteet esiselvitysvaiheessa
2. Avoin lähdekoodin vaihtoehdon mahdollistavan tarjouspyynnön tekeminen
3. Tarjousten arvioimisessa huomioon otettavat asiat, kuten toimittajariippuvuuden riskin arviointi

#### 5.1 Toimintamalleja esiselvitysvaiheeseen

##### 5.1.1 Perehdy avoimien ohjelmien mahdollisuuksiin ja erityispiirteisiin

Julkisissa hankinnoissa ajatellaan käytännössä myös hankintaprosessiin itseensä käytettäviä resursseja. Hankinta pyritään tekemään tasapuolisesti ja niin, että todennäköisyys oikeusriidalle minimoituu. Resurssien rajallisuuden vuoksi hankinnat tehdään usein jo aiemmin hyväksi havaitulla tavalla. Avoimien ohjelmien tapauksessa tähän liittyy haaste: totutut mallit ovat kehittyneet miltei yksinomaan suljettujen ohjelmistojen hankinnasta opittujen kokemusten kautta. Avoimiin ohjelmiin, niiden lisensointiin ja yritysten toimintakulttuuriin liittyy kuitenkin suljettujen ohjelmien maailmasta eroavia tekijöitä, jotka on syytä ottaa huomioon.

##### 5.1.2 Vaatimusmäärittely ja esiselvitysvaihe ovat olennaisia onnistuneen ohjelmisto- tai järjestelmähankinnan kannalta

Vaatimusmäärittelyn ja esiselvitysvaiheen kohdalla on syytä huomata, että monilta osin valinta hankittavan tuotteen tyypistä tehdään jo silloin. Ongelmaa voisi kuvata rinnastamalla valinta tilanteeseen, jossa halutaan palkata urheilija. Tarjolla on kuulantyoöntäjä ja maratoonari. Kun määritellään vaatimuksia eli ominaisuuksia, määritellään käytännössä samalla myös se kumpi valitaan. Olennaista on, kuinka hyvin määritely tuote tai hankinta soveltuu suunniteltuun käyttöön.

Vaatimusmäärittelyn ja esiselvityksen merkitys korostuu kun otetaan huomioon, mitä Valtion hankintakäsikirja toteaa hankinnoista:

*10. Tarjousten käsittelyssä tarkistetaan ensin tarjoajan soveltuvuus ja tarjouksen tarjouspyynnön mukaisuus. Sen jälkeen tarjoukset vertaillaan. Kokonaistaloudellisen edullisuuden vertailu ja sen mahdollinen pisteytys on perusteltava yksityiskohtaisesti.*

Vertailuperusteet ovat lukittuja siinä vaiheessa, kun julkinen tarjouspyyntö lähtee. Sen jälkeen arviointiperusteita ei enää voida muuttaa. Valmistelutyö on siksi tehtävä kunnolla.

Erilaiset sopimusehdot hankaloittavat eri tarjousten vertailua. Helppointa on, jos kaikki tarjoajat velvoitetaan noudattamaan tilaajan määrittelemiä sopimusehtoja.

#### 5.2 Seitsemän askelta avoimen lähdekoodin ohjelmiston mahdollisuuden arviointiin

Ohjelmistoa hankittaessa tärkeintä on määritellä hankittavan ohjelmiston vaatimukset siten, että se sopii hyvin käyttötarkoitukseensa. Useimmiten paras, mutta myös raskain tapa on määritellä tarkasti ne toiminnallisuudet, mitä ohjelmalla pitäisi olla ja tehdä hyvä vaatimusmäärittely.

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

Joissakin tilanteissa voidaan selvittää huomattavasti edullisemmin seuraavaa seitsemän askeleen hankintaprosessia käyttäen.

1. Selvitetään mitä ohjelman pitäisi pystyä tekemään käytännön tasolla. Tämä voidaan toisinaan tehdä ottamalla verrokiksi tunnettu suljettu ohjelma ja lähteä hakemaan vähintään sen tasoista avoimen lähdekoodin ohjelmaa.
2. Suoritetaan ohjelman koeasennus ja tehdään ohjelmalla niitä tosielämän työtehtäviä, joita sillä tehtäisiin myös tuotantokäytössä. Kokemukset kirjataan ja arvioidaan ohjelmiston soveltuvuus. Monista avoimista ohjelmista, erityisesti palvelinkäyttöön tehdyistä, on internetissä esittelyversio, jolla voi kokeilla käyttöä. Käytetään koeasennuksen yhteydessä internetissä mahdollisesti saatavilla olevia käyttäjäfoorumeita vastausten ja tuen etsimiseen ja arvioi sen laatu. Yritetään etsiä muita ohjelmistoa käyttäviä organisaatioita ja tiedustellaan käyttökokemuksia.
3. Arvioidaan tuen tarve ja sen aiheuttamat lisäkustannukset. Arvioidaan organisaation henkilöstön tarvitsema lisäkoulutus jos siirrytään käyttämään uutta ohjelmistoa.
4. Jos kysymys on palvelinohjelmistosta, selvitetään kannattaako organisaation ylläpitää palvelinta itse vai kannattaako työ ostaa muualta. Sovellusvuokraus ja SaaS-malli (engl. Software as a Service) kannattaa tutkia vaihtoehtoina. Jokin IT-palveluyritys voi myydä ohjelmiston käyttöoikeuden kiinteään kuukausihintaan huolehtien myös ylläpidollisista töistä.
5. Arvioidaan, mitä sellaisia hyötyjä harkitsemasi avoin ohjelmisto tarjoaa, joita suljetut vaihtoehdot eivät voi tarjota. Arvioidaan niiden merkitys käytännön toiminnan kannalta.
6. Käytetään immateriaalioikeuden ja ohjelmistolisenssien asiantuntijaa arvioimaan harkittavana olevan ohjelmiston lisenssin vaikutukset tuotteisiisi, toimintaasi ja liiketoimintaasi. Kannattaa muistaa, että toiminnan laajuus voi muuttua ja ohjelmistojen lisensseillä saattaa olla suuri vaikutus siihen, mikä on mahdollista tai taloudellisesti järkevää tulevaisuudessa.
7. Arvioidaan kokonaisuuden hyödyt, haitat sekä kustannukset. Verrataan niitä muihin vaihtoehtoihin ja tehdään päätös tarjouspyynnön sisällöstä. Päätöksen tulee perustaa mahdollisimman konkreettisiin faktoihin, trendeillä ratsastaminen ei kannata. Jos paljastuu, että jossakin edeltävässä askeleessa ei ole otettu kaikkea tarpeellista huomioon, palaa siihen.

### 5.3 Avoimen ohjelmiston ja suljetun ohjelmiston hankinnan erot

Oheiseen taulukkoon on koottu tärkeimmät avoimien ja suljettujen ohjelmistojen hankintaan liittyvät erot.

Taulukko 5. Avoimen ja suljetun ohjelmiston hankinnan vertailu.

	<b>Avoim ohjelmisto</b>	<b>Suljettu ohjelmisto</b>
<b>Hankinnan sisältö</b>	Koska ohjelmisto on lähtökohtaisesti vapaasti saatavilla, hankinnan kohde on avoimeen ohjelmistoon kohdistuvat palvelut. Käytön laajuus ei vaikuta hintaan.	Käyttöoikeus tiettyyn määrään asennettuja ohjelmiston kopioita tai tiettyyn käytön laajuuteen. Hinnanmuodostus on yleensä sidottu käytön laajuuteen.
<b>Keneltä hankitaan</b>	Keneltä tahansa toimittajalta, joka tarjoutuu toimittamaan ratkaisun kyseisellä ohjelmistolla. Toimittajia voi olla monia. Tavallisesti ohjelmistot on ladattavissa verkosta, jolloin myös omatoiminen käyttöönotto on mahdollinen.	Käyttöoikeus ostetaan ohjelmiston valmistajalta tai valtuutetulta jälleenmyyjältä.
<b>Kuka vastaa tuesta</b>	Tuesta vastaa yleisesti järjestelmän toimittaja, jos on kyse ostetusta palvelusta. Tuotteiden osalta tuen toimittajan voi kilpailuttaa vapaasti. Tukea saa myös avoimilta foorumeilta tai erikoistuneilta kaupallisilta tukiorganisaatioilta.	Useimmiten jälleenmyyjiä, joilla on tukiorganisaatio. Vaikeissa teknisissä kysymyksissä sen on kuitenkin käännyttävä ohjelmiston valmistajan puoleen.
<b>Miten korjaukset</b>	Ohjelmiston toimittaja toimittaa. Jos näin ei tapahdu, asiakas voi itse tehdä tai teettää	Ohjelmiston toimittaja toimittaa. Jos näin ei tapahdu, asiakas ei voi tehdä korjausta

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

<b>ohjelmistoon tehdään?</b>	korjauksen.	omatoimisesti. Yleensä vain ns. escrow-sopimusehdon laukeaminen mahdollistaa pääsyn lähdekoodiin ja oikeuteen muuttaa sitä.
<b>Integrointi muihin tuotteisiin</b>	Integroitavissa muihin avoimiin ohjelmistoihin. Integrointityön määrä riippuu mm. ohjelmistojen teknisistä toteutuksista ja dokumentaatioista. Integrointi suljetun ohjelmiston kanssa riippuvainen suljetun ohjelmistovalmistajan tahdosta avata ohjelmistonsa rajapintoja.	Integrointi saman valmistajan ohjelmistoihin ja järjestelmiin yleensä saumatonta. Muissa tapauksissa tapauskohtaista riippuen määritellyistä rajapinnoista ja lisenssiehdoista.

### 5.4 Esiselvitysvaiheessa avoimiin ohjelmistoihin liittyvät erityispiirteet

Esiselvitysvaiheessa on lähtökohtana seuraava tilanne:

- On tieto siitä, mitä ohjelmistoja organisaatiolla on jo olemassa
- Uuden hankinnan tarve pohjautuu uusiin tarpeisiin ja sen pitää nivoutua jo olemassa olevaan
- Näin ollen on käsitys siitä tarpeesta, joka uuden ohjelmiston tai ohjelmistokokonaisuuden pitää täyttää

Esiselvitysvaiheessa on mahdollista joustavasti vertailla erilaisten toteutustapojen ominaisuuksia ja kustannuksia. Samalla voidaan kartoittaa mahdollisuudet yhteistyöhön ja yhteisiin hankintoihin muiden organisaatioiden kanssa. Vaikka käytössä ei vielä olekaan tarjouksia, hankintojen hintojen suuruusluokka on kuitenkin pystyttävä selvittämään, muun muassa kilpailutusprosessin laajuuden arvioimisen takia.

Parhaan vaihtoehdon löytämiseksi pyritään kuvaamaan tarvittava ohjelmisto tai ratkaisu mahdollisimman prosessi- ja tarvelähtöisesti. Näin säilytetään mahdollisuus ratkaista asia tavoilla, jotka muuten voivat rajautua määrittelijän mielikuvituksen rajojen tai tietojen rajallisuuden takia pois. Olennaisinta on pyrkiä kuvaukseen, joka ei ole pelkästään näennäisesti etäännytetty kuvaus yleisimmästä ratkaisutavasta. Tämä periaate on hyödyllinen, valittiinpa avoin tai suljettu ohjelmisto, koska näin mahdollistetaan totutusta poikkeavat ratkaisut.

Ohjelmistohankintaa tekevän kannalta avoimien ohjelmistojen maailma on hankala, koska toimintatavat ja liiketoimintamallit ovat jossain määrin erilaiset kuin suljettujen ohjelmien puolella. Esimerkiksi TIEKEN *Onnistunut julkinen ICT-hankinta* -julkaisun ilmaisu ”katsaus markkinoilta löytyviin ratkaisuihin” herättää kysymyksen: mitä käsite ”markkinat” tarkoittaa avoimien ohjelmien tapauksessa? Mahdollisia vastauksia on äärimuodoissaan kaksi:

- ”markkinat” sisältää kaikki muodossa tai toisessa saatavilla olevat ohjelmistot;
- ”markkinat” sisältää vain sellaiset ohjelmistot, joita kaupalliset toimijat tarjoavat.

Ero on olennainen siksi, että on olemassa paljon käyttökelpoisia avoimen lähdekoodin ohjelmistoja, joita mikään ohjelmistotoimittaja ei kuitenkaan Suomessa tue tai toimita. Jos organisaatiossa on riittävästi tietoteknistä osaamista, tällaisen ohjelmiston valinta voi olla perusteltua. Tällainen tapaus voi olla esimerkiksi silloin, kun ohjelmisto on tehty johonkin harvinaiseen käyttötarkoitukseen. Jälkimmäinen tulkinta on realistinen niiden organisaatioiden kannalta, joilla ei ole omaa tietoteknistä osaamista tai jotka edellyttävät takaajan ohjelmiston toimivuudelle.

Avoimen lähdekoodin käyttöä tulee verrata tasavertaisena vaihtoehtona muihin ratkaisuihin käyttäen kokonaistaloudellisuutta kriteerinä. Koko elinkaaren kustannusten laskeminen on tosin haasteellinen tehtävä. Lisäksi avoimen lähdekoodin myötä tulee uusia kysymyksiä, esimerkiksi:

- Mikä arvo on sillä, että jos saman ratkaisun käyttöä organisaatiossa laajennetaan, ei lisenssimaksujen määrä kasva?

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

- Mikä arvo on sillä, että käyttäjäkunnan laajentuminen ei aiheuta uutta tarjouskilpailua?
- Mikä arvo on sillä, että mikäli tarjouspyynnössä mainittuja toiminnallisuuksia halutaan oleellisesti laajentaa, niin laajennusta ei tarvitse kilpailuttaa?
- Mikä arvo on sillä, että organisaatio voi antaa ohjelmiston vastikkeetta henkilöstönsä tai vaikka toisen valtionlaitoksen käyttöön?

Esimerkki:

Virasto A on kilpailuttanut itselleen projektinohjausjärjestelmän.

- Virasto B ja kunta C haluaisivat saman järjestelmän.
  - **Suljettu lähdekoodi:** Niiden tulee kilpailuttaa järjestelmänsä hankinta ja tuleva järjestelmä määräytyy kilpailutuksen perusteella.
  - **Avoimien lähdekoodi:** Ne voivat ottaa järjestelmän käyttöönsä ja kilpailuttavat mahdollisesti tarvittavan työn.
- Virasto A haluaa laajentaa projektinohjausjärjestelmää dokumentinhallinnan ja arkistoinnin suuntaan.
  - **Suljettu lähdekoodi:** Dokumentinhallinta ja arkistointi kilpailutettava ja tuleva järjestelmä määräytyy kilpailutuksen perusteella.
  - **Avoimien lähdekoodi:** Järjestelmän toiminnallisuuksia voidaan laajentaa kilpailuttamalla tehtävä työ.

Avoimien lähdekoodien ohjelmistojen tarjoamien vaihtoehtojen selvittäminen on luonteeltaan erilainen prosessi kuin suljettujen ohjelmistojen hankinta. Avoimien ohjelmien kohdalla yksi olennainen ongelma liittyy niiden suljettujen vastineitaan huonompaan tunnettuuteen. Pääsyy tähän on se, että kun yksi yhtiö ei omista ohjelmistoa ja useampi toimittaja voi tarjota samaa ohjelmistoa, niin yhdelläkään toimittajalla ei ole intressiä tehdä suuria markkinointiponnistuksia tuotteen eteen.

Hyvä esimerkki on WWW-palvelinohjelmisto Apache HTTP Server, jolla on merkittävä markkinaosuus maailman WWW-palvelimista. Vaikka Suomessakin Apache HTTP Server on yksi vakioratkaisuista WWW-palvelimista toteutettaessa, suomalaisessa lehdistössä ei ole näkynyt kyseisen ohjelmiston mainoksia. Ohjelmiston suosio perustuu sen teknisistä ominaisuuksista ammattilaisten keskuudessa leviävään tietoon ja osaltaan myös siihen, että ohjelmistoa käyttävät harrastajat kasvavat ammattilaisiksi jatkaen sen käyttöä.

Näistä tekijöistä johtuen avoimien lähdekoodien ratkaisuihin tietoa täytyy aktiivisesti hankkia, jotta sitä saisi yhtä paljon verrattuna perinteisiin kaupallisiin ohjelmistoihin. Parhaita tiedonlähteitä avoimien ohjelmistojen kohdalla ovat toiset samaa ohjelmistoa käyttävät organisaatiot - mitä samankaltaisempi organisaatio sitä parempi. Tehokas tapa on etsiä viitteitä ohjelmiston käyttäjäorganisaatioista ohjelmiston käyttäjien kehittäjä- ja tukifoorumeilta. Ne ovat yleensä julkisia koska ohjelmiston kehityskin tapahtuu yleensä julkisesti. On kuitenkin syytä muistaa, että nämä käyttäjät ovat jo sitoutuneet valinnallaan ohjelmiston käyttöön eikä ole lainkaan varmaa kuinka hyvin he tuntevat muita vaihtoehtoja. Siksi heille esitettävät kysymykset kannattaa esittää operationalisoidussa muodossa: esimerkiksi ”Miten hoidatte tiedon vaihtamisen rajapinnassa x kyseisellä ohjelmalla?” sen sijaan, että kysyttäisiin ”Toimiiko se teillä hyvin x:n kanssa?”

Avoimista ohjelmistoista löytyy myös koottua, luokiteltua tietoa. Tietolähteitä ja koottuja hakemistoja on listattu liitteessä 1. Jos tällaiset hakemistot ovat sisällöllisesti liian vaikeita tai haastavia, kannattaa organisaation antaa ohjelmiston valinta jonkin konsultoivan osapuolen tehtäväksi. Erityisesti tällaisessa tapauksessa kannattaa kartoittaa yhteistyömahdollisuudet muiden samankaltaisten organisaatioiden kanssa.

Tässä yhteydessä kannattaa miettiä, olisiko vastaavalla ratkaisulla käyttöä muissa julkisissa organisaatioissa. Hyvä esimerkki on kuntien yhdistämiset, joissa avoimilla ohjelmistoilla toteutetusta ratkaisusta on se etu, että se voidaan lisenssiehtoja noudattaen ”monistaa” käyttöön ilman ohjelmistoista koituvia lisäkustannuksia.

Jos ”monistettavuuden” vaatimus tai arviointiperuste esitetään tarjouspyynnössä, se tulee esittää yleisellä tasolla viittaamatta tiettyihin ohjelmistolisensseihin. Näin vertailu on tasapuolinen sekä perinteisten suljettujen ja avoimien lähdekoodien ratkaisujen tarjoajille. Suositeltava muotoilu on esimerkiksi: ”Tilaajalla

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

on oikeus muokata ohjelmistoa tarkoituksiinsa sopivaksi ja niin halutessaan jakaa ohjelmistoa lähdekoodeineen eteenpäin kolmansille osapuolille ilman rojalteja tai muita maksuja.”

Hankintalaki ei erikseen mainitse tietokoneohjelmia tai käyttöoikeuksia vaan puhuu tavaroiden tai palveluiden hankkimisesta. Alankomaissa toimivan Nederland Open in Verbinding (NOiV) -järjestön julkaisemassa *The acquisition of (open-source) software* -oppaassa tulkitaan EU-lainsäädäntöä siten, että itse tehty maksuttomien (gratis) ohjelmien käyttöönotto ei ole lain tarkoittama hankinta lainkaan. Organisaatio voi näin ollen tehdä päätöksen ottaa käyttöön tietty maksuton ohjelmisto ja kilpailuttaa käyttöönottoon liittyvät palvelut erikseen.

On huomattava, että hankintayksikön on ongelmallista käsitellä rinnakkain vastikkeetta tapahtuvaa itse tehtävää avoimen ohjelmiston käyttöönottoa ja sille vaihtoehtoista suljetun ohjelmiston ohjelmistohankintaa. Hankinnan kohde on vaihtoehtoisissa eri ja kahden eri hankinnan kohteen vertailua samassa kilpailutuksessa on hankala tehdä.

Käytännössä avoimen lähdekoodin ohjelmistoja harkitsevalla on yleisten hankintaperiaatteiden ja hankintalain mukaan kaksi mahdollisuutta:

1. Valitaan avoimen lähdekoodin ohjelmisto itse ja otetaan se käyttöön omin toimin. Tällöin ohjelmisto ja sen käyttöönotto ei itsessään ole hankintalain tarkoittama hankinta. Hankinnan kohteeksi voi kuitenkin muodostua avoimeen ohjelmistoon kohdistuva työ, esimerkiksi ohjelmistoon hankittavat tukipalvelut. Hankintayksikkö voi tehdä päätöksen maksuttoman ohjelmiston käyttöönotosta ja sen jälkeen kilpailuttaa esimerkiksi käyttöönottoon liittyvät asennus- ja tukipalvelut. Näin on menetelty esimerkiksi Oikeusministeriön OpenOffice.org-käyttöönotossa, jossa suurin osa noin 10 000 työasemasta on varustettiin avoimen lähdekoodin OpenOffice.org-toimisto-ohjelmistolla.
2. Tehdään julkinen tarjouspyyntö, jossa hankinnan kohde on ohjelmiston, käyttöönoton ja integroinnin muodostama kokonaisuus. Tällöin hankitaan ohjelmiston lisäksi myös käyttöönottoon liittyvät toimet ja esimerkiksi integrointi jo olemassa olevaan tietojärjestelmään ja laaditaan tarjouspyyntö siten, että siihen voidaan vastata myös avoimilla ohjelmistoilla. Tällöin hankinta tehdään yleisten hankintaperiaatteiden mukaan.

Jos esiselvitysvaiheessa tullaan siihen tulokseen, että jokin avoin ohjelma täyttää tarpeet ja on kokonaisuutena tarkastellen järkevä ja kokonaistaloudellinen vaihtoehto, käyttöönotto voidaan tehdä ilman kilpailuttamista. Pilotointi on toki järkevää, jotta mahdolliset ongelmat tunnistetaan.

Taulukossa 6 on vertailtu ohjelmistojen omin toimin tehtyä käyttöönottoa ohjelmistojen hankintaan.

Taulukko 6. Ohjelmiston oman käyttöönoton ja hankinnan erot.

<b>Ohjelmiston oma käyttöönotto</b>	<b>Ohjelmiston hankkiminen</b>
Markkinoiden ja vaihtoehtojen kartoittamisella suuri merkitys.	Vaatimusmäärittely keskeisessä asemassa.
Sopivan ohjelmiston etsiminen ja lataaminen vaatii erityisosaamista henkilöstöltä.	Tarjouspyynnön valmisteleminen vaatii paljon tietoa. Tarjoajat antavat osan siitä.
Mahdolliset palvelut pitää kilpailuttaa erikseen.	Ohjelmistot ja palvelut voidaan sisällyttää samaan tarjouspyyntöön.

### 5.5 Avoimiin ohjelmistoihin liittyvät erityiskysymykset tarjouspyyntöä laadittaessa ja arvioitaessa

Hankinnan kohteen kuvaus tulee tehdä niin riippumattomasti, että se ei sisällä tarpeettomia viittauksia tiettyihin ohjelmistoihin tai teknologioihin. Tämä pätee myös avoimiin ohjelmistoihin liittyviin käsitteisiin. ”Avoin lähdekoodi” ja ”vapaa ohjelma” eivät ole sisällöllisesti kovin tunnettuja eivätkä ne ole tarkkarajaisia tai yksiselitteisiä.

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

Sama pätee erilaisiin avoimien ohjelmien lisensseihin. Sen sijaan, että edellytettäisiin hankittavaa ohjelmistoa esimerkiksi GPL-lisenssoituna, voidaan kirjoittaa GPL-lisenssin ehdot auki halutussa määrin. Voidaan esimerkiksi edellyttää, että lähdekoodi tulee toimituksen mukana ja että hankintayksikkö tai kuka tahansa muu saa hyödyntää lähdekoodia rajoituksetta, kunhan julkaisee tekemänsä muutokset, mikäli ohjelmistoa levitetään. Näille vaatimuksille tulee kuitenkin olla perustelut.

Tarjousten vertailuperusteisiin kannattaa kiinnittää huomiota. Erityisesti sekahankinnoissa ongelmaksi saattaa muodostua, miten hinnan osuutta voidaan painottaa siten, että järkevä hankintalain mukainen kilpailutus saadaan tehtyä.

Hankintalain mukainen lähtökohta tarjouksia arvioitaessa on se, että valinta tulee tehdä joko halvimman hinnan tai kokonaistaloudellisen edullisuuden perusteella. Jos valinta tehdään kokonaistaloudellisuuden perusteella, vertailuperusteet on ilmoitettava. Jos näin ei menetellä, tulee valita hinnaltaan halvin.

Edelleen hankintakäsikirja toteaa vertailukriteereistä:

*Vertailuperusteet tulee yksilöidä riittävästi, koska ne eivät saa antaa hankintayksikölle rajoittamatonta harkintavaltaa vertailtavien seikkojen sisällön suhteen. Tarjoajien tulee saada tietää, millä seikoilla on merkitystä vertailussa. Tästä syystä kunkin vertailuperusteen osalta on ilmoitettava ne yksittäiset seikat, joihin kyseisen vertailuperusteen vertailussa kiinnitetään huomiota. Laatukäsité tulee yksilöidä ja jakaa osatekijöihin, ja käyttö- tai elinkaarikustannusten osalta on ilmoitettava arvioidut käyttömäärät ja muut kustannusten laskemisessa tarvittavat tiedot. Hankintayksikön tulee korostaa, että tarjoajan on toimitettava tarjouksessaan vertailun tekemisessä tarvittavat selvitykset, koska vertailussa ei voida käyttää muita tietoja kuin tarjouksesta ilmeneviä tietoja.*

Kaikkien vertailuperusteiden on siis oltava mitattavissa yksiselitteisesti. Sinänsä erilaisten kriteerien suhteen organisaatiolla on liikkumavapautta, kunhan ne pystytään perustelemaan.

Vertailuperusteista hankintakäsikirjassa todetaan:

- Ilmoitettava kansallisissa hankinnoissa tärkeysjärjestyksessä. Myös painoarvojen käyttäminen on mahdollista.
- EU-hankinnoissa vertailuperusteiden suhteellinen painotus eli ns. painoarvojen tai niiden vaihteluvälin ilmoittaminen on pakollista. Jos näiden ilmoittaminen ei ole perustellusta syystä mahdollista, vertailuperusteet EU-hankinnoissa on ilmoitettava tärkeysjärjestyksessä.
- Käytettäessä painoarvoja tulisi hinnan painoarvon yleensä olla vähintään noin 35-40 %. Massatavaran osalta hinnan painoarvo voi puolestaan olla edellä esitettyä korkeampikin. Jos hinnan painoarvoksi harkitaan yli 50 %, on toinen vaihtoehto tällöin käyttää valintaperusteena vain hintaa ja tuotteen huolellisella vaatimusmäärittelyllä varmistaa tarpeet täyttävän tuotteen valinta. Yleensä yksittäisen vertailuperusteen painoarvon tulisi olla vähintään 20 %, jotta sillä olisi todellista merkitystä.
- Palveluhankintojen osalta hinnan painoarvo voi poikkeustapauksissa olla alle 35 %. Hinnan liian vähäinen painottaminen voi kuitenkin merkitä hankinnalle huomattavaa lisähintaa, varsinkin jos yhden hankintayksikön laadullinen etumatka muihin tarjoajiin muodostuu suureksi tai jos tarjousten väliset hintaerot ovat suuret.
- Mitä merkittävämpi palvelu on hankintayksikölle strategisesti, sitä tärkeämpää on painottaa laatutekijöiden merkitystä. Vertailuperusteen painoarvonsuuruudella osoitetaan tarjoajille, kuinka tärkeäksi hankintayksikkö arvostaa kyseisen seikan tarjousten vertailussa.
- Laajoissa hankinnoissa pisteytysperusteet ja vertailusuunnitelma on syytä laatia aina ennen tarjousten vertailua, jos pisteytysperusteita ei ole ilmoitettu tarjouspyynnössä.

Näillä perusteilla voidaan laatia hankinnan arvioinnille pisteytysperusteet. Hankinnan laatuun liittyviä kriteereitä on luokiteltu seuraavassa.

### 5.6 Kelpoisuuskriteereihin kuuluvat laatutekijät

#### 5.6.1 Referenssit ja henkilöstön pätevyys

Referenssien ja henkilöstön osaamisen määrittelyssä on otettava huomioon avoimen lähdekoodin maailman erilainen toimintakulttuuri (mm. hajautettu ohjelmistokehitys) ja esittää myös sellaisia kysymyksiä, jotka mittaavat tarjoajan aktiivisuutta suhteessa tarjottavaan ohjelmistotuotteeseen.

Kysymyksillä pitäisi vähintään pystyä erottamaan tarjoajista ne, jotka todella osallistuvat ohjelmiston kehittämiseen tavalla tai toisella niistä, jotka hiljaisesti seuraavat kehitystä ja tarjoavat ohjelmistoon perustuvia ratkaisuja. Kehitykseen osallistuvilla yrityksillä on kokemuksesta karttunut laajempi ohjelmiston osaamistaso ja tiiviimmät yhteydet kehittäjäyhteisöön. Vaatimuksissa esitettyjen yhteiskuntavastuuta mittaavien edellytysten tavoin voidaan kysyä asioita, jotka mittaavat yrityksen vastuunottoa avoimen lähdekoodin yhteisössä.

Esimerkkikysymyksiä:

- Onko yrityksenne osallistunut tarjotun ohjelmiston kehittämiseen luovuttaen siihen lähdekoodia? Millä tavoin ja kuinka paljon?
- Onko yrityksenne esittänyt ohjelmiston kehittäjille kehitys- tai korjausehdotuksia? Mitä ja milloin? Miten ehdotuksille kävi?
- Osallistutteko ohjelmiston kehittämiseen liittyvään organisoituun toimintaan? Mihin ja missä roolissa?

On oleellista tietää, miten tarjoajana olevan aktiivisen toimijan kehittämis ehdotuksiin on suhtauduttu. On hyvä merkki, jos mielekkäät kehittämis ehdotukset on toteutettu ja tarjotut koodimuutokset hyväksytyt. Mikäli näin ei ole, toimija ei välttämättä ole vakiinnuttanut asemaansa ohjelmiston kehittäjäyhteisössä tai kehittämis ehdotukset tai -panostukset eivät ole olleet riittävän hyviä.

Tällaisia kysymyksiä voidaan kuitenkin kritisoida kahdesta syystä.

1. Ne mittaavat asioita, jotka ovat nimenomaan tyypillisiä avoimen lähdekoodin ohjelmistotuotannolle, kuten kehitysehdotusten tai korjausten tekemistä.
2. Oma suljettua ohjelmistoaan tarjoava yritys saa näistä täydet pisteet.

Tämän vuoksi ehdotetut kysymykset toimivatkin vain vertaillen tarjouksia avoimen lähdekoodin ratkaisusta. Tilanne osoittaa myös sen, millaisiin vaikeuksiin ”sekakilpailutukset”, joihin voidaan vastata sekä avoimilla että suljetuilla ohjelmistoilla, saattavat johtaa.

##### 5.6.1.1 Referenssit

Toimittajalta odotetaan näyttöä aiemmista vastaavanlaisista toteutuksista. Toimittajan eduksi katsotaan kokemus ratkaisukuvauksen laatimisesta ja toteutuksesta sekä alustaratkaisun räätälöinnistä vastaamaan asiakkaan tarpeita. Samoin eduksi katsotaan toimittajan kokemus järjestelmien integroinnista.

Referenssien kohdalla kannattaa ottaa huomioon, että avoimia ohjelmistoja voi tarjota kuka vain. Ostajaa voi kiinnostaa esimerkiksi se, mitkä samankaltaiset organisaatiot käyttävät kyseistä ohjelmistoa vaikka toimitus ei olisikaan tarjoajan toteuttama. Näistä tiedoista on hyötyä kun selvitetään esimerkiksi soveltuvuutta ostajan oman hallinnon alan käyttöön. Toisaalta tarjoajien oma osaaminen kyseisen ohjelmiston tapauksessa on syytä selvittää.

Esimerkkikysymyksiä:

- Mitkä ovat yrityksenne toimitusreferenssit tarjoamanne ohjelmiston osalta?
- Onko mielestänne muita kannaltamme vertailukelpoisia kyseisen ohjelmiston tai ratkaisun käyttäjiä?

### 5.6.1.2 Henkilöstön pätevyys

Tarjoajan tulee esittää työhön käytettävät henkilöstöresurssit ja nimetä työstä vastaava henkilö. Resurssien on oltava koulutukseltaan ja kokemukseltaan projektiin sopivia.

Tämän arvioinnissa avoimien ohjelmistojen kohdalla kannattaa ottaa huomioon henkilöstön koulutustason lisäksi myös osallistuminen erilaisiin avoimiin ohjelmistohankkeisiin, sillä ne kartuttavat kokemusta.

Tässä kohtaa toimittajan tulisi pystyä nimeämään toimitukseen osallistuvat henkilöt, jotta voidaan arvioida sekä henkilöstöresurssien laatua että riittävyttä.

### 5.6.1.3 Oikeus levittää ohjelmaa

Esiselvitysvaiheessa tehty harkinta muista mahdollisista vastaavan järjestelmän tarvitsijoista kannattaa huomioida. Jos niitä on, oikeus muokata ja levittää ohjelmaa edelleen voidaan nimetä pakolliseksi vaatimukseksi.

## 5.7 Tarjousten arviointiin kuuluvat laatutekijät

### 5.7.1 Kokonaisratkaisun soveltuvuus

Arviointitapa: verrataan tarjousta tarjouspyynnössä esitettyihin vaatimuksiin, visioon ja tahtotilaan. Arvioinnissa ei ole merkittäviä avoimiin ohjelmistoihin liittyviä ongelmia. Jos määrittely ja tarjous on tehty siten, että se ei sisällä viittauksia tiettyihin käyttöjärjestelmiin tai ohjelmistoihin, arviointi on suoraviivaista toiminnallisuuden arviointia.

#### 5.7.1.1 Hinta

Laatuvertailun jälkeen lasketaan hintapisteet tarjouspyynnössä annetun kaavan mukaisesti.

Hintoja arvioitaessa on syytä ottaa huomioon, että avoimilla ohjelmistoilla ei tyypillisesti ole maksullisia lisenssejä. Niille perustuvan tarjouksen hinnasta suurin osa koostuu yrityksen omasta työstä.

#### 5.7.1.2 Toimitusvarmuus

Suljettujen ohjelmien kohdalla voidaan kysyä toimittajalta, kuinka pitkä elinkaari ja tuki ohjelmistolle taataan. Niiden kohdalla on syytä myös arvioida ja pyrkiä minimoimaan toimittajariippuvuudesta aiheutuva riski. Jos toimittaja poistuu markkinoilta, asiakkaan tilanteesta voi tulla hankala, ellei olla suojauduttu escrow-järjestelyllä, jossa tiettyjen ehtojen täytyessä ostaja saa ohjelmiston lähdekoodin haltuunsa.

Avoimien ohjelmistojen kohdalla vastaavaa riskiä ei ole. Niiden kohdalla toimittajalta on hyvä pyytää lähdekoodit osana kokonaistoimitusta. Tällöin on mahdollista ryhtyä toimenpiteisiin heti, jos toimittajan kanssa tehtävässä yhteistyössä ilmenee vakavia ongelmia.

Suljetun ohjelmiston ostaminen lähdekoodeineen ja tekijänoikeuksineen on myös mahdollista. Esimerkiksi EU:n tasolla tullin käyttämissä järjestelmissä on joissain tapauksissa menetelty näin.

Tarjoajien toimitusvarmuuden arvioiminen voi olla avoimen lähdekoodin ohjelmia tarjoavien yritysten kohdalla ongelmallista. Mittarina on perinteisesti käytetty yrityksen vakavaraisuutta ja sen mittaamisessa suurena liikevaihtoa. Pelkkää työtä tai konsultointia myyvän yrityksen liikevaihto jää alemmalle tasolle kuin esimerkiksi sellaisen, joka tekee samalla laite- ja lisenssikauppaa.

Avoimen lähdekoodin yritykset myyvät lähes pelkästään omaa työtään ja siksi keskisuuren yrityksenkin liikevaihto voi jäädä huomattavasti alemmalle tasolle kuin työntekijämäärältään huomattavasti pienemmän, tuotteiden myynnillä ansaitsevan yrityksen. Lisäksi pienikin yritys voi jollain erityisalalla, esimerkiksi roskapostisuodatinjärjestelmissä, olla alansa huippu. Liikevaihtoa olisikin hyvä käyttää harkiten kriteerinä ja keskittyä projektiorganisaation kvalitatiivisiin ominaisuuksiin, kuten projektiin nimettävien henkilöiden

osaamiseen. Jos liikevaihtoa käytetään mittarina, siitä pitäisi käyttää vain sitä osaa, joka koskee kyseessä olevan hankinnan luonteista liiketoimintaa.

### 5.8 Ohjelmiston hankkiminen avoimen lähdekoodin lisenssillä

Ohjelmiston hankinta tulee tehdä pääsääntöisesti siten, että hankittava ohjelmisto voidaan jakaa hallinnon ja kansalaisten keskuudessa. Parhaiten ohjelmiston jakamisen edellytykset turvataan käyttämällä avoimen lähdekoodin lisenssejä. Tähän on kolme vaihtoehtoa:

1. Määritellään tarjouspyynnössä, että ohjelmisto toimitetaan avoimen lähdekoodin oikeudet antavilla lisenssiehdoilla. Mitään yksittäistä avoimen lähdekoodin lisenssiä ei kannata mainita nimellä, vaan listataan tarjouspyyntöön avoimen lähdekoodin määritelmästä tärkeimmät oikeudet ja mahdollisesti lisäksi joitakin toivotun lisenssin erityispiirteitä. Esimerkiksi:
  - o Oikeus ajaa ohjelmaa millä tahansa laitteistolla ja käyttää sitä mihin tahansa tarkoitukseen.
  - o Oikeus jaella ohjelmistoa lähdekoodi- ja binäärimuodossa. Kaikilla käyttäjillä on yhtäläiset oikeudet ohjelmistoon.
  - o Oikeus muokata ohjelmistoa haluamallaan tavalla ja jaella muutettua ohjelmistoa.
  - o Oikeus käyttää ohjelmaa minkä tahansa ohjelmiston kanssa. Lisenssi ei saa rajoittaa sitä, minkälaisien ohjelmistojen kanssa sitä käytetään olivatpa ne suljettuja tai avoimia. Lisenssi ei saa myöskään olla riippuvainen isommasta ohjelmistokokonaisuudesta, jonka osana ohjelma toimitetaan.
2. Määritellään tarjouspyynnössä, että määräysvalta ohjelmiston lisensoinnista on hankintayksiköllä.
3. Hankintaan ohjelmisto tekijänoikeuksineen niin, että oikeudet siirtyvät hankintayksikölle (kallein vaihtoehto).

Kun avoimeen ohjelmistoon teetetään muutoksia, olisi kehitystulokset hyvä saada avoimen lähdekoodin lisenssillä ja julkaista alkuperäisen ohjelmiston projektiin ja muiden hyödynnettäviksi. Tästä on useita etuja:

- Muut saavat kehitystulokset käyttöönsä ja voivat testata niitä ja kehittää edelleen. Tuote voi kehittyä entistä paremmaksi ilman uusia hankintoja.
- Alkuperäinen projekti voi ottaa kehitystulokset ohjelmiston viralliseen jakeluun, jolloin hankintayksikön ohjelmisto on yhteensopiva alkuperäisen ohjelmiston ja sen tulevien päivitysten kanssa. Näin voidaan välttää ohjelmiston haarautuminen alkuperäisestä teoksesta, mikä voi vaikeuttaa ylläpitoa.

## 6 Opas juridisiin aiheisiin

### 6.1 Tekijänoikeus ja avoimen lähdekoodin ohjelmistolisenssit

#### 6.1.1 Nykytila

Tietokoneohjelmat on suojattu tekijänoikeudella. Tämä tarkoittaa, että ohjelmien käyttämiseen ja kopioimiseen tarvitaan aina lupa tekijänoikeuden haltijalta.

Perinteisissä kaupallisissa ohjelmistoissa käyttö- ja kopiointiluvan saaminen on tapahtunut ostamalla lisenssejä, jotka oikeuttavat käyttämään ohjelmaa tietyissä rajatuissa puitteissa. Tyypillisesti yhtä lisenssiä vastaan ohjelman on voinut asentaa yhteen koneeseen. Joissakin tapauksessa ohjelmistoihin on voinut ostaa myös esimerkiksi kuntakohtaisia lisenssejä, jotka sallivat ohjelmiston asentamisen kaikkiin yksikön koneisiin. Kokonaan oma lukunsa on vielä palvelinkoneilla pyörivät ohjelmistot. Näissä lisenssiehdot perustuvat esimerkiksi yhtäaikaisten käyttäjien maksimimäärään tai koneesta löytyvien prosessorien lukumääriin.

Yhteistä näille kaikille on, että lisenssinsaaja ei saa tyypillisesti muokata ohjelmistoja, vaan muutokset ja korjaukset tulevat alkuperäiseltä ohjelmistoyritykseltä. Koska ohjelmiston lähdekoodi ei tule pääsääntöisesti ohjelmistojen mukana, muokkaaminen olisikin käytännössä lähes mahdotonta.

### 6.1.2 Avoimen lähdekoodin lisenssit

Suurin periaatteellinen ero avoimien ja suljettujen ohjelmien lisensseissä on, että suljettujen ohjelmien lisensseissä tekijänoikeuden suomia mahdollisuuksia käytetään lisenssinsaaajan vapauksien rajoittamiseen ja avoimien ohjelmien lisensseissä tekijänoikeuden suomia mahdollisuuksia käytetään lisenssinsaaajan vapauksien turvaamiseen. Open Software Initiativen (OSI) mukaan avoimen lähdekoodin lisenssien tulee täyttää OSI:n avoimen lähdekoodin määritelmän ehdot, jotka on käyty läpi luvussa 4.

OSI on standardoinut kymmeniä lisenssiä, jotka täyttävät tämän määritelmän. Näistä tärkeimmät esitellään seuraavassa kohdassa. On syytä kuitenkin huomioida, että tekijänoikeus avoimilla lisensseillä julkaistuihin ohjelmistoihin pysyy niiden tekijöillä. Mikäli avointen lisenssien ehtoja ei noudateta, kyse on vastaavasta rikkomuksesta kuin kaupallistenkin tuotteiden kohdalla – tosin sillä erolla, että tilanne on useimmiten paljon helpommin korjattavissa esimerkiksi julkaisemalla ohjelmiston lähdekoodi.

Tavallisessa työpöytäkäytössä avoimien lisenssien yksityiskohtaisiin ehtoihin ei yleensä ole varsinaista tarvetta tutustua. Tärkeintä on tieto ohjelmien vapaasta käyttämisestä ja kopioimisesta. Jos tiettyä ohjelmistoa halutaan kuitenkin lähteä muokkaamaan esimerkiksi useamman kunnan yhteisprojektina, tällöin asiaan on kuitenkin syytä perehtyä huolellisemmin.

### 6.1.3 Lisenssityyppejä

Avoimen lähdekoodin lisenssejä löytyy tällä hetkellä melkein joka lähtöön. Tilanne ei ole tältä osin erityisen toivottava, koska "lisenssitulva" tekee lisenssien hallinnoinnin hankalaksi ohjelmistoyrityksille. Käyttäjälle asti tämä ongelma ei kuitenkaan yleensä heijastu, sillä yhteensovittamiseen liittyvät ongelmalliset kohdat tulevat yleensä vastaan vain niissä tilanteissa, kuin ohjelmia levitetään edelleen organisaation ulkopuolelle.

Avoimet lisenssit kategorisoidaan usein kolmeen eri ryhmään niiden ominaisuuksien mukaan:

- sallivat lisenssit
- vastavuoroisuutta edellyttävät lisenssit
- vahvaa vastavuoroisuutta edellyttävät lisenssit

#### 6.1.3.1 Sallivat lisenssit

Sallivia lisenssejä ovat esimerkiksi MIT-, BSD- ja Apache-lisenssit. Näissä lisensseissä ohjelmaa saa muokata ja levittää vapaasti ja lähdekoodin voi, muttei välttämättä tarvitse, laittaa levitettävän ohjelman mukaan. Nämä lisenssit ovat hyvin lyhyitä ja selviä. Esimerkiksi MIT-lisenssi mahtuu kolmeen kappaleeseen:

The MIT License

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Tunnetuimpia tähän kategoriaan kuuluvia lisenssejä käyttäviä ohjelmistoja ovat BSD-käyttöjärjestelmä, PostgreSQL-tietokanta ja Apache www-palvelin.

### 6.1.3.2 Vastavuoroisuutta edellyttävät lisenssit

Toisen lisenssikategorian muodostavat vastavuoroisuutta edellyttävät lisenssit. Näissä lisensseissä edellytetään, että ohjelmistiedostoihin tehdyt muutokset tulee julkaista tietyin edellytyksin, joista ylivoimaisesti yleisin on ohjelmiston julkinen levittäminen muutoksen jälkeen. Näillä lisensseillä julkaistuja ohjelmistoja saa kuitenkin yhdistää vapaasti muihin, eri lisensseillä tehtyihin ohjelmistoihin esimerkiksi linkittämällä ohjelmakirjastoihin. Yleisimpiä tämän kategorian lisensseistä ovat Eclipse Public License, Mozilla Public License ja GNU Lesser General Public License (LGPL).

Tunnetuimpia tähän kategoriaan kuuluvia lisenssejä käyttäviä ohjelmistoja ovat OpenOffice.org ja Mozilla Firefox.

### 6.1.3.3 Vahvaa vastavuoroisuutta edellyttävät lisenssit

Kolmannen kategorian muodostavat vahvaa vastavuoroisuutta edellyttävät lisenssit. Nämä ovat oikeastaan vastavuoroisuutta edellyttävien lisenssien erityiskategoria, sillä nämä lisenssit edellyttävät lähdekoodin julkaisemista alkuperäisellä lisenssillä kaikissa niissä tilanteissa, joissa alkuperäistä ohjelmistoa muokataan tai siihen yhdistetään uusia elementtejä esimerkiksi linkittämällä.

Tämän hetken kaikista yleisin avoimen lähdekoodin lisenssi, GNU General Public License (GPL), kuuluu tähän kategoriaan. Toisena esimerkkinä voidaan mainita Common Public License.

Näiden lisenssien kohdalla ongelmana on yhteensopivuus muiden avoimen lähdekoodin lisenssien kanssa. Esimerkiksi GPL 2.0 ei ole yhteensopiva Apache 2.0 -lisenssin kanssa, mikä estää niillä lisensoitujen ohjelmistojen yhdistämisen toistensa kanssa. Myöskään GPL 2.0 ja GPL 3.0 eivät ole keskenään yhteensopivia.

Eräitä tunnetuimpia tähän kategoriaan kuuluvia lisenssejä käyttäviä ohjelmistoja ovat Linux-ydin ja MySQL-tietokantajärjestelmä.

Taulukko 7. Ohjelmistojen jakelumallien ja lisenssien ominaisuuksia.

Lisenssi	Vapaa levitys	Vapaa käyttö	Vapaa lähdekoodi	Normaali vastavuoroisuus	Vahva vastavuoroisuus
Suljettu ohjelmisto	-	-	-	-	-
Shareware	x	-	-	-	-
Freeware	x	x (voi sisältää rajoituksia)	-	-	-
BSD, MIT, Apache	x	x	x	-	-
LGPL, MPL ...	x	x	x	x	-
GPL, GCL ...	x	x	x	x	x

### 6.1.4 Vastuukysymykset ja avoimen lähdekoodin lisenssit

Käytännössä kaikissa avoimen lähdekoodin lisensseissä on mukana vastuurajoituslausekkeet, jotka pyrkivät estämään ohjelmien tekijöihin kohdistuvat vahingonkorvausvaatimukset. Tilanne ei tältä osin poikkea

merkittävästi kaupallisista ohjelmistoista. Suomalaisessa oikeuskäytännössä tällaiset vastuurajoitusehdot ovat lähtökohtaisesti sitovia muissa paitsi niissä tilanteissa, joissa on kyse törkeistä laiminlyönneistä tai tahallisista teoista aiheutuvista vahingoista. Asiaa käsitellään yksityiskohtaisemmin seuraavassa luvussa.

### 6.2 Avoimen lähdekoodin riskit ja niiden hallinta

#### 6.2.1 Riskianalyysin osa-alueet

Vaikka avoimen lähdekoodin ottaminen mukaan organisaation ohjelmistopalettiin ei välttämättä tarkoita suurta muutosta aikaisempaan, kyse on kuitenkin muutoksesta, jonka hallinnan tulee olla suunnitelmallista toimintaa. Yksi keskeinen osa tätä prosessia on riskinhallinta. Riskinhallinnasta perinteisemmissä ohjelmistoprojekteissa on saatavilla jo valmiiksi runsaasti erilaista materiaalia niin teknisestä, kaupallisesta kuin oikeudellisesta näkökulmasta. Turhan toiston välttämiseksi tässä osioissa pyritään nostamaan esiin erityisiä riskikohtia avoimen lähdekoodin hankinnoissa ja vertailemaan niiden hallintaa perinteisissä ja avoimen lähdekoodin ohjelmistohankinnoissa.

Avoin lähdekoodi ja erityisesti sen hankinta on ilmiönä varsin uusi monille hankintayksiköille, mikä korostaa hyvän valmistautumisen merkitystä. Todellisten riskien hahmottaminen ei välttämättä ole helppoa tilanteessa, jossa monet aikaisemmin opitut periaatteet on käännetty pääläelleen. Lisäksi, kuten kaiken uuden kohdalla, myös avoimen lähdekoodin riskit voivat alkuun vaikuttaa suuremmilta kuin mitä ne todellisuudessa ovat. Vasta käytännön kautta voidaan oppia, mihin asioihin tulisi kiinnittää huomiota ja mitkä ovat toisarvoisia.

Oikeilla menettelytavoilla avoimen lähdekoodiin liittyvien riskien taso voidaankin säätää useimmiten organisaation haluamalle tasolle. Käytännön toiminnassa erot perinteisiin kaupalliseen tuotteisiin jäävät, jos niin halutaan, hyvin pieniksi. Tosin avoin lähdekoodi tarjoaa monesti jopa joustavamman lähestymisen riskinhallintaan ja tässä mielessä syntyvät erot ovat lähinnä avoimille ohjelmille eduksi.

##### 6.2.1.1 Vaihtoehtoisten toimittajien saatavuus

Yhteen toimittajaan sitoutuminen on yksi suurimmista taloudellisista riskeistä, mitä ohjelmistoprojekteissa joudutaan usein ottamaan. Ostajan vahva lukittautuminen tiettyyn toimittajaan mahdollistaa tuotteen elinkaaren aikana sen kustannustason merkittävän kohottamisen. Vieläkin ongelmallisempia ovat tilanteet, joissa toimittaja lopettaa yllättäen käytössä olevan tuotteen tukemisen ja korvaavaan tuotteeseen siirtyminen ei ole mahdollista esimerkiksi puuttuvan tiedostomuotodokumentaation vuoksi. Riskiä voidaan osittain rajata ns. escrow-ratkaisuilla, mutta turva jää kauas täydellisestä. Escrow-järjestelyä sovittaessa järjestely toteutetaan kuviteltujen ongelmatilanteiden varalta. Käytännössä on kuitenkin vaikea varautua kaikkiin mahdollisiin tilanteisiin ja monesti toimittaja ei myöskään suostu esimerkiksi rajapintojen oikeuksien avaamiseen missään olosuhteissa.

Avoimen lähdekoodin ratkaisuihin näihin ongelmiin törmätään huomattavasti harvemmin. Ensinnäkin avoimen lähdekoodin luonne on omiaan suosimaan aitojen avoimien standardien käyttöä rajapinnoissa ja tiedostomuodoissa. Toiseksi koska lähdekoodi on vapaasti saatavilla ja muokattavissa, hankintayksikkö pystyy ainakin periaatteessa aina etsimään toisen toimittajan ohjelmiston jatkokehittämiselle, jos käytössä olevan toimittajan kanssa ei päästä yksimielisyyteen hinta- tai palvelutasosta.

Kannattaa kuitenkin muistaa, että tarvittavaa tietotaitoa ei välttämättä ole harvinaisempien tuotteiden kohdalla kovinkaan monella toimittajalla, mikä rajaa todellista kilpailua. Samoin hyvin uusien tuotteiden kohdalla osaamista ei välttämättä ole ehtinyt syntymään monelle toimittajalle, ja jos kysyntää ei ole riittävästi, intoa tarvittavien oppimisinvestointien tekemiseen ei välttämättä ole. Ennen syvempää sitoutumista tiettyyn ohjelmistoon tai tekniikkaan on markkina-analyysin tekeminen siis järkevää myös avoimen lähdekoodin tuotteiden osalta.

### 6.2.1.2 Takuut ja vastuut

Vastuukysymysten järjestäminen koetaan usein yhdeksi isoimmista ongelmista avoimen lähdekoodin ohjelmistohankinnoissa. Eihän kukaan voi vastata netistä ladattavasta koodista, jonka jokin epämääräinen yhteisö on tuottanut? Käytännössä tilanne ei välttämättä poikkea lainkaan perinteisistä kaupallisista tuotteista. Paljon riippuu siitä, kuinka laajan takuun ostaja haluaa ja kuinka paljon siitä ollaan valmiita maksamaan.

Ensinnäkin on totta, että jos organisaatio hakee internetistä ja asentaa itse jonkin avoimen lähdekoodin tuotteen, ohjelmiston toiminta ja ongelmat jäävät sen kannettavaksi. Avoimen lähdekoodin projektit tarjoavat toki usein vapaaehtoisvoimin neuvontaa esimerkiksi sähköpostilistoilla ja keskustelualueilla, mutta tarjottavan avun laatu vaihtelee voimakkaasti projektien koon ja resurssien mukaan, tukeen liittyviä oikeudellisia sitoumuksia ei anneta ja useimmissa avoimen lähdekoodin lisensseissä nimenomaisesti sanoudutaan irti vastuusta ohjelmiston toiminnasta (tai toimimattomuudesta). Esimerkiksi suosituissa BSD-lisenssissä on seuraava vastuunrajoitusehto:

*THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.*

Jos hankintayksikkö ottaa käyttöönsä BSD-lisensoidun ohjelmiston, sen kehittäneet ohjelmoijat eivät kannaltaan vastuuta ohjelman toimivuudesta tai sen virheistä tai niiden aiheuttamista vahingoista. On kuitenkin syytä pitää mielessä, että tilanne ei tältä osin juuri poikkea kaupallisista valmisohjelmistoista, joiden lisenssiehdoista löytyy normaalisti vastaavat vastuunrajoitusehdot. Merkittävin ero valmisohjelmistoihin tulee siitä, että avoimen lähdekoodin ohjelmat voidaan ottaa käyttöön ilman lisenssimaksuja ja niiden toimivuutta haluttuun käyttötarkoitukseen voidaan testata ilman rajoituksia ennen varsinaista tuotantokäyttöönottoa. Koska lähdekoodi on saatavilla ja muokattavissa, havaittuihin ongelmiin voidaan myös tarvittaessa ja tietotaidon riittäessä hakea itsenäisesti ratkaisuja.

Mikäli hankintayksikkö haluaa käyttää kolmatta tahoja varmistamaan ohjelmiston toiminnan, tämä on myös mahdollista. Avoimen lähdekoodin ratkaisut voivat joissakin turvallisuuskriittisissä tapauksissa olla koodin escrow-menettelyn ohella ainoita vaihtoehtoja toteuttaa ohjelmiston tarkistus esimerkiksi haittaohjelmien ja piilotettujen toiminnallisuuksien osalta. Kaikki kaupallisten ohjelmistojen rajoitetut koodin katselmoitumahdollisuudet eivät kuitenkaan mahdollista sovelluksen kääntämistä luovutetuista lähdekoodeista. Tällöin lopullista varmuutta ajettavan ohjelman ominaisuuksista ei suljetulla puolella voida saavuttaa.

Avoimen lähdekoodin lisenssit eivät estä lisenssiehtoja parempien takuu- ja vastuu- ja vastuuehtojen tarjoamista. Näin esimerkiksi tarjouspyyntöön voidaan määritellä haluttu takuutaso, johon toimittajan tulee sitoutua. Mikään ei myöskään estä takuupalveluiden erillistä ostamista jo käyttöön otetulle avoimen lähdekoodin ratkaisulle. Ongelmaksi saattaa muodostua sellaisen toimittajan löytäminen, joka on kiinnostunut tämänkaltaisesta sopimuksesta.

Oma lukunsa on koodia koskevien oikeudellisten takuiden antaminen. Hankintayksikön kannalta olisi toivottavaa, että toimittaja lupaisi korvata tai hoitaa mahdolliset tekijänoikeuteen ja ohjelmistopatentteihin liittyvät vaatimukset. Tekijänoikeuden osalta eri avoimen lähdekoodin projekteissa koodin oikeuksien hallinta vaihtelee voimakkaasti, mikä aiheuttaa varteenotettavia uhkakuvia erityisesti huomioiden projektien hajautetun luonteen. Onkin lähes mahdotonta varmistua kattavasti, että tietyn projektin lähdekoodissa ei ole sinne luvatta lisättyä materiaalia. Tosin koodin vapaa saatavuus on omiaan vähentämään luvaton

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

plagiointia, sillä julkisesti saatavilla olevasta lähdekoodista luvatta kopioidun koodin löytäminen on huomattavasti helpompaa kuin suljetusta koodista. On myös olemassa kaupallisia toimijoita, jotka tarjoavat työkaluja lähdekoodin auditointiin.

Ohjelmistopatenttien osalta tilanne on vieläkin huonompi. On lähes varmaa, että kaikki laajemmat avoimen lähdekoodin tuotteet rikkovat ainakin jotakin myönnettyistä kymmenistä tuhansista patenteista. Jälleen kerran on kuitenkin syytä pitää mielessä, että ero tältä osin perinteisiin kaupallisiin tuotteisiin on olematon: yhtä todennäköisesti myös suljetun koodin tuotteet rikkovat patenteja. Avoimen lähdekoodin osalta patenttiriskiä tosin nostaa jonkin verran koodin vapaa saatavuus, joka helpottaa loukkauksien löytämistä. Toisaalta Suomessa ohjelmistopatentteihin liittyviä oikeudenkäyntejä ei ole esiintynyt ja täältä puuttuvat kokonaan erityisesti Yhdysvalloissa yleiset patenttivedätykset, jossa pyritään epämääräisillä patenteilla tapahtuvaan mahdollisimman laajaan rahastukseen.

Joka tapauksessa on syytä tiedostaa, että määräaikaista toimenpidekieltoa (käyttökielto) voidaan hakea ketä tahansa patenttia käyttävää tahoa vastaan, jonka vuoksi julkishallinnon järjestelmäkään eivät ole immuuneja. Euroopan Unionin alueella ei ole kuitenkaan tiedossa yhtään julkishallintoa vastaan nostettua kannetta tai patenttiriitaa. Yhdysvalloissa harjoitettava patenttikiistoilla rahastaminen ei leviä todennäköisesti julkiselle sektorille, koska osavaltiot on suojattu IPR-kiistoilta lailla.

Ei liene kovinkaan yllättävää, että toimittajat ovat varsin haluttomia antamaan oikeudellisia takuita avoimen lähdekoodin tuotteiden osalta. Voidaan myös kysyä, mikä merkitys olisi ylipäättänsä pienempien toimittajien antamalla takuulla, jos yrityksiensä taseet eivät kuitenkaan kestä täysimittaista patenttilitigaatiota. Yksi vaihtoehto tilanteen ratkaisemiseen on erityisten IPR-vakuutusten käyttö joko toimittajan tai hankintayksikön toimesta. Vakuutusten osalta ongelmana nähdään kuitenkin yleisesti kalliit hinnat ja huono kattavuus ongelmatilanteissa. Onkin varsin kyseenalaista, saavutetaanko vakuutusten käytöllä todellista apua potentiaalisiin ongelmatilanteisiin.

### 6.2.1.3 Avoimen lähdekoodin tuotteiden elinkaari

Yksi keskeinen ero avoimen lähdekoodin ja perinteisten suljettujen tuotteiden välillä on julkaisu nopeus. Suljetut tuotteet julkaistaan yleensä vasta siinä vaiheessa, kun ne ovat toimittajan mielestä valmiita tarkoitettuun käyttötarkoitukseen. Avoimen lähdekoodin ohjelmistot ovat yleensä vapaasti saatavilla yhteisöjen kotisivulta aivan alusta alkaen, jolloin ohjelmien toiminnallisuus voi olla hyvinkin puutteellista. Tämä ero on tosin osittain pienentynyt viime aikoina perinteisten toimittajien lisättyä erilaisten beta-versioiden tarjoamista tuotteistaan ja välillä kaupalliset paineet pakottavat julkaisemaan keskeneräisiä tuotteita.

Alkuvaiheen avoimen lähdekoodin ohjelmistoille tyypillisiä piirteitä ovat:

- Nopeat muutokset niin koodikannassa kuin myös rajapinnoissa
- Ohjelmien asentaminen vaatii omatoimista lähdekoodin kääntämistä ja voi olla myös muuten hankalaa
- Uusia versioita julkaistaan viikoittain tai jopa päivittäin
- Dokumentaatiota on minimaalisesti
- Käyttäjää on vähän, joista suurin osa on itse mukana kehitystoiminnassa
- Kaupallisia tukipalveluja ei ole saatavilla ja neuvonta tapahtuu postilistalla tai IRC-kanavilla
- Ohjelmistosta puuttuu suunniteltuja toiminnallisuuksia kuten esimerkiksi graafinen käyttöliittymä

Näiden ns. alfa-vaiheen ohjelmistojen käyttöä ei voi suositella kuin kaikista riskisietoisimmille ja osaavimmille organisaatioille. Hyvänä nyrkkisääntönä kannattaisi odottaa ainakin 1.1-version julkaisemista ohjelmistosta ennen kuin laajempaa käyttöönottoa alettaisiin harkita. Tosin tämäkään sääntö ei ole ilman poikkeuksia eli kaikissa tuotteissa numerointi ei välttämättä kerro tuotteen kypsyydestä (poikkeuksia on molempiin suuntiin).

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

Mikäli avoimen lähdekoodin tuote selviää alfa-vaiheesta, kehitystyön vauhti yleensä rauhoittuu ja käytettävyyden hiominen ja dokumentaatio saavat enemmän huomiota osakseen. Ohjelman suosion kasvaessa sen ympärille alkaa kasvaa "ekosysteemi", joka tuottaa myös kaupallisia palveluita. Tässä vaiheessa ohjelmistojen käyttöönotto ei yleensä muodosta kovin suurta haastetta. Hieman tuotteesta riippuen tämä tasaisen kehityksen vaihe saattaa kestää vuosia. Yksinkertaisemmissa tapauksissa tuote voidaan saada myös toiminnallisuuksien osalta valmiiksi, jonka jälkeen kyse on enää ylläpidosta ja virheiden korjaamisesta.

Ylläpitovaiheessa olevalla tuotteella tyypillisiä piirteitä ovat:

- Vakaus ja kypsytys kaikilta tuotteen osilta
- Yleinen luottamus tuotteen toimintaan pitkäaikaisten käyttökokemusten perusteella
- Uusia virheitä tai tietoturva-aukkoja paljastuu harvoin
- Olemassa oleva käyttäjäkunta pitää yllä kysyntää tukipalveluille ja tuotteen ylläpidolle

Kehitystyön hiljentyminen voi myös merkitä riskiä, koska välttämättä kovinkaan moni yhteisön jäsen ei ole kiinnostunut pelkästä ylläpidosta ja näin projekti voi hiljaksen hiipua pois. Tämä on toisaalta helppo tarkistaa esimerkiksi katsomalla viimeisten julkaisujen päivitysten ajankohtia tai hankkeen sivustolla käytävien keskustelujen ikää. Muilta osin tämän kehityksasteelle päässeen avoimen lähdekoodin tuotteen käyttö on varsin riskitöntä ja soveltuu kaikille hankintayksiköille.

Yhteenvedona avoimen lähdekoodin ohjelmistojen käyttöönotossa on keskeistä tuntee projektin kehityksaste ja suhteuttaa se sitten omaan osaamiseen. Osaavan organisaation käsissä epäkypsäkin projekti voi tarjota tehokkaan ratkaisun ongelmaan ja parhaassa tapauksessa yhteistyö hankkeen kanssa voi johtaa tuotteen kehittämiseen hyvin omaa tarkoituspäästä tukevaan suuntaan. Kyseessä tulisi olla tällöin aina harkitusta ja tietoisesta riskinotosta. Myös huomattavan vanhojen tuotteiden käyttöön voi liittyä riskejä ylläpidon jatkuvuuden suhteen. Sama tilanne on myös perinteisissä kaupallisissa tuotteissa sillä poikkeuksella, että avoimen lähdekoodin kohdalla korjauksien tekeminen ja ylläpito ei ole sidottu yhden toimittajan päätöksentekoon tuotteen tulevaisuudesta.

### 6.2.1.4 Lähdekoodin saatavuuden merkitys

Vaikka avoimen lähdekoodin yksi nimenomaisia vahvuuksia onkin lähdekoodin saatavuus, tilanne ei ole yksinomaan positiivinen. Organisaation sisällä pitää sopia koodin käyttöön liittyvistä menettelyistä ongelmien välttämiseksi.

Ensinnäkin helposti käsillä oleva lähdekoodi mahdollistaa omien, pienten korjausten tekemisen järjestelmiin. Mikäli tätä toimintaa ei koordinoita mitenkään ja korjauksista ei jää dokumentaatiota, voidaan nopeasti päätyä tilanteeseen, jossa ylläpito muuttuu lähes mahdottomaksi suoksi. Toisaalta omien korjausten tekemisellä voidaan saavuttaa merkittäviäkin kustannussäästöjä eli sitä ei missään tapauksessa kannata kokonaan kieltää. Paras ratkaisu olisi luoda erillinen kehitysympäristö, jonne koottaisiin muutokset ja niiden dokumentaatio ennen käyttöönottoa.

Toimiviksi todetut muutokset ja dokumentaatio kannattaa lähettää ohjelmiston kehittäjäyhteisölle siinä toivossa, että ne liitettäisiin alkuperäiseen ohjelmistoon. Tällöin oma muokattu ohjelmisto olisi yhteensopiva yhteisön kehittämän ohjelmiston kanssa ja siten riskittömämmin päivitettävissä uusiin ohjelmistoversioihin.

Edellä mainitun kanssa rinnakkainen kysymys on ohjelmistojen asentamiseen käytettävien tiedostojen hakeminen. Jos ohjelmistojen lataaminen sallitaan vapaasti verkosta (minkä lisenssit sinänsä sallivat), tietoturvan ja versionhallinnan kanssa tulee nopeasti ongelmia. Olisikin järkevintä, että organisaatio määritteli ohjelmistopankin (oman tai jonkin laajemmassa käytössä oleva), joka olisi ainoa sallittu paikka asennusmedioiden hakemiseen. Tämä kuitenkin edellyttää, että valittu pankki pitää asennusmediat ajan tasalla, koska muuten kiusaus hakea toimivampi versio muualta voi olla liian suuri. Jos avoimen lähdekoodin ratkaisulle käytetään kaupallista toimittajaa, keskitetty ratkaisu on usein myös ainoa mahdollinen.

### 6.2.1.5 Teknologiariskit

Kaupallisten tuotteiden toimittajilta on yleensä saatavissa markkinointimateriaalia, josta käy ilmi enemmän tai vähemmän yksityiskohtaisesti tuotteiden tekniset ominaisuudet. Samoin yritykset tiedottavat usein (reilustikin) etukäteen tulossa olevista uusista versioista ja niihin tulevista ominaisuuksista. Vaikka tällaiset tiekartat eivät aina toteudu ainakaan esitetyssä muodossa, ne antavat kuitenkin kuvaa siitä, mihin suuntaan tietty tuote on kehittymässä.

Kuten kohdassa 4.3 kuvattiin, avoimen lähdekoodin tuotteiden osalta vastaavaa suoraa tuoteinformaatiota on paljon vaikeampi löytää, mutta verkosta löytyy melko kattavia tuote-esittelyjä ja -vertailuja. Toisaalta kaikkea informaatiota ei ole yksinkertaisesti olemassa. Vain isoimmat projektit julkaisevat selviä tiekarttoja tulossa olevista versioista.

Vaikka suunnitelmia olisi olemassa, niistä ei välttämättä pidetä kovin tarkasti kiinni. Yhteisöjen prioriteeteissa keskeisempää onkin saada tuote julkaisukelpoiseen kuntoon, ei niinkään pitää kiinni enemmän tai vähemmän sattumanvaraisesti määritellyistä määräajoista. Jälleen kerran tässäkin on poikkeuksia ja osa avoimen lähdekoodin projekteista noudattaa tarkasti ennalta määriteltyä julkaisuaikataulua.

### 6.2.2 Menettelytavat riskien hallintaan

Mikäli hankintayksikkö päättää ottaa käyttöön itsenäisesti tietyn avoimen lähdekoodin tuotteen, on useimmissa tapauksissa lähes välttämätöntä tehdä huolellinen ennakoarviointi. Ennakoarvion ja varsinaisen käyttöönoton välinen ero ei tosin ole kuitenkaan yhtä jyrkkä avoimen lähdekoodin tuotteiden kohdalla kuin mitä se on perinteisten kaupallisten ohjelmistojen kohdalla. Koska avoimen lähdekoodin lisenssit sallivat rajoittamattoman kokeilun, ainoastaan hankintayksikön resurssit ja mahdolliset aikataulurajoitteet muodostavat rajat käyttökokemusten hakemiselle ennen varsinaiseen tuotantokäyttöön siirtymistä. Käyttökokemusten hakemisen lisäksi kannattaa aina kuitenkin tutustua tuotteen käyttämiin teknologioihin, projektin historiaan, käyttöprofiiliin sekä tuotteesta verkosta löytyvään palautteeseen.

Tilanne on luonnollisesti eri, jos kyse on kilpailutetusta hankinnasta. Tällöin toimittaja kantaa (ainakin lähtökohtaisesti) vastuun valinnasta. Tosin tällöinkin hankintaa edeltävässä esiselvitysvaiheessa hankintayksikön olisi järkevää käydä läpi ainakin suurin piirtein samat asiat kuin mitä tuotteen itsenäisessä käyttöönotossa. Käytettäväksi harkittavasta ohjelmistosta tulisi ainakin varmistaa seuraavat seikat:

- Ohjelmisto toimii, kuten sen dokumentaatio ja verkkosivuilla jaettava informaatio antavat ymmärtää
- Ohjelmiston todellinen suorituskyky riittää tai ylittää suunnitellussa tehtävässä tarvittavan suorituskyvyn
- Ohjelmistoa ylläpitävä yhteisö (tai yhtiö) on ”hengissä”. Positiivisia merkkejä tästä ovat esimerkiksi
  - Ohjelmiston päivitysversioiden ja tietoturva-aukkojen säännölliset julkaisut
  - Ohjelmiston verkkosivut ovat päivittyneet säännöllisesti
  - Ohjelmistoa koskevilla sähköpostilistoilla ja foorumeilla on tuoreita viestejä
  - Ohjelmiston tulevaisuudesta löytyy suunnitelma (tiekartta, engl. roadmap)
- Ohjelmistolle on tarjolla useampia kaupallista ylläpitopalvelua tarjoavia toimittajia

Avoimen lähdekoodin tuotteet koostuvat usein monesta erillisestä komponentista, joiden kehittäjä- ja ylläpitoyhteisöt voivat toimia täysin itsenäisesti. Tällöin tarkistukset tulisi tehdä jokaisen komponentin osalta erikseen. Kaikissa tapauksissa tämä ei ole luonnollisesti mahdollista: ääriesimerkinä voidaan ajatella Linux-jakelu Debiania, joka koostuu tuhansista itsenäisesti ylläpidetyistä, eri ohjelmistoja sisältävistä paketeista. Näiden läpikäyminen tarkoittaisi melkein koko avoimen lähdekoodin kentän läpikäyntiä. Hankintayksikön kannattaakin tällöin harkita, mitkä komponentit ovat keskeisiä hankinnalle tai käyttöönotolle ja mitkä taas niin merkityksettömiä, etteivät niiden mahdolliset ongelmat kumuloidu kokonaisuuteen.

Avoimen lähdekoodin tuotteiden osalta tarvittavien tietojen löytäminen on yleensä helppoa. Koska kehitystyö tapahtuu pääsääntöisesti avoimilla sähköpostilistoilla ja foorumeilla, hakukoneilla on löydettävissä niin positiiviset kuin negatiiviset keskustelut tuotteista.

### 6.2.3 Hankintatapaan liittyvien riskien arviointi

Kuten tässä luvussa on jo aikaisemmin kerrottu, avoin lähdekoodi mahdollistaa varsin joustavan riskinhallinnan hankintayksikön preferenssien perusteella. Samoin kriteerejä hankintatavan valitsemiselle on jo käyty läpi luvussa 5. Joka tapauksessa itsenäisen käyttöönoton ja täysin toimittajalähtöisen hankinnan välillä on luonnollisesti merkittäviä eroja riskiprofiilissa.

#### 6.2.3.1 Itsenäinen hankinta

Kun hankintayksikkö asentaa itse Internetistä hankittuja ohjelmistoja, tähän liittyy luonnollisesti omat riskinsä niin tietoturvan kuin oikeudellisten kysymysten osalta.

Laajasti käytössä olevien tuotteiden kohdalla tietoturvaan liittyvät riskit ovat pienempiä ja aktiivinen käyttäjäyhteisö havaitsee yleensä nopeasti tahattomat tai tahalliset tietoturva-aukot. Pienempien projektien kohdalla on kuitenkin mahdollista, että tuotteeseen on jäänyt joko tahattomia tai tahallisia aukkoja, joita voidaan hyödyntää verkkohyökkäyksissä. Jälleen kerran tilanne ei kuitenkaan tältä osin eroa merkittävästi perinteisistä suljetun lähdekoodin tuotteista. Tosin osaava hankintayksikkö pystyy ainakin periaatteessa auditoimaan avoimen lähdekoodin tuotteiden sisällön käymällä läpi niiden lähdekoodin. Kaikista tietoturvakriittisimmissä sovelluksissa tämä on myös usein jopa välttämätöntä. Auditoinnin voi suorittaa myös kolmas osapuoli.

Itsenäiseen ohjelmistojen ylläpitoon liittyy myös aina kysymys hankintayksikön tietotaidosta. Vaikka normaaleista ongelmatilanteista selvittäisiinkin itsenäisesti, oudompien ongelmien ratkaisemiseen taidot eivät välttämättä riitä. Avoimen lähdekoodin yhteisöt saattavat kyetä auttamaan tällaisissa tilanteissa. Toinen vaihtoehto on, että tarvittava erityisosaaminen ostetaan ulkopuoliselta palveluntarjoajalta. Tämä luonnollisesti edellyttää, että käytettävän tuotteen ylläpitoon ja ongelmanratkaisuun löytyy kaupallisia palveluntarjoajia. Näiden palveluiden tarjonta olisi syytä varmistaa etukäteen ainakin kriittisempien tuotteiden osalta.

Kokonaan oma lukunsa on tilanteet, joissa hankintayksikkö alkaa itsenäisesti kokoamaan eri avoimen lähdekoodin komponenteista uusia tuotteita tarpeisiinsa. Tämä edellyttää jo hyvin huomattavaa osaamista. Tämän lisäksi huomioitavaksi tulee erityisesti rakennettavien järjestelmien hallinnointi tilanteissa, joissa käytettävien komponenttien kehityssuunnat eivät vastaa välttämättä hankintayksiköiden tarpeita.

#### 6.2.3.2 Ulkoinen toimittaja

Ulkoinen toimittajien valintaan liittyviä kysymyksiä on jo käsitelty varsin laajasti luvussa 5. Riskinhallinnan näkökulmasta keskeistä on ensinnäkin, että valittava toimittaja sitoutuu noudattamaan hankintayksikön käyttämiä standardeja ja menettelytapoja mm. tietoturvan ja laadunhallinnan osalta.

Ideaalitilanteessa hankintayksikön tulisi edellyttää toimittajalta täydellistä käytettävien komponenttien auditointia niin tietoturvan kuin IPR:ien osalta. Käytännössä tämä ei kustannussyistä ole useinkaan mahdollista ainakaan jos käytetään laajempia avoimen lähdekoodin kokonaisuuksia. Huomiota tulisi myös kiinnittää siihen, että toimitettavien tuotteiden dokumentaatio on niin laadukasta, että hankintayksikkö pystyy tarvittaessa käyttämään kolmansia tahoja mahdollisten ongelmien ratkaisemiseen sekä tuotteiden jatkokehittämiseen. Erityisen kriittistä on, että mahdolliset toimittajan tuotteisiin lisäämät uudet rajapinnat ovat ensinnäkin dokumentoitu riittävällä tarkkuudella, mutta myös niin avoimia IPR-mielessä, että ne ovat toteutettavissa myös muiden kuin toimittajan toimesta.

Mikäli on nähtävissä, että toimittajan ja hankintayksikön välinen suhde tulee olemaan pidempiaikaisempi kuin yksittäinen kertatoimitus, toimittajan luotettavuuden arviointi on riskinhallintamielessä järkevää. Hankintaprosessissa tämä olisi hyvä tehdä mahdollisuuksien mukaan jo esiselvitysvaiheessa. Keskeisiä kysymyksiä, jotka tosin soveltuvat myös perinteisiin toimittajiin, ovat mm.:

- Toimittajan taloudellinen vakaus
- Tunnustettu asema tuotteen toimittajana

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

- Toimivat prosessit riskinhallinnassa
- Liiketoimintastrategian realismi
- Kolmansien osapuolten vaikutusmahdollisuudet (mm. vihamielinen fuusio)
- Liikkeenjohdollinen osaaminen
- Tietotaito ja tietotaidon pysyvyys

### 6.2.4 Tuotteen elinkelpoisuuden arviointi

Avoimen lähdekoodin projektien ”kuoleminen” ei ole mitenkään erityisen harvinainen tapahtuma. Vaikka kyseessä ei ole yhtä radikaali ilmiö kuin perinteisen ohjelmistoyrityksen konkurssi lähdekoodin jäädessä julkisesti saataville, tilanne voi aiheuttaa huomattavia ongelmia kyseisen tuotteen valinneelle organisaatiolle.

Vaikka ylläpitoa voidaankin näissä tapauksissa periaatteessa jatkaa itsenäisesti tai käyttämällä kolmansien tahojen tarjoamia palveluja, tarvittavan osaamisen kerryttäminen vie aikaa ja aiheuttaa mahdollisesti merkittäviä kustannuksia. Mikäli kyse on tarpeeksi arvokkaasta tai helposti hallittavasta tuotteesta, tämäkin vaihtoehto voi kuitenkin olla realistinen. Avoimen lähdekoodin ominaispiirteet, pysyvästi saatavilla oleva lähdekoodi ja täysi muokkaus- ja jatkokehittämisoikeus, muodostuvatkin tässä merkittäväksi riskinrajaukseksi. Useimmiten järkevämpi vaihtoehto on kuitenkin etsiä korvaava tuote ja mahdollisesti koettaa saada siihen integroiduksi edellisen tuotteen erityisen tärkeät osat.

### 6.2.5 Tuotteen jatkokehityksen arviointi

Avoimen lähdekoodin tuotteet eivät lähtökohtaisesti eroa merkittävästi perinteisistä suljetuista tuotteista teknologia-tiekartan suhteen. Esimerkiksi mahdollisuus vaikuttaa tietyn projektin suuntaan ei juuri poikkea hankintayksikön mahdollisuudesta vaikuttaa tietyn yrityksen jatkokehityssuunnitelmiin – pienissä projekteissa tämä on usein helppoa kuten myös PK-yritysten kohdalla. Tämä edellyttää kuitenkin useimmiten aktiivisia toimia kuten koodiparannusten lähettämistä projektiin. Toisaalta yksittäisen kunnan mahdollisuus vaikuttaa esimerkiksi Linux-ytimen kehitykseen on yhtä todennäköistä, tai paremminkin epätodennäköistä, kuin vaikuttaminen kilpailevien käyttöjärjestelmien kehitykseen.

Selkeä merkittävä ero on kuitenkin siinä, että mikäli hankintayksikön intressi on tarpeeksi suuri, se voi muokkaus-oikeutta hyväksikäyttäen tehdä oman räätälöidyn version tuotteesta. Suljettujen tuotteiden osalta tätä mahdollisuutta ei ole.

### 6.2.6 Riskien tarkistuslista

Seuraavaa listaa, joka on koottu tässä kappaleessa esiin nousseista asioista, voidaan käyttää apuna avoimen lähdekoodin hankintojen riskinhallinnassa. Kuten listasta käy ilmi, useimmat riskinhallintaan liittyvät asiat eivät suoraan liity avoimeen lähdekoodin erikoispiirteisiin sinänsä, vaan soveltuvat aivan yhtäläisesti myös perinteisiin suljettuihin ohjelmistotuotteisiin. Lisäksi kyse on ennen kaikkea vain esimerkistä ja ennen listan laajempaa käyttöä olisi keskeistä miettiä hankintayksikön sisällä, kuvaako listan sisältö niitä riskejä, jotka ovat relevantteja kyseisissä tapauksissa.

Lista on esitetty aika- ja tärkeysjärjestyksessä. Jos ensimmäisissä kohdissa tarkastelun tulos on kielteinen, kannattaa harkita jatkon mielekkyyttä.

- Vastaako ohjelmisto siitä annettuja kuvauksia?
- Vastaako ohjelmisto sen dokumentaatiota?
- Onko ohjelmisto jaossa tunnetulla ja yleisesti saatavilla olevalla verkkosivustolla?
- Onko ohjelmiston takana yritys, ei-kaupallinen yhteisö vai vain muutamia kehittäjiä?
- Onko hankintayksikkö riippuvainen kehityssuunnitelmassa mainitusta, tulossa olevasta toiminnallisuudesta?
- Noudattaako ohjelmisto myös avoimia rajapintoja, protokollia ja tiedostomuotoja?
- Onko ohjelmistosta julkaistu opaskirjoja?
- Onko ohjelmiston käyttäjämäärä kasvussa?

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

- Onko ohjelmiston kehittäjä määrä kasvussa?
- Löytyykö ohjelmistolle paikallisia toimittajia?
- Ovatko ohjelmistojen toimittajat yritykset toimittaneet tuotetta aikaisemmin julkiselle sektorille?
- Onko ohjelmistolla olemassa kehityssuunnitelmaa (engl. roadmap)?
- Voiko ohjelmistoon esittää lisättäväksi uusia toiminnallisuuksia?
- Onko ohjelmiston virheiden raportointiin ja korjaamiseen olemassa selvä prosessi?
- Onko ohjelmiston julkaisuhistoria julkisesti saatavilla?
- Onko ohjelmiston tietoturva-aukoista ja niiden paikkaamisesta saatavilla julkista informaatiota?
- Onko ohjelmistolla useampia ja projektista riippumattomia käyttäjäfoorumeita tai postilistoja?
- Mikäli ohjelmisto vaatii toimiakseen kolmansien osapuolten komponentteja (avoimia tai suljettuja), onko niistä tehty riskianalyysi?
- Onko ohjelmistolla menettelyohjetta patenttiloukkausten suhteen?
- Onko ohjelmistolla prosessia tekijänoikeuksien hallintaan?
- Onko ohjelmistolla omaa Usenet-ryhmää (kertoo projektin iästä)?

### 6.3 JIT 2007 ja avoin lähdekoodi

#### 6.3.1 Yleistä

Mahdollisimman vakioitujen ehtojen käyttäminen hankinnoissa on ehdottoman järkevä strategia. Se helpottaa sekä toimittajan että hankintayksikön elämää lisäten erityisesti varmuutta tehdyn oikeustoimen sisällöstä. Avoimen lähdekoodin osalta tilanne ei luonnollisesti ole erilainen. Julkisen hallinnon tietohallinnon neuvottelukunta (JUHTA) suosittelee, että *“valtion virastot, liikelaitokset, laitokset ja rahastot sekä kunnat ja kuntayhtymät käyttävät ...Julkisen hallinnon IT hankintojen sopimusehtoja (JIT 2007) hankkiessaan IT- tuotteita ja palveluita.”*

JIT 2007 valmisteltiin valtionvarainministeriön alaisessa työryhmässä, jossa olivat edustettuina sekä toimittajat että valtio ja kunnat. Ehdoista järjestettiin myös laaja kuulemiskierros, jolla saatiin varsin kattava näkemys sektorin ehtoihin asettamista toiveista ja vaatimuksista. Muutamien iteraatiokierrosten jälkeen ehdot saatiin valmiiksi keväällä 2007 ja ne otettu 8.10.2007 käyttöön JHS-suosituksena. Ehdot korvaavat valtion tietotekniikkahankintojen yleiset sopimusehdot 1998 (VYSE 1998) sekä niihin liittyvät erityisehdot.

Ehdot koostuvat Yleisistä ehdoista ja niitä täydentävistä liitteistä:

- (i) Yleiset sopimusehdot (JIT 2007 – Yleiset ehdot)
- (ii) Erityisehtoja tilaajan sovellushankinnoista (JIT 2007 – Sovellukset)
- (iii) Erityisehtoja palveluista (JIT 2007 – Palvelut)
- (iv) Erityisehtoja konsultointipalveluista (JIT 2007 – Konsultointi)
- (v) Erityisehtoja valmisohjelmistohankinnoista (JIT 2007 – Valmisohjelmistot)
- (vi) Erityisehtoja laitehankinnoista (JIT 2007 – Laitteet)

On syytä huomioida, että JIT 2007 ei sisällä varsinaista sopimusta, jossa määriteltäisiin hankinnan osapuolet, kohde, toimituksen tai palvelun sisältö hinnat ja ym. vastaavat yksilöivät seikat. Tällainen tulee siis aina laatia erikseen. Toiseksi JIT 2007:ssä esiintyy myös yleisesti toteamus ”*jollei toisin ole kirjallisesti sovittu*”. Näissä kohdissa on haluttu antaa signaali, että asiasta ehkä todellakin kannattaisi sopia toisin tai ainakin asiaa tulisi harkita huolellisesti.

#### 6.3.2 JIT ja avoimen lähdekoodin erityisehdot

JIT 2007:n soveltamisohje kommentoi avoimen lähdekoodin ja JIT 2007:n suhdetta seuraavasti:

*“Avoimen lähdekoodin käyttö on lisääntynyt IT-projekteissa. Näissä sopimusehdoissa on lähdetty siitä, ettei avoimen lähdekoodin käyttö yleensä poikkea muiden kolmansien osapuolten komponenttien käyttämisestä toimituksessa. Koska avoimen lähdekoodin lisenssiehdot eivät ole kovin hyvin tunnettuja, sopimusehtoihin on sisällytetty ehto, jonka mukaan toimittajan tulee selvittää tilaajalle avoimen lähdekoodin käyttöön liittyvät*

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

*ehdot ja varmistaa, etteivät kyseiset ehdot tule sovellettavaksi tilaajan muihin ohjelmistoihin, jollei siitä ole sovittu. Mikäli tilaaja edellyttää tuotteen avoimen lähdekoodin luomista tilaajaa varten, sopijapuolten tulee sopia tapauskohtaisesti avoimen lähdekoodin käyttöön ja lisensointiin liittyvistä kysymyksistä, kuten lisenssiehdoista ja lähdekoodin julkaisemisesta.”*

JIT 2007:n peruslähdekohtana on siis, että avoimen lähdekoodin ohjelmistot eivät poikkea muista ohjelmistotuotteista, joiden oikeudet kuuluvat kolmansille osapuolille. Niihin sovelletaan täysin yhteneväisesti toimittajan vastuuta koskevia säännöksiä, testaus ja hyväksymismenettelyitä jne. Avoin lähdekoodi on määritelty JIT 2007:ssä seuraavasti (Liite 2: Yleiset sopimusehdot):

*"Avoin lähdekoodi tarkoittaa ohjelmistoja, joiden käyttöä koskevat osoitteessa [http://www.opensource.org/yksilöidyt käyttöoikeusehdot](http://www.opensource.org/yksilöidyt_käyttöoikeusehdot) tai joiden käyttöoikeusehtoihin sisältyy vaatimus julkaista tai muuten luovuttaa lähdekoodi, jos tilaaja levittää sitä kolmansille osapuolille."*

Muotoilu kattaa kaikki yleisimmät avoimen lähdekoodin lisenssit (OSI-hyväksytyjä) ja tämän lisäksi kohdassa 4.2. käytetyn jaottelun mukaisesti kaikki heikkoa ja vahvaa vastavuoroisuutta edellyttävät lisenssit. Sen sijaan sallivat lisenssit, jotka eivät ole saaneet OSI-hyväksyntää eivät ole määritelmän piirissä.

Käytännön seurauksia on kaksi. Yleisissä sopimusehdoissa Takuu-kohdan alle sijoitetussa ehdossa todetaan seuraavaa:

*(9) Jos sopijapuolet sopivat avoimen lähdekoodin käyttämisestä, toimittaja ilmoittaa tilaajalle etukäteen avoimeen lähdekoodiin liittyvistä käyttöoikeutta koskevista ehdoista sekä muista ehdoista ja rajoituksista, joita tilaajan tulee avoimen lähdekoodin käytössä noudattaa. Mikäli toisin ei ole sovittu, toimittaja vastaa siitä, että avoimen lähdekoodin sopimuksen mukainen käyttö ei johda siihen, että tilaajan muihin ohjelmistoihin sovellettaisiin avoimen lähdekoodin käyttöoikeutta koskevia ehtoja.*

Kohdan mukaan toimittajan tulee kertoa mitä oikeuksia ja velvollisuuksia sopimus tuo tullessaan. Tilanne ei tältä osin poikkea merkittävästi perinteisten kaupallisten tuotteiden käytöstä, joiden mukana toimitaan yleensä näiden lisenssiehdot. Yksinkertaisimmillaan ehto täyttyy toimittamalla tuotteen mukana siinä käytetyt avoimen lähdekoodin lisenssit.

Käytännössä useimmissa tapauksissaärkevin tapa kohdan täyttämiseen on luettelo, jossa listataan toimituksen mukana tulevat ohjelmistot ja niistä löytyvät lisenssit. Avoimeksi jää sellainen tilanne, jossa avoimen lähdekoodin käytöstä ei ole erikseen sovittu. Lähtökohtaisesti tällöin erillistä velvollisuutta käyttöehtoja koskevien ehtojen luovuttamiseen ei ole.

Kohdan toinen edellytys on huomattavasti haasteellisempi toimittajalle. Sen tarkoituksena on estää vahvaa vuorovaikutusta edellyttävien lisenssien (esimerkiksi GPL) ehtojen laajeneminen kattamaan tilaajan olemassa olevia ohjelmistoja. Tilannetta tosin helpottaa se seikka, että läheskään kaikissa toimituksissa tämä ei ole edes teoriassa mahdollista. Kohta ei esimerkiksi luonnollisestikaan sovellu ohjelmistoihin, joiden lisenssit ovat sallivia (esimerkiksi BSD) tai edellyttävät ainoastaan normaalia vastavuoroisuutta (esimerkiksi Mozilla Public License). Samoin jos ohjelmaa ei levitetä hankintayksikön ulkopuolelle, vastavuoroisuusvelvollisuutta ei synny.

Käytännössä voidaankin olettaa, että toimittajat pyrkivät rajaamaan "sopimuksen mukaisen käytön" hankintayksikön sisäiseksi käytöksi, jolloin kohdan merkitys katoaa. Tätä ei voida myöskään pitää täysin kohtuuttomana, koska erilaisten avoimen lähdekoodin integrointitapojen määrittely ennakoivasti sopimuksissa olisi melkoisen haasteellinen tehtävä.

### 6.3.3 JIT:n yleiset ehdot

JIT:n ehdoissa on myös muita kohtia, joilla on suoraa merkitystä avoimen lähdekoodin hankintaprosessille. Erityisen keskeinen on immateriaalioikeuksia koskeva kohta.

## JUHTA - Julkisen hallinnon tietohallinnon neuvottelukunta

Toimittaja on ehtojen mukaan täysin vastuussa myös toimittamistaan avoimen lähdekoodin tuotteista. Osalle toimittajista tämä saattaa olla liian "kova pala" nieltäväksi, koska kuten edellä on selostettu, täydellinen IPR-auditointi on usein käytännössä mahdotonta ainakin laajempia kokonaisuuksia käsittävissä avoimen lähdekoodin toimituksissa. Tällaisissa tilanteissa hankintayksikön kannattaa suorittaa oma riskianalyysi ja miettiä kummasta aiheutuu enemmän ongelmia, potentiaalisten toimittajien vähenemisestä vai kustannustason noususta (varsin teoreettisissa) loukkaustilanteesta.

On myös syytä huomioida, että pienempien toimittajien tosiasiallinen kestävyys esimerkiksi laajempaa patenttioikeudenkäyntiä vastaan on varsin rajallinen. Onkin pelättävissä, että toimittajat sitoutuvat helposti ehtoihin, joita eivät tositalanteen tullessa kykenisi kantamaan. Tilanne ei tältä osin jälleen kerran poikkea merkittävästi verrattuna perinteisten suljetun lähdekoodin tuotteisiin.

### 7 Suositukset

- Ohjelmiston hankinta tulee tehdä pääsääntöisesti siten, että hankittava ohjelmisto voidaan jakaa hallinnon ja kansalaisten keskuudessa. Parhaiten ohjelmiston jakamisen edellytykset turvataan käyttämällä avoimen lähdekoodin lisenssejä.
- Kun avoimeen ohjelmistoon teetetään muutoksia ja lisäyksiä, tulisi kehitystulokset julkaista avoimen lähdekoodin lisenssillä, ellei kehitystyön sulkemiseen ole erityistä perustetta.
- Avoimen ohjelmiston omatoimisessa käyttöönnotossa on huolehdittava, että käyttäjäorganisaatiolla on riittävästi omaa osaamista ja resursseja evaluoida vaihtoehtoiset ratkaisut ja suorittaa käyttöönotto. Useimmissa tapauksissa ulkopuolisen asiantuntijapalvelun hankkiminen on suositeltavaa.
- Ohjelmiston tulee noudattaa avoimia standardeja ja rajapintoja. Näin varmistetaan mahdollisimman suuri liikkumavara tulevaisuudessa.
- Muiden organisaatioiden kokemuksia ja suunnitelmia vastaavista hankinnoista kannattaa kartoittaa. Yhteistyöhön kannattaa pyrkiä – voidaan esimerkiksi sopia kustannusten jakamisesta.
- Hankittavan ohjelmiston lisensointimallista riippumatta on syytä edellyttää lähdekoodin toimittamista osana kokonaistoimitusta.
- Avointa ohjelmistoa hankittaessa on tutustuttava, että ohjelmiston lisenssi täyttää Open Source Initiatiivin avoimen lähdekoodin määritelmän. Markkinoilla on ohjelmistoja, joita markkinoidaan avoimina, vaikka niitä ei saa muokata tai jaella edelleen. OSI:n hyväksymien avoimen lähdekoodin lisenssien lista on osoitteessa <http://www.opensource.org/licenses>.

### 8 Opastavat tiedot

#### 8.1 Suosituksen ylläpito

Tätä suositusta ylläpitää Julkisen hallinnon tietohallinnon neuvottelukunta JUHTA, puh (09) 16001, sähköposti: [jhs-sihteerit@jhs-suositukset.fi](mailto:jhs-sihteerit@jhs-suositukset.fi)

JHS-järjestelmän verkkosivut: <http://www.jhs-suositukset.fi/>

#### 8.2 Liitteet

- Liite 1: Linkkejä tietovarantoihin ja ohjelmistohakemistoihin

## 9 Liite 1: Linkkejä tietovarantoihin ja ohjelmistovarastoihin

### Tietovarantoja

- **OSOR, <http://www.osor.eu/>**  
Tietoa Euroopassa tehdyistä avoimen lähdekoodin hankinnoista ja käyttöönotoista sekä hyvistä käytännöistä. Sisältää myös julkishallinnon ohjelmakoodi- ja dokumenttihakemiston.
- **Open Source Initiative, <http://www.opensource.org/>**  
OSI hallinnoi avoimen lähdekoodin määritelmää ja ylläpitää listaa hyväksymistään avoimen lähdekoodin lisensseistä.
- **FOSSBazaar, <http://www.fossbazaar.org/>**  
Yhteisö avoimen lähdekoodin hallittuun hyödyntämiseen. FOSSBazaar tarjoaa tietoa esimerkiksi juridisista kysymyksistä ja hyvistä käytännöistä avoimen lähdekoodin hyödyntämisessä.
- **Suomen open source -keskus COSS, <http://www.coss.fi/>**  
Uutisia ja tapahtumia avoimen lähdekoodin maailmasta. Tietoa avoimen lähdekoodin ratkaisujen ja palvelujen toimittajista.

### Ohjelmistohakemistoja

- **FLOSS guide ja FLOSS catalog, <http://guide.conecta.it/>**  
EU:n komission rahoittamien Flossmetrics- ja OpenTTT -hankkeiden yhteydessä kootut avoimen lähdekoodin opas ja noin 100-sivuinen avoimen lähdekoodin ohjelmistojen luettelo. Opas on lähinnä pk-yrityksille tarkoitettu mutta se soveltuu perusteellisuutensa ja laajuutensa vuoksi myös julkishallinnon käyttöön. Ohjelmistot on luetteloitu käyttötarkoituksen mukaan ja niiden soveltuvuus ja kypsyys on arvioitu.
- **OSOR, <http://www.osor.eu/>**  
Julkishallinnossa hyödynnettävien ohjelmistojen ja ohjelmistodokumenttien hakemisto.
- **Enterprise Open Source Directory, <http://www.eosdirectory.com>**  
Yrityskäyttöön soveltuvia avoimen lähdekoodin ohjelmistoja kategorioihin jaoteltuina. Ohjelmistot on arvioitu mm. toiminnallisuuden, kypsyiden ja ohjelmistoa kehittävän yhteisön aktiivisuuden mukaan.
- **Open Source as Alternative, <http://www.osalt.com/>**  
Avoimen lähdekoodin ohjelmistoja voi hakea sillä perusteella, mitä suljettuja ohjelmistoja ne korvaavat tai vastaavat.
- **Open Source Enterprise Software Directory, <http://www.opensourcexs.info/>**  
Pyrkii listaamaan parhaita yrityskäytössä olevia avoimen lähdekoodin ohjelmistoja.
- **SourceForge.net, <http://www.sourceforge.net/>**  
Erittäin laaja ohjelmistohakemisto ja alusta avoimen lähdekoodin projektin ylläpidolle.
- **Launchpad, <http://www.launchpad.net/>**  
Ohjelmistohakemisto ja alusta avoimen lähdekoodin projektien ylläpidolle.