

DRAFT

To: OASIS Open Projects Advisory Committee
From: Jamie Clark, OASIS Open
Date: 21 September 2021
Re: Consultation on Emerging IP issues and practices in Open Projects

This note is an update on some emerging intellectual property rules and practices issues associated with our Open Project (OP) outputs and repos. These are internal, preliminary views only, not policy – we’re still thinking this stuff through – so this represents no-one’s official position so far.

We crafted our original rules for the hybrid open source-open standards OP program in 2017, with a lot of help from many major IPR experts on various sides of the issues. The program’s intended to provide the best of both FOSS and standards methodology. We’ve been running it in production for 3+ years now, which makes us somewhat unusual. As we gain experience we learn more about how the program works, when it doesn’t, what it’s good for and what people want from it.

To thrive, any stable licensing scheme that’s widely relied upon by developers, governments, regulators and users – like ours – needs to work reasonably well and reasonably predictably for *all* stakeholders: proposers, governing participants, casual contributors, other license-givers, implementers, end users, and other license-reliers. So there always are balancing issues We try to keep that balance in mind when evolving our rules, too. Here are a few emerging issues that we think might drive future tweaks to rules, tools, or infrastructure. Some may mature into rule change proposals to our OASIS Board of Directors. At this stage, these are rough notes, not policy positions. Your thoughts are warmly welcome. You’re also warmly welcome to tell us that all or part of this thinking is wrong. Wouldn’t be the first time :) We’re designing and running novel stuff here, and evolution is expected.

1. **Additional licenses?** As you know, we allow our open projects to declare repos under license terms from a controlled list²: Apache v2, Eclipse v1, Eclipse v2, BSD 3-clause, MIT, CCBY v2, CCBY v4 and CC-0. These are the terms under which all input contributions¹ may be made. We’ve always assumed we will add more license choices over time.

Right now there are a few likely candidates based on demand, including the CLDA and AGPL. The question of which licenses we permit or encourage in works that are *shared or submitted to external SDOs* is separate (see Q3) – this is just about which input licenses we will support for *any* hosted collaborative work. Adding new license options does raise some policy questions, though.

1 As you probably know, there are additional *nonassert* obligations, beyond the simple FOSS input commitments, borne by governing board members. Those are discussed separately below.

Traditionally our approach at OASIS has been to give our members a choice of license modes, and allow each project or committee to negotiate its own terms, within our parameterized choices. By and large for two decades that's worked well.² However, we do have some interest in only hosting projects with *stable* licensure. So:

- Does it matter if a license does not have OSI approval? We already have three CC non-OSI licenses. In 2018 it seemed to us that Creative Commons easily showed long-term stability and neutrality.
- Does it matter if the source is non-proprietary, is administered by a vendor-neutral host, and/or has a stability track record? Is there some other kind of diligence or filter that we should apply?³
- Does it matter if the license includes copyleft features? Our adding the AGPL, and also GPL if sought, obviously would raise that question. Though opinions varied, some argued that the EPL has copyleft aspects as well. Whether copyleft is suitable *for external submissions* (such as to ISO) is a separate question. See Q3. Here the simpler question is, would OASIS encounter any difficulties with having work under copyleft licenses “in the tent” at all? My personal bias is that there's no harm, and some benefit, in doing so.

2. Multiple licenses in a single project or output? In our old-style technical committees, there's a *one-minimum-license-mode per-committee* rule per committee, so IPR life is simpler: few issues about license compatibility or application come up within a TC's outputs. With OPs, though, multiple licenses within a single body of work are permitted. That mirrors coder community reality, but it does add additional governance questions.

- *Compatibility*: OASIS does not offer advice or assurance about which licenses work acceptably with which others. As with single choices of license, we think those issues are best left to the participants to determine and negotiate. (If we think that obvious license problems are ignored, we're certainly able to mention possible omissions or collisions to projects; we just don't make ourselves the deciders.) Also, bluntly, experts have genuine disagreements on these “combinatorics” issues; many other stable, well-managed SDOs and FOSS hosts similarly avoid giving implied legal advice.
- *Governance*: Who makes the licensing decisions? They may affect (a) what outputs can be assembled; (b) what kind of contributions may be accepted; and (c) recruiting

2 OASIS as a host does not push participants into a specific license choice. We expect founders and prospects, and then participants, to work out what's best for their project. Sometimes this is a subject of negotiation among stakeholders in project launch planning. We try to keep participants and leadership panels aware that these choices are their karma burden and expertise, not ours as host. By design, we are in the *fair-reporting-of-openly-negotiated-rule-based-outcomes* platform business, not the *dictating-licenses-to-stakeholders* business. This is as it should be, because ultimately it's the licensing from the voluntary contributors that creates value for the output. (There may be an analogy to publishing platforms for user-generated content here.)

3 Bearing in mind both (a) our bias that these should be participant decisions when possible, see fn 6, and (b) the risk that unilateral hosting of some licenses, like some ephemeral code or APIs, might just go poof.

– that is, which stakeholders are willing to work under, or to be obligated to support, what kinds of licenses? Each Open Project is governed by its governing board (PGB), whose board members have an elevated patent nonassertion duty, and vote to approve any final project outputs. So our design provided that those board members also carry the responsibility for license choices.

In contrast to our TCs, most PGBs can revisit their project’s license decisions at any time, simply by opening another repo under a new license – and doing so essentially is an invitation to submit contributions under that license. This is why we put PGBs, not single contributors or maintainers, in charge of that process, so that the parties with governance and higher license obligations retain some democratic control over expansion (or splintering) of the licensing ground rules for the outputs for which they’re responsible.

3. Licensure for external submissions. We originally designed the OP program in 2017 to provide for licensing at two levels:

- PGB members provide a TC-like, standards-level patent nonassert⁴ in favor of any conformant implementation of a finally approved specification; and
- each input contribution also bears the FOSS license terms associated with its repo.

As a regular submitter to other SDOs such as ISO6, in order for OASIS to provide the required assurances we make to sharing partners, both of those licensing tracks must meet or exceed the other SDO’s licensing requirements. The nonassert does, and we have been using it readily since 2009. Many of the input contribution FOSS licenses also do. But maybe not all:

- It’s possible that contributions bearing copyleft-style licenses like the GPL or AGPL would not be suitable for contributions under the ISO/ITU/IEC Common Patent Policy – as we would be required, and unable, to convey the aggregate work free of some of the copyleft requirements imposed on derivative works.
- “Implementer” and “non-implementer” classes were set up in our original OP rules⁵, to set aside a special class for licenses that may not be suitable for external submission, intended for documentation or ancillary work, but not core standards. Right now CC-0 is the only “non-implementer” license. Specially, contributions under a non-implementer-class license: (a) cannot be included⁶ in machine executable code in a finally approved OP specification⁷; and (b) do not have the benefit of the nonassertion

4 The “Specification NonAssertion Covenant” defined in our OP Rules at <https://www.oasis-open.org/policies-guidelines/open-projects-process/#ss1403>

5 <https://www.oasis-open.org/policies-guidelines/open-projects-process/#repository-specification-licenses-licenseList>

6 OP Rules Sec. 13.4(b): <https://www.oasis-open.org/policies-guidelines/open-projects-process/#project-specifications-approval-criteria> So work licensed as CC-0 is ineligible for final standards inclusion, if it’s code.

7 Our rules call this stage “Project Specification.” PS-stage work is considered a final, ready-to-implement standard, with the full benefit of all nonassertion obligations. These PSs are the OP equivalent of what our TC program calls “Committee Specifications.”

covenant from all PGB members⁸ So aggregate work licensed as CC-0 receives much less patent protection, and by itself may be ineligible for many external submissions.

Our initial assumptions may have been too simple: we saw the CC license options as likely to be used mostly for documentation or ancillary materials. We also may have expected clearer distinctions between textual materials and machine-executable methods. What is the most sensible approach, or at least what options should we leave available, for applying FOSS licenses to specification-type mixed material?

4. **CLAs in 2021.** Different hosts take different approaches to Contributor License Agreements or the equivalent. In 2017 our program was roughly modeled on Apache Foundation practices at the time. Some other projects (such as Linux kernel development) use a simpler, less legalistic and some argue less enforceable) “Developer Certificate of Origin”.⁹

The DCO sets a bar for simplicity that arguably makes some FOSS contributors reticent to agree to anything more. Recently, some hosts, reacting to that market demand for simplicity, have been softening their CLA language to make it more palatable.¹⁰ Is this a good trend, or a race to the bottom? Should we be collecting simpler commitments from casual contributors? Should we be collecting distinct, more acute commitments from repo maintainers with repo write access?

If we are collecting separate, corporate- or entity-level agreements from companies who employ a contributor, then is a DCO enough documentation at the individual dev to satisfy that responsibility?

As a standards host on which many of our members, and collaborating SDOs, depend, we have a responsibility to maintain clear, auditable evidence of all input license commitments we receive. Still, OASIS must continue to review how to offer the simplest, least burdensome terms necessary to collect the minimum commitments we need, and the manner in which we present them.

JBC

8 OP Rules Sec. 15.3 (under “PGB Covenant for Specifications”): <https://www.oasis-open.org/policies-guidelines/open-projects-process/#ss1403>

9 <https://developercertificate.org/>

10 See the 2021 rewrite of Eclipse’s “Contributor Agreement,” which pared back their prior form, and now embeds the DCO: <http://www.eclipse.org/legal/ECA.php>. Note, though, that it is written for the distinct role of a casual or intermittent contributor, not a “committer” with write permission.