

HELLO FROM TIDELIFT

The definitive guide to professional open source

Best practices for responsibly using open source components in your organization

August 2018

TIDELIFT

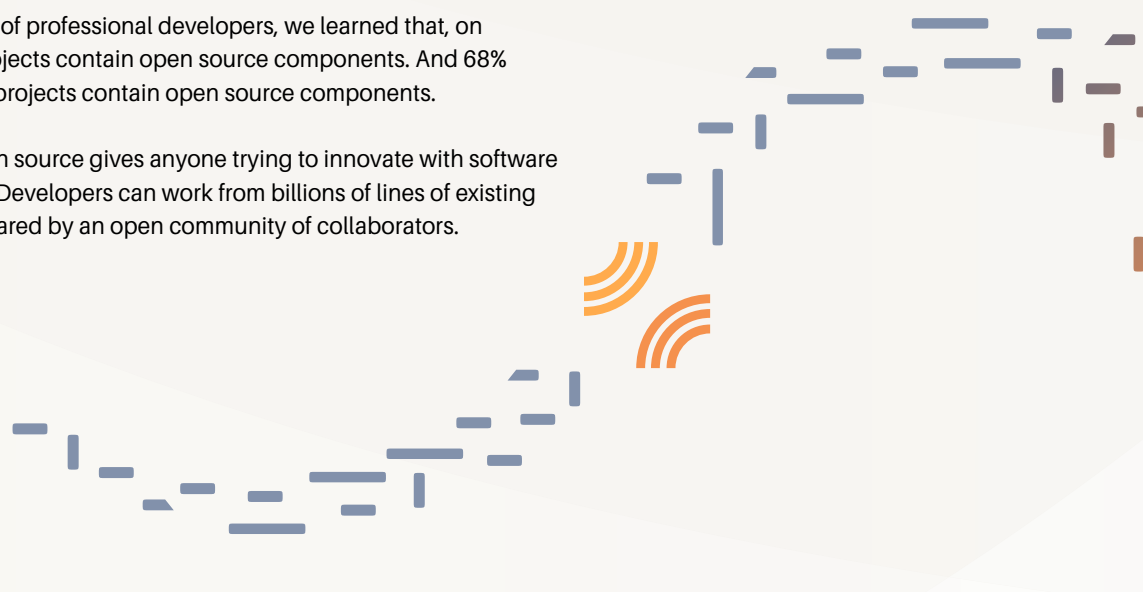
INTRODUCTION

Open source software is everywhere

Open source has come a long way. No longer a fringe activity, open source software is now everywhere you look. In fact, open source is the foundation of most of the software projects being created by commercial software development teams, from startups to large enterprises.

In a recent Tidelift study of professional developers, we learned that, on average, 92% of their projects contain open source components. And 68% reported that *all* of their projects contain open source components.

And it's no wonder. Open source gives anyone trying to innovate with software a formidable head start. Developers can work from billions of lines of existing code, developed and shared by an open community of collaborators.



“As open source components lower costs, improve quality and speed software development processes, they’ve quickly become the building blocks of the modern software supply chain. In fact, open source components make up about 80 percent of today’s typical applications.”

—Heather Meeker, [Wired Insights](#)

We mean, it is—quite literally—everywhere

Many people know open source in the context of standalone systems-level software like Linux, or applications like the Apache web server or MongoDB database. But often organizations don't realize that the software applications their in-house teams develop are also primarily composed of open source software in the form of software libraries and components.

Unless your organization is dramatically different from most, it's very likely that a high percentage of the underlying code behind the applications your software development teams build comes from third-party open source projects.

That's fantastic for many reasons—speed of development and cost savings being two of the main ones. But it does require some attention and diligence to deeply understand the software you use and how it's secured, maintained, and documented.

You may already be familiar with commercial services for open source provided by companies like Red Hat, Cloudera, and MongoDB. But even if you subscribe to services from vendors like these, the vast majority of the open source libraries that your business critically depends on aren't covered by any of these vendors. For example, Red Hat Enterprise Linux covers only 2,600 packages out of the more than 2.6 million in the broader open source ecosystem.

THE LIMITED REACH OF COMMERCIAL SERVICES FOR OPEN SOURCE



2600 packages supported by Red Hat¹

In addition:

- 2.6 million open source libraries across 36 different application-level package managers including npm (686k), Maven (112k), PyPI (124k)³
- 37% of websites include an open source library with a known open source vulnerability⁴

¹ Red Hat Software Repository

² GitHub "About Us"

³ Libraries.io

⁴ Northeastern University research study

Open source is 100, but it's got issues...

Open source is awesome. But what are the tradeoffs for being able to make use of all of that amazing open source code?

The pain of making it all work together

First, your highly compensated software development team wastes valuable time assembling, integrating, and testing open source components—ensuring they play well together. This is work that could be done better and at a lower cost by a dedicated third party outside your organization. Ideally it would be done by someone who can focus on these tasks as their primary responsibility, investing the effort necessary to do the work once comprehensively, with many organizations sharing the benefits.

The pain of keeping it secure

Second, your development team must take on the significant burden of policing the security status of the hundreds of individual components you integrate, or accept the risk of breaches like the one that impacted Equifax (and there are countless similar horror stories).

The pain of staying compliant

All of these great open source components come with a confusing array of different types of licenses. To safely use open source software, your business must also be willing to investigate and police the licensing status of hundreds of individual components you integrate— or accept the risk of accidentally infringing other parties' intellectual property rights and licenses by using them incorrectly.

What we've achieved with open source is incredible. But there is an opportunity we can't afford to miss to make it even better for everyone—and it's hiding in plain sight.

WHAT YOU'LL LEARN IN THIS GUIDE

Our goals for this guide:

- ▶ Help you understand the issues faced by professional teams building on open source
- ▶ Show how Tidelift can help you assess the use of open source in your organization
- ▶ Explain how to define an open source policy for your organization and implement it with a Tidelift Subscription



HOW DOES TIDELIFT HELP?

We created Tidelift in part because we wanted to make it easier for software development teams to continue to get the amazing value out of the open source components they already use—while at the same time helping to manage risk from security, maintenance, and license issues.

Tidelift leads the Libraries.io project, which maintains the largest database of open source components on the web, with comprehensive information on 3.1 million open source packages. We've used this massive dataset as the backbone of the Tidelift Subscription, a paid service that gives software development teams a single source for purchasing and maintaining their software with professional-grade assurances from the experts who know it best.

STEP ONE

Understand open source software risk areas and opportunities

What is software package management?

Open source software components or libraries are managed by tools called package managers. In open source, there is typically a distinct package manager for each programming language ecosystem or “stack.”

For example, the Javascript/Node.js ecosystem mostly uses the npm package manager, the Java ecosystem predominantly uses Maven, and the Python ecosystem generally uses the Python Package Index (PyPI).

While the collaborative process of open source software development typically happens on hosted source code collaboration platforms like GitHub, GitLab, and Bitbucket, most open source software is then assembled into package releases that go into the public package manager repositories.

To save time and make it easier to ensure the software is installed and runs properly, most commercial users of open source actually acquire their software from the package manager repositories, not directly from these code collaboration platforms.

WHAT IS A PACKAGE MANAGER?

A package manager is a set of software tools that automates the process of installing, upgrading, configuring, and removing computer programs in a consistent manner. Package managers help ensure everything is in place for software to run properly.

HOW DOES TIDELIFT HELP?

- Tidelift gives you a comprehensive view of these package managers and packages, including the vast majority of programming languages and open source communities.
- Tidelift has built the most complete index of open source packages in the world. We’ve done it by indexing more than 36 package manager ecosystems as well as GitHub, GitLab, and Bitbucket.

- Through our open-source project and public service, Libraries.io, Tidelift tracks more than 3.1 million open source packages.

Think of it as almost like a search engine for open source software. A way to find almost every available open source package and understand how they all relate to each other. Which brings us to...



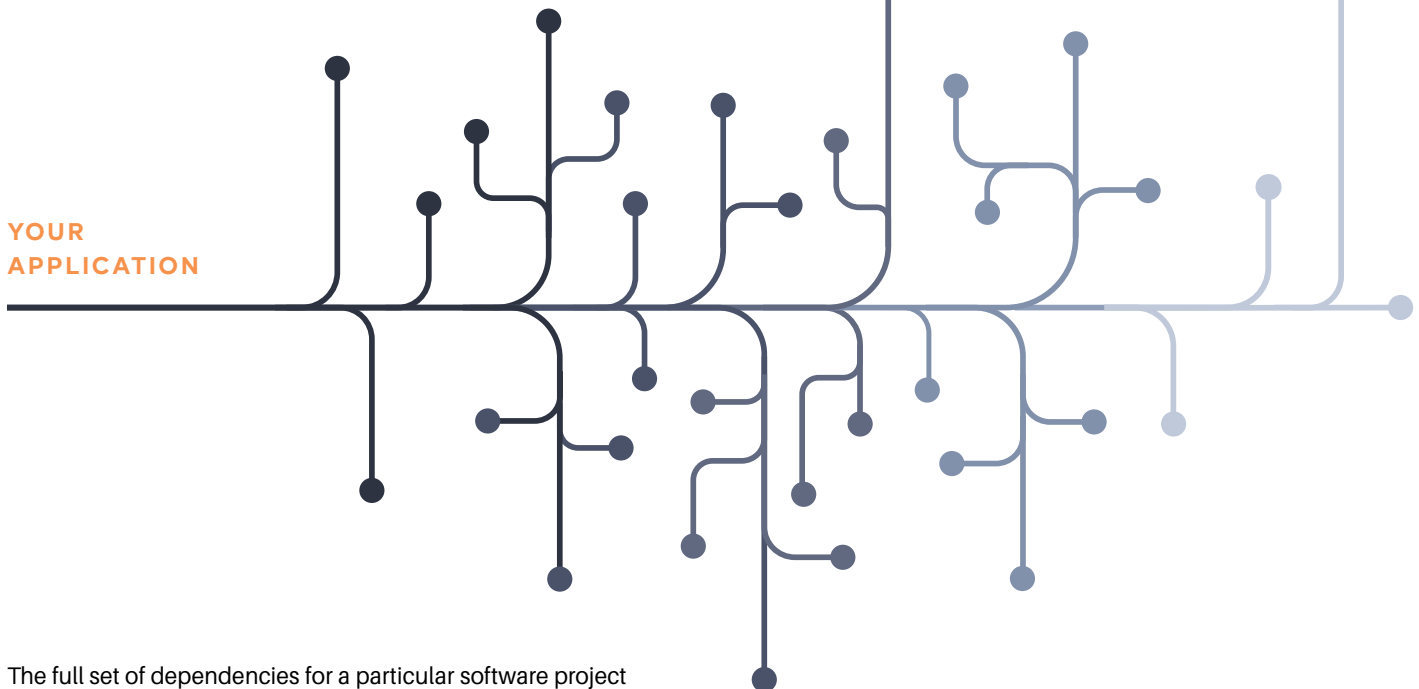
Understanding dependencies in software projects

Software applications have always been built by assembling many small pieces of software, generally delivered as software packages—and alternatively called components or libraries. When a component is added to your software application, your application is said to “*depend on*” it, and the component is now a “*dependency of*” your application.

Each individual component may have many dependencies of its own, each of which may have many dependencies, and so on. The full set of software components forms a *dependency tree*.

HOW MANY DEPENDENCIES?

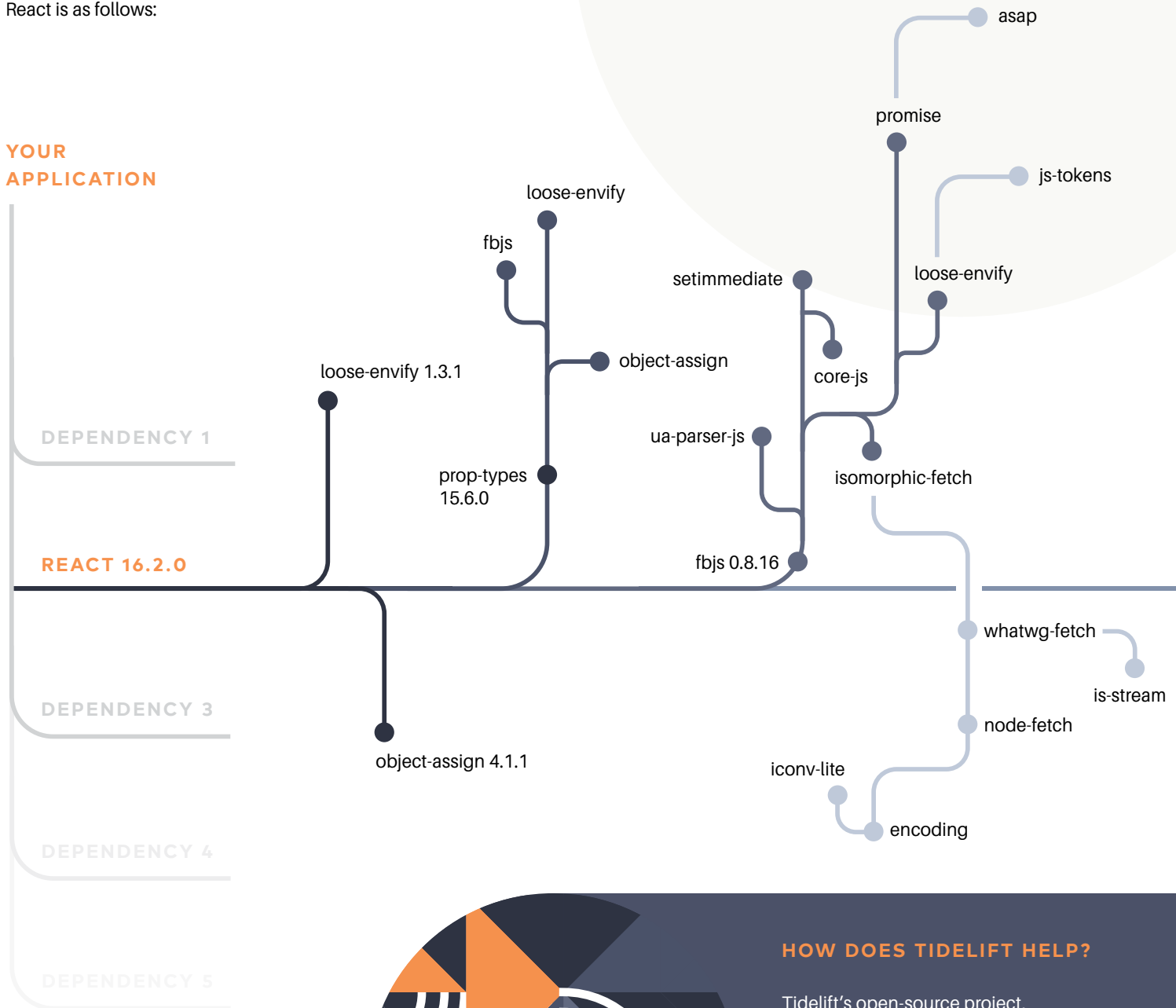
Direct dependencies are those you’ve explicitly chosen to rely on; but direct dependencies in turn pull in other packages, called transitive dependencies. Typical JavaScript applications might have less than ten direct dependencies, but around a thousand transitive dependencies. These transitive dependencies often haven’t been reviewed by anyone on the application team.



The full set of dependencies for a particular software project or application, including dependencies of dependencies and so on, is known as its transitive dependencies.

For example, the dependency tree for the Javascript front-end package React is as follows:

YOUR APPLICATION



HOW DOES TIDELIFT HELP?

Tidelift's open-source project, Libraries.io, tracks and analyzes more than 300 million open source package dependency relationships. This helps software development teams better understand the complex set of relationships underlying the software they use in their applications.

Open source software is amazing, but it's still software

Open source is a fabulous approach to software development, but it doesn't magically make the challenges that apply to all kinds of software go away. Each open source dependency that your application uses represents a distinct piece of software with its own license, release lifecycle, security and maintenance practices, and so on.

It just takes one major issue in an open source package anywhere in the dependency tree for your application to cause a security breach. That risks exposing customer data, creating a customer-facing systems outage, or leading to an expensive and time-consuming intellectual property infringement claim against your organization.

Open source doesn't magically make the challenges that apply to all kinds of software go away.

HOW DOES TIDELIFT HELP?

Tidelift helps overcome these obstacles to professional success with open source. In the following section, we'll dive into issues around security, maintenance, and licensing—and demonstrate how Tidelift can help.

RISK AREA

Security

Open source software components, like all software, are susceptible to security vulnerabilities caused by programming oversights, or even intentional sabotage. Government-sponsored projects like [Common Vulnerabilities and Exposures](#) (CVE®) compile lists of common identifiers for publicly known cyber security vulnerabilities, including those in open source software.

However, few commercial organizations have robust and effective processes in place to ensure that all of their software is continually assessed for known vulnerabilities.

One painfully familiar example is the [Equifax exploit of 2017](#).



“Equifax has confirmed that attackers entered its system in mid-May through a web-application vulnerability that had a patch available in March. In other words, the credit-reporting giant had more than two months to take precautions that would have defended the personal data of 143 million people from being exposed. It didn’t.”

—Wired Magazine, [“Equifax Has No Excuse”](#), September 2017

While it’s easy to point fingers at Equifax, it’s also worth asking whether your own organization has the tools and processes in place to track vulnerabilities in all of your applications and their open source dependencies. As we’ve already seen, the number of dependencies in a single application can easily run into the hundreds of packages.

And even once a vulnerability is identified, fixing it is another matter.

“Patching the security hole was labor intensive and difficult, in part because it involved downloading an updated version of Struts and then using it to rebuild all apps that used older, buggy Struts versions. Some websites may depend on dozens or even hundreds of such apps, which may be scattered across dozens of servers on multiple continents. Once rebuilt, the apps must be extensively tested before going into production to ensure they don’t break key functions on the site.”

—Ars Technica, [Failure to patch two-month-old bug led to massive Equifax breach](#), September 2017

RISK AREA

Maintenance

Virtually all developers are familiar with the experience of *dependency hell*.

The business impact of issues like dependency hell is that software teams waste incalculable amounts of time chasing and addressing software component compatibility issues, instead of solving their actual business problems.

The resulting business impacts are quiet but painfully real:

- Reduced developer productivity
- Expensive software team salaries wasted on trivial but time-consuming issues
- Developer frustration and resulting employee retention challenges
- Software projects that take longer to complete and overrun budgets

WHAT IS DEPENDENCY HELL?

Dependency hell is the term software developers use when they've installed software packages which have dependencies on specific versions of other software packages. The "hell" happens when packages have dependencies depending on different and incompatible versions of the shared packages. If the shared package or library can only be installed in a single version, the developer may need to obtain newer or older versions of the dependent packages—possibly breaking other dependencies and pushing the problem to another set of packages.

RISK AREA

Licensing

Software licensing and intellectual property law are key “technologies” that enable open source software to exist in the first place. Every piece of open source software guarantees you the ability to use it, but also has at least some minimal restrictions on how it can be used. Around 10% of packages have even stronger restrictions, usually called “copyleft.”

However, especially in large software projects with many dependencies, open source license compliance can become a major complicating factor.

WHAT IS A COPYLEFT LICENSE?

“Copyleft” or “reciprocal” licenses require sharing of modifications under certain conditions. Examples include the well-known GNU General Public License and a spectrum of others, including the “network” Affero GPL (whose conditions may be triggered by use in services) and a variety of “weak” copylefts like the Eclipse and Mozilla licenses (whose conditions generally require sharing of fewer classes of changes).

“Open source and licenses can be a massive headache even for companies with massive resources at their disposal, given so many different licenses can be incompatible with each other. Just trying to get accurate and complete license information for anything can be a time-consuming headache especially on a large scale. At the same time, companies also need to better understand the issues around complying with open source licenses.”

—The New Stack, [SPDX Could Help Organizations Better Manage Their Thickets of Open Source Licenses](#)

Often, individual software developers are tasked with making critical decisions about the software licenses of their open source components—a complex and nuanced area of expertise for which they are not trained.

The resulting business impacts can be painful, and include:

- Exposure to litigation by third parties over breach of license or copyright
- Accidental loss of company intellectual property rights, such as the ability to enforce patents
- Expensive and unscheduled costs when components must be removed late in the software development process to remediate licensing issues
- Delay or outright cancellation of financings, mergers & acquisitions, initial public offerings, and other corporate transactions
- A legal injunction against continuing to distribute a product while it has an infringement claim outstanding
- Requirements to distribute source code for all or parts of your company's proprietary software (depending on the licenses involved), potentially revealing intellectual property or trade secrets
- Customer service and support costs associated with changing a product that is already deployed

“Compliance is also a bit of a headache for companies that are not adept at dealing with open source. Ensuring that your company distributes the source and complies fully with the GPL, including making sure that if the GPL is combined with other works that it is license-compatible, can require a fair amount of attention.”

— CIO Magazine, [How Open Source Licenses Affect Your Business and Your Developers](#)

HOW DOES TIDELIFT HELP?

Tidelift helps your software teams avoid all three of these risk areas by:

- Mapping the software components your application depends on (including open source dependencies) using Tidelift's catalog of over 3.1 million open source components
- Continually providing timely insights, alerts, and notifications about the security status of your software components
- And perhaps most importantly, working with open source developers to provide timely fixes for vulnerabilities in open source components, as well as necessary context on compatibility required for rolling those fixes out with confidence

STEP TWO

Assess your organization's current use of open source

Once you understand the issues involved with using open source in a professional environment, the next step is to get a comprehensive view of your organization's open source usage by mapping your dependency trees. While this was once a near-Herculean task, it is now much easier thanks to Tidelift.

Get an open source assessment from Tidelift

With support for Javascript, Java, Python, PHP, and 20 more languages and package managers, our open source software assessment will give you a unified view of all open source components your organization is already using.

On a call with one of our open source experts, we'll:

- Check for deprecated and unmaintained packages
- Highlight missing licenses
- Identify security vulnerabilities
- Dive into direct and transitive dependencies

To schedule an open source assessment, go to:

<https://tidelift.com/subscription/request-an-ossa>



Developing a plan

Now that you have a comprehensive list of the open source components your organization uses, and a clearer view of any current issues, it's time to come up with a plan to address those problems.

One path forward is to start trying to remediate issues by upgrading individual packages, contributing your own pull requests to upstream open source packages, or removing problematic dependencies from your application and choosing other packages, or implementing similar functionality from-scratch yourself. Then, keep an eye on the dashboard and rinse, repeat.

But at Tidelift, we're working to provide a more comprehensive—and scalable—model. When you purchase the Tidelift Subscription, you can rely on expert maintainers partnered with Tidelift to do that work for you, in a professional manner—fixing issues that are visible today, and proactively avoiding and resolving issues that come up in the future.

We recommend thinking about the Tidelift Subscription in the context of an organization-wide open source policy.

At Tidelift, we're working
to provide a more
comprehensive—and
scalable— model.



STEP THREE

Define an open source policy and implement it with Tidelift

Define an open source policy for your organization

To ensure good business outcomes and to comply with necessary regulatory and risk management requirements, most organizations implement some form of a software vendor management program.

Often enforced by a procurement or legal function, the purpose of a vendor management program is to ensure that software used within the organization is dependable.

Vendor management programs require vendors to provide maintenance and updates for the software over a defined life cycle. They also typically require the vendor to make representations about the provenance of the intellectual property that comprises the software itself.

WHAT IS A VENDOR MANAGEMENT PROGRAM?

A vendor management program requires software vendors to provide representations about issues that are discovered or may arise in the future—and put in place formal means to correct them.



But who provides representations for open source software?

Strangely, while these requirements are universally enforced for purchased software, many organizations overlook these basics for the open source components they download for free.

Why? Sometimes, this responsibility falls in the cracks between engineering and functional areas like procurement and legal. Practically speaking, if there is no cost for the software, then there is no reason for procurement to get involved. And if there is no agreement to sign, then there is no reason for the legal team to get involved.

So, purely because of the way procurement of software is handled, open source components often skirt around the requirements of the vendor management program.

But—and this is the important part—this does not mean that the open source components are any less likely to suffer from the same issues that lead organizations to develop vendor management practices in the first place.

There are still maintenance issues to be addressed. Representations about the security, dependability, and intellectual property ownership of the software that are needed.

And historically, in the context of open source software, there has typically been no one to provide them.

Sure, there are a few vendors covering a very small part of the open source landscape. Some of them—like Red Hat for the Linux operating system—have been widely adopted.

But for the vast majority of open source components, the kind of representations responsible organizations are looking for simply don't exist today. This is a missed opportunity that saddles enterprise organizations with long-term maintenance and upkeep headaches, as well as vulnerability to lots of bad outcomes.

HOW DOES TIDELIFT HELP?

With the Tidelift Subscription, Tidelift makes it simple for organizations to get open source software they can depend on.

- We bring together the maintainers of the components you are already using.
- We work with them to deliver a standardized set of maintenance, security, and licensing services as part of the Tidelift Subscription.
- Through the Tidelift Subscription, we provide your organization with the same representations of dependability you would expect of any other software vendor.
- Your legal and procurement teams engage with one vendor—Tidelift—saving both time and money. And you get broad coverage for your open source dependencies.

Incorporate the Tidelift Subscription into your software development process

With the Tidelift Subscription in place, organizations can now get many of the same assurances around open source software they could from other commercial software vendors.

What's more, they can safely monitor their dependencies and stay up to date with notifications about updates, license incompatibilities, or deleted dependencies.

We're continuing to add coverage for more parts of the open source landscape all the time. When you use Tidelift to monitor your open source dependencies, you'll be alerted when the subscription adds coverage for packages you use.



It's time to professionalize how we use open source software

Open source has fundamentally changed software development for the better. It allows us to do things with software that were never before possible, accelerating progress in our organizations, and improving our world along the way.

But using open source also introduces challenges—specifically in the areas of security, maintenance, and licensing.

Our mission at Tidelift is to make open source work better—for everyone.

If this professional approach to using and managing open source software sounds right for your business and you'd like to learn more:

- [Contact us](#) to learn more about Tidelift subscriptions, start a free trial or to get a demo
- [Request an open source assessment](#) to better understand your current dependencies
- [Get a demo](#) of the Tidelift Subscription
- Visit [our website](#) to sign up for updates or map your dependencies with our free dependency analysis

