Understanding Open Source in an AVCDL Context

Revision

Version 3
4/22/24 2:02 PM

Author

Charles Wilson

Abstract

This document describes how open source software is considered within the context of the **AVCDL**.

Audience

The audience of this document are the cybersecurity development lifecycle practice leads who will be guiding **AVCDL** adoption within their organization.

Note: This document is not subject to certification body review.

License

Overview

The open source software [2] (**open source**) has become a large part of the total software contributing to any product. Because open source is not subject to the rigor required of safety-critical software, additional steps must be taken to enable its use within safety-critical products. Although the **AVCDL** [1] does not specifically discuss open source, it is useful to elaborate on how it might be managed within an **AVCDL** context.

Definition of Terms

The following are a some commonly seen terms [23, 24, 25, 26, 27, 28]:

Term	Working Definition
proprietary	Software that is owned by an individual or company. There are almost always major restrictions on its use. Its source code is almost always kept secret.
third-party	Software developed by an entity other than the original vendor of the development platform.
open-source	Software whose source code is made available under a license in which the copyright holder specific provides the rights to study, change, embody, and redistribute the software.
COTS (commercial off-the-shelf)	Items, including services, available in the commercial marketplace that can be bought and used under government contract.
public domain	Software that has been donated to the public domain by its copyright holder. Thus, it is no longer copyrighted. As such it is completely free and may be used by anyone for any purpose without restriction.
freeware	Proprietary software offered for use free of monetary charges. There are generally severe restrictions on its use and the source code is kept secret.
intellectual property	Property (such as an idea, invention, or process) that derives from the work of the mind or intellect.
copyright	The exclusive legal right to reproduce, publish, sell, or distribute the matter and form of something.
license	A grant by the holder of a copyright or patent to another of any of the rights embodied in the copyright or patent short of an assignment of all rights.

Note: Common term definitions are from Merriam-Webster online dictionary [29].

Open Source Use Tracking

When using open source software, it is important to track its use. The following table shows the minimal information required by the engineering and legal groups:

Element	Engineering	Legal	Notes
name	x	x	software name
version	x	x	as precise as possible
release date	x	x	needed in order to resolve some version discrepancies
form	x	x	source, library, executable, ... (impacts how the material is considered)
distribution	x	x	integration, static library, dynamic library, executable
language	x		C, C++, JavaScript, ...
license		x	legal will need to determine whether the license is acceptable on its own as well as in concert with the other licenses in use within the context of the product
consumer	x	x	project / release (where the material is to be used)
contact		x	URL, email, phone number
origin	x	x	URL (source of the material)
purpose	x		what it will be used for (user interface, data transfer, database, ...)

Shown in each row is information indicating the software's element of interest, whether engineering and/or legal would desire the information, and notes related to the element.

Major Licenses

The inclusion of open source software requires that we attend to the stipulations contained within their various licenses. The following table shows the obligations and usage allowed for various open source licenses:

License	Description
None	Without a license, the code is copyrighted by default. People can read the code, but they have no legal right to use it. To use the code, you must contact the author directly and ask permission.
Public Domain	Anyone may use the code for any purpose whatsoever. Nothing is in the public domain by default; you must explicitly put your work in the public domain if you want it there. Otherwise, you must be dead a long time before your work reverts to the public domain.
GPL	Code may never be statically linked to proprietary programs.
LGPL	Code may be statically linked to proprietary programs under certain very specific circumstances.
MIT/X11	Generic legal disclaimer of liability.
BSD	Requires legal disclaimer of liability with explicitly named organization.
Apache	Requires derivative works to provide notification of any licensed or proprietary code in a common location.
Eclipse	Derivative works choose their own license for their contributions.
Mozilla	Allows liberal mixing with proprietary software.
Microsoft	Requires all contributed code to be returned to the community.

Note: References to the abovementioned licenses are provided in the **References** section of this document.

License Compliance

The inclusion of open source software requires that we attend to the stipulations contained within their various licenses. The following table shows the obligations and usage allowed for various open source licenses [18, 19, 20]:

License	Obligations				Usage		
	Attribution Required	Code Availability	Document Changes	Modification Submission	Static Linkage	Dynamic Linkage	Code Derivation
None	†	†	†	†	†	†	†
Public domain	×	×	×	×	✓	✓	✓
GPL	✓	✓	✓	✓	×	✓	×
LGPL	✓	✓	✓	✓	‡	✓	×
MIT/X11	✓	×	×	×	✓	✓	✓
BSD	✓	×	×	×	✓	✓	✓
Apache	✓	×	✓	×	✓	✓	✓
Eclipse	✓	×	×	×	✓	✓	✓
Mozilla	✓	✓	×	✓	✓	✓	✓
Microsoft	✓	✓	×	✓	✓	✓	✓

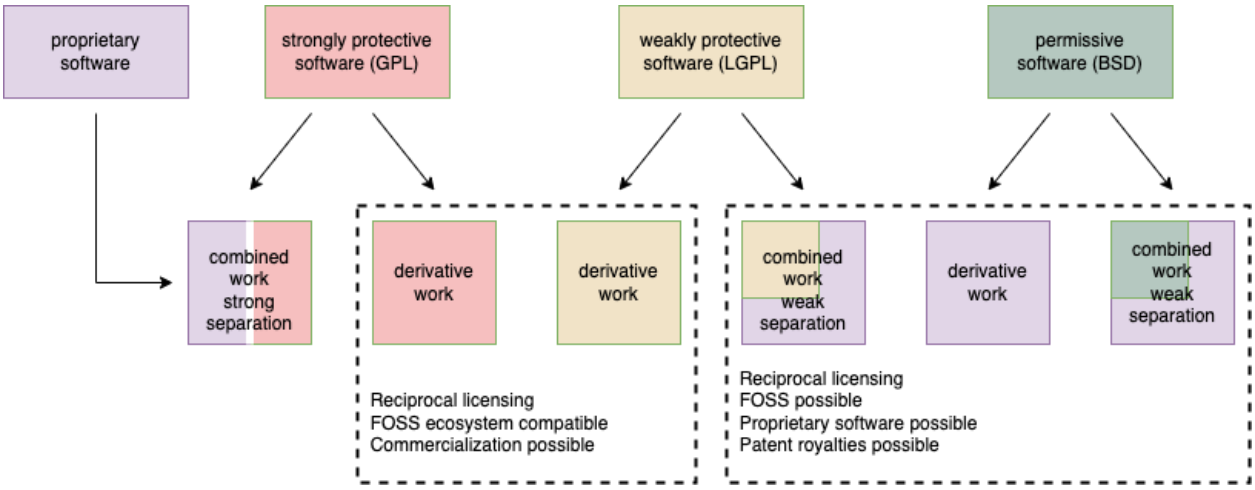
- ✓ required / allowed
- × not required / not allowed
- † explicit permission must be obtained from the copyright holder
- ‡ linking object must itself be provided either as source or as a linkable object

In order to simplify the process of open source software adoption, it is recommended that the GPL/LGPL obligations be satisfied regardless of the license. Doing so will ensure that errors in license compliance are minimized.

Note: Pay special attention to the GPL and LGPL licenses when static linking, or code derivation are intended use cases.

Derived Works

The following diagram [21] illustrates the various modalities of derived works:



The top row indicates the sources from which the derived work is created. The bottom row shows the various derived work modalities. These modalities fall into three categories:

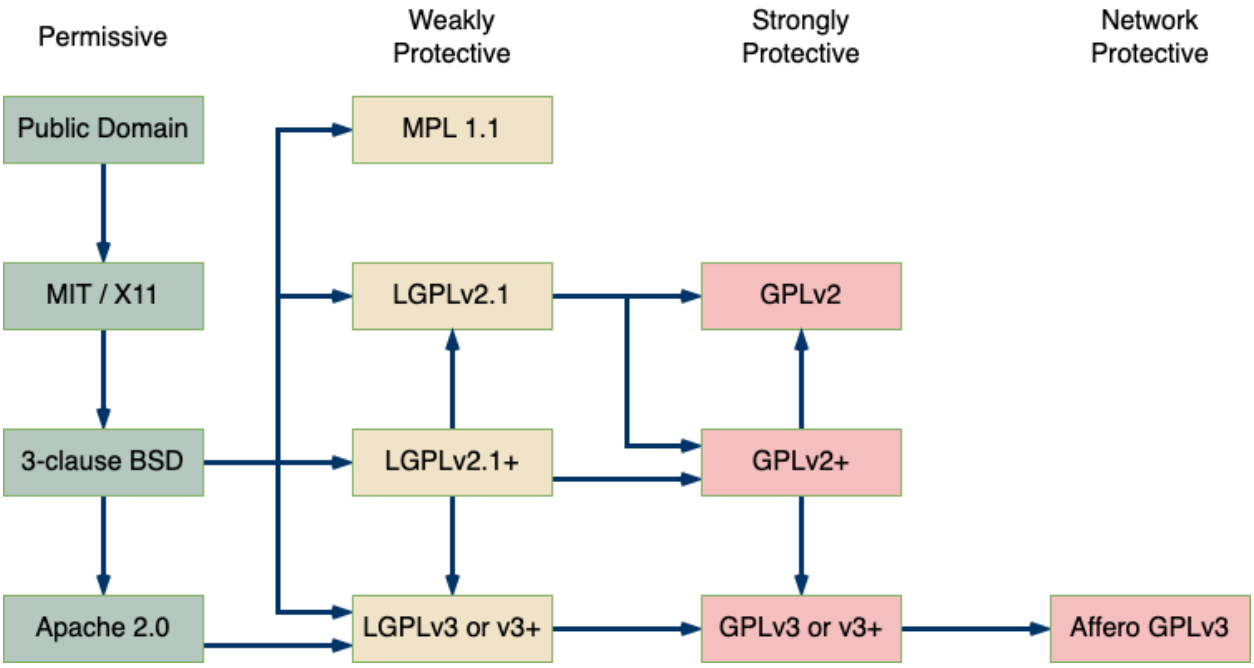
- Purely derivative from open source
- Combined derivative with strong separation (dynamic linkage)
- Combined derivation with weak / no separation (static linkage / source inclusion)

Note: It is critical to not use open source software with strongly protective licenses when the intent is to create derived software with weak separation. This would create a situation where the proprietary software was subject to code disclosure under the terms of the protective software's license [4].

(use case intentionally not shown)

Licenses Restriction Flow

When multiple licenses are in play in the creation of derived software, the most restrictive should be adopted. The following diagram [21] shows the flow from least to most restrictive:



Workflows

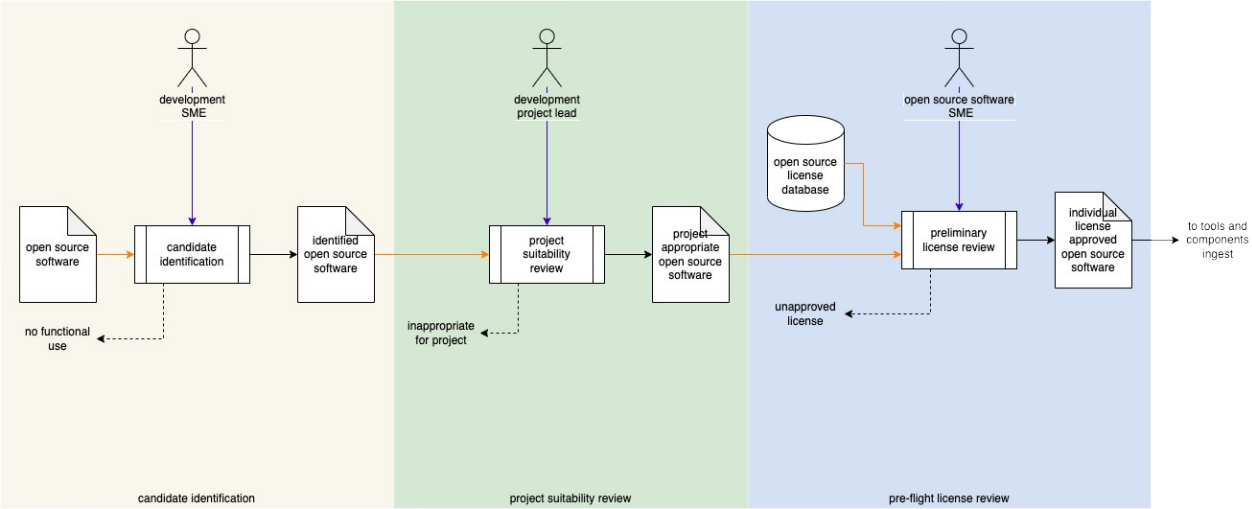
There are several workflows considered in this document. They are:

- Open Source Inclusion
- Open Source Modification

These shall be addressed in order.

Open Source Inclusion

Before open source material is included in a safety-critical product it needs to be reviewed. The following show the recommended workflow.

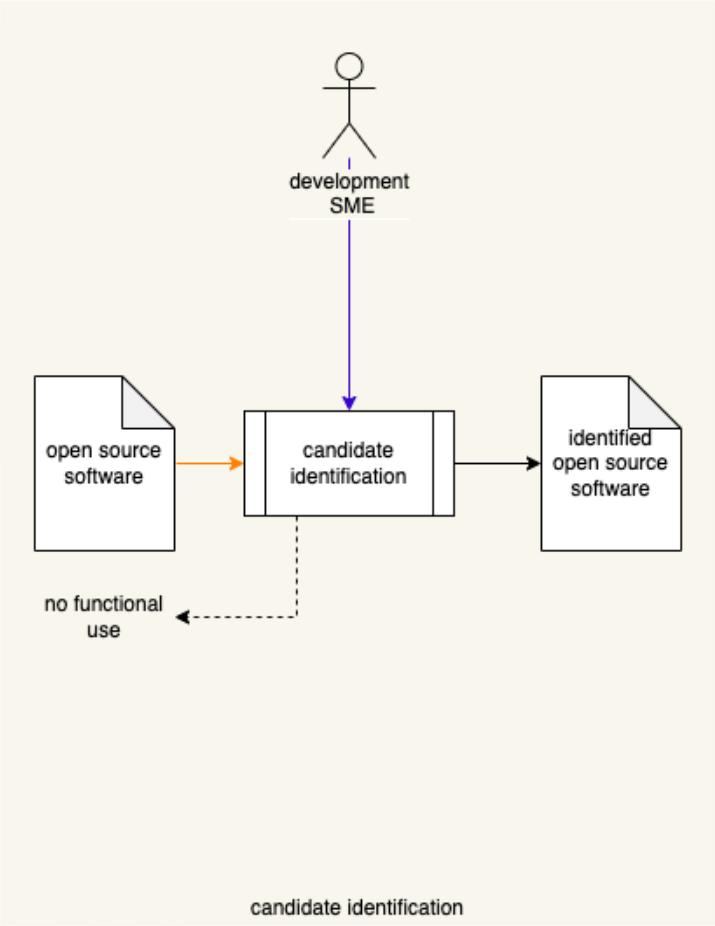


Note: This workflow augments the one described in the **List of Approved Tools and Components** [3] **AVCDL** secondary document.

Inclusion Process

Candidate Identification

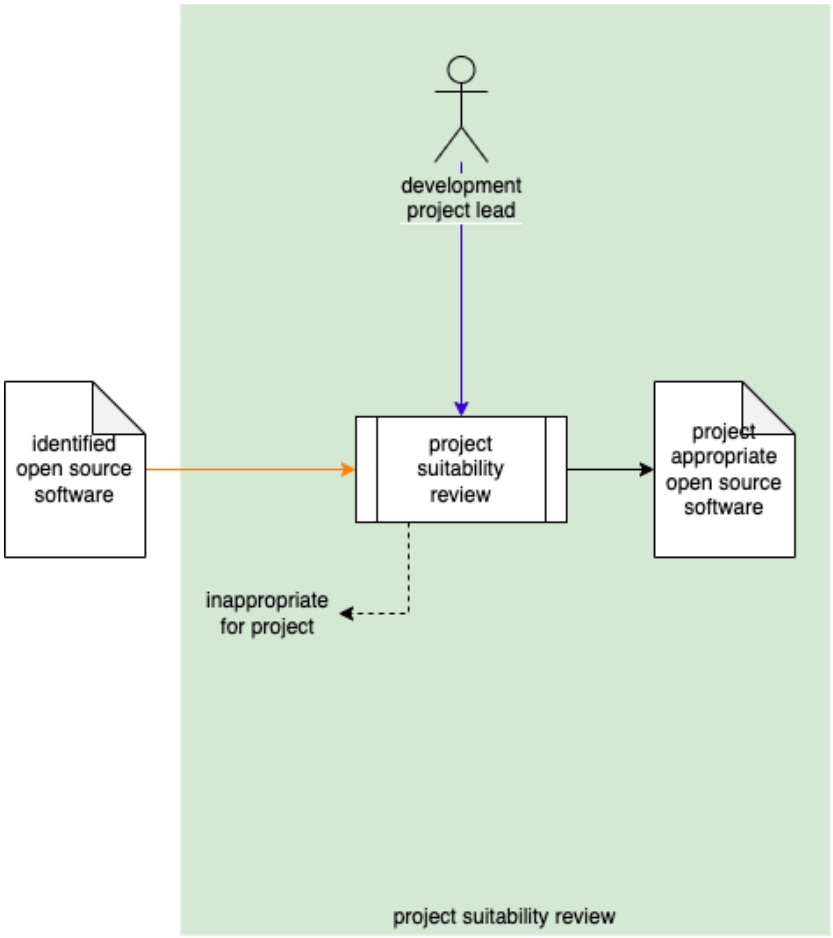
Inputs	Open source software
Outputs	Identified open source software
Participants	Development SME



The Development SME reviews the **open source software** to determine whether it is a good candidate for the desired use case. What specifically qualifies as a “good candidate” is based on the **open source software** and its application within the context of the project under consideration. If the **open source software** is deemed unacceptable, it is rejected as having **no functional use**. Otherwise, it will be considered to as **identified open source software**.

Project Suitability Review

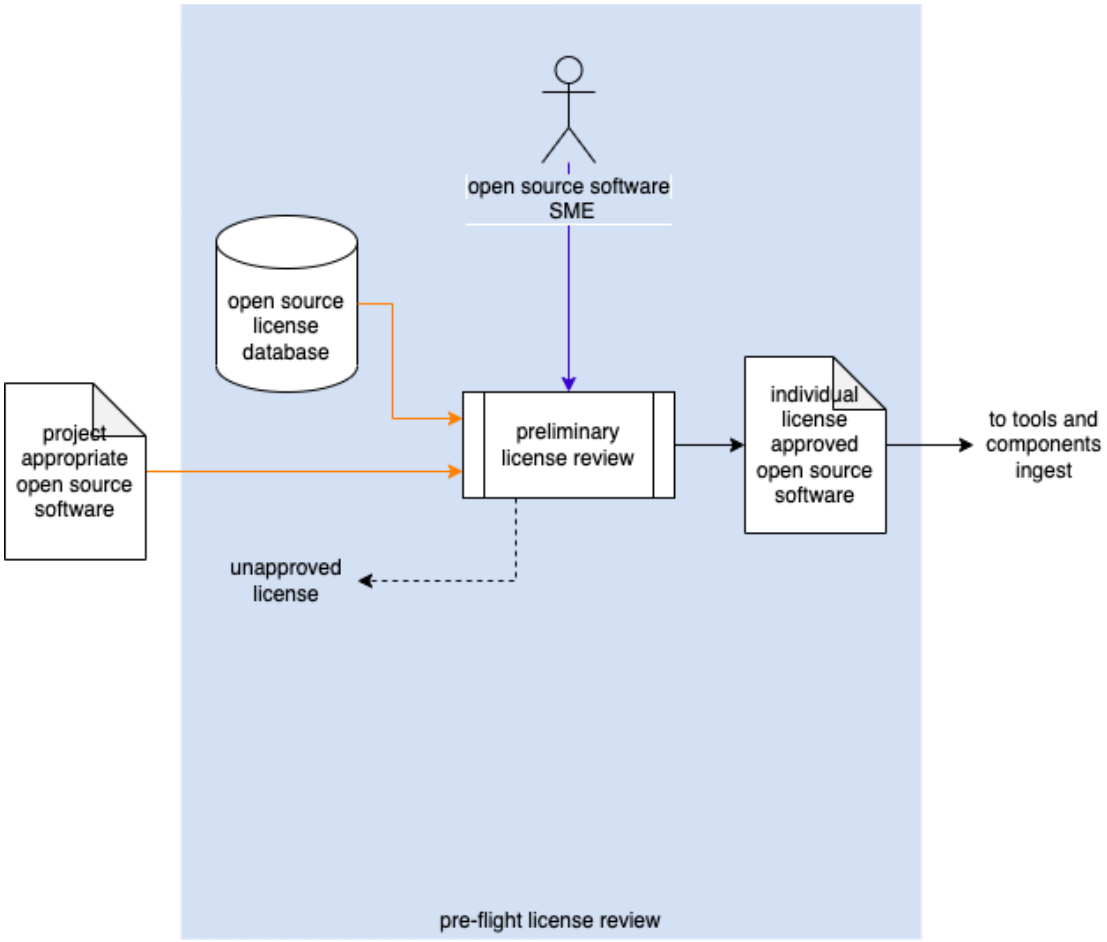
Inputs	Identified open source software
Outputs	Project appropriate open source software
Participants	Development project lead



The Development Project Lead reviews the **identified open source software** to determine whether it is suitable for inclusion in the project under consideration. What specifically qualifies as “suitable” is based on the **identified open source software** and its application within the context of the project under consideration. If the **identified open source software** is deemed unsuitable, it is rejected as being **inappropriate for project** use. Otherwise, it will be considered **project appropriate open source software**.

Pre-flight License Review

Inputs	Project appropriate open source software
Outputs	Individual license approved open source software
Participants	Open source software SME



Using the **open source license database**, the Open Source Software SME conducts a preliminary license review of the **project appropriate open source software** to determine whether its license has been approved for use. If the **project appropriate open source software** has an unapproved, it is rejected as having an **unapproved license**. Otherwise, it will be considered to as **individual license approved open source software**. It will then proceed to tools and components ingest as documented in the **List of Approved Tools and Components AVCDL** secondary document.

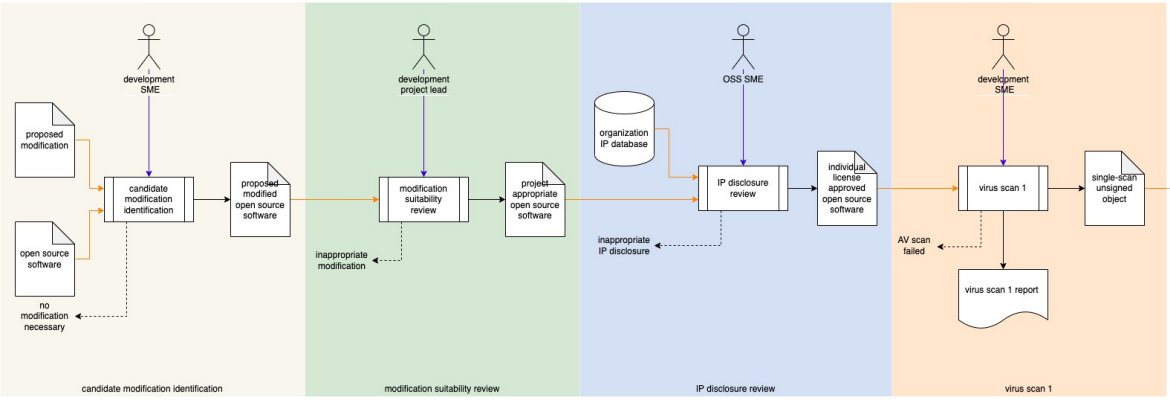
Note: It is presumed that the **open source license database** is managed under the guidance of the organization's legal department, who would be responsible for the approval of individual licenses and the final determination as to their appropriate usage within the context of the organization's products.

Note: It is presumed that if the **project appropriate open source software** has a novel license (one not in the **open source license database**), that the organization's process for reviewing novel licenses for possible inclusion into the **open source license database** will be undertaken.

Note: It is important to keep in mind that that the approval of an individual license does not imply that the aggregate of all licenses of all components within a project will be suitable. A review of the aggregate is necessary prior to finalization of the project for release.

Open Source Modification

Care must be taken when modifying open source material, especially when it is included in a safety-critical product. The following show the recommended workflow.

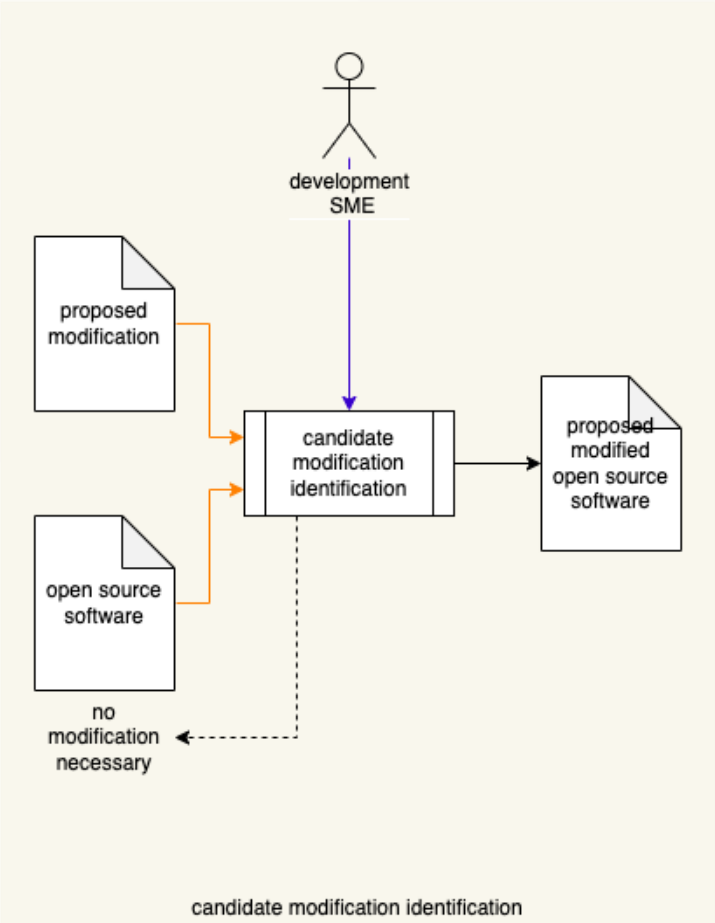


Note: This workflow is intended to augment the organization's existing source code / artifact management processes.

Modification Process

Candidate Modification Identification

Inputs	Open source software Proposed modification
Outputs	Proposed modified open source software
Participants	Development SME

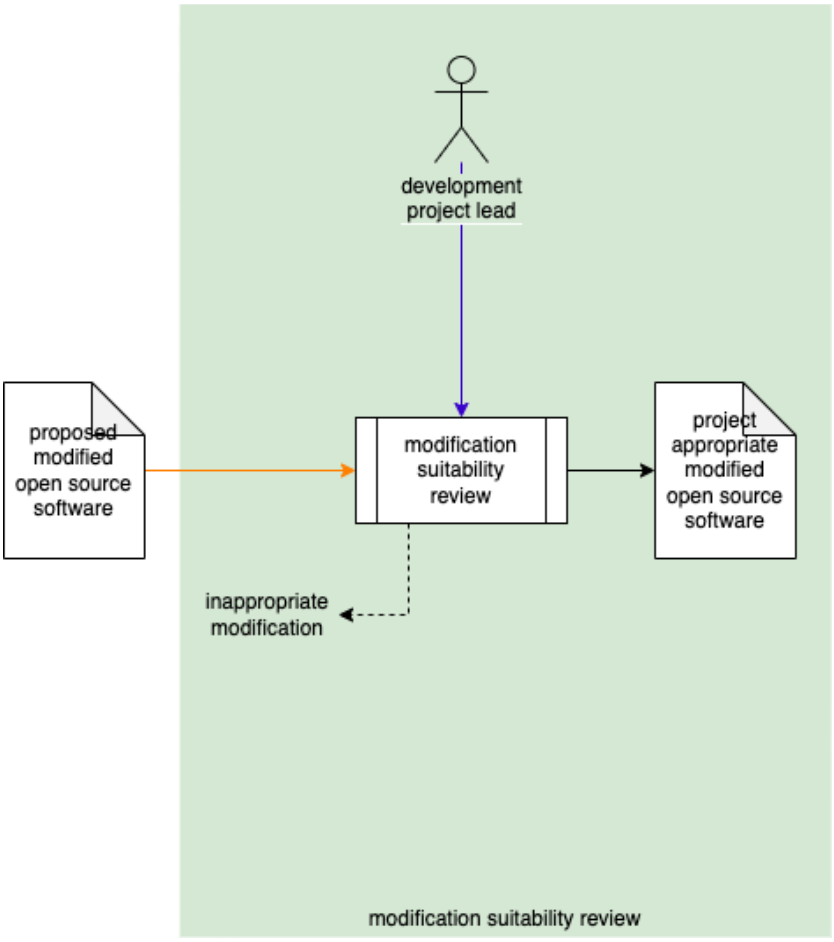


candidate modification identification

Using the **proposed modification**, the Development SME reviews the **open source software** to determine whether the **proposed modification** is necessary. What specifically qualifies as “necessary” is based on the **open source software** and its application within the context of the project under consideration. If the **proposed modification** to the **open source software** is deemed unnecessary, it is rejected as there is **no modification necessary**. Otherwise, it will be considered **proposed modified open source software**.

Modification Suitability Review

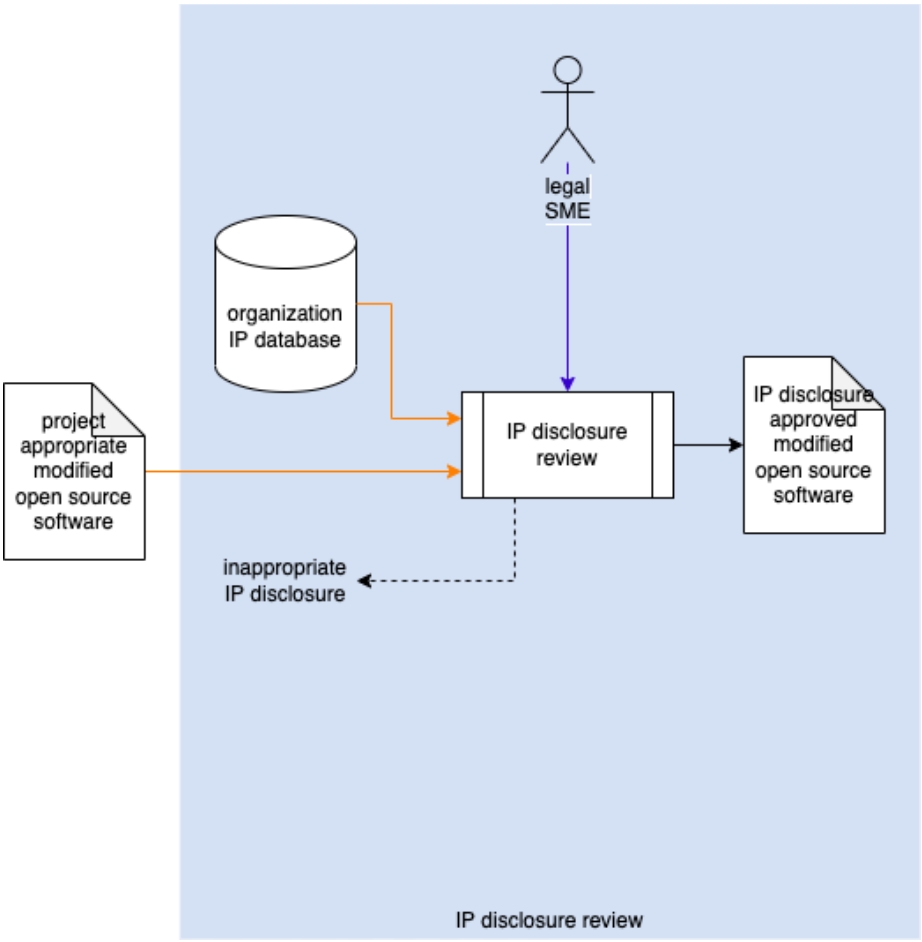
Inputs	Proposed modified open source software
Outputs	Project appropriate modified open source software
Participants	Development project lead



The Development Project Lead reviews the **proposed modified open source software** to determine whether it is suitable for inclusion in the project under consideration. What specifically qualifies as “suitable” is based on the **proposed modified open source software** and its application within the context of the project under consideration. If the **proposed modified open source software** is deemed unsuitable, it is rejected as being an **inappropriate modification**. Otherwise, it will be considered **project appropriate modified open source software**.

IP Disclosure Review

Inputs	Project appropriate modified open source software
Outputs	IP disclosure approved modified open source software
Participants	legal SME

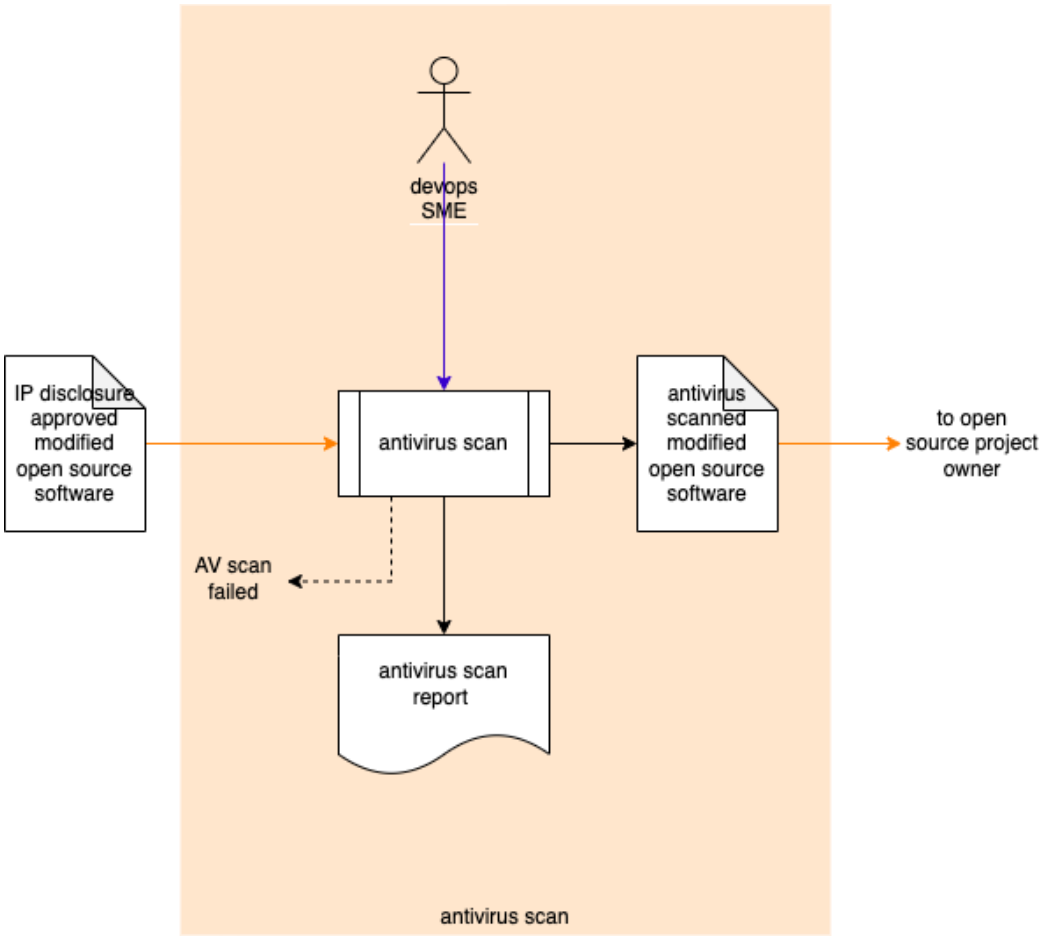


Using the **organization IP database**, the Legal SME conducts a review of the **project appropriate modified open source software** to determine whether contains any organizational IP. If the **project appropriate modified open source software** contains organizational IP, it is rejected as making **inappropriate IP disclosure**. Otherwise, it will be considered **IP disclosure approved modified open source software**.

Note: It is presumed that the **organization IP database** is managed under the guidance of the organization's legal department, who would be responsible for the organizational IP available for use within the context of the organization's products.

Antivirus Scan

Inputs	IP disclosure approved modified open source software
Outputs	Antivirus scanned modified open source software Antivirus scan report
Participants	devops SME



The devops SME runs an antivirus scan on the **IP disclosure approved modified open source software** to determine whether contains harmful code. If the **IP disclosure approved modified open source software** is determined to contain harmful code, it is rejected as the **AV scan failed**. Otherwise, it will be considered to as **antivirus scanned modified open source software**.

At this point, the **open source software**'s **proposed modification** can be conveyed to the open source project's owner.

References

1. **AVCDL** (primary document)
2. **Open-source software**
   https://en.wikipedia.org/wiki/Open-source_software
3. **List of Approved Tools and Components** (AVCDL secondary document)
4. **GNU General Public License – Linking and derived works**
   https://en.wikipedia.org/wiki/GNU_General_Public_License#Linking_and_derived_works
5. **GNU General Public License version 2**
   https://opensource.org/licenses/GPL-2.0
6. **GNU General Public License version 3**
   https://opensource.org/licenses/GPL-3.0
7. **Apache License, Version 2.0**
   https://opensource.org/licenses/Apache-2.0
8. **The 2-Clause BSD License**
   https://opensource.org/licenses/BSD-2-Clause
9. **The 3-Clause BSD License**
   https://opensource.org/licenses/BSD-3-Clause
10. **GNU LGPL**
    https://opensource.org/licenses/lgpl-license
11. **The MIT License**
    https://opensource.org/licenses/MIT
12. **Mozilla Public License 2.0 (MPL-2.0)**
    https://opensource.org/licenses/MPL-2.0
13. **Common Development and Distribution License 1.0**
    https://opensource.org/licenses/CDDL-1.0
14. **Eclipse Public License 1.0 (EPL-1.0)**
    https://opensource.org/licenses/EPL-1.0
15. **Boost Software License 1.0 (BSL-1.0)**
    https://opensource.org/licenses/BSL-1.0
16. **The zlib/libpng License (Zlib)**
    https://opensource.org/licenses/Zlib
17. **Licenses by Name**
    https://opensource.org/licenses/alphabetical
18. **Comparison of free and open-source software licenses**
    https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licenses
19. **Open Source Licenses Comparison [Guide]**
    https://itsfoss.com/open-source-licenses-explained/
20. **Pick a License, Any License**
    https://blog.codinghorror.com/pick-a-license-any-license/
21. **License compatibility**
    https://en.wikipedia.org/wiki/License_compatibility
22. **Frequently Asked Questions about version 2 of the GNU GPL**
    https://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.html

23. Licenses
https://choosealicense.com/licenses/
24. Proprietary Software Definition
http://www.linfo.org/proprietary.html
25. Proprietary Software
https://en.wikipedia.org/wiki/Proprietary_software
26. Commercial off-the-shelf
https://en.wikipedia.org/wiki/Commercial_off-the-shelf
27. Third-party software component
https://en.wikipedia.org/wiki/Third-party_software_component
28. Licenses and Copyright
http://www.linuxjournal.com/article/1297
29. Merriam-Webster Dictionary
https://www.merriam-webster.com/dictionary/