OpenStack: Case for Adopting the DCO

This memo discusses problems with OpenStack's current contributor license agreement (CLA) system and proposes adoption of an approach using the Developer Certificate of Origin (DCO).

Background

Overview of CLAs

Open source projects approach the issue of legal management of inbound contributions in various ways. The vast majority of projects make no distinction between inbound contribution licensing and outbound project licensing (for example, contributions to an Apache License 2.0 (ALv2) project are licensed under ALv2). A minority of projects depart from the typical approach by using formal contributor agreements that either supplement or replace the use of the project's outbound open source license. CLAs, which themselves take a variety of forms, are one category of formal contributor agreement. The DCO is another type of contributor agreement.

A CLA is a license agreement between the contributor and some inbound entity (generally a company or nonprofit organization). A CLA at a minimum provides for copyright licensing of contributions submitted to the project and often also specifies patent licensing. Typically-encountered CLAs, including the CLAs used by OpenStack which are based on those used by the Apache Software Foundation (ASF), have a number of features that give them an *asymmetric* quality in relation to the outbound open source license of the project:

- They name a specific inbound licensee, while open source licenses are grants to everyone
- They require the contributor to grant a broader set of rights to the inbound licensee than that licensee in turn grants to the general public through the project's open source license
- They must be signed by the contributor, while open source licenses are not signed instruments
- They contain contractual provisions not found in standard open source licenses, or omit conditions found in the project's outbound open source license, in ways that benefit the inbound licensee at the expense of the contributor.

Like the similar mechanism of copyright assignment, asymmetric CLAs have been politically controversial in open source developer communities. One reason is that typical CLAs reflect or reinforce inequalities in political or legal power in the project community. Another reason is that they invariably introduce some red tape and delay into the open source development process, which is ordinarily relatively frictionless. Politically-controversial contributor agreements have sometimes played a key role in driving developer preferences for alternative technologies and have sometimes been a factor leading to project forks.

There is reason to believe that asymmetric contributor agreements (including copyright assignment agreements as well as ASF-style CLAs), never characteristic of most open source projects to begin with, are now in declining use, possibly due to an upsurge in criticism of such agreements in the open source community during the past several years.

Note on ASF use of CLAs

As noted above, OpenStack uses CLAs based on those used by the ASF. The standard ASF CLA is the Individual Contributor License Agreement (ICLA), signed by individual developers. The ASF also uses a Corporate CLA (CCLA) which is intended to be signed by a representative of an employer of several developers contributing to a project, who are expected to be named in an appendix to the CCLA. While ASF-style CLAs have been controversial outside the ASF, the ASF's CLA system seems well accepted by the Apache developer community. This may be because of the lenient administration of the CLA requirement by ASF projects, a fact which appears not to be well known outside the ASF.

For ASF projects, only contributors who attain *committer* status on a project are normally expected to sign an ICLA. Newer or casual contributors to a project are not expected to sign CLAs at all, instead licensing their contributions directly under ALv2. (The ASF points to a provision of ALv2, section 5, as a legal basis for this form of direct licensing, but it merely reflects the standard practice of most open source projects regardless of license.) As for the CCLA, the ASF regards it as something that is signed *instead of* a set of ICLAs, and only if a corporate contributor prefers to do so. The ASF does not request that CCLAs be signed. The ASF also grants significant discretion to project leads to determine how best to administer the CLA requirement.

OpenStack has not followed the liberal approach to CLAs employed by the ASF itself.

OpenStack's use of CLAs

OpenStack has imposed an ICLA and CCLA requirement since the project's launch in July 2010. Initially, Rackspace followed an approach used by all other known projects using ASF-style CLAs (including Apache projects themselves), which we will call an *either/or* approach: contributors were expected to sign an ICLA or else have their employer sign a CCLA. In November 2010, for reasons that have not been publicly recorded, Rackspace began to require that all contributors sign the ICLA and, in addition, that employers of all contributors sign a CCLA, departing from the either/or practice. The unusual dual ICLA/CCLA requirement continues to this day under the OpenStack Foundation's management. (A special carveout was made early on for U.S. government agency employees: these contributors are not required to sign the ICLA, but their employers must sign a U.S. government CLA.) The texts of Rackspace's CLAs were carried over when the Foundation was formed with no substantive changes, and they are incorporated into the Foundation's bylaws as part of its intellectual property policy.

There appears to be some uncertainty about the scope of the CLA requirement in the OpenStack technical community, but today it is assumed to be quite broad, applying to most development activities even where the material being developed never finds its way in a production distribution or deployment of OpenStack. The CLA requirement is applied not only to projects in the integrated release and projects under incubation, but also documentation, project infrastructure, QE tools, developer tools, and deployment tools. Over 90 percent of the projects on stackforge, including such things as Puppet and Chef recipes, impose the CLA requirement.

We noted above that OpenStack is unusual in having a dual ICLA/CCLA requirement per contributor. Another way in which OpenStack's administration of its CLA system is relatively

strict is its absence of any exception mechanism for small or trivial contributions. In our experience, most projects that impose a contributor agreement requirement have explicit, de facto or discretionary policies that allow minor patches in even if a contributor agreement is not signed, although in some cases this only occurs following some negotiation between the contributor and the project administrator. (One good example is the ASF, since Apache projects normally expect ICLAs to be signed only by those who have earned committer status.) The Legal Affairs Committee informally recommended against adoption of a procedure to exempt trivial contributions from the CLA requirement. The absence of any CLA carveout for trivial contributions is something that mainly affects new contributors to OpenStack.

Problems with OpenStack's CLA system

Excessive barriers for new contributors

OpenStack has placed importance on its ability to attract new contributors, and it has been successful in building a large and diverse developer community, drawn mainly from employees of the vendors collaborating on OpenStack. However, we are concerned that the CLA system is a factor that is limiting OpenStack's success in attracting other kinds of contributors that it needs.

There is growing realization that for its longer-term viability OpenStack must improve its ability to get feedback from OpenStack operators and to enable operators to influence the project's technical direction. One important form of operator feedback consists of patches to code, documentation and deployment tools. Any excessive friction introduced by OpenStack into its contribution process conflicts with the goal of increasing operator feedback. As Stefano Maffulli put it in on legal-discuss in April: *"Having to sign a Corporate CLA and Individual CLA for a trivial patch, from an operator (whose job is to run clouds, resulting in small and rare patches, not to develop large features) can conflict with our effort to get more operators involved in OpenStack."*

Over the past year we have seen what seems like an increase in anecdotal evidence of new contributors being unreasonably delayed or blocked from participation in OpenStack due to hurdles in the contribution process, often (though not exclusively) related to the CLA requirement. While the anecdotes may seem to involve a limited set of cases, we suspect they are indicative of a larger problem, one in which potential new contributors silently walk away from OpenStack without contributing, or, at best, complete the process but are left with a bad first impression of a project that seems actively hostile to new contributors.

Some of these anecdotal cases seem to reveal an inclination by contributors and project leads to find authorized ways to circumvent the hurdles of the CLA process. For example, both the ICLA and CCLA have a provision allowing for the CLA signer to submit contributions "on behalf of" a third party, which simply requires "identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware". We anticipate that if the CLA system is retained in its current form, such provisions will be used creatively as CLA evasion mechanisms in order to reduce the friction for new contributors caused by the process.

Recruitment of skilled developers and OpenStack's "image problem"

For continued technical success, OpenStack must attract the highest-skilled developers with expertise in areas relevant to the project, but such developers are in short supply. The OpenStack technical community tries its best to be make the project welcoming to new contributors, but here too we have received anecdotal evidence of a problem rooted in the CLA system: it appears to be contributing to an "image problem" for OpenStack. In conversations with OpenStack developers at conferences, peer developers not otherwise affiliated with an OpenStack corporate member question the legitimacy of OpenStack as a true open source project, pointing to the CLA as proof that OpenStack is controlled by narrow corporate agendas rather than a drive for technical excellence (since this is how CLA use is commonly perceived in open source development communities). In addition, the administration of the CLA system is regarded as proof that the project is unreasonably hostile to new and unaffiliated individual contributors. We do not claim that this is a large set of potential OpenStack contributors, but each one who turns away from participating in the project is one who has the skillset to have made a significant positive impact on OpenStack development.

Organizational costs

The Board should be aware that its CLA system, as currently administered, entails significant organizational costs. The OpenStack Infrastructure team has had to put a significant amount of effort into integrating Gerrit's CLA support with the OpenStack Foundation member database. The system must be maintained, tested and reintegrated with new versions of Gerrit. Moreover, when the system breaks down (as it inevitably will, usually due to external factors), new contributors are prevented for purely technical reasons from having patches accepted until the system is fixed. If the Foundation should wish to evolve the current system rather than replace it with an easily-administered DCO approach, the work needed to be done by the Infrastructure team to implement such changes should not be underestimated.

Legal complexity

By requiring a system of three different CLAs in addition to ALv2, the Foundation has increased the legal complexity of OpenStack for its corporate and individual developers and users. We have noticed that even some lawyers involved in OpenStack seem confused about the nature of the differences between the CLAs and ALv2. Use of one familiar open source license for both inbound contributions and outbound releases, which can be achieved through the DCO, would eliminate this complexity.

The DCO and its benefits

Despite the problems associated with the current CLA system, we acknowledge that the CLAs are intended to achieve two valuable objectives: (1) have some explicit, affirmative licensing act by contributors beyond merely contributing explicitly or implicitly under ALv2; (2) enable distribution of OpenStack releases under ALv2. These objectives do not require continued use of OpenStack's CLA system. We recommend use of a DCO system instead.

Overview of the DCO

The DCO was adopted by the Linux kernel project in the wake of the SCO lawsuits. (It is thus approximately the same age as the ASF's CLAs.) The original version was drafted by Diane Peters, then general counsel of Open Source Development Labs, one of the predecessors of the Linux Foundation. James Bottomley has described the DCO as a "best practices CLA", though most developers view the DCO as a different category of contribution mechanism. A fraction of the length of each of the OpenStack CLAs, the DCO as used by the Linux kernel project can be quoted here in full:

```
By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I have the
right to submit it under the open source license indicated in the file; or

(b) The contribution is based upon previous work that, to the best of my
knowledge, is covered under an appropriate open source license and I have the
right under that license to submit that work with modifications, whether
created in whole or in part by me, under the same open source license (unless
I am permitted to submit under a different license), as indicated in the file;
or

(c) The contribution was provided directly to me by some other person who
certified (a), (b) or (c) and I have not modified it.

(d) I understand and agree that this project and the contribution are public
and that a record of the contribution (including all personal information I
submit with it, including my sign-off) is maintained indefinitely and may be
redistributed consistent with this project or the open source license(s)
involved.
```

Note that the DCO does not include a special license grant to a foundation or other organization, but instead incorporates by reference the open source license of the project. For a GPLv2 project like the Linux kernel, the DCO includes a certification of licensing under GPLv2. For an ALv2 project like OpenStack, use of the DCO would constitute a certification of licensing under ALv2.

The certification of the DCO is referenced in a "signoff" included by the contributor in the commit message for a given patch. For example:

```
From: Doug Hellmann <doug.hellmann@dreamhost.com>
Date: Wed, 12 Jun 2013 15:05:13 +0000 (-0400)
Subject: Update stevedore to 0.9
X-Git-Url:
https://review.openstack.org/gitweb?p=openstack%2Frequirements.git;a=commitdif
f_plain;h=e154cadf3eb545e310a4527a9631f3c27b75c321
```

```
Update stevedore to 0.9

Use a version of stevedore that does not try to install
distribute.

Change-Id: Ib3d635b1899e1a59a08e10df103d3e93e4bb0ec6
Signed-off-by: Doug Hellmann <doug.hellmann@dreamhost.com>
```

Growth in use of the DCO

We have noticed an increase in use of the DCO by projects in recent years. In at least some cases the motivation seems to be to use a lighter-weight, politically noncontroversial alternative to use of a CLA. In addition to the Linux kernel, other well-known projects using the DCO today include Docker, Ceph, Qemu, OpenDaylight, Open vSwitch, GlusterFS, Git, Samba, and oVirt. Docker (which recently switched from use of ASF-style CLAs to the DCO) and Open vSwitch are examples of projects that use the DCO in conjunction with ALv2. The Linux Foundation appears to be standardizing on use of the DCO with all of its “Collaborative Projects”.

Most of the corporate members of the OpenStack Foundation are already contributing to DCO-using projects. It therefore cannot be claimed that the DCO is new, unfamiliar, or per se objectionable to these participants.

Advantages of DCO over CLAs

The DCO achieves the same realistic legal objectives as the OpenStack CLAs but in a form that entails no friction or contribution barriers, fits naturally in existing developer workflows, and has no history of political controversy among developers; indeed the drive to use the DCO typically comes from developers. This, in a nutshell, is the main benefit to use of the DCO. As noted, the DCO does not have the legal asymmetry and complexity inherent in the OpenStack CLAs, because under the DCO all contributors license contributions on an equal basis to the public under ALv2. Administration of the DCO is zero-cost for both the Foundation and first-time contributors, since it is integrated into git and the signoff can also be included in contributions made through email or a bug-tracking system.

Another benefit in using the DCO is that the certification applies separately to each contribution, while a CLA is signed once by each contributor and is then presumed to cover all contributions. With the DCO there is thus greater legal certainty concerning the licensing of each contribution (much as there might be if a CLA were signed anew each time a patch was submitted).

The DCO’s detailed record of patch signoffs has also been described as establishing a “chain of custody” for contributions going beyond what is evident in inspection of commit logs, including cases where a patch is authored by more than one individual. The CLA system is not capable of providing such information.

Comments on general objections and issues raised by Foundation counsel

General comments on risk

Arguments in favor of retention of the CLA system seem to be focused on two perceived areas of legal risk that the CLA system is thought to address: (1) potential repudiation or revocation of copyright licenses granted by contributors, and (2) patent assertions against OpenStack by present-day OpenStack participants or future acquirers of their patents.

Leaving aside the topic of corporate authority, raised by Mark Radcliffe, which we address below, we note that the scope of the copyright and patent license grants in the CLAs and in a DCO + ALv2 system are similar, if not identical, and it must be acknowledged that some uncertainty in license grant scope is unavoidable in either approach. As for the matter of repudiation or revocation of copyright licenses, there is no known history of this occurring in three decades of active use of open source licensing, so we cannot take this risk very seriously relative to the costs of the CLA system. Furthermore, it is difficult to see why the certification in the DCO is not as powerful as any representations made through a CLA.

As for the more serious matter of patent risk, we must point out that neither the CLAs nor ALv2 can do very much about this problem. Neither are designed to prevent patent assertions where the patent claim in question is not somehow captured by a code contribution to the project. The likelihood that a given patent claim assertable against OpenStack, even if currently or formerly owned by an OpenStack contributor, will be traceable to a code contribution is vanishingly small. Serious efforts to contain patent risk traceable in some way to OpenStack participants must rely on mechanisms external to contributor agreements and open source licenses. It is common sense that the bulk of patent risk faced by a project such as OpenStack comes from patents acquired and held by entities having no contributor connection to OpenStack whatsoever.

Another important point about risk is that OpenStack forms only a small portion of the code needed for a production deployment of OpenStack. Even if CLAs could somehow affect the risk exposure of that small portion, it would do nothing about the hundreds of library and external dependencies needed to use OpenStack. Very little of that software originates in projects using similar CLAs (or DCOs, for that matter). We see no basis for the view that the minority *OpenStack* portion of the stack needed to run OpenStack is especially vulnerable to risk compared to the rest of the stack.

Perhaps the risk containment argument is based on purely psychological or marketing benefits assumed to result from use of CLAs. We do not think this is a strong enough argument when balanced against the problems associated with the CLA system and the potential benefits of a DCO system.

“Required intermediary” interpretation of ALv2

Much of Mark Radcliffe’s draft “legal framework” memo rests on an interpretation of ALv2 that is inconsistent with ten years of prevailing industry and community understanding of that license, now one of the most popular open source licenses and widely used in commerce.

The theory seems to be that ALv2 assumes the existence of a system (thought incorrectly to be typified by the ASF) in which IP is licensed in to a foundation that itself does not hold such IP but instead sublicenses contributor IP under ALv2. The contention in other words is that ALv2 cannot be used as a direct license; this matters because a key feature of the DCO is that it involves the contributor directly licensing the contribution under the terms of the project’s open source license.

We do not agree with the textual interpretation of ALv2 that leads to the conclusion that direct licensing is somehow dubious or impossible. If it were true that only non-IP-owning foundations could use ALv2, ALv2 would not legitimately be considered an open source license. But actual use of ALv2 in the real world -- including by the OpenStack Foundation itself and its predecessor Rackspace -- conflicts with the theory that ALv2 cannot be used as a direct license. As we noted above, non-committer contributors to ASF projects normally directly license their contributions under ALv2. ALv2 is a popular license today for projects launched by one vendor (countless examples come to mind, including well-known projects maintained by Google, Red Hat, VMWare, Microsoft, and others). Even though some such projects use CLAs, most of the code for such projects is directly licensed under ALv2 by the dominant vendor. There is code copyrighted by the OpenStack Foundation in current releases of OpenStack; it is obviously being directly licensed by the Foundation under ALv2.

Scope of ALv2 vs. CLA license grants

Mark Radcliffe’s legal framework memo suggests that ALv2’s copyright and patent license grants can be read as having narrower scope than the corresponding grants in the CLAs because of the manner in which the term “Work” is used. This seems to imply that the Foundation is not granting as broad a license to its users as it ought to be. We do not agree with this interpretation, but it can easily be addressed by adding a sentence to the DCO adapting language from the CLAs: “For purposes of this certification, ‘Work’ as used in the Apache License 2.0 means any of the projects owned or managed by the OpenStack Foundation.”

Corporate Authority Issue

Mark Radcliffe’s memo raises concerns about the authority of individual developers to make DCO certifications in a manner that will bind their employers.

We note that this concern provides no basis for arguing that the *ICLA* is preferable to the DCO. It is only a basis for preferring the *CCLA* to the DCO.

The existing CLA system does not itself avoid the corporate authority issue, and it may not be practical for it to do so. This is because, while the ICLA requirement is enforced strictly through Gerrit (such that no patch can enter a CLA-using OpenStack project unless an ICLA has been signed by the contributor), the CCLA requirement apparently cannot be practically enforced today other than through mere documentation of the requirement as presented to new individual

contributors. In addition, it is not practical to police the requirement for CCLA-signing employers to keep the list of authorized employee contributors up to date. And of course there is no way of knowing whether the person signing a CCLA actually has corporate authority to do so.

This difference in administration of the ICLA and CCLA requirements actually suggests that the entire OpenStack CLA system is upside-down in its approach: effort is focused on ensuring ICLA signature when in fact the more realistic source of risk is associated with employers rather than individual contributors.

In any event, there is a way to address the concern about corporate authority through use of the DCO. In addition to per-patch signoffs from individual developers, an authorized corporate representative from each employer could do a per-release signoff covering all contributions from employees reflected in an entire integrated release. Such a requirement would likely be easier for Foundation staff to administer than the CCLA. OpenDaylight has considered implementing a two-level DCO approach of this sort.