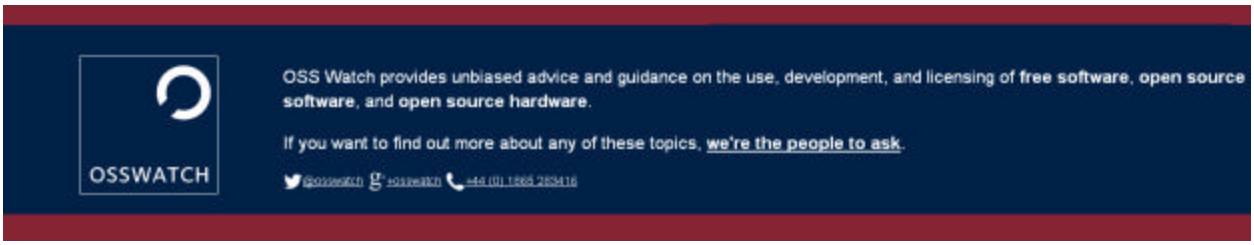

OSS Watch team blog
open source software advisory service

Locked into Free Software? Unpicking Kuali's AGPL Strategy

Posted on **September 10, 2014** by **Scott Wilson**

Chuck Severance recently published a post entitled How to Achieve Vendor Lock-in with a Legit Open Source License – Affero GPL where he criticises the use of AGPL licenses, particularly its use – or at least, intended use – by Kuali. Chuck's post is well worth reading – especially if you have an interest in the Kuali education ERP system. What I'm going to discuss here are some of the details and implications of AGPL, in particular where there are differences between my take on things and the views that Chuck expresses in his post.

Copyleft licenses such as GPL and AGPL are more restrictive than the so-called permissive licenses such as the Apache Software License and MIT-style licenses. The intent behind the additional restrictions is, from the point of view of the Free Software movement, to ensure the continuation of Free Software. The GPL license requires any modifications of redistributed code to also be released as GPL.

With the advent of the web and cloud services, the nature of software distribution has changed; GPL software can – and is – used to run web services. However, using a web service is not considered distributing the software, and so companies and organisations using GPL-licensed code to run their site are not required to distribute any modified source code.

Today, most cloud services operate what might be described as the "secret source" model. This uses a combination of Open Source, Free Software and proprietary code to deliver services. Sometimes the service provider will contribute back to the software projects they make use of, as this helps improve the quality of the software and helps build a sustainable community – but they are under no obligation to do so unless they actually choose to distribute code rather than use it to run a service.

The AGPL license, on the other hand, treats deployment of websites and services as "distribution", and *compels* anyone using the software to run a service to also distribute the modified source code.

AGPL has been used by projects such as Diaspora, StatusNet (the software originally behind Identi.ca – it now uses pump.io), the CKAN public data portal software developed by the Open Knowledge Foundation, and MIT's EdX software.

We've also discussed before on this blog the proposition – made quite forcefully by Eben Moglen – that the cloud needs more copyleft. Moglen has also spoken in defence of the AGPL as one of the means whereby Free Software works with cloud services.

So where is the problem?

The problem is that the restrictions of AGPL, like GPL before it, can give rise to bad business practice as well as good practice.

In a talk at Open World Forum in 2012, Bradley Kuhn, one of the original authors of AGPL, reflected that, at that time, some of the most popular uses of AGPL were effectively "shakedown practices" (in his words). In a similar manner to how GPL is sometimes used in a "bait and switch" business model, AGPL can be used to discourage use of code by competitors.

For example, as a service provider you can release the code to your service as AGPL, knowing that no-one else can run a competing service without sharing their modifications with you. In this way you can ensure that all services based on the code have effectively the same level of capabilities. This makes sense when thinking about the distributed social networking projects I mentioned earlier, as there is greater benefit in having a consistent distributed social network than having feature differentiation among hosts.

However, in many other applications, differentiation in services is a good thing for users. For an ERP system like Kuali, there is little likelihood of anyone adopting such a system without needing to make modifications – and releasing them back under AGPL. It would certainly be difficult for another SaaS provider to offer something that competes with Kuali using their software based on extra features, as any improvements they could make would automatically need to be shared back with Kuali anyway. They would need to compete on other areas, such as price or support options.

But back to Chuck's post – what do we make of the arguments he makes against AGPL?

If we look back at the four principles of open source that I used to start this article, we quickly can see how AGPL3 has allowed clever commercial companies to subvert the goals of Open Source to their own ends:

- ***Access to the source of any given work** – By encouraging companies to only open source a subset of their overall software, AGPL3 ensures that we will never see the source of the part (b) of their work and that we will only see the part (a) code until the company sells itself or goes public.*
- ***Free Remix and Redistribution of Any Given Work** – This is true unless the remixing includes enhancing the AGPL work with proprietary value-add. But the owner of the AGPL-licensed software is completely free to mix in proprietary goodness – but no other company is allowed to do so.*
- ***End to Predatory Vendor Lock-In** – Properly used, AGPL3 is the perfect tool to enable predatory vendor lock-in. Clueless consumers think they are purchasing an "open source" product with an exit strategy – but they are not.*
- ***Higher Degree of Cooperation** – AGPL3 ensures that the copyright holder has complete and total control of how a cooperative community builds around software that they hold the copyright to. Those that contribute improvements to AGPL3-licensed software line the pockets of commercial company that owns the copyright on the software.*

On the first point, access to source code, I don't think there is anything special about AGPL. Companies like Twitter and Facebook already use this model, opening some parts of their code as Open Source, while keeping other parts proprietary. Making the open parts AGPL makes a difference in that competitors also need to release source code, so I think overall this isn't a valid point.

On the second point, mixing in other code, Chuck is making the point that the copyright owner has more rights than third parties, which is unarguably true. Its also true of other licenses too. I think its certainly the case that, for a service provider, AGPL offers some competitive advantage.

Chuck's third point, that AGPL enables predatory lock-in, is an interesting point. There is nothing to prevent anyone from forking an AGPL project – it just has to remain AGPL. However, the copyright owner is the only party that is able to create proprietary extensions to the code without releasing them, which can be used to give an advantage.

However, this is a two-edged sword, as we've seen already with MySQL and MariaDB; Oracle adding proprietary components to MySQL is one of the practices that led to the MariaDB fork. Likewise, if Kuali uses its code ownership prerogative to add proprietary components to its SaaS offering, that may precipitate a fork. Such a fork would not have the ability to add improvements without distributing source code, but would instead have to differentiate itself in other ways – such as customer trust.

Finally, Chuck argues that AGPL discourages cooperation. I don't think AGPL does this any more than GPL already does for Linux or desktop applications; what is new is extending that model to web services. However, it certainly does offer less freedom to its developer community than MIT or ASL – which is the point.

In the end customers do make choices between proprietary, Open Source, and Free Software, and companies have a range of business models they can operate when it comes to using and distributing code as part of their service offerings.

As Chuck writes:

> *It never bothers me when corporations try to make money – that is their purpose and I am glad they do it. But it bothers me when someone plays a shell game to suppress or eliminate an open source community. But frankly – even with that – corporations will and should take advantage of every trick in the book – and AGPL3 is the "new trick".*

As we've seen before, there are models that companies can use that take advantage of the characteristics of copyleft licenses and use them in a very non-open fashion.

There are also other routes to take in managing a project to ensure that this doesn't happen; for example, adopting a meritocratic governance model and using open development practices mitigates the risk of the copyright owners acting against the interests of the user and developer community. However, as a private company there is nothing holding Kuali to operate in a way that respects Free Software principles other than the terms of the license itself – which of course as copyright owner it is free to change.

In summary, there is nothing inherently anti-open in the AGPL license itself, but combined with a closed governance model it can support business practices that are antithetical to what we would normally consider “open”.

Choosing the AGPL doesn’t automatically mean that Kuali is about to engage in bad business practices, but it does mean that the governance structure the company chooses needs to be scrutinised carefully.