How to Achieve Vendor Lock-in with a Legit Open Source License – Affero GPL

September 8, 2014, 10:26 pm

Note: In this post I am not speaking for the University of Michigan, IMS, Longsight, or any one else. I have no inside information on Kuali or Instructure and am basing all of my interpretations and commentary on the public communications from the kuali.org web site and other publically available materials. The opinions in this post are my own.

Before reading this blog post, please take a quick look at this video about Open Source:

http://youtu.be/a8fHgx9mE5U

The founding principles of Open Source from the video are as follows:

1. Access to the source of any given work
2. Free Remix and Redistribution of Any Given Work
3. End to Predatory Vendor Lock-In
4. Higher Degree of Cooperation

A decade ago efforts like Jasig, Sakai, and Kuali were founded to collaboratively build open source software to meet the needs of higher education to achieve all of the above goals. Recently Kuali has announced a pivot toward Professional Open Source. Several years ago the Sakai and Jasig communities decided to form a new shared non-profit organization called Apereo to move away from Community Source and toward pure Apache-style open source. So interestingly, at this time, all the projects that coined the term “Community Source”, no longer use the term to describe themselves.

In the August 22 Kuali announcement of the pivot from non-profit open source to for-profit open source, there was a theme of how much things have changed in the past decade since Kuali was founded:

> …as we celebrate our innovative 2004 start and the progress of the last decade, we also know that we live in a world of change. Technology evolves. Economics evolve. Institutional needs evolve. We need to go faster. We need a path to a full suite of great products for institutions that want a suite. So it is quite natural that a 10-year-old software organization consolidates its insights and adapts to the opportunities ahead.

There were many elements in the August 22 announcement that merit discussion (i.e. here and here) but I will focus on these particular quotes from the FAQ that accompanied the August 22 announcement:

> This plan is still under consideration. The current plan is for the Kuali codebase to be forked and re-licensed under Affero General Public License (AGPL).
>
> The Kuali Foundation (.org) will still exist and will be a co-founder of the company. … The Foundation will provide initial capital investment for the company out of its reserves.

In a follow-up post five days later on August 27 they clarified the wording about licensing and capital:

> All software that has been released under the current, Open Source Initiative approved Educational Community License (ECL) will and can continue under that license.
>
> The software license for work done by the new entity and from its own capital will be the Open Source Initiative approved Affero GPL3 license (AGPL3).

While the details and overall intent of the August 22 and August 27 announcements from the Kuali Foundation may seem somewhat different, the AGPL3 license remains the central tenet of the Kuali pivot to professional open source.

The availability of the AGPL3 license and the successful use of AGPL3 to found and fund "open source" companies that can protect their intellectual property and force vendor lock-in *is* the "change" that has happened in the past decade that underlies both of these announcements and the makes a pivot away from open source and to professional open source an investment with the potential for high returns to its shareholders.

Before AGPL3

Before the AGPL3 license was created, there were two main approaches to open source licensing – Apache-style and GPL-style. The Apache-like licenses (including BSD, MIT, and ECL) allow commercial companies to participate fully in both the active development of the code base and the internal commercial use of that code base without regard to mixing of their proprietary code with the open source code.

The GNU Public License (GPL) had a "sticky" copyleft clause that forced any modifications of redistributed code to also be released open source. The GPL license was conceived pre-cloud and so its terms and conditions were all about distribution of software artifacts and not about standing up a cloud service with GPL code that had been modified by a company or mixed with proprietary code.

Many companies chose to keep it simple and avoided making any modifications to GPL software like the Linux kernel. Those companies could participate in Apache projects with gusto but they kept the GPL projects at arms length. Clever companies like IBM that wanted to advance the cause of GPL software like Linux would hire completely separate and isolated staff that would work on Linux. They (and their lawyers) felt they could meet the terms of the GPL license by having one team tweak their cloud offerings based on GPL software and a completely separate team that would work on GPL software and never let the two teams meet (kind of like matter and anti-matter).

So clever companies could work closely with GPL software and the associated projects if they were very careful. In a sense because GPL had this "loophole", while it was not *easy* for commercial companies to engage in GPL projects when a company tweaked the GPL software

for their own production use, it was *possible* for a diverse group of commercial companies to engage constructively in GPL projects. The Moodle project is a wonderful example of a great GPL project (well over a decade of success) with a rich multi-vendor ecosystem.

So back in 1997, the GPL and Apache-like licenses appeared far apart – in practice as the world moved to cloud in the past decades the copyleft clause in GPL became less and less of a problem. GPL licensed code could leverage a rich commercial ecosystem almost as well as Apache licensed code. The copyleft clause in GPL had became much weaker by 2005 because of the shift to the cloud.

AGPL – Fixing the "loophole" in GPL

The original purpose of the GPL license was to insist that over time all software would be open source and its clause to force redistribution was a core element.

> For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

The fact that these cloud vendors could "have their cake and eat it too" could be easily fixed by making the AGPL3 license tighter than the GPL license by adding this clause:

> The GNU Affero General Public License is designed specifically to ensure that, in such cases, the modified source code becomes available to the community. It requires the operator of a network server to provide the source code of the modified version running there to the users of that server. Therefore, public use of a modified version, on a publicly accessible server, gives the public access to the source code of the modified version.

This seems simple enough. Fix the flaw. The GPL license did not imagine that someday software would not be "distributed" at all and only run in the cloud. The AGPL3 license solves that problem. Job done.

But solving one problem in the GPL pathos causes another in the marketplace. AGPL3 ensures that we can "see" the code that those who would remix and run on servers would develop, but it creates an unfortunate asymmetry that can be exploited to achieve a combination of vendor lock-in and open source.

AGPL3 = Open Source + Vendor Lock-In

The creators of GPL generally imagined that open source software would have a diverse community around it and that the GPL (and AGPL) licenses were a set of rules about how that community interacted with each other and constrain companies working with GPL software to bring their improvements back to the commons. But just like the GPL founders did not imagine the cloud, the AGPL creators did not imagine that open source software could be created in a proprietary organization and that the AGPL license would ensure that a diverse community would never form (or take a really long time to form) around the open source software.

These days in Educational Technology it is pretty easy to talk to someone on your Caltrans commute and get $60 Million in venture capital for an educational technology startup. But your

VC's want an exit strategy where they make a lot of money. I think that there are likely no examples of VC-funded companies that used an Apache-like license in their core technology that were funded let alone successful. That hippie-share-everything crap just does not cut it with VC's. Vendor lock-in is the only way to protect asset value and flip that startup or go public.

Clever company founders figured out how to "have their cake and eat it too". Here is the strategy. First take VC money and develop some new piece of software. Divide the software into two parts – (a) the part that looks nice but is missing major functionality and (b) the super-awesome add-ons to that software that really rock. You license (a) using the AGPL3 and license (b) as all rights reserved and never release that source code.

You then stand up a cloud instance of the software that combines (a) and (b) and not allow any self-hosted versions of the software which might entail handing your (b) source code to your customers.

Since the (a) portion is incomplete it poses no threat to their commercial cloud offering. And since the (a) part is AGPL it is impossible for a multi-vendor commercial ecosystem to emerge. If a small commercial competitor wants to augment the (a) code to compete with the initial vendor that has (a)+(b) running in the cloud, they are bound by the AGPL3 license to publish all of their improvements. This means that if the second company comes up with a better idea than the original company – the original company gets it and any and all competitors of the second company get the improvement for free as well. But if the original company makes an improvement – they keep it hidden and proprietary thus extending their advantage over all other commercial participants in the marketplace:

You can see this theme in the August 22 Kuali FAQ where they talk about "What happens to the Kuali Commercial Affiliates (KCAs)?":

> There will be ample and growing opportunities for the KCAs to engage with Kuali clients. The company would love for KCAs to take on 80% or more of the installation projects. The Kuali platform will continue to become more and more of a platform that KCAs can augment with add-ons and plugins. In addition, KCAs will likely be used to augment the company's development of core code and for software projects for Kuali customers.

Reading this carefully, the role for companies other than Kuali, Inc. is to install the software developed by the new "Kuali, Inc." company or perhaps develop plugins. With the source code locked into AGPL3, the greatest role that a community of companies can do is be "Kuali Inc's little helpers". The relationship is not a peer relationship.

When a company builds a proprietary product from scratch and releases a portion of it under APGL3, there never was a commons and the AGPL3 license is the best open source license the company can use to insure that **there never will be a true commons**.

Revisiting – AGPL – Fixing the "bug" in GPL (oops)

Now the AGPL3 advocates actually achieve their goals when the original company goes out of business because even though we never see the (b) component of the software, since the (a) part is open source and a truly open ecosystem could emerge around the carcass of the company – but by the time the company failed – it is not likely that their "half-eaten code carcass" would be all that useful.

What is far more likely is that the company using the AGPL strategy would get a few rounds of VC, thrive and sell themselves for a billion dollars or go public for a few billion dollars. After the founders pocket the cash, there would no longer need to market themselves as "open source" so they would just change the license on (a) from AGPL3 to a proprietary license and stop redistributing the code. Since the (b) code was always proprietary – after a few months of improvements to the (a) code in a non-open source fashion and the deep interdependence of the (a) and (b) code, the open copy of (a) has effectively died on the vine. The resulting company has a wonderfully proprietary and closed source product with no competitors and the VC's have another half-billion dollars to give to some new person on a Caltrans ride. And the "wheel of life" goes on.

Each time open source loses and VCs and corporations win, I am sure somewhere in the world, about ten Teslas get ordered and a puppy cries while struggling to make it to the next level.

Proprietary Code is a Fine Business Model

Probably by this time (if you have read this far) you probably have tagged this post as **#tldr** and **#opensourcerant** – it might indeed warrant **#tldr** – but it is not an open source rant.

I am a big fan of open source but I am also a big fan of proprietary software development. The educational technology market is made up of well over 90% of its software that is proprietary. Excellent proprietary offerings come from companies like Blackboard, Coursera, Instructure (part b), Piazza, Microsoft, Google, Edmodo, Flat World Knowledge, Pearson, McGraw Hill, Apple and many others. Without them open source efforts like Sakai and Moodle would not exist. I am not so foolish that I believe that purely open source solutions will be sufficient to meet the need of this market that I care so much about.

The right combination in a marketplace is a combination of healthy and competitive open source and proprietary products. This kind of healthy competition is great because choices make everyone stronger and keep teams motivated and moving forward:

- Linux and Microsoft Windows
- Microsoft Office and LibreOffice
- Sakai and Blackboard
- Apache HTTPd and Microsoft IIS
- ….

The wisest of proprietary companies even see fit to invest in their open source competitors because they know it is a great way to make their own products better.

The reason that the "open source uber alles" strategy fails is that proprietary companies can raise capital far more effectively than open source efforts. This statement from an earlier Kuali blog post captures this nicely:

> We need to accelerate completion of our full suite of Kuali software applications, and to do so we need access to substantially more capital than we have secured to date to meet this need of colleges and universities.

The problem is also why it is very rare for an open source product to dominate and push out proprietary competitors. Open source functions best as a healthy alternative and reasonably calm competitor.

AGPL3 + Proprietary + Cloud Strategy in Action

To their credit, Instructure has executed the AGPL3 open/closed hybrid strategy perfectly for their Canvas product. They have structured their software into two interlinked components and only released one of the components. They have shaded their marketing the right way so they sound "open source" to those who don't know how to listen carefully. They let their fans breathlessly re-tell the story of "Instructure Open Source" and Instructure focuses on their core business of providing a successful cloud-hosted partially open product.

The Kuali pivot of the past few weeks to create Kuali Inc., (actual name TBD) is pretty clearly an attempt to replicate the commercial success of the Instructure AGPL3 strategy but in the academic business applications area. This particular statement from the August 22 Kuali announcement sums it up nicely:

> From where will the founding investment come?
>
> The Foundation will provide initial capital investment for the company out of its reserves. Future investment will come from entities that are aligned with Kuali's mission and interested in long-term dividends. A first set of investors may be University foundations. There is no plan for an IPO or an acquisition.

Read this carefully. Read this like a lawyer, venture capitalist, or university foundation preparing to invest in Kuali, Inc. would read it. The investors in Kuali, Inc. may be more patient than the average investor – but they are not philanthropic organizations making a grant. The AGPL license strategy is essential to insuring that an investment in Kuali, Inc. has the potential to repay investors investments as well as a nice profit for its patient investors.

Is there any action that should be taken at this time? If I were involved in Kuali or on the board of directors of the Kuali Foundation, I would be very suspect of any attempted change to the license of the code currently in the Kuali repository. A change of the kind of license or a change to "who owns" the code would be very significant. The good news is that in the August 27 Kuali post it appears that at least for now, a board-level wholesale copyright change is off the table.

> All software that has been released under the current, Open Source Initiative approved Educational Community License (ECL) will and can continue under that license.

I think that a second issue is more about the individual Kuali projects. There are lots of Kuali projects and each project is at a different maturity level and has its own community and its own leadership. I think that the approach to Kuali, Inc. might be different across the different Kuali Foundation projects. In particular if a project has a rich and diverse community of academic and commercial participants, it might be in that communities' best interest to ignore Kuali Inc. and just keep working with the ECL licensed code base and manage its own community using open source principles.

If you are a member of a diverse community working on and using a Kuali project (Coeus and KFS are probably the best examples of this) you should be careful not to ignore a seemingly innocuous board action to switch to AGPL3 in any code base you are working on or depending

on (including Rice). Right now because the code is licensed under the Apache-like Educational Community License, the fact that the Foundation "owns" the code hardly matters. In Apache-like licenses, the owner really has no more right to the code than the contributors. But as soon as the code you are working on or using is switched to AGPL3, it puts all the power in the hands of the copyright owner – not the community.

A worrisome scenario would be to quietly switch the license to AGPL3 and then have the community continue to invest in the Kuali Foundation version of the code for a year or so and then a year from now, the Kuali Foundation Board could then transfer ownership of the code to someone else and then you would have to scramble and pick through the AGPL3 bits and separate them out if you really wanted to continue as a community. This is usually so painful after a year of development that no one ever does it.

The Winter of AGPL3 Discontent

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.

So AGPL3 is the perfect open source license for a company that thinks open source sounds great but an actual open community is a bad idea. The saddest part is that most of the companies that were using the "loophole" in GPL were doing so precisely so they could participate in and contribute to the open source community.

Conclusion

As I wrote about MySQL back in 2010, a copyright license alone does not protect an open source community:

> Why an Open Source Community Should not cede Leadership to a Commercial Entity – MySql/Oracle

> Many people think that simply releasing source code under an open license such as Instructure or GPL is "good enough" protection to ensure that software will always be open. For me, the license has always been a secondary issue – what matters is the health and vitality of the open community (the richness and depth of the bazaar around the software).

Luckily, the MySQL *community* saw the potential of the problem and made sure that they had a community-owned version of the code named MariaDB that they have actively developed from the moment that Oracle bought MySQL. I have not yet used MariaDB – but its existence is a reasonable insurance policy against Oracle "going rogue" with MySQL. So far, now over four years later Oracle has continued to do a reasonable job of managing MySQL for the common good so I keep using it and teaching classes on it. But if MariaDB had not happened, by now the game would likely be over and MySQL would be a 100% proprietary product.

While I am sure that the creators of the Affero GPL were well intentioned, the short-term effect of the license is to give commercial cloud providers a wonderful tool to destroy open source communities or at least ensure that any significant participation in an open-source community is subject to the approval and controls of the copyright owner.

I have yet to see a situation where the AGPL3 license made the world a better place. I have only seen situations where it was used craftily to advance the ends of for-profit corporations that don't really believe in open source.

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".

Instead of hating corporations for being clever and maximizing revenue – we members of open source communities must simply be mindful of being led down the wrong path when it comes to software licensing.

Note: The author gratefully acknowledges the insightful comments from the reviewers of this article.