



YETIFORCE
CHOOSE OPEN SOURCE



A GUIDE TO OPEN SOURCE PROJECTS LICENSING

T A B L E O F C O N T E N T S



- 04 INTRODUCTION
- 05 WHY IS IT SO DIFFICULT?
- 07 WHERE SHOULD YOU START?
- 08 IDENTIFYING LICENSES IN AN ORGANIZATION
- 10 SECURING THE CONTRACT
- 10 THE NATURE OF THE PROBLEM
- 11 HOW TO PROTECT YOURSELF?
- 12 ADVICE FOR STARTUPS
- 13 LICENSING OF COMPLEX PROGRAMS
- 13 PROGRAM'S PRIMARY LICENSE
- 14 ADDITIONAL LICENSES
- 15 AUDIT
- 16 LICENSE TYPES AND CATEGORIES
- 16 BREAKDOWN BY CATEGORY
- 18 BREAKDOWN BY RISK LEVEL
- 19 LICENSE COMPATIBILITY
- 20 COMBINING LICENSES IN PRACTICE
- 20 LICENSE COMPATIBILITY MATRIX
- 21 ADDITIONAL EXPLANATIONS FOR THE LICENSE COMPATIBILITY MATRIX
- 22 LIST OF LIBRARIES COMPATIBLE WITH GPL v2
- 23 LIST OF LIBRARIES COMPATIBLE WITH GPL v3
- 24 LIST OF LIBRARIES INCOMPATIBLE WITH GPL v2
- 26 MOST COMMON MISTAKES
- 26 DISORGANIZED LIBRARIES AND LICENSES
- 28 DISTRIBUTION OF THE LICENSE WITH A COPYLEFT LIBRARY
- 29 COMBINING LIBRARIES WITH COPYLEFT LICENSES
- 31 OVERVIEW OF COPYLEFT LICENSES
- 31 OSLv3
- 32 AGPLv3
- 32 GPLv3
- 33 LICENSING TRAPS
- 35 EXTERNAL MATERIALS

INTRODUCTION

FOREWORD

ALTHOUGH WE GENERALLY LIKE SHARING SIMPLE AND TRIVIAL THINGS THAT DO NOT REQUIRE TOO MUCH INVOLVEMENT, WHEN IT COMES TO SHARING KNOWLEDGE, PEOPLE ARE NOT THAT EAGER TO DO SO. WE LIVE IN TIMES WHEN THE ACCESS TO KNOWLEDGE IS COMMON AND THERE ARE BILLIONS OF INFORMATIVE WEBSITES, YET WE STILL FEEL THAT PIECES OF IMPORTANT INFORMATION ARE TABOO.

We have to understand that sharing our knowledge and experience is the key to a better tomorrow, and will not result in losing some kind of an "invisible advantage". How comes that in a time where 99% of all companies in the world use open source software directly or indirectly, so little can be found about its licensing? If all these companies were more aware would there be less work or just the opposite?

In the past 10 years, I have participated in more than 300 projects that used open source software. Among these 300 clients, only one of them (yes, just one!) was interested in the licensing details. It is hard to believe that so few companies want to learn more about licensing; a license is nothing else than an agreement between parties. Do companies not read agreements the same way programmers do not read manuals?

When I started my adventure with open source licenses I had no experience, no financial resources, no substantive support, yet I wanted to create a large project based on open source software.

Why are companies so reluctant to learn and understand licenses? After all, it does not cost much, in contrary to investing into software released under some unknown to us license, which might turn out to be very risky. It is worth mentioning that the problem does not apply only to open source projects, it applies to all software in the company. The IT and legal departments are not very aware of the types of software and licenses, which causes problems as far as identifying the software in the company is concerned.

Quite a while ago a good friend of mine told me that the biggest challenge in his company (he worked for a company that hired almost 50000 people in Poland) was a detailed audit of the hardware and software. For more than 8 years of his employment at a high level in the IT department nobody had tried to face the challenge, or even mention it during their meetings with the board.

WHY IS IT SO DIFFICULT?

FOR OVER 6 YEARS I HAVE BEEN TRYING TO ANALYZE, CONSULT AND DISCUSS OPEN SOURCE LICENSING. THIS IS DUE TO TWO FACTORS: THE FIRST IS THAT I AM A SYSTEM ARCHITECT AND I NEED THE KNOWLEDGE BECAUSE THE SOFTWARE USES ABOUT 100 LIBRARIES LICENSED UNDER 20 DIFFERENT LICENSES. SECONDLY, LEGAL CONSULTATIONS ARE SO EXPENSIVE THAT WE ONLY USE THEM WHEN WE CONSIDER IT NECESSARY.

However, once we reach the point where we want to take advantage of lawyers' experience, we encounter several problems.

- ▶ **The first** is that there are not many specialists in the field of licensing open source projects (or at least I had a problem to find them from the perspective of 6 years and a wide range of lawyers on LinkedIn). In most cases, the lawyers we found had less knowledge in the field of open source licenses than we did, or specialized in only one license, e.g. GPL, but not AGPL, OSL, MPL, etc.
- ▶ **The second** problem is the license complexity. Some licenses are so extensive and consist of so many paragraphs that a detailed analysis is required to understand them. For this reason, many law experts have doubts about the interpretation of particular fragments of the license.
- ▶ **The third** problem is the cost of legal opinions. Very often, we received quotes which exceeded our budget several times - e.g., we needed a legal opinion on the AGPL, we had 10 questions and the budget of 10,000 Polish zloty (\$2,600). After receiving the quote, it turned out that we would need 50-100 thousand zloty (\$13,000- \$26,000), even after resigning from some of the questions.
- ▶ **The fourth** problem, which is also the worst one, is the lack of willingness to share knowledge. Even if there is a company with a lawyer who understands open source licenses, it is unlikely that his knowledge and experience will be publicly available. This practice is understandable in the case of private corporations, which do not share knowledge, however, knowledge and experience should not be concealed in the public sector.

I believe that the government that has the knowledge, resources and can benefit from sharing its knowledge about the use of open source software in the organization should be the most helpful here. Small and medium-sized companies would benefit the most from it because they often can not afford additional expenses for legal opinions and the analysis of software licenses in a company.



EACH PERSON ON THE TEAM SHOULD LEARN SOMETHING NEW AND WORK ON THEIR FLAWS EVERY DAY, IT IS THE ONLY WAY TO CREATE A STRONG AND CREATIVE TEAM.

WHERE SHOULD YOU START?



IN MOST OF THE COMPANIES I COOPERATED WITH (ALSO AS AN IT SPECIALIST) NO ONE HAS EVER ASKED ME ABOUT LICENSES AND HOW TO DEAL WITH THEM. AND EVEN IF SOMEONE ASKED ME 10 YEARS AGO, WHAT WAS I SUPPOSED TO ANSWER? MY KNOWLEDGE ABOUT LICENSING BACK THEN WAS SO LIMITED. THEREFORE, IT CAN BE ASSUMED THAT THIS IS REALLY A TABOO SUBJECT. IF A COMPANY BUYS A PROGRAM (WINDOWS, OFFICE, SAP, ETC.) OR DOWNLOADS FROM THE INTERNET (FIREFOX, LIBREOFFICE, NOTEPAD ++ ETC.), THEN NOBODY WILL GO DEEPER INTO THE LICENSES BECAUSE THEY ARE COMMONLY USED APPLICATIONS.

In a small company like ours (currently 20 employees), I receive at least one inquiry per day about a license for the software created or used in the organization. I suppose there would be no questions about the license if there was no internal training on licenses and no verification of licenses at every stage of software development. Awareness seems to come with time and is the result of the organization's maturity.

In order to avoid responding to each individual and analyzing the same thing every single time, it is worth creating a document/guide that will answer the most frequently asked questions and will grow with the organization. When I was looking for such a document in the Polish public sector, unfortunately, I could not find any interesting materials. Therefore, I started searching in Google and I came across a very interesting documentation: <https://opensource.google.com/docs/>, which describes how to handle software on different licenses.

The website offers a wide variety of useful information, however, it provides us with one important rule - a user must receive guidelines on how to deal with any new software/project in the organization. For example:

1. Check if the program/application has already been verified in our database and whether we have already issued recommendations for it. Your company's database can be a simple Excel file available online, or an application adapted for this purpose, e.g. YetiForce CRM.
2. Enter all the information you have, including:
 - program/application name,
 - description/purpose,
 - primary license,
 - licenses of external libraries,
 - other relevant information
3. Ask for approval (make sure the person you are asking for approval has proper authorization and access).
4. After receiving the approval, start working.

IDENTIFYING LICENSES IN AN ORGANIZATION

EACH EMPLOYEE MUST UNDERSTAND THE PROCESS OF IDENTIFYING AND APPROVING LICENSES TO BE USED IN THE COMPANY, REGARDLESS IF IT IS A SMALL ONLINE PROGRAM OR ADVANCED SOFTWARE USED FOR ACCOUNTING. IF A COMPANY USES SOFTWARE THAT HAS NOT BEEN IDENTIFIED THEN SOONER OR LATER IT MIGHT FACE SOME UNPLEASANT SITUATIONS (FOR EXAMPLE, AN EMPLOYEE WHO IS LEAVING COULD REPORT THE COMPANY FOR BREACHING A LICENSE OR FOR NOT PURCHASING SOME MANDATORY LICENSE).

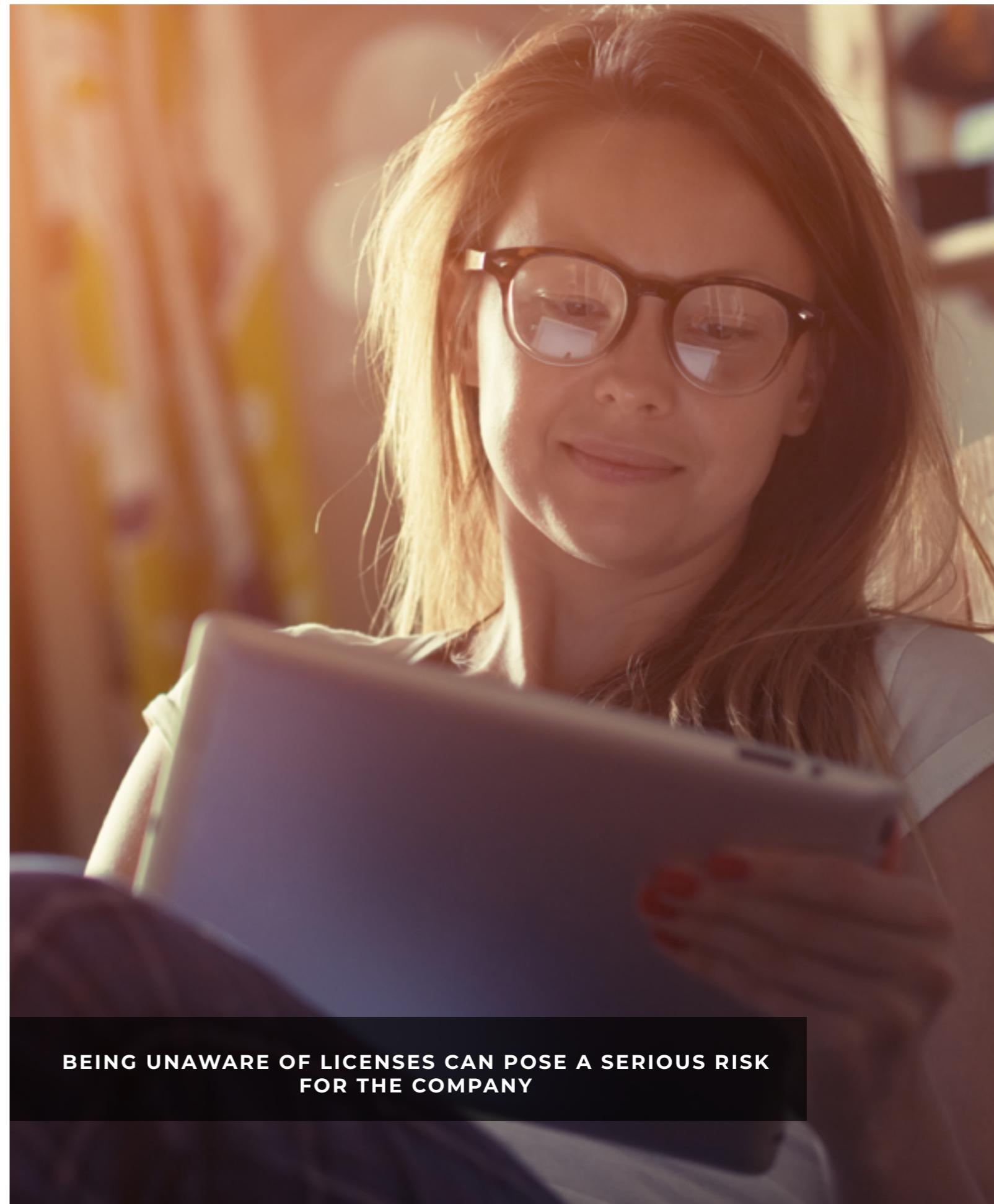
In case of software that is entirely created by only one producer, we usually deal with one (main) license that we must accept - this is a relatively simple procedure, an employee asks for approval for the software to be used.

In case of complex software, apart from the main license, we have to approve (or at least be aware of) a number of third party licenses. We should get familiar with all the libraries and their licenses and understand the rules of combining licenses, since in practice some licenses cannot be combined (it is described later on in this guide).

There is a number of disorganized applications that do not provide one central list of third party licenses they include. You should think twice about using them, the better choice might be to give up on them. However, if the application in question is crucial for the company you should consider auditing it.

Properly organized licenses reflect the maturity of the software and its producer, and what is more important, they lower the risk of using the program. I believe that companies that are looking for a software vendor should request a full list of licenses used by the software in order to be able to verify its compliance with the company's regulations prior to choosing a particular vendor.

It happens very often that files or libraries do not contain any information about the license. Such code should either be immediately removed from the system or we should verify its source and license. It is essential to make sure if the library is compatible with other libraries.



BEING UNAWARE OF LICENSES CAN POSE A SERIOUS RISK FOR THE COMPANY

SECURING THE CONTRACT

THE NATURE OF THE PROBLEM

"If we do not comply with the license, then the author is protected by the provisions of the Act of 4 February 1994 on copyright and related rights and may seek protection under both civil and criminal law. The author is also protected on the basis of European Union law, including Directive 2009/24 /EC of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs"

The law primarily protects the author and that is why we must be very careful, in particular that:

"The initial licensor may assert his rights in court proceedings against all parties who violate his rights, even if they do so unknowingly"

The quote from the legal opinion shows how complex the problem of respecting the license from the business side is. If we download the software (regardless of whether it is open or proprietary software) and the software violates the license, then the person who has the original rights to the license may sue us.

Description of the case in which an open source license was violated: <https://www.fsf.org/blogs/licensing/update-on-artifex-v-hancom-gnu-gpl-compliance-case-1>.

Creating good software often takes years, which in case of many applications means that the license and the origin of the code are unknown. Of course, this should not happen in a mature software development company.

In addition, you should always consider the possible risk that a programmer used a forbidden library and hid the information about its author. That is why an external audit or static code analysis is so important. Quality must be a process and you should pay special attention to licensing at every stage of software design and development.

"Seventy-four percent of audited codebases contained components with license conflicts, the most common of which were GPL license violations, found in 44% of codebases."

Source: 2018 Open Source Security and Risk Analysis - Synopsys

HOW TO PROTECT YOURSELF ?

IT IS CRUCIAL TO PROTECT YOURSELF FROM THE POTENTIAL LEGAL EFFECTS OF THIRD PARTIES CLAIMING THEIR RIGHTS TO SOFTWARE USED BY YOU. THE BEST WAY TO ACHIEVE THIS IS TO APPLY APPROPRIATE CONDITIONS BETWEEN THE PARTIES. BELOW YOU CAN FIND A DOCUMENT THAT WE USE WITH OUR CLIENTS:

1. The Parties confirm that the Customer has the right to use the YetiForce Software ("Software") distributed by the Vendor, including all its modifications, according to the public license available at the following address: "<https://github.com/YetiForceCompany/YetiForceCRM/tree/developer/licenses>".
2. The Vendor confirms that according to the terms of the license provided by the Vendor referred to in paragraph 1 above under no circumstances and on any legal basis, the Customer shall be liable to any third party for any damage (direct and indirect) related to the use of the Software
3. In the case of claims made by third parties indicating that the Software provided by the Vendor infringes the intellectual property rights of these third parties, the Vendor shall defend the Customer against such claims at Vendor's own expense and shall cover all costs, damages and costs of legal services ordered by the court or included in the settlement approved by the Vendor, provided that the Customer:
 - ▶ will inform the Vendor in writing of such a claim without delay,
 - ▶ will allow the Vendor to take control and cooperate with the Customer in the field of defense against such claims and in the conduct of any negotiations to settle the dispute amicably.
4. If the Software is found to infringe intellectual property rights or copyrights of a third party, the Vendor at his own expense and at his own choice will obtain the right for the Customer to use the Software under the terms described in the license agreement or will replace the Software with another software that does not violate the rights referred to in paragraph 3 above, in such a way that the Ordering Party continually uses the Software or other software that fulfills the same functions and requirements as the Software.
5. For the avoidance of doubt, the Vendor shall not be liable for any claims related to any code supplied by the Customer.

ADVICE FOR STARTUPS



FIRST DECISIONS THAT YOU MAKE MIGHT BE CRUCIAL

IN 2013, AS A COMPANY THAT WANTED TO CREATE AN OPEN SOURCE FORK, WE WERE WONDERING WHICH SYSTEM WE SHOULD USE. THE MAIN PROBLEM WE HAD TO SOLVE WAS THE LICENSING OF THE SYSTEMS WE CONSIDERED AND THE QUALITY OF THE CODE. WE HAD TO DECIDE BETWEEN TWO SYSTEMS: SUGARCE AND VTIGERCRM.

Although the company was small and had little funds for legal opinions, we commissioned an independent analysis for these two systems and spent about \$2,600 (it was a small amount from today's perspective, but back then it was all we had and we could spend). This legal opinion confirmed that we had made a good decision and from the perspective of years, we can engage both financial and human resources in the development of the system under our own brand. What is important, the opinion was issued by a law firm specializing in open source licenses.

If we had chosen SugarCE, then most likely we would not be able to participate in most tenders (due to the license) and most of the customers who know about the consequences of using the AGPL license would not be interested in doing business with us.

A wrong decision made in the past would have cost us millions in losses today and would significantly limit the company's development. It should be noted that as a company we are responsible not only for ourselves, but also for our customers.

Currently, YetiForce is one of the most well-organized CRM systems in terms of libraries and licensing. Not only can the administrator verify the list of libraries at any time, but he can also check their versions and licenses. In many cases, a user has access to licenses in both Polish and English. The latest version of the system has been published under the **YetiForce Public License 3**, which is a slightly modified version of the MIT license.

LICENSING OF COMPLEX PROGRAMS

PROGRAM'S PRIMARY LICENSE

IT ESSENTIAL TO REMEMBER THAT THE LICENSE IS AIMED AT LIMITING OUR RIGHTS TO THE WORK IT CONCERN. HOWEVER, IF SOMEONE HAS ALREADY CREATED A LICENSE FOR THE SOFTWARE, IT IS ONLY TO LET US KNOW WHAT WE CAN, WHAT WE CAN NOT AND WHAT WE MUST. A LICENSE IS ALSO A CONTRACT BETWEEN THE PARTIES, SO IT IS IMPORTANT TO KNOW THE CONTENT OF THE LICENSE (CONTRACT) IN ORDER NOT TO BREAK ITS TERMS.

There are currently hundreds of open source licenses, and there are over 80 licenses, which have been accepted by the Open Source Initiative. In many cases, open source licenses are milder than proprietary software licenses, but this is not always a rule.

There are websites with a list of open source licenses, for example: <https://opensource.org/licenses/alphabetical>.

In addition, we can find many places like for example <https://tldrlegal.com/> that helps us understand different types of licenses by summarizing and organizing in three categories: what you can, what you cannot and what you must.

If our software includes an open source license (e.g. MIT) and we do not have a lawyer specializing in the licensing of open applications, we can try to match the license in the tool given above: <https://tldrlegal.com/license/mit-license#summary>. This way, we will receive the most important information about this license.

You should be careful in any case, because it may turn out that the application is licensed differently than you or the producer thought because it is combined with another license (e.g. a copyleft license). Therefore, third-party libraries are so important to us.

To succeed in today's market, companies must innovate! The rapid pace of innovation excludes building everything from scratch. It is estimated that modern applications contain between 80-90% of open source code!

Source: [forbes.com](https://www.forbes.com/sites/forbestechcouncil/2018/02/27/the-most-common-open-source-laws-and-licensing-mistakes-to-avoid/)

ADDITIONAL LICENSES

ADDITIONAL LICENSES ARE THIRD PARTY LIBRARY LICENSES THAT ARE USUALLY PLACED IN “THIRD_PARTY” OR “VENDOR” FOLDERS. THESE LICENSES ARE AS IMPORTANT AS THE MAIN LICENSE OF THE SOFTWARE. IT IS SIGNIFICANT TO READ ALL THE LICENSES INCLUDED IN THE APPLICATION BECAUSE THERE IS A CHANCE THAT SECONDARY LICENSES MIGHT AFFECT THE MAIN LICENSE.

Each program should manage its libraries and licenses in an orderly manner. If it is not the case then we recommend either discarding the software, or conducting an audit.

If we are dealing with mature software we will get a full list of all libraries (together with licenses and version numbers) in a readable form with the option to sort and read each license. Below is a screenshot from the YetiForce system:

Nazwa	Wersja	Licencja
YetiForce	4.4	YetiForce Public License v3
Vigen (Oryginalnego kodu Vigena zostało około 10%. Pozostały kod został przepisany i rozszerzony)	6.40 rev. 14548	VPL 1.1
Sugar CRM (Viger bazował na Sugar CRM. Obejście w YetiForce pozostało kilka plików po Sugar CRM)		SPDX-1.1.2
creditor/creditor	4.10.0	MPL-1.1+ (GPL-2.0+, LGPL-2.1+, MPL-1.1+)
composer/ca-bundle	1.1.1	MIT
dg-mos/php	v1.3	BSD-3-Clause
doctrine/collections	v1.5.0	MIT
doctrine/inflector	v1.3.0	MIT
ezzyang/htmlpurifier	v4.18.0	GPL
giggsey/laravelnumber-form-php	8.8.11	Apache-2.0
giggsey/locale	1.5	MIT
illuminate/contracts	v5.6.29	MIT
illuminate/support	v5.6.29	MIT
keyknight/php-cs-parser	1.4.1	MIT
minify/base64	5.6	GPLv3.0
nesbot/carbon	1.32.0	MIT
phalcon/phalcon	v6.0.5	GPL-2.1
phoffice/phpspreadsheet	1.3.1	GPL-2.1-or-later
psr/container	1.0.0	MIT
psr/log	1.0.2	MIT
psr/simple-cache	1.0.1	MIT
rmeocut/requests	v1.7.0	ISC
salome/dav	3.2.2	BSD-3-Clause
salome/event	3.0.0	BSD-3-Clause
salome/http	v4.2.4	BSD-3-Clause
salome/uri	1.2.1	BSD-3-Clause
salome/object	4.1.6	BSD-3-Clause
salome/xml	1.5.0	BSD-3-Clause
securitec/security-checker	v1.1.8	MIT

As you can see on the screenshot above, we provide a full list of libraries and their versions, license names, and the option to download their content and preview their details. If a library has been released under different licenses then you can see which one we selected.

The producer of SugarCRM also provides information about libraries and licenses for the commercial versions of the software in a separate PDF document:
<https://support.sugarcrm.com/files/d38fd368-35dd-11e7-93de-06ac51574995.pdf>

AUDIT

WE MIGHT FIND SOFTWARE WHOSE LIST OF LIBRARIES AND THEIR LICENSES IS NOT MANAGED PROPERLY. IF WE DO NOT WANT TO OPT OUT OF SUCH SOFTWARE, WE CAN TRY CONDUCTING ITS AUDIT. IF WE ARE DEALING WITH OPEN SOURCE SOFTWARE THERE IS NO PROBLEM TO PERFORM AN AUDIT (WE CAN DO IT ON OUR OWN).

If it is proprietary software, then the audit should be performed by an independent company, and even though we will not have access to the source code, we should get access to the audit report. If this is not possible, then unfortunately we discourage you from using this type of software.

The audit can be done either manually (by assigning a person responsible for opening every folder and file of the application, and writing down all found libraries and licenses), or automatically by using ready-made tools, for example: <https://fossa.io>.

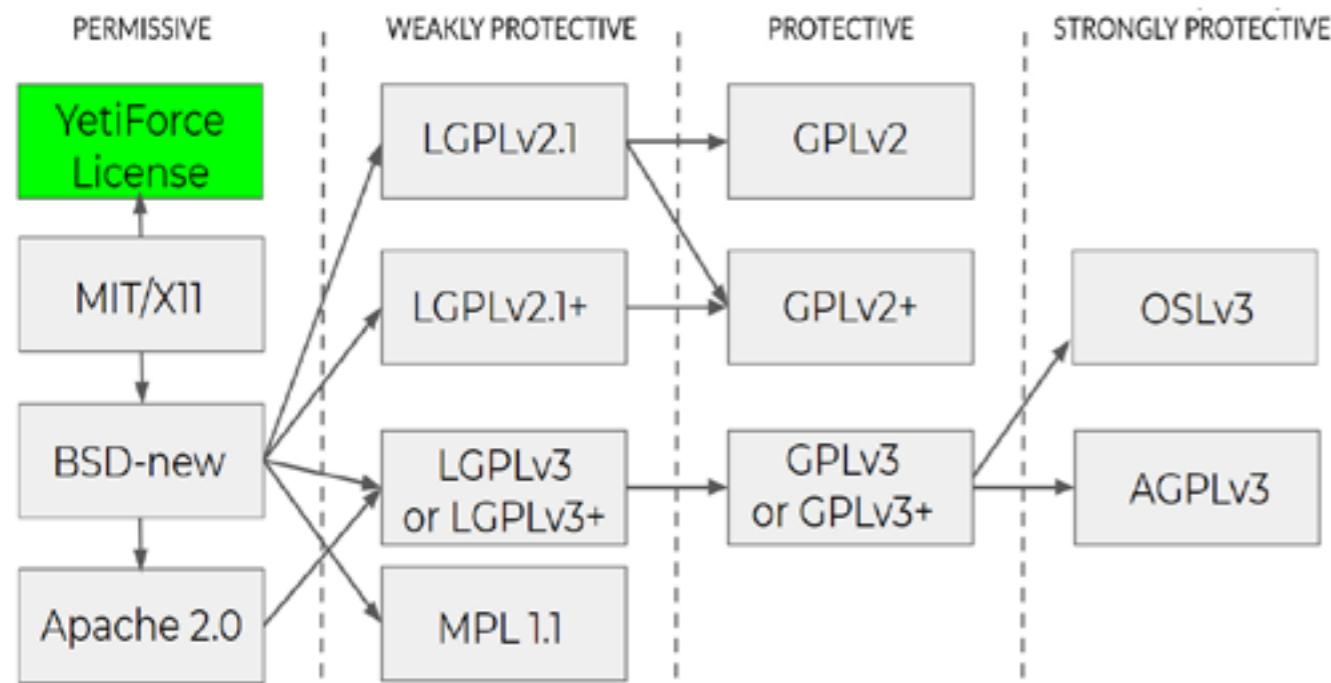
FOSSA				
Open source management powered by FOSSA, Inc. © 2018 v0.7.0 - revid: n/a				
<h1>YetiForce Sp. z o.o.</h1>				
 ADD PROJECTS	 RESCAN ALL			
<input type="checkbox"/>	TITLE	STATUS	STATS	LAST UPDATED
<input type="checkbox"/>	EPESI	3 Issues Found	 6 deps + 21 licenses	6 days ago
<input type="checkbox"/>	SuiteCRM	3 Issues Found	 37 deps + 22 licenses	6 days ago
<input type="checkbox"/>	VtigerCRM	2 Issues Found	 83 deps + 34 licenses	6 days ago
<input type="checkbox"/>	YetiForceCRM	No Issues Found	 29 deps + 21 licenses	6 days ago

As you can see in the image above, Fossa found a few problems in EPESI, SuiteCRM, and VtigerCRM. Each of these problems should be analyzed manually because Fossa might misinterpret some licenses, especially if the library has been released under several different licenses simultaneously.

The majority of business systems (in particular these less popular ones) struggle with licensing, for example: SuiteCRM, VtigerCRM, coreBOS, joForce, EPESI, EspoCRM, or Dolibarr. Unfortunately, tools that analyze systems automatically do not find all of the problems that can be found during a manual audit. This problem will be discussed in more detail later on in this guide.

LICENSE TYPES AND CATEGORIES

BREAKDOWN BY CATEGORY



01 PERMISSIVE

Does not impose using the same license in our code, however, you should keep the license information and a note in a file (it is just business simplification to help you understand the differences between the categories of licenses).

02 WEAKLY PROTECTIVE

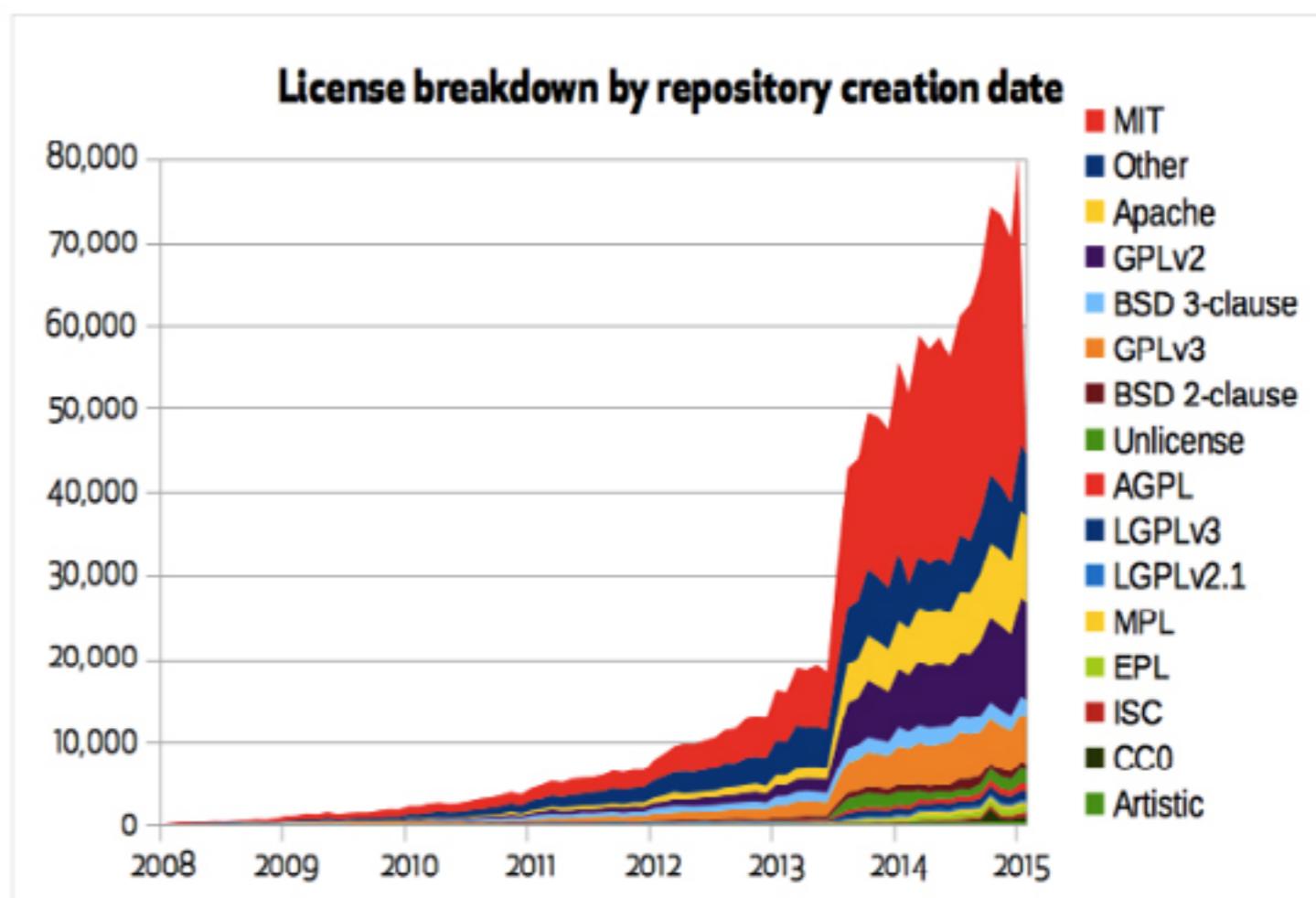
Does not impose using the same license in our code, unless this code is part of the library and the library itself is not the basic functionality of the application. For example, if our system wants to use a library under LGPL license to print a PDF, and the rest of the application is under a different license, then we can use the LGPL license (it is just business simplification to help you understand the differences between the categories of licenses).

03 PROTECTIVE

It forces us to have the whole application under the same license if at least a few lines of code are used in the application and distributed with it (it is just business simplification to help you understand the differences between the categories of licenses).

04 STRONGLY PROTECTIVE

Licenses similar to the category of strongly protective with the difference that the distribution is already either at the stage of online access to the application or when uploading changes to it (it is just business simplification to help you understand the differences between the categories of licenses).



Źródło: <https://blog.github.com/2015-03-09-open-source-license-usage-on-github-com/>

BREAKDOWN BY RISK LEVEL

COMPANIES SHOULD CONSIDER DRAFTING AN “INITIAL” LIST OF LICENSES AND WAYS OF DEALING WITH THEM, IN CASE NEW SOFTWARE IS NEEDED. BELOW YOU CAN FIND A LIST USED BY OUR COMPANY, WE CONSTANTLY EXPAND AND VERIFY THIS LIST.

LOW RISK	MODERATE RISK	HIGH RISK
The software can be used in an organization, it does not always require internal consent to use the library if the source of the application is secure and no extensive internal analysis is needed.	The software creates additional risks that may depend on the way the software is used or the way a developer uses it. It always requires internal consent to use the library.	The software creates a high risk due to the copyleft licensing, where new code must be distributed under the same license. It always requires internal consent to use the library.
<input checked="" type="checkbox"/> Apache 1.0, 1.1 <input checked="" type="checkbox"/> Apache 2.0 <input checked="" type="checkbox"/> BSD (all versions) <input checked="" type="checkbox"/> Internet Software Consortium <input checked="" type="checkbox"/> JDOM license <input checked="" type="checkbox"/> MIT (all versions) <input checked="" type="checkbox"/> Python Software Foundation license <input checked="" type="checkbox"/> YetiForce Public License <input checked="" type="checkbox"/> WTFPL	<input checked="" type="checkbox"/> Apple Public source <input checked="" type="checkbox"/> Artistic License 1.0/2.0 <input checked="" type="checkbox"/> Code Project Open license (all versions) <input checked="" type="checkbox"/> Common Development and Distribution (CDDL) 1.0/1.1 <input checked="" type="checkbox"/> Common Public Attribution License (CPAL) 1.0 <input checked="" type="checkbox"/> Commons Public License <input checked="" type="checkbox"/> Eclipse Public License v1 license <input checked="" type="checkbox"/> IBM Public License v1 <input checked="" type="checkbox"/> Microsoft Public License <input checked="" type="checkbox"/> Mozilla Public License v1.1 <input checked="" type="checkbox"/> Mozilla Public License v2 license <input checked="" type="checkbox"/> SVNkit license <input checked="" type="checkbox"/> Sugar Public License 1.1.2 <input checked="" type="checkbox"/> Vtiger Public License 1.1	<input checked="" type="checkbox"/> Affero GNU Public License v3 <input checked="" type="checkbox"/> Less GNU Public License v2 <input checked="" type="checkbox"/> Less GNU Public License <input checked="" type="checkbox"/> GNU Public License v2 <input checked="" type="checkbox"/> GNU Public License v3 <input checked="" type="checkbox"/> Open Software License (OSL) All versions license

The risk of using the license may depend on e.g. how we use the software. For example, if we choose the software under the LGPL license and further develop it, then everything we add must also be under the LGPL license. If the same software under the LGPL license is added as a library and it will be only a small part of our main software, then our software does not have to be under LGPL license (it is only a simplification in order to better understand the general rule, each case requires a legal analysis).

LICENSE COMPATIBILITY



THE PROBLEM ARISES WHEN WE HAVE MANY DIFFERENT LICENSES IN THE SYSTEM AND THEY MUST BE COMPATIBLE WITH EACH OTHER. THERE ARE LICENSES THAT ARE VERY PERMISSIVE AND ARE NOT PROBLEMATIC BOTH FROM THE LEGAL SIDE AS WELL AS FROM THE BUSINESS SIDE, BUT THERE ARE ALSO LICENSES THAT REQUIRE A LOT OF ATTENTION, BECAUSE THEY ARE VERY RESTRICTIVE.

Each producer must settle the issues of licensing at the very beginning of the project, so that he can have full control over the code at all times. If you talk to a software vendor, make sure to ask for a list of all libraries and licenses that are used in the software.

In addition, you can use tools that have been designed to analyze the code and search all libraries in it. A good example of this might be <https://fossa.io/>, which is free for open source projects.

In addition to the standard list of licenses used in the software, you should also look for the so-called dependencies between libraries and license conflicts, because different versions of licenses are compatible with each other or not.

For example, the AGPL license in version 2 is not compatible with the GPL license version 2, while the AGPL and GPL in version 3 are compatible.

Most copyleft licenses are based on the GPL [General Public License]. I recommend using version 3 of this license, but there are projects which consider version 2 to be sufficient (<https://www.cnet.com/news/torvalds-no-gpl-3-for-linux/>). The main advantage of version 3 of this license has been described here: <http://www.ifross.org/en/what-difference-between-gplv2-and-gplv3>, and what is even more important, this version allows linking with more other licenses such as AGPL v3 and Apache License v2.0.

COMBINING LICENSES IN PRACTICE

LICENSE COMPATIBILITY MATRIX

THE VARIOUS GNU LICENSES ARE USUALLY COMPATIBLE WITH EACH OTHER, WITH ONE EXCEPTION - YOU CAN'T COMBINE OLDER VERSION CODE WITH NEWER VERSION CODE. BELOW IS A DETAILED COMPATIBILITY MATRIX FOR VARIOUS COMBINATIONS OF THE GNU LICENSES. THE STRUCTURE INDICATES IF YOU ARE ALLOWED TO USE SOMEONE ELSE'S CODE UNDER CERTAIN LICENCE WHILE AT THE SAME TIME USING A DIFFERENT LICENSE FOR YOUR CODE.

		I want to license my code under:					
		GPLv2 only	GPLv2 or later	GPLv3 or later	LGPLv2.1 only	LGPLv2.1 or later	LGPLv3 or later
I want to copy code under:	GPLv2 only	OK	OK [2]	NO	OK: Combination is under GPLv2 only [2]	OK: Combination is under GPLv2 only [2][3]	NO
	GPLv2 or later	OK [1]	OK	OK	OK: Combination is under GPLv2 or later [2]	OK: Combination is under GPLv2 or later [2]	OK: Combination is under GPLv3 [8]
	GPLv3	NO	OK: Combination is under GPLv3 [3]	OK	OK: Combination is under GPLv3 [2]	OK: Combination is under GPLv3 [2]	OK: Combination is under GPLv3 [8]
	LGPLv2.1 only	OK: Convey copied code under GPLv2 [2]	OK: Convey copied code under GPLv2 or later [2]	OK: Convey copied code under GPLv3 [2]	OK	OK [6]	OK: Convey copied code under GPLv3 [2][8]
	LGPLv2.1 or later	OK: Convey copied code under GPLv2 [2][11]	OK: Convey copied code under GPLv2 or later [2]	OK: Convey code under GPLv3 [2]	OK [5]	OK	OK
	LGPLv3	NO	OK: Combination is under GPLv3 [8][3]	OK: Combination is under GPLv3 [8]	OK: Combination is under GPLv3 [2][8]	OK: Combination is under GPLv3 [4]	OK
I want to use a library under:	GPLv2 only	OK	OK [2]	NO	OK: Combination is under GPLv2 only [2]	OK: Combination is under GPLv2 only [2][3]	NO
	GPLv2 or later	OK [1]	OK	OK	OK: Combination is under GPLv2 or later [2]	OK: Combination is under GPLv2 or later [2]	OK: Combination is under GPLv3 [8]
	GPLv3	NO	OK: Combination is under GPLv3 [3]	OK	OK: Combination is under GPLv3 [2]	OK: Combination is under GPLv3 [2]	OK: Combination is under GPLv3 [8]
	LGPLv2.1 only	OK	OK	OK	OK	OK	OK
	LGPLv2.1 or later	OK	OK	OK	OK	OK	OK
	LGPLv3	NO	OK: Combination is under GPLv3 [2]	OK	OK	OK	OK

Source: <https://www.gnu.org/licenses/gpl-faq.en.html#AllCompatibility>

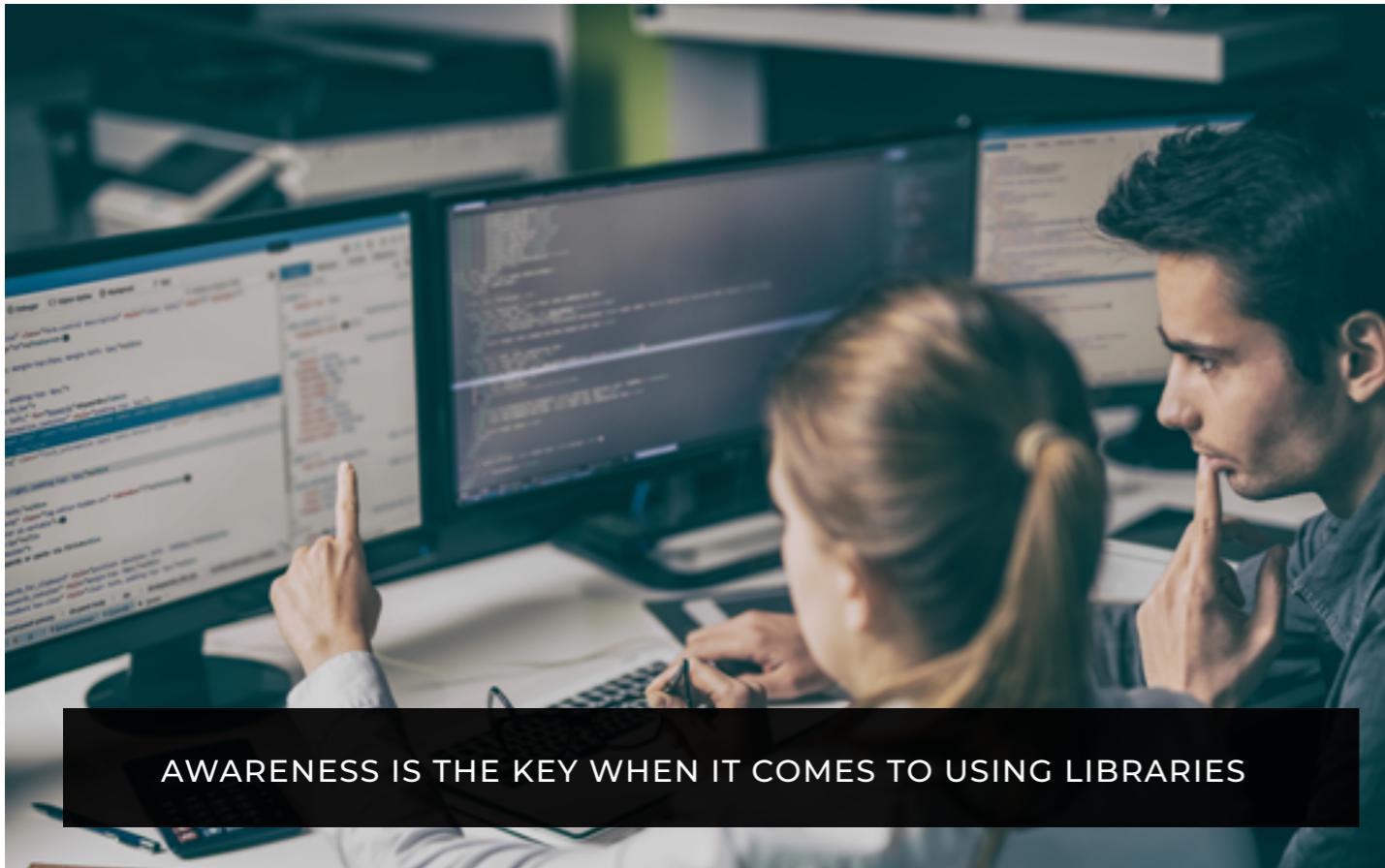
Find the license for your project in a column at the top of the table, and the license for the other code in a row on the left. The cell where they meet will tell you whether or not this combination is permitted.

When we say "copy code," we mean just that: you're taking a section of code from one source, with or without modification, and inserting it

into your own program, thus forming a work based on the first section of code. "Use a library" means that you're not copying any source directly, but instead interacting with it through linking, importing, or other typical mechanisms that bind the sources together when you compile or run the code. Each place that the matrix states GPLv3, the same statement about compatibility is true for AGPLv3 as well.

ADDITIONAL EXPLANATIONS FOR THE LICENSE COMPATIBILITY MATRIX

1. You must follow the terms of GPLv2 when incorporating the code in this case. You cannot take advantage of terms in later versions of the GPL.
2. While you may release your project (either your original work and/or work that you received and modified) under GPLv2-or-later in this case, note that the other code you're using must remain under GPLv2 only. As long as your project depends on that code, you won't be able to upgrade the license of your project to GPLv3-or-later, and the work as a whole (any combination of both your project and the other code) can only be conveyed under the terms of GPLv2.
3. If you have the ability to release the project under GPLv2 or any later version, you can choose to release it under GPLv3 or any later version—and once you do that, you'll be able to incorporate the code released under GPLv3.
4. If you have the ability to release the project under LGPLv2.1 or any later version, you can choose to release it under LGPLv3 or any later version—and once you do that, you'll be able to incorporate the code released under LGPLv3.
5. You must follow the terms of LGPLv2.1 when incorporating the code in this case. You cannot take advantage of terms in later versions of the LGPL.
6. If you do this, as long as the project contains the code released under LGPLv2.1 only, you will not be able to upgrade the project's license to LGPLv3 or later.
7. LGPLv2.1 gives you permission to relicense the code under any version of the GPL since GPLv2. If you can switch the LGPLed code in this case to using an appropriate version of the GPL instead (as noted in the table), you can make this combination.
8. LGPLv3 is GPLv3 plus extra permissions that you can ignore in this case.
9. Because GPLv2 does not permit combinations with LGPLv3, you must convey the project under GPLv3's terms in this case, since it will allow that combination.



AWARENESS IS THE KEY WHEN IT COMES TO USING LIBRARIES

- License of the iMatix Standard Function Library
- License Javascript of Netscape company (#NetscapeJavaScript)
- License of Perl version 5 and earlier (#PerlLicense)
- License of Python 1.6a2 and earlier versions
- License of Python 2.0.1, 2.1.1, and newer versions
- License of Ruby (#Ruby)
- License of Vim, version 6.1 or later
- License of ZLib
- BSD license without the advertising clause
- Mozilla Public License (MPL) version 2.0 - The more recent MPL version 2.0 is compatible with the GPL, due to a lot of hard license analysis work... making the MPL compatible with the GPL was one of the main reasons the MPL was updated. However, MPL version 2.0 has optional "incompatible with secondary licenses" clause, and if you include it the software is no longer GPL-compatible (so do not include that clause).
- OpenLDAP License, version 2.7
- Public Domain
- Sleepycat Software Product License (#BerkeleyDB)
- Standard ML of New Jersey Copyright License
- W3C Software Notice and License
- X11 License / MIT
- Zope Public License, version 2.0
- License of Guile

Source: http://gplv3.fsf.org/wiki/index.php/Compatible_licenses#GPLv3-compatible_licenses

LIST OF LIBRARIES COMPATIBLE WITH GPL V2:

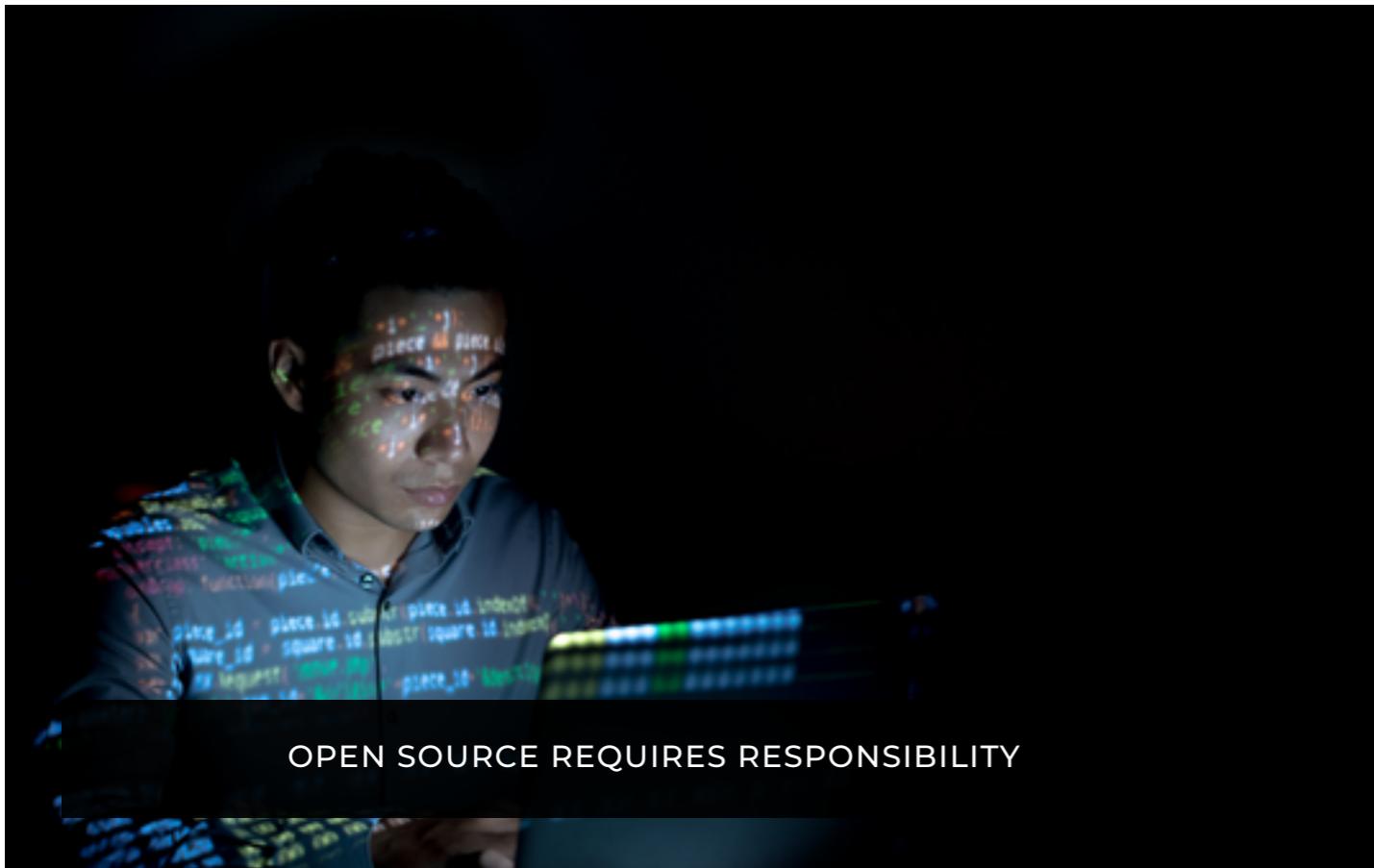
Below is a list of licenses compatible with GPL v2. Please note that compatibility does not work both ways, but this part only discusses the compatibility with the GPL v2 license.

- Artistic License 2.0 (#ArtisticLicense2)
- Boost Software License
- CeCILL version 2
- Clarified Artistic License
- Cryptix General License
- Eiffel Forum License, version 2
- EU DataGrid Software License
- Intel Open Source License (#intel)
- LGPL – Lesser General Public License, version 2.1 (#LGPLv2.1)
- License of eCos, version 2.0 (#eCos2.0)
- License of Expat (#Expat)
- License of FreeBSD (#FreeBSD)
- License of FreeType (#freetype)

LIST OF LIBRARIES COMPATIBLE WITH GPL V3:

Below we present a list of licenses compatible with GPL v3, keep in mind that this compatibility does not work both ways, it is only compatible when we consider GPL v3.

- Apache License, Version 2
- Affero General Public License, Version 3 (see sec. 13 of GPLv3)
- Lesser General Public License, Versions 2, 2.1 and 3 (LGPL)
- + licenses from GPL v2.



OPEN SOURCE REQUIRES RESPONSIBILITY

LIST OF LIBRARIES INCOMPATIBLE WITH GPL V2:

- Academic Free License, version 1.1.
- Academic Free License, version 2.1.
- Affero General Public License, version 1.0, 2.0
- Aladdin Free Public License (#Aladdin)
- Apache License, version 1.0, 1.1, 2.0
- Apple Public Source License, version 1.0, 1.1, 1.2, 2.0
- Artistic License 1.0 (#ArtisticLicense)
- Common Development and Distribution License
- Common Public License, version 1.0
- Creative Commons
- Eclipse Public License, version 1.0
- GNU Free Documentation License, version 1.2
- GPL for Computer Programs of the Public Administration (#GPL-PA)
- Hacktivismo Enhanced-Source Software License Agreement (HESSLA) (#HESSLA)
- IBM Public License, version 1.0
- Interbase Public License, version 1.0
- Jabber Open Source License, version 1.0

- Jahia Community Source License (#Jahia)
- LaTeX Project Public License 1.2, 1.3a
- License eCos, version 1.1 (#eCos1.1)
- License Lha (#Lha)
- Shared Source License of Microsoft company, used for CLI, C# and Jscript (#Ms-SS)
- License of PINE (#PINE)
- License of Python 1.6b1 and later versions, through 2.0 and 2.1
- Utah University Public License (#UtahPublicLicense)
- License of Scilab program (#Scilab)
- License of xinetd
- YaST License (#YaST)
- Lucent Public License version 1.02 (Plan 9 license)
- Mozilla Public License version 1.0, 1.1
- NASA Open Source Agreement
- Netizen Open Source License, version 1.0
- Netscape Public License
- Nokia Open Source License
- Old OpenLDAP License, version 2.3
- Open Public License (#OpenPublicL)
- Open Software License, version 1.0
- OpenSSL license
- Original BSD license
- PHP license, version 3.01
- Phorum License, version 2.0
- Q Public License, version 1.0
- Reciprocal Public License (#RPL)
- Old license of ksh93 (#ksh93)
- Old license of Plan 9 (#Plan9)
- Old license of Squeak (#Squeak)
- Sun Industry Standards Source License 1.0
- Sun Public License
- Sun Community Source License (#SunCommunitySourceLicense)
- SunSolarisSourceCode(FoundationRelease)License, Version1.1(#SunSolarisSourceCode)
- The Condor Public License
- XFree86 1.1 License
- Zend License, version 2.0
- Zope Public License, version 1

Source: http://gplv3.fsf.org/wiki/index.php/Compatible_licenses#GPLv3-compatible_licenses

MOST COMMON MISTAKES

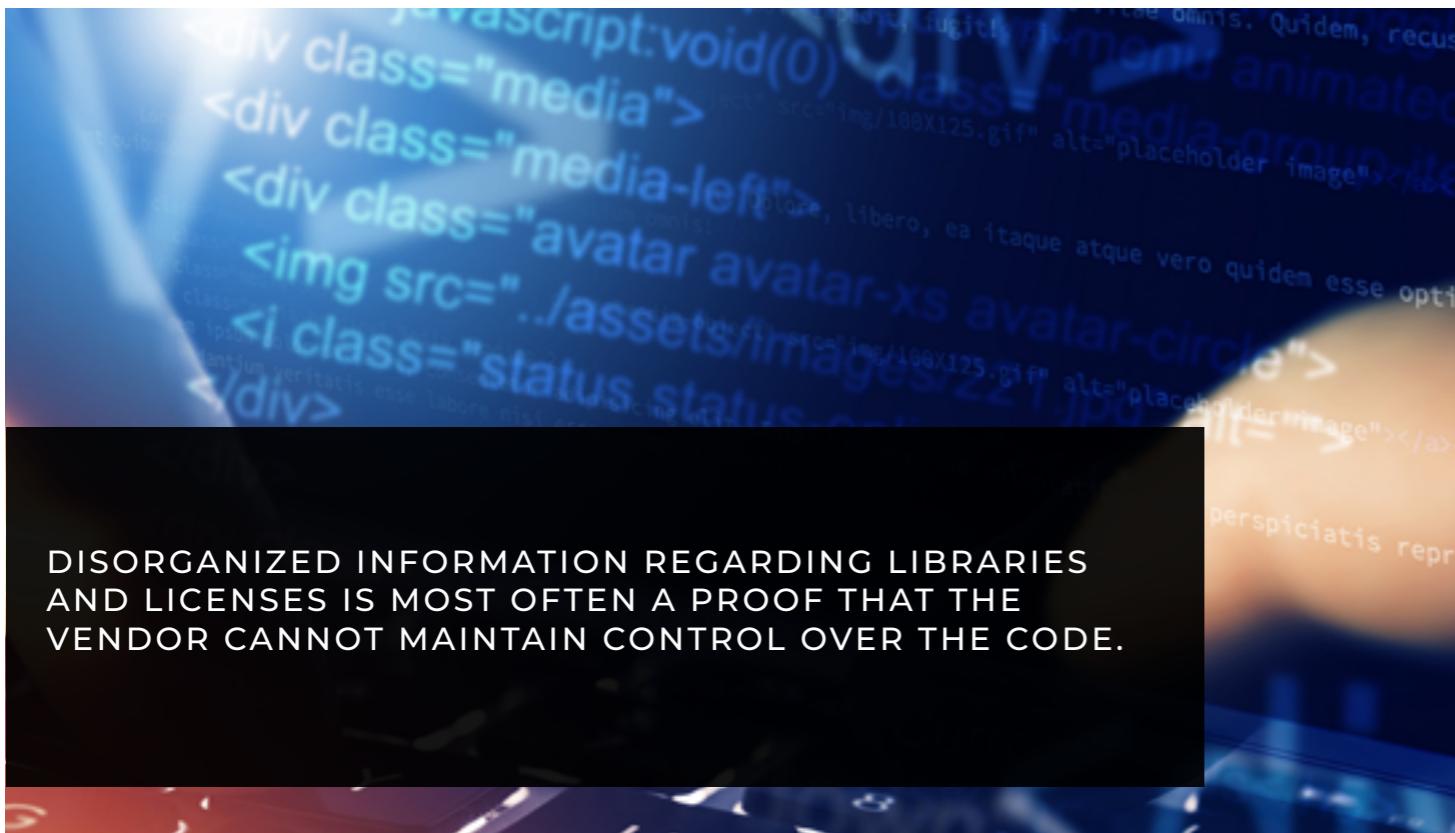
AS A SOFTWARE VENDOR, WE OFTEN ANALYZE THE CODE AND LICENSES OF OTHER SYSTEMS TO MAKE SURE THAT WE CAN USE PART OF THE CODE IN OUR SOLUTION.

DISORGANIZED LIBRARIES AND LICENSES

Searching for libraries in open source projects should be simple because well-designed PHP/JS software uses:

- composer [<https://getcomposer.org/>] - an organized set of PHP libraries
- yarn [<https://yarnpkg.com/lang/en/>] - an organized set of JavaScript libraries
- npm [<https://www.npmjs.com/>] - a file manager

On the basis of the data, you can easily create a list of libraries, dependencies, and licenses used in the system.



We verified several open source CRM systems, the results are as follows:

• **SuiteCRM**

- composer: only 10% of all used libraries
- yarn: no
- npm: no

• **VtigerCRM**

- composer: no
- yarn: no
- npm: no

• **EPESI**

- composer: only 10% of all used libraries
- yarn: no
- npm: no

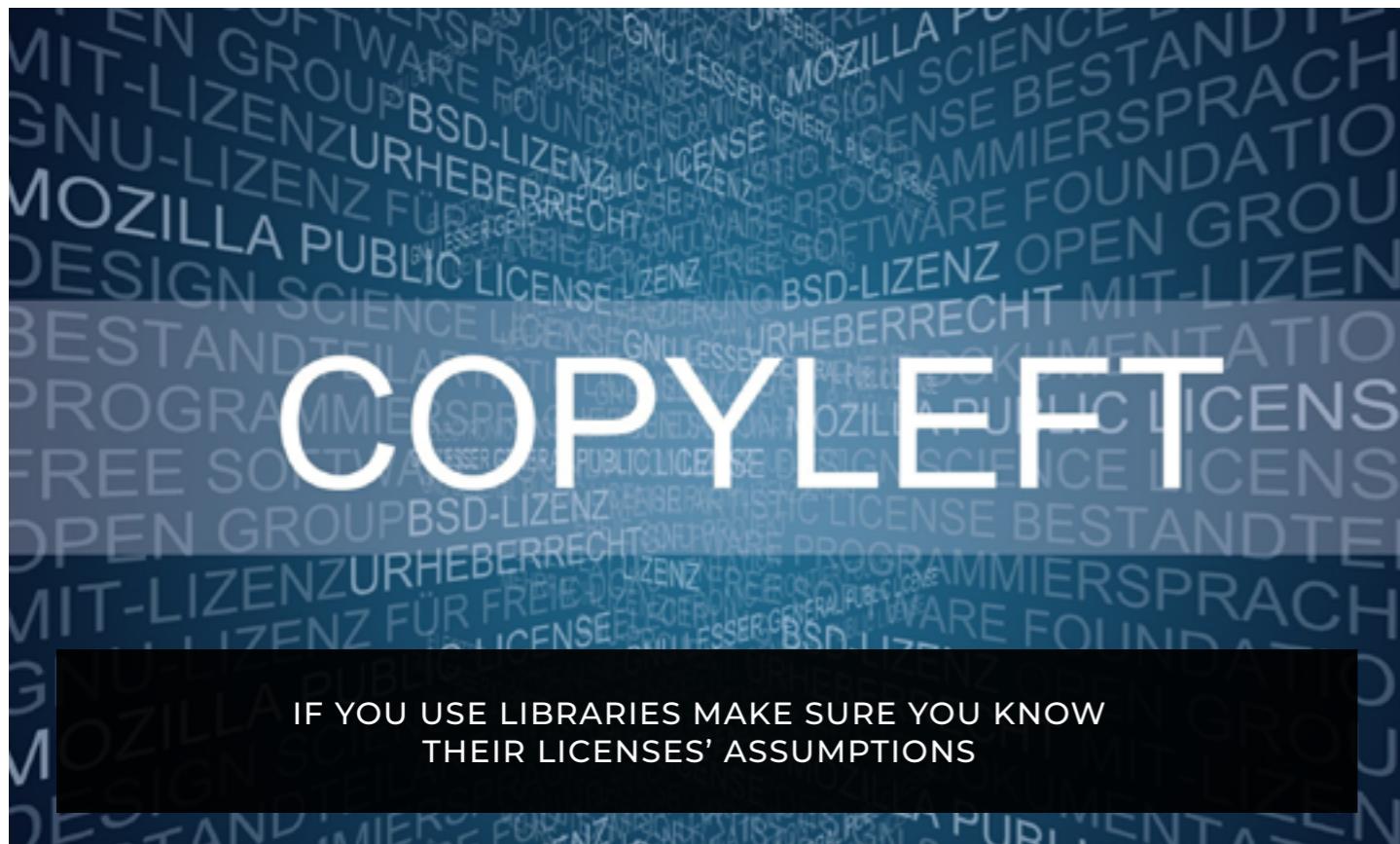
• **JoForce**

- composer: only 3% of all used libraries
- yarn: no
- npm: only 1 library

• **coreBOS**

- composer: only one library
- yarn: no
- npm: no

It is worth considering if changes in the system, libraries, and licenses can actually be controlled if it is not properly managed for the application globally.



COPYLEFT

IF YOU USE LIBRARIES MAKE SURE YOU KNOW
THEIR LICENSES' ASSUMPTIONS

DISTRIBUTION OF THE LICENSE WITH A COPYLEFT LIBRARY

ONE DISTINCTIVE FEATURE OF COPYLEFT LICENSES IS USUALLY THE RULE THAT IF YOU USE A CROSS-LICENSE LIBRARY IN YOUR SOFTWARE THEN THE ENTIRE SYSTEM MUST ALSO BE RELEASED UNDER THE SAME LICENSE.

EPESI

This problem is visible for example in the EPESI system that distributes the software under the MIT license but includes libraries that are incompatible license-wise. One of these libraries is [Roundcube](#) - an email client. Additionally, other viral licenses can be found, eg. [UiUIKit library](#) released under AGPL.

The only solution is to ask the EPESI vendor what the license of their product is - is it MIT? GPLv3? AGPLv3?

joForce

This case is similar to the problems found in EPESI, but on a much larger scale. The system is licensed under "JoForce Public License 1.0" by default, however, it seems to be the same license as VPL 1.1. Without going into the details, this definitely is not GPL or AGPL license. What we can see at the "first glance":

1. Libraries licensed under GPLv3:

- Roundcube:
[- jbPivot:
\[libraries/jquery/pivot/js/formatter_default.js\]\(https://github.com/joForce/JoForceCRM/blob/270dbecb208904af73e505d65974253b33de1e30/libraries/jquery/pivot/js/formatter_default.js\)
 - aggregate_distinct:
\[libraries/jquery/pivot/js/aggregate_distinct.js\]\(https://github.com/joForce/JoForceCRM/blob/270dbecb208904af73e505d65974253b33de1e30/libraries/jquery/pivot/js/aggregate_distinct.js\)](https://github.com/joForce/JoForceCRM/tree/e40d7df2f5774c0c4fb1d380d8eb20407dd55145/modules>EmailPlus/roundcube</div><div data-bbox=)

2. Libraries licensed under GPLv2:

- mPDF:
[modules/PDFMaker/mpdf/LICENSE.txt](https://github.com/joForce/JoForceCRM/blob/80cb61182f95174e96830465dab956a80f772369/modules/PDFMaker/mpdf/LICENSE.txt)

3. Libraries licensed under AGPLv3:

- ODFViewerPlugin:
[libraries/jquery/Viewer.js/PluginLoader.js](https://github.com/joForce/JoForceCRM/blob/270dbecb208904af73e505d65974253b33de1e30/libraries/jquery/Viewer.js/PluginLoader.js)
- webodf:
<https://github.com/joForce/JoForceCRM/blob/master/libraries/jquery/Viewer.js/webodf.js>
[VtigerCRM](#)

VtigerCRM

1. Libraries licensed under GPLv3:

- jbPivot:
<http://code.vtiger.com/vtiger/vtigercrm/blob/master/libraries/jquery/pivot/js/jbPivot.min.js>
- freefont:
<http://code.vtiger.com/vtiger/vtigercrm/blob/master/libraries/tcpdf/fonts/freefont/CREDITS>

2. Libraries licensed under AGPLv3:

- ODFViewerPlugin:
<http://code.vtiger.com/vtiger/vtigercrm/blob/master/libraries/jquery/Viewer.js/ODFViewerPlugin.js>
- webodf:
<http://code.vtiger.com/vtiger/vtigercrm/blob/master/libraries/jquery/Viewer.js/webodf.js>

Conclusion

These conflicts indicate how much these vendors still have to change in order not to infringe the licenses included in their software. Luckily, there are many great tools (eg. <https://fossa.io/>), that allow the clients to verify what the vendors have stated.

To solve licensing problems it is important to change the approach to software development so that licensing is always supervised and audited at each stage of development.

COMBINING LIBRARIES WITH COPYLEFT LICENSES

IN THE PREVIOUS PARAGRAPH, WE DISCUSSED A SITUATION WHERE WE ADDED A COPYLEFT LIBRARY TO OUR SYSTEM. IN THIS PARAGRAPH WE WILL DISCUSS A MORE COMPLEX SUBJECT, NAMELY ADDING A LIBRARY TO A SYSTEM RELEASED UNDER A COPYLEFT LICENSE. THERE ARE PLENTY OF SYSTEMS UNDER THESE LICENSES, FOR EXAMPLE:

- ▶ OroCRM [OSLv3]
- ▶ SuiteCRM [AGPLv3]
- ▶ Personal CRM [GPLv3]
- ▶ ERPNext [GPLv3]
- ▶ Dolibarr [GPLv3]

In theory, the matter of combining these libraries with other libraries is quite simple (from the formal side), you can find a list of copyleft licenses that are compatible with each other on the Internet. For example, a set of libraries compatible with each other has been described in the "List of GPL compatible libraries" section.

Let's take a look at how this compatibility looks in practice:

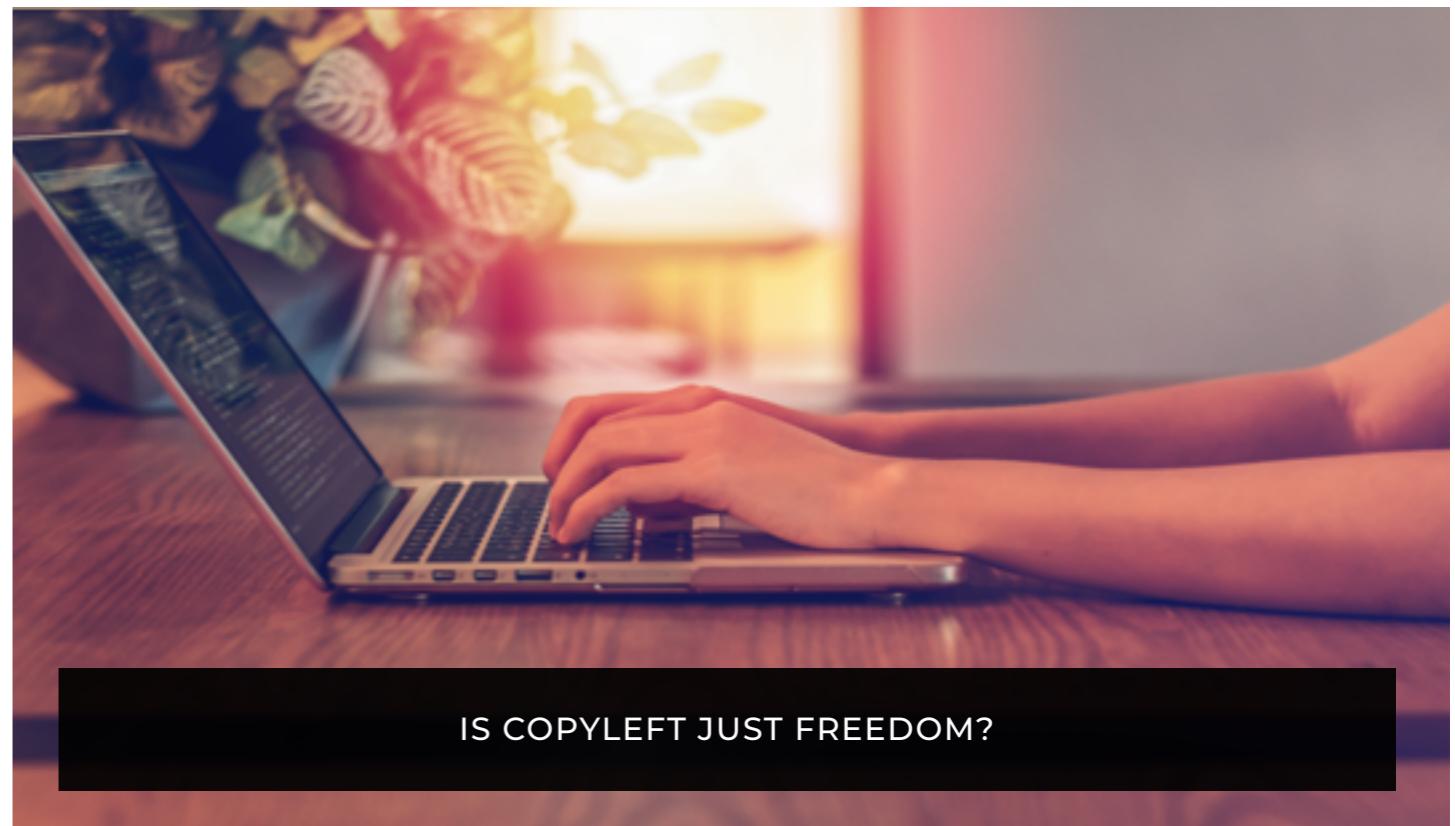
SuiteCRM

SuiteCRM, licensed under AGPLv3, includes the following libraries:

1. rgraph [GPLv2] [incompatible]:
<https://github.com/salesagility/SuiteCRM/blob/master/include/SuiteGraphs/rgraph/license.txt>
2. HTTP_WebDAV_Server [PHPv3] [incompatible]:
https://github.com/salesagility/SuiteCRM/blob/master/include/HTTP_WebDAV_Server/Server.php
3. Crypt_Blowfish [PHPv3] [incompatible]:
https://github.com/salesagility/SuiteCRM/tree/master/include/Pear/Crypt_Blowfish
4. XML_HTMLSax3 [PHPv3] [incompatible]:
https://github.com/salesagility/SuiteCRM/tree/master/include/Pear/XML_HTMLSax3
5. class.soap_transport_http.php [LGPL] [incompatible]:
https://github.com/Seven-7/website-gnuworld/blob/master/php_includes/SOAP/class.nusoap_base.php
6. pclzip [GNU/LGPL] [incompatible]:
<https://github.com/salesagility/SuiteCRM/blob/master/include/pclzip/pclzip.lib.php>

There are many libraries like these in SuiteCRM. Unfortunately, due to the lack of one central place to store the list of all libraries, it is very difficult to assess the severity of the problem.

OVERVIEW OF COPYLEFT LICENSES



IS COPYLEFT JUST FREEDOM?

OSLV3

Recent versions of the Open Software License have a term which requires distributors to try to obtain explicit assent to the license. This means that distributing OSL software on ordinary FTP sites, sending patches to ordinary mailing lists, or storing the software in an ordinary version control system, is arguably a violation of the license and would subject you to possible termination of the license. Thus, the Open Software License makes it very difficult to develop software using the ordinary tools of free software development. For this reason, and because it is incompatible with the GPL, we recommend that no version of the OSL is used for any software.

Using this license is very risky and requires a deep legal analysis in order to determine who and when has to approve the license. What is interesting, this license is not recommended even by copyleft advocates, as it significantly limits the development of the software: "We urge you not to use the Open Software License for software you write. However, there is no reason to avoid running programs that have been released under this license."

AGPLv3

If the program is used only internally by a company or by a private person for personal use, there is no need to make the source code of such software publicly available. AGPL v.3 requires that the user of the modified program be allowed to download the source code in the event of the code being downloaded "from an external source". However, this option cannot be used to charge a fee for sharing the program's source code with potential users outside the organization. In this case, the software is released "outside", so the condition provided by the license (and hence the obligation to share the source code free of charge) is fulfilled.

What may cause problems with the distinction between making software available "outside" and "inside" an organization cannot be clearly defined. On the one hand, an organization with thousands of members using the software may not make software available "externally" yet use it on hundreds of their employees' workstations. On the other hand, individual persons transferring their software may, in the light of AGPL v.3, be obliged to share the code.

This is due to the fact that organizations (here: companies) are treated by the law as persons separate from their employees. Therefore, if a given organization employs one hundred people, it is still one organization (one person). Doubts arise when access to the server is granted to persons who are not formally the Company's employees. In Poland, it will often be people running a sole proprietorship or people hired under a contract for a specific work. In most cases, such circumstances will not qualify as "internal" use of the software.

GPLv3

Out of these three copyleft licenses, GPLv3 is the most business-friendly one, even though all changes must be introduced as GPLv3, the moment when we have to provide someone with the source code is much less troublesome; for example when we modify the system and we distribute it further using a website.

Using an AGPLv3 licensed system in an organization is quite risky if we plan to expand and further develop the system because each change has to be made available for its users. Unfortunately, the majority of companies hide this information and do not provide any legal opinion about the license to its clients.

"Providing non-employees (sole proprietor, contract person) with access to a corporate intranet or internal network will be tantamount to software distribution and will therefore require that the code be made available. It should be remembered that each case of such actions requires a separate assessment and legal qualification

It should be remembered that in practice there are situations which are much more complicated. An example may be the creation of affiliated organizations (subsidiaries), where one company will be the "carrier" of intellectual property (licenses, marks, etc.) and the other company will be the "carrier" of the employees. The former entity may hire the employees only on a part-time basis in order not to activate the "copyleft virus" and the latter entity will actually pay the remuneration. Instances of this kind, however, are too complicated to be analyzed here on a step-by-step basis. In such cases it is therefore necessary to contact a competent specialist."

The AGPLv3 and OSLv3 licenses are very unfavorable business-wise and we strongly discourage you from using software released under these licenses without consulting a lawyer.

LICENSING TRAPS

AGPL

Google forbids using AGPL licensed software entirely in the whole company, regardless of the purpose of using the software. It also applies to regular applications downloaded onto the desktops of our computers

This page is part of Google's [open source documentation](#).

AGPL Policy

WARNING: Code licensed under the GNU Affero General Public License ([AGPL](#)) MAY NOT be used at Google.

The license places restrictions on software used over a network which are extremely difficult for Google to comply with. Using AGPL software requires that anything it links to must also be licensed under the AGPL. Even if you think you aren't linking to anything important, it still presents a huge risk to Google because of how integrated much of our code is. **The risks heavily outweigh the benefits.**

- Do not attempt to check AGPL-licensed code into google3 or use it in a Google product in any way.
- Do not install AGPL-licensed programs on your workstation, Google-issued laptop, or Google-issued phone without explicit authorization from the Open Source Programs Office.

In some cases, we may have alternative licenses available for AGPL licensed code.

Except as otherwise noted, the content of this page is licensed under [CC-BY-4.0](#) license. Third-party product names and logos may be the trademarks of their respective owners.

SISSL

Code licensed under Sun Industry Standards Source License (SISSL) should not be used in the organization. This license includes conditions that are very difficult to meet (even Sun, before the acquisition, stopped using or recommending this license).

WTFPL

Code released under WTPL should not be used in organizations. This license has many legal disadvantages (no warranty waiver, granting rights is very vague)

SuiteCRM

The company that stands behind SuiteCRM demonstrated an exceptional appetite for risk, as they decided to modify the original license terms published in the code by SugarCRM. Practically all license files have been altered by the following:



While SuiteCRM has the right to protect its ownership of the modifications, it is not allowed to interfere with the original license, in particular, if no changes have been made to the file.

“AGPL v.3 license does not permit modification of its terms. If such a modification is introduced, the so called attribution will apply, which means that the primary license terms on the basis of which the „basic” program was distributed, will be binding.

In practice, however, there are cases of such modifications. It should be kept in mind that changing the license terms may involve significant risks. The original licensor can enforce his rights by way of court proceedings against any entities violating his rights, even if they do so without awareness of the fact."

EXTERNAL MATERIALS:

GPLv3 Final Discussion Draft Rationale:
<http://gplv3.fsf.org/gpl3-dd4-rationale.pdf>

GNU GPL FAQ:
<https://www.gnu.org/licenses/gpl-faq.html#VersionTwoOrLater>

An example of an external library list for SugarCRM 7.9.0.0:
<https://support.sugarcrm.com/files/d38fd368-35dd-11e7-93de-06ac51574995.pdf>

License compatibility matrix:
<https://www.gnu.org/licenses/gpl-faq.en.html#AllCompatibility>

Make Your Open Source Software GPL-Compatible:
<https://www.dwheeler.com/essays/gpl-compatible.html>

Legal opinion regarding AGPL:
https://public.yetiforce.com/pdf/opinia_prawna_dla_licencji_agpl.pdf

Independent opinion:
<https://opensource.stackexchange.com/questions/5637/use-lapl-code-in-an-agpl-project>

Difference between GPL v2 and GPL v3:
<http://www.ifross.org/en/what-difference-between-gplv2-and-gplv3>

Torvalds: No GPL 3 for Linux:
<https://www.cnet.com/news/torvalds-no-gpl-3-for-linux/>

The Lesser GPL:
<https://copyleft.org/quide/comprehensive-gpl-quidech11.htm>

Open software licenses - practical guide:
<http://lookreatywni.pl/kategorie/prawo-autorskie/licencje-otwartego-oprogramowania/>

An example of an open source license infringement case:
<https://www.fsf.org/blogs/licensing/update-on-artifex-v-hancom-gnu-qpl-compliance-case-1>

Google's description regarding open source licenses
<https://opensource.google.com/docs/releasing/>

10 most frequently answered questions regarding GPL:
<https://resources.whitesourcesoftware.com/blog-whitesource/top-10-gpl-license-questions-answered>



YETIFORCE

CHOOSE OPEN SOURCE

CONTACT

PHONE

+48 884 310 123

WEB PAGE

www.yetiforce.com

www.yetiforce.shop

E-MAIL

info@yetiforce.com

YETIFORCE is the vendor of one of the best CRM systems in the world. The system is based on open source code and is constantly developed by its users. Thanks to its distinct and exceptionally liberal license, YetiForce is very business-friendly. It can be freely developed and modified according to the company's needs. As one of the most secure CRMs, YetiForce can cope with even the most demanding integrations with open and commercial systems.