VECNet - Open Source License

Alexander Vyushkov, James Domingo, Notre Dame Center for Research Computing

This document describes the rationale behind the selection of Mozilla Public License 2.0 as the open source license for VECNet software developed by VECNet institutions.

Current Status

This document is a draft. It currently only accounts for Notre Dame's policies, and therefore, it currently only applies to VECNet software written by Notre Dame employees. The policies of other VECNet institutions need to be reviewed, so this document can be updated to ensure compliance with all relevant policies.

Executive Summary

The project's proposal / contract requires the use of an open-source software license approved by the Open Source Initiative (http://opensource.org/). Per Notre Dame's IP policy (http://policy.nd.edu/policy_files/IntellectualPropertyPolicy.pdf), the University of Notre Dame holds copyright to the software written by staff programmers at CRC. Therefore, project staff at ND consulted the Office of Technology Transfer (OTT, http://ott.nd.edu/), whose mission includes "providing guidance in matters relating to University Intellectual Property".

Per OTT's guidance, the project leadership at ND should choose among the OSI's list of Popular Licenses (http://opensource.org/licenses) since VECNet is a new project (as opposed to contributing software to an existing open-source project which already has a license). The list contains 7 popular, widely-used licenses and 2 licenses with strong, active communities. Since VECNet software is not part of either of the latter two communities, then its license should be one from the 7 popular licenses.

These 7 licenses were categorized into 3 groups based on their requirements for sharing of source code when software is distributed:

- Reciprocal (GPL)
- Limited Reciprocal (LGPL, MPL)
- Permissive (Apache, BSD 2-clause, BSD 3-clause, MIT)

The Reciprocal group was eliminated (rejected) because of legal ambiguities associated with GPL (cite Van Lindberg's book). Both of the remaining license groups -- Limited Reciprocal and Permissive -- would be satisfactory/acceptable for the project. In order to make a license recommendation, we consulted other project documents for guidance.

In Master Collaborative Agreement (MCA), the "Data Sharing Guiding Principles" in Appendix B[1]

[1] Appendix B "VECNet DATA ACCESS AND CONTRIBUTION GUIDING PRINCIPLES"

state:

> "The intent of VECNet is to create a publically available facility for analysis and dissemination of data and information about vector-borne diseases and that the results of their work should similarly be broadly and publicly available."

In the spirit of keeping our work (our source code) as broadly and publicly available, we recommend a Limited Reciprocal license. This type of license will require external developers that modify VECNet software and distribute it, to contribute their VECNet modifications back to the project, thus ensuring these modifications continue to benefit the public.

Either one of the two Limited Reciprocal licenses (LGPL or MPL) would be satisfactory/ acceptable for the project. We recommend MPL 2.0 because it is much shorter and simpler to understand, than LGPLv3. LGLPv3 is "a set of additional permissions added to version 3 of the GNU General Public License", so the GPLv3 has to be distributed with LGPLv3. Furthermore, Lindberg recommends MPL because:

> *"The careful wording and excellent legal work that went into the MPL make it a good choice for open source licensing in a corporate context." (pg 211)*

Note that Lindberg is talking about MPL 1.1; MPL 2.0 was a major revision in 2012 that made the license even better. Lindberg also notes:

> *"In practice, the MPL is most often used for larger platform or infrastructural projects, whereas the LGPL is used for libraries and small pieces of well-defined accessory code." (pg 211)*

Given the scale of the VecNet project, MPL is the more appropriate license.

Rationale Details

Additional information that expands on stuff presented in the executive summary.

OSI's Popular Licenses

The Open Source Initiative (OSI) approves licenses for open-source software. Approved licenses are grouped into several categories. The categories were developed by OSI's License Proliferation Committee because it "... *thought that these more descriptive categories would help people initially picking a license to use one of the more popular licenses, thereby helping to reduce the numbers of different licenses commonly used.*"[2]

Accordingly, the licenses in the popular category are also listed on OSI's license home page:

[2] Section 4 "The Groups" in the OSI License Proliferation Report

● Apache License 2.0 ● BSD 3-Clause "New" or "Revised" license ● BSD 2-Clause "Simplified" or "FreeBSD" license ● GNU General Public License (GPL) ● GNU Library or "Lesser" General Public License (LGPL) ● MIT license ● Mozilla Public License 2.0	popular and widely used
● Common Development and Distribution License ● Eclipse Public License	have strong communities

In its 2006 License Proliferation Report, the committee explained why popularity was considered a category:

> *"While it might at first sight not seem appropriate for the popularity of a license to be significant in categorizing it, popular and long-established licenses have an important thing going for them: the existence of an established interpretive tradition and a well-developed set of expectations about correct behavior with respect to them. This is significant in reducing confusion and (especially in common-law countries) is even likely to condition judicial interpretation of the licenses."*

To identify "licenses that are popular and widely used or with strong communities", the committee "... *used statistics obtained from public sources to determine which licenses are widely used. We believed that there were a few licenses that, while not the most popular, were widely used within their communities and that these also belonged in this group."* The few licenses "with strong communities" are the Common Development and Distribution License (CDDL) and the Eclipse Public License (EPL).

The CDDL was developed[3] by Sun Microsystems in 2004, and is used by various open- source projects that originated at Sun, e.g., OpenSolaris (operating system), NetBeans (IDE[4]), GlassFish (Java application server). The Eclipse Public License (EPL) is used within the Eclipse community, which formed in 2001 when IBM released the source code for its Eclipse IDE under the Common Public License (CPL). Initially, the project was stewarded by the Eclipse Consortium. In 2004, the Eclipse Foundation replaced the consortium, and the EPL was developed to replace the CPL.

[3] From Mozilla Public License 1.1

[4] Integrated Development Environment = set of integrated tools used by programmers to write software

If a new open source project was developing software for one of these two communities, then the license used within that community is definitely a reasonable candidate for the new project's license. But in the case of VecNet, since its software is not part of either of those communities, its open-source license should, instead, be one of the 7 popular and widely-used licenses.

3 Groups of Popular Licenses

The 7 popular and widely-used licenses can be grouped based on their requirements for sharing source code.

Reciprocal *Strong CopyLeft*	**Limited Reciprocal** *Weak Copyleft*	**Permissive** *Academic, Research*
GPL	LGPL MPL (Mozilla)	Apache BSD (2-clause) BSD (3-clause) MIT

The alternative names for the groups (shown in italics above) are described in this very useful article: *A Quick Guide to Software Licensing for the Scientist-Programmer*.

To illustrate their requirements about sharing source code, consider the scenario where one developer, Harry, has written and distributed a software library outside his organization. Another developer at a different organization, Sally, has written a program that uses Harry's library. Sally wants to distribute her program outside her organization. The license that Sally can distribute her program with depends upon which license Harry distributed his library with, and whether Sally made any modifications to his library.

Sally uses Harry's code as is (unmodified)

If Harry distributes his library with this license:	**Reciprocal**	**Limited Reciprocal**	**Permissive**
Does Sally have to share the source code for her program?	Yes	Her decision	Her decision
If so, under what license?	Same	Her choice	Her choice

Sally modifies Harry's code

If Harry distributes his library with this license:	**Reciprocal**	**Limited Reciprocal**	**Permissive**

Does Sally have to share the source code for **her program**? If so, under what license?	Yes Same	Her decision Her choice	Her decision Her choice
Does Sally have to share the source code for **her changes to Harry's library**? If so, under what license?	Yes Same	Yes Same	Her decision Her choice

Legal Ambiguities with GPL

This section should summarize the legal ambiguities that Lindberg describes in his book (chapter 12). We don't want to replicate the whole chapter; just present the key points:

- Legal experts disagree whether linking to GPL code creates a derivative work under copyright law.
  - Free Software Foundation says in GPL that it does.
  - But Lindberg, and a Stanford law professor he cites, disagree (linking does NOT create a derivative work)

- *"In considering the differences between the LGPLv2 and LGPLv3, the best answer is probably dual licensing under both version 2 and version 3 as discussed relative to the GPL below." (pg 212)* Dual licensing means more effort to explain why there are two licenses to developers.