

Leven met open source licenties

Software licenties...het woord alleen al roept bij menig software engineer een reactie op variërend van totale onverschilligheid tot afschuw. Hoe er ook tegenaan wordt gekeken: zodra er ook maar één if-statement op het scherm wordt gezet dan krijgt men hiermee te maken aangezien er dan meteen auteursrecht op zit. Zodra men tooling, libraries of voorbeelden van het internet wil gebruiken, krijgt men alsnog met een licentie van doen. Dat in het achterhoofd houdende, is het verstandig om meer te weten over dit onderwerp.

Nu de software industrie aan het groeien is, wordt dit onderwerp dus steeds belangrijker. Naast de gesloten licenties en het publieke domein, is het speelveld de laatste jaren uitgebreid met open source. Voor programmeurs heel handig en leerzaam, maar voor bedrijfsjuristen wordt het een groeiende hoofdpijn. Want naast een aantal rechten, die men bij open source licenties krijgt, zitten er ook een aantal plichten aan vast. En juist die plichten maakt het voor een hoop bedrijven lastig.

Een voorbeeld: de BSC licentie betekent voor veel mensen "doe wat je wilt, zo lang onze naam er maar bij staat". In clause 2 staat echt ook: "Redistributions in binary form must reproduce above copyright". Deze regel ziet een gebruiker al snel over het hoofd, maar hier moet wel degelijk rekening mee worden gehouden. Even een stukje code knippen en plakken kan dus consequenties hebben. In bovenstaand geval kan men bijvoorbeeld in een 'about' scherm een print doen, maar voor echte embedded software kan het lastiger worden als de binary ineens een licentie af moet kunnen drukken.

Een andere uitdaging is het bundelen van software. Een distributie kan vaak veel licenties door elkaar heen bevatten. Sterker nog: een package kan zelfs een verwijzing naar een licentie hebben, maar dat wil niet zeggen dat alle bestanden erin dezelfde licentie hebben. Om bijvoorbeeld een distributie voor een router samen te stellen dient, naast het zorgen voor binaire compatibiliteit, ook rekening te worden

gehouden met licentie compatibiliteit. GPLv2 stelt dat men geen verdere restricties mag opleggen, anders dan in de licentie voorgeschreven. De BSD licenties stelt echter dat men, op reclame voor producten met die code, zonder geschreven toestemming vooraf de namen van de auteurs mag gebruiken. Het mag dan wel samen compileren, maar juridisch gezien passen deze twee niet samen. En om het nog spannender te maken: in een lap code als de Linux Kernel, heeft iedere schrijver een patch een stukje auteursrecht. Wil men de kernel van licentie laten veranderen, dan zal men deze stuk voor stuk om toestemming moeten vragen.

Om dit laatste op te lossen zijn veel projecten aan het overschakelen naar een overdrachtsconstructie. Zodra met code oplevert, geeft men ook toestemming om het onder een organisatie te laten vallen, of wordt er een standaard licentie aan toegekend. Debian heeft bijvoorbeeld een 'social contract' en GNU biedt de mogelijkheid om de code bij de Free Software Foundation in beheer te brengen.

In de kernel is een ander opgelost door het verplicht stellen van een 'signed off by' regel in patches. Hiermee garandeert de inzender dat de code van de inzender zelf is, of ongewijzigd van iemand afkomt het ook heeft getekend. Daarnaast betekent die handtekening ook dat de code onder GPLv2 of compatible licentie valt.

Hoewel dit voor de projecten afdoende is, kan het voor een bijdrager wel een probleem opleveren. Durf je je code af te geven aan een ander? In het geval van de Free Software Foundation zijn de intenties duidelijk, maar in hoeverre kun je grote multinationals vertrouwen? Wie garandeert dat ze morgen niet ineens closed source gaan met jouw code?

Zelfs de copyright regel kan al een uitdaging vormen voor de gebruikers. Deze bestaat doorgaans uit een jaartal, naam en email adres. Hieronder een drietal voorbeelden:

©2010 Micha Hergarden (Micha.hergarden@altenpts.nl)

©2010 Alten PTS bv (info@altenpts.nl)

©2010 Micha Hergarden (Micha@home.nl)

Zo'n op het oog eenvoudige regel kan al een aantal vragen oproepen. In het eerste geval zou ik de software op eigen titel geschreven kunnen hebben. Het email adres doet echter vermoeden dat ik het in opdracht van Alten PTS heb geschreven. Als dit zo is, dan zou ik beter de tweede variant kunnen gebruiken. Wanneer ik het privé geschreven zou hebben, dan kan ik beter de laatste versie gebruiken. Voor een gebruiker van een file waar de eerste variant in staat, kan het dus heel lastig zijn om de juiste bedoeling van de auteur te begrijpen.

Om het omgaan met al die licenties een beetje te vergemakkelijken lopen er een aantal initiatieven om de boel te standaardiseren. Met name het Open Source Initiative doet er alles aan om zoveel mogelijk mensen te overtuigen een standaard goedgekeurde licentie te gebruiken. Er zijn momenteel zo'n 66 goedgekeurde licenties beschikbaar, dus de kans dat een licentievoorwaarde hier niet in past is klein.

Tot slot nog een paar tips voor bedrijven om een zo makkelijk mogelijk een keuze te maken:

- Kies voor duidelijkheid en standaarden. Bij een dubbelzinnig licentie wordt vaak gekozen om de software niet te gebruiken. GPLv2 is overal bekend, maar met een 'pizzalicensie' (als je de software goed vindt, moet je mij een pizza geven) wordt het distribueren van de code lastig (en wat te doen met 12.000 quattro stagioni's?)
- Vraag alle licenties op als er een leverancier aan te pas komt. Degene die distribueert is ook degene die de aanklacht aan zijn broek krijgt bij problemen.
- Open source licenties zijn niet anders dan gewone licenties. Zorg dat er een jurist naar kijkt, dan weet je waar je aan toe bent.

Ondanks dit alles biedt open source grote kansen en nieuwe mogelijkheden. En dat maakt het voor een hoop bedrijven toch interessant. Dit blijkt wel uit het feit dat een bedrijf als Microsoft dit jaar op de Linux Tag niet alleen een dubbele stand had ingekocht, maar zelfs als

Keynote spreker was uitgenodigd. De duivel haalt z'n schaatsen uit het vet!

N.B. dit artikel is geschreven naar aanleiding van de presentatie: 'The State of Open Source licensing and ways to improve it' op LinuxTag 2010 van Martin Michlmayr (Hewlett-Packard).

Micha Hergarden