Open Source Compliance

Prof. Dr. Dirk Riehle

FAU Erlangen / Bayave GmbH

Open Source @ Siemens 2026

Open Source: Safe and Easy?

Open-source software is great! You get high-quality software for free

- Most products and projects are 80%-99% open source code

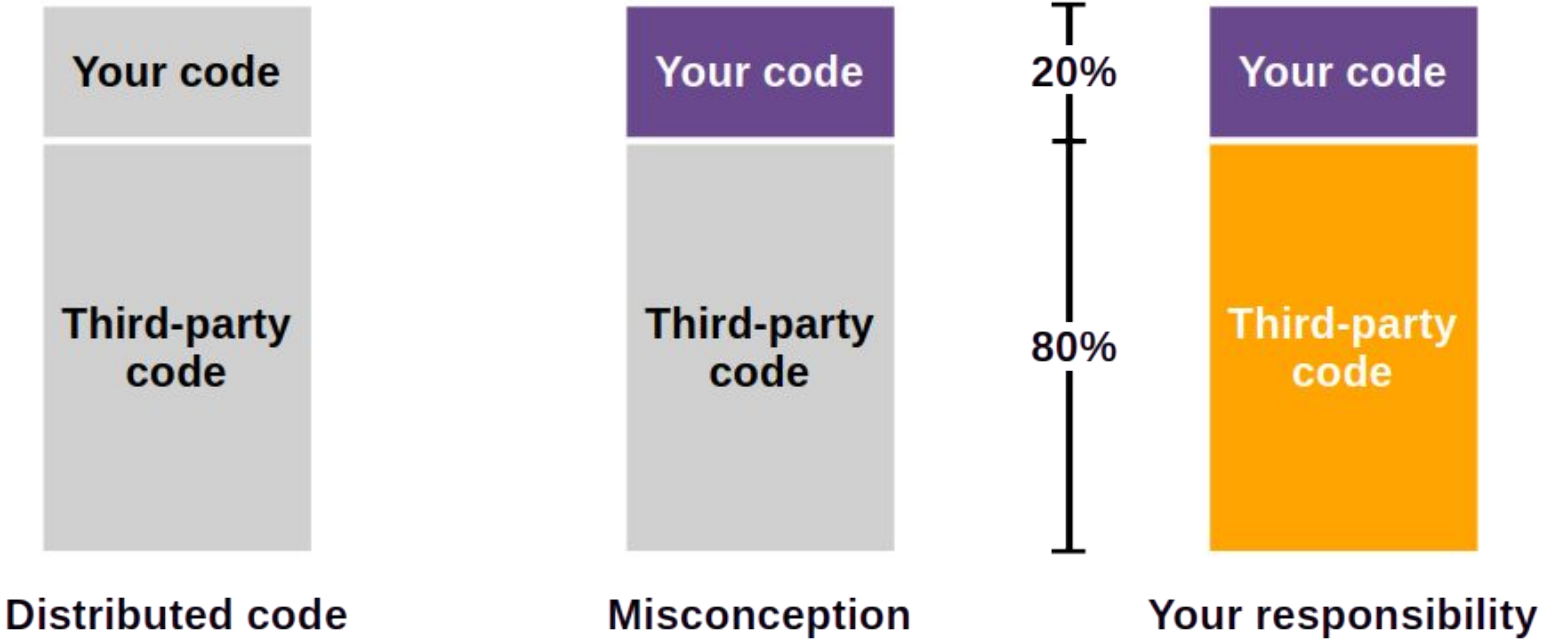
But: Building products from open-source software comes with risks

- You might lose your intellectual property rights → business model
- You might get sued for license incompliance → license compliance
- Your products might not be secure → customer embarrassment
- Your products might not comply with the laws → regulatory compliance

We want to make using open source (= open source compliance) safe and easy

- More at https://scatool.com

The Scope of the Compliance Responsibility



Agenda

1. SBOM management
2. Open source governance
3. License compliance
4. Vulnerability management
5. Regulatory compliance

Projects
Issues
Settings

Projects

+ New

Search...

Project	Issues
Demo Project	0
Vercel Serve	0

Support
Feedback
Collapse

1. SBOM Management

Software Bill of Materials (SBOM)

A list of all code components (snippets, stand-alone units) in your product

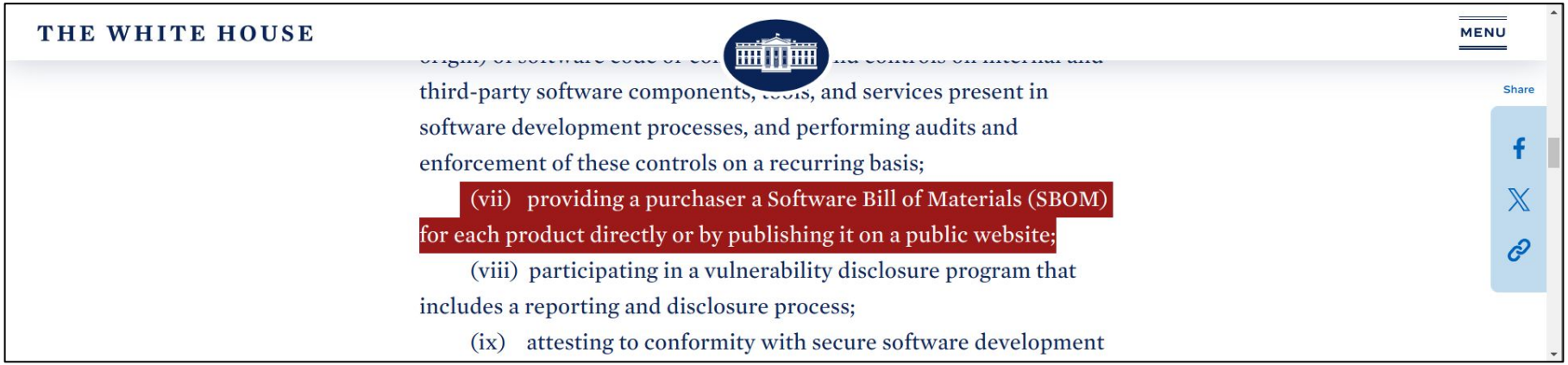
- A snippet is a piece of code embedded in other (licensed) code
- A stand-alone unit is its own compilation unit (file)

Minimal record for each code component

- Code component identification
  - Context name
  - File name
  - Version
- License
- Source

SBOMs are a Purchasing Requirement [1]

From the U.S. White House executive order on improving the nation's cybersecurity



(Not just the U.S. government: Most large customers long wanted to know.)

[1] See https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

SBOMs are a Critical Data Structure for Any Product

You need SBOM Management [1]; without an SBOM

1. Your intellectual property isn't safe [2]
2. You might get sued for license incompliance [3]
3. You won't know about new vulnerabilities [4]
4. You don't how to comply with regulation

But no two tools today will give you the same SBOM

[1] See https://scatool.com/solutions/sbom-management/
[2] See https://scatool.com/solutions/open-source-governance/
[3] See https://scatool.com/solutions/license-compliance/
[4] See https://scatool.com/solutions/vulnerability-management/

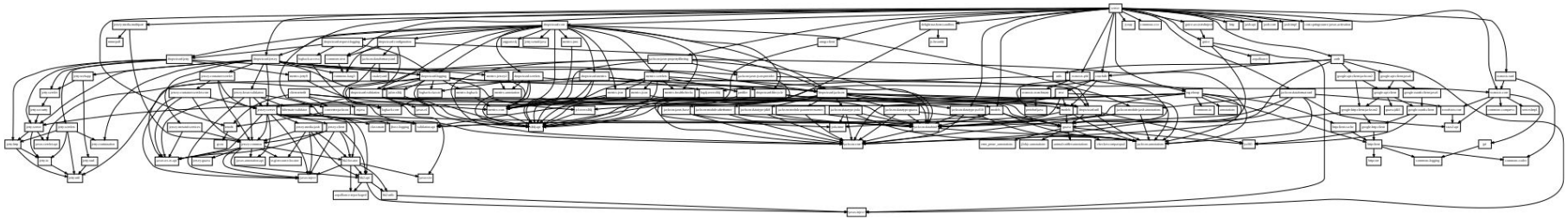
Software Composition Analysis (SCA)

Software composition analysis is

- The analysis of your code base for any included third-party code
- Resulting in an SBOM

The focus of software composition analysis is

- All code reachable as part of your dependency graph



Standards for Software Bills of Materials

SPDX standard

- ISO / IEC 5962:2021: https://www.iso.org/standard/81870.html

Other formats

- OWASP CycloneDX: https://owasp.org/www-project-cyclonedx/

Why Software Composition Analysis is so Laborious [1]

A build tool typically gives developers

- All needed compilation units to build your product

Beyond this, for a complete SBOM, you need to include

- **All embedded third-party code snippets**
- **For all such code components, license texts**
- **For all such code components, copyright notices**

Package managers often

- Don’t have the information
- Are hopelessly out of date

[1] See https://bayave.com/sca for software composition analysis services

Overview
Packages
Develop
Deliver
Monitor

Creation
05/16/2026, 11:00
Git repository

Statistics
1 module
691 dependencies
36 vulnerabilities

Configuration
github.com/vercel/serve
f3c702c Version Packages (#846)
03/03/2026, 20:04

Reference
Version: main

Dependency Graph Analysis → Source Code Analysis → Analysis Completed

3. Analysis Completed

The analysis has been successfully completed. You can now view the results on their respective pages.

Support
Feedback
Collapse

Overview
Packages
Develop
Deliver
Monitor
Support
Feedback
Collapse

Packages

Package	Type	Scope	Clearance	Security Risk
@babel / helper-simple-access 7.25.7	transitive	development	0%	NONE
@babel / helper-string-parser 7.25.7	transitive	development	0%	NONE
@ba… / helper-validator-ide… 7.25.7	transitive	development	0%	NONE
@ba… / helper-validator-opti… 7.25.7	transitive	development	0%	NONE
@babel / helpers 7.25.7	transitive	development	0%	MEDIUM 6.2
@babel / highlight 7.25.7	transitive	development	0%	NONE
@babel / parser 7.25.8	transitive	development	0%	NONE
@babel / runtime-corejs3 7.25.7	transitive	development	0%	MEDIUM 6.2
@babel / runtime 7.25.7	transitive	development	0%	MEDIUM 6.2
@babel / template 7.25.7	transitive	development	0%	NONE

Rows per page: 10 11–20 of 692

color-convert 1.9.3 69%

Start Curation

Pending (31%) Blind Accept (69%) Reviewed (0%) Excluded (0%)

Package Metadata

PURL	pkg:npm/color-convert@1.9.3
Homepage	https://github.com/Qix-/color-convert#readme
Declared License	MIT
Discovered License	MIT

Source Code

Type	VCS
Provider	Git
Repository URL	https://github.com/Qix-/color-convert.git
Revision	99dc5da127d3d17d0ff8d13a995fd2d6aab404aa

View on GitHub

Source Code Analysis Data

Status	SUCCEEDED
Total Files	13
Last Analysis	2026-02-09T08:21:31.415384Z

Search...

test			0-0
.editorconfig			0-0
.gitignore			0-0
.npmrc			0-0
.travis.yml			0-0
CHANGELOG.md			0-0
component.json			0-0
conversions.js		MIT	0-1
index.js			0-0
LICENSE		MIT	1-1

2. Open Source Governance

Open Source Licenses

Open source licenses grant you the rights to

- Use, modify, distribute, modified or not, the software, all free of charge
- If and only if you comply with the obligations of the licenses

Fail to comply with the obligations, and

- **You lose the right to use the software**

Open Source License Obligations

Open source license obligations depend on the use case

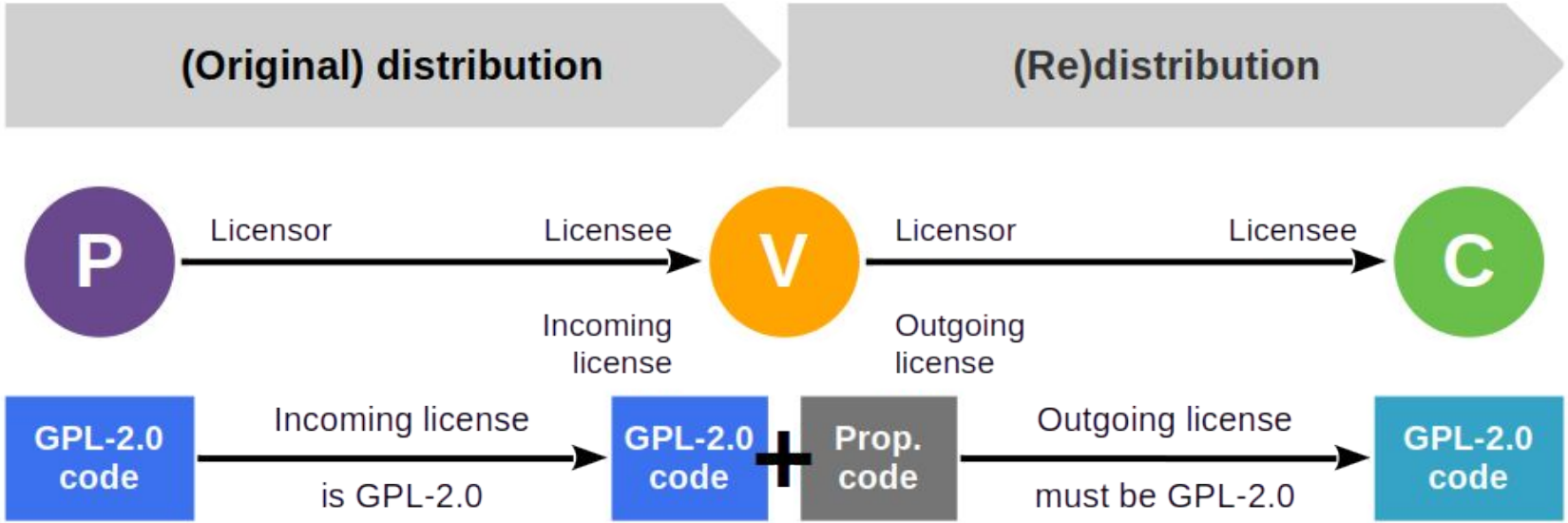
- End-user: You are just using the open-source software
- Distributor: You are distributing the open-source software [1]

License obligations typically only apply to distributors

- Many license obligations seem benign (attribution, provision of license texts)
- **Some license obligations may pose a problem (copyleft clauses)**

[1] Distributor is a general term that applies to anyone who passes on open source code

The Copyleft License Strategy [1]



P = Programmer
V = You (developer / vendor)
C = Customer

[1] See https://bayave.com/osc for open source compliance training

Get the latest update on our Vizio court case →

software freedom conservancy

News | About | Our Work | Tools | Learn | Donate

Copyleft Compliance
We defend and uphold the rights of software users and consumers under copyleft licenses.

Impact Litigation
We defend the legal rights of software users. Learn the details, status, and stakes of our court cases.

Give Up GitHub
We urge FOSS Developers to *Give Up GitHub!* Learn why.

Outreachy
We offer internships for anyone who faces underrepresentation, systemic bias, or discrimination in the tech industry.

FOSSY
Our annual community-oriented conference focused on the creation and impact of free and open source software.

Software Freedom Conservancy is a nonprofit organization centered around ethical technology. Our mission is to ensure the right to repair, improve and reinstall software. We promote and defend these rights through fostering free and open source software (FOSS) projects, driving initiatives that actively ma technology more inclusive, and advancing policy strategies that defend FOSS (such as copyleft). Learn more.

VISIT FULL GLOSSARY OF TERMS...

NEWS

Open Source Governance

Open source governance is

- The process of governing the use of open-source software in products [1]
- To meet the organization's needs (typically match the business model)
- Typically carried out by an open source program office (OSPO)

At a minimum, open source governance includes the

- Review
- Approval
- Management

of open source components and the compliance with its licenses

[1] Later also not just use of, but contribution to and leadership of open source projects

Overview
Packages
Develop
Curation
Governance
Deliver
Monitor

Edit Governance Rules

Template

On-Premises Software

Allowed

Copyleft (File-Level)
Copyleft (Module-Level)
Permissive
Public Domain

Must ask

Copyleft (LGPL)
Strong Copyleft

Prohibited

Commercial
Network Clause
Unstated

Cancel
Save Configuration

Support
Feedback
Collapse

Overview
Packages
Develop
Curation
Governance
Deliver
Monitor
Support
Feedback
Collapse

Open Source Governance

Edit Rules

Problems found
There are 28 packages that violate your governance rules.

Package	Classification	License Integrity	Effective License
chai 5.1.1	Prohibited	Medium	Apache-2.0 AND LicenseRef-scancode-generic-cla AND MIT
eslint-... 2.25.4	Prohibited	Medium	LicenseRef-scancode-unknown-license-reference AND MIT
espree 9.6.1	Prohibited	Medium	BSD-2-Clause AND BSD-3-Clause AND LGPL-2.0-or-later AND LGPL-2.1-or-later AND LicenseRef-scancode-ecma-no-patent AND LicenseRef-scancode-generic-cla AND MIT
typescript 5.6.3	Prohibited	Medium	Apache-2.0 AND BSD-3-Clause AND ISC AND LicenseRef-scancode-generic-cla AND LicenseRef-scancode-unknown-license-reference AND MIT AND ODbL-1.0
consola 3.2.3	Prohibited	Medium	LicenseRef-scancode-unknown-license-reference AND MIT
fast-lev... 2.0.6	Prohibited	Medium	LicenseRef-scancode-generic-cla AND MIT
eslint-v... 3.4.3	Prohibited	Medium	Apache-2.0 AND LicenseRef-scancode-unknown-license-reference
regen... 0.14.1	Prohibited	Medium	LicenseRef-scancode-generic-cla AND MIT
eslint-u... 3.0.0	Prohibited	Medium	LicenseRef-scancode-unknown-license-reference AND MIT
eslint-p... 3.2.0	Prohibited	Medium	LicenseRef-scancode-unknown-license-reference AND MIT

3. License Compliance

The MIT License

1. Copyright <YEAR> <COPYRIGHT HOLDER>

2. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

3. The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

4.

5. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The Open Source Legal Notices [1]

The two most common open source license obligations are

- The provision of all copyright notices to recipients (a.k.a. attribution)
- The provision of license texts to recipients

As an industry best practice, these text snippets are compiled into

- The (open source/third-party) legal notices and
- Provided as one document to recipients

Is it easy to create open source legal notices?

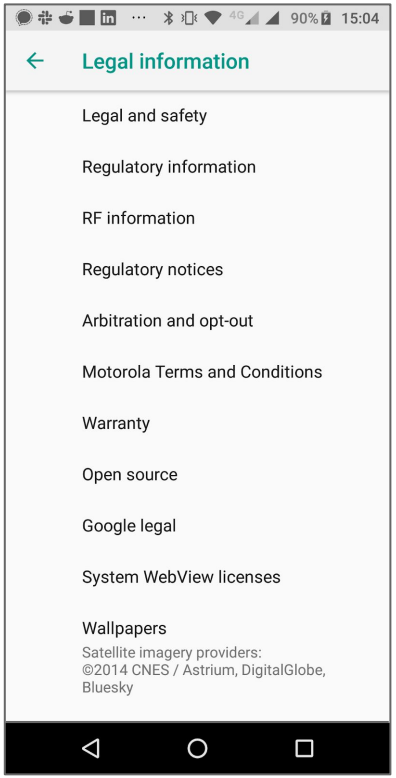
[1] Also known as third-party notices or just legal notices

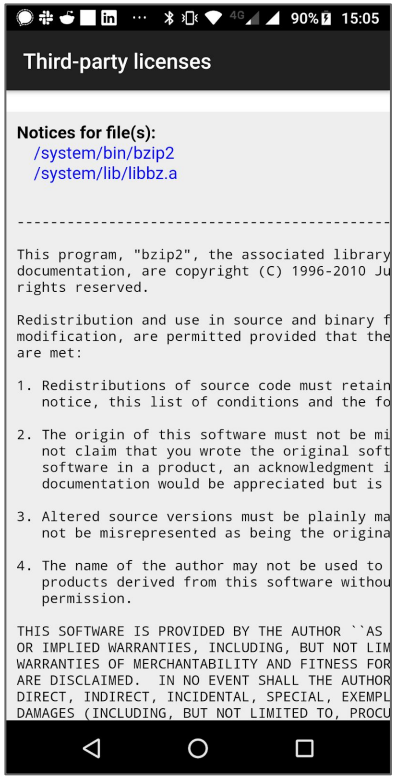
Index [1] [2]

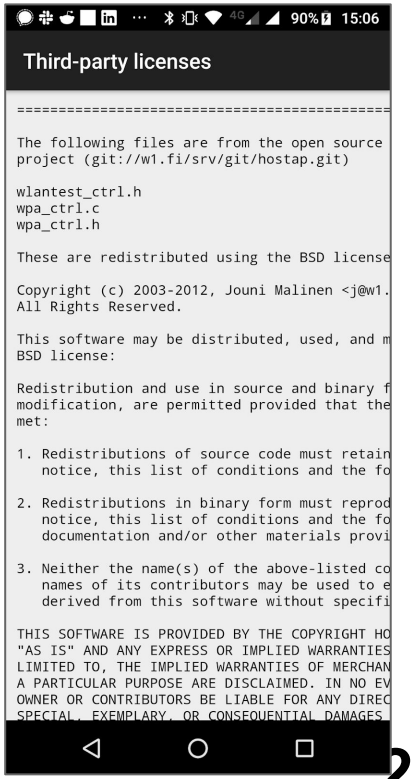
Contents

[1] See http://www.embedded.it/?q=content/daimler-mercedes-benz-and-open-source-software
[2] See https://moba.i.daimler.com/bai-cars/ba/foss/content/en/licence_agreement.html

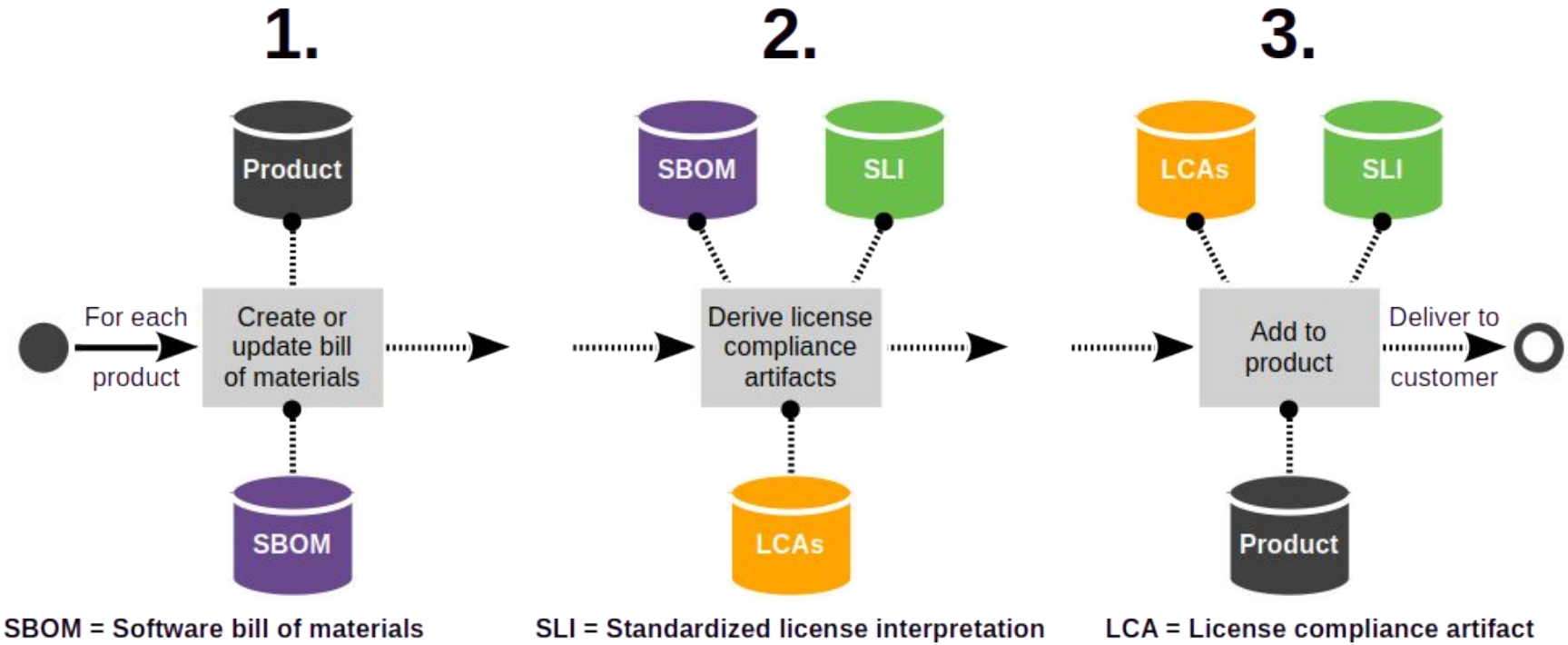
Android Open Source Legal Notices







The License-Compliant Distribution Workflow [1]



[1] See https://bayave.com/osc for open source compliance training

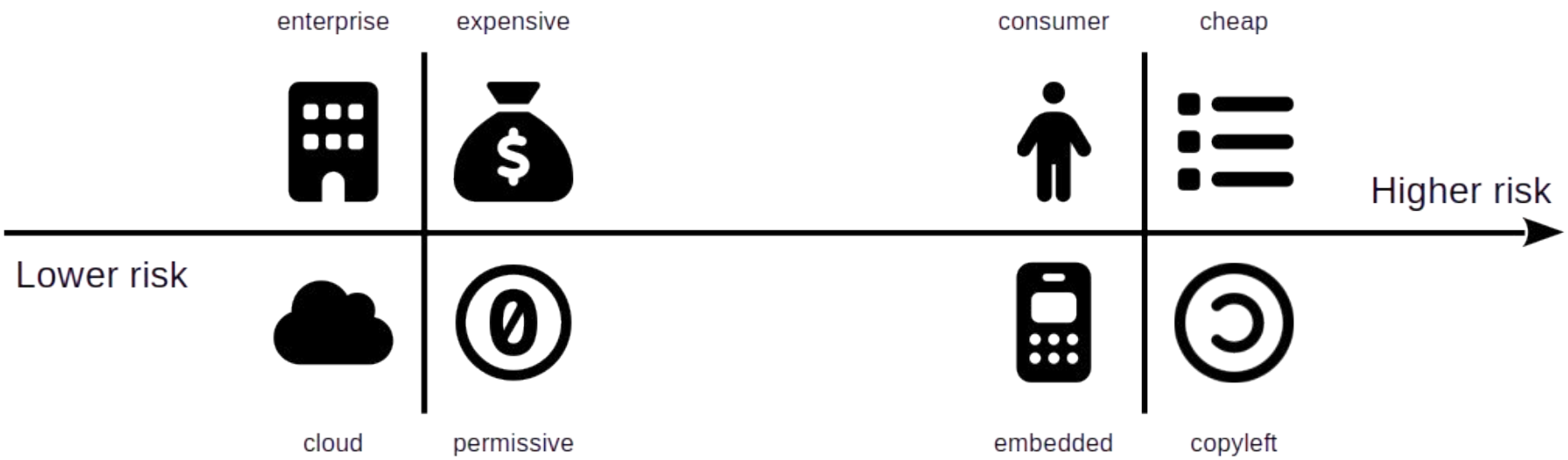
Copyright and Patent Trolls

	Enforcers	Trolls
Copyright	x	x
Patents		x

How a Copyright Troll Operates

1. Searches the web for license-incompliant software they hold rights in
2. Documents all violations (more than one)
3. Contacts company and requests fixing of one violation; asks for a modest fee
4. **Also asks for a cease-and-desist declaration with a penalty clause**
5. Upon receipt of declaration, returns with other violations and now hefty penalties

Discovery Risk Based on Product Properties



Legally Defective Products in Germany

Lawsuit against importer of automotive OEM for license incompliance in Germany [1]

Press Release: Lawsuit filed against MG importer for violation of open source terms

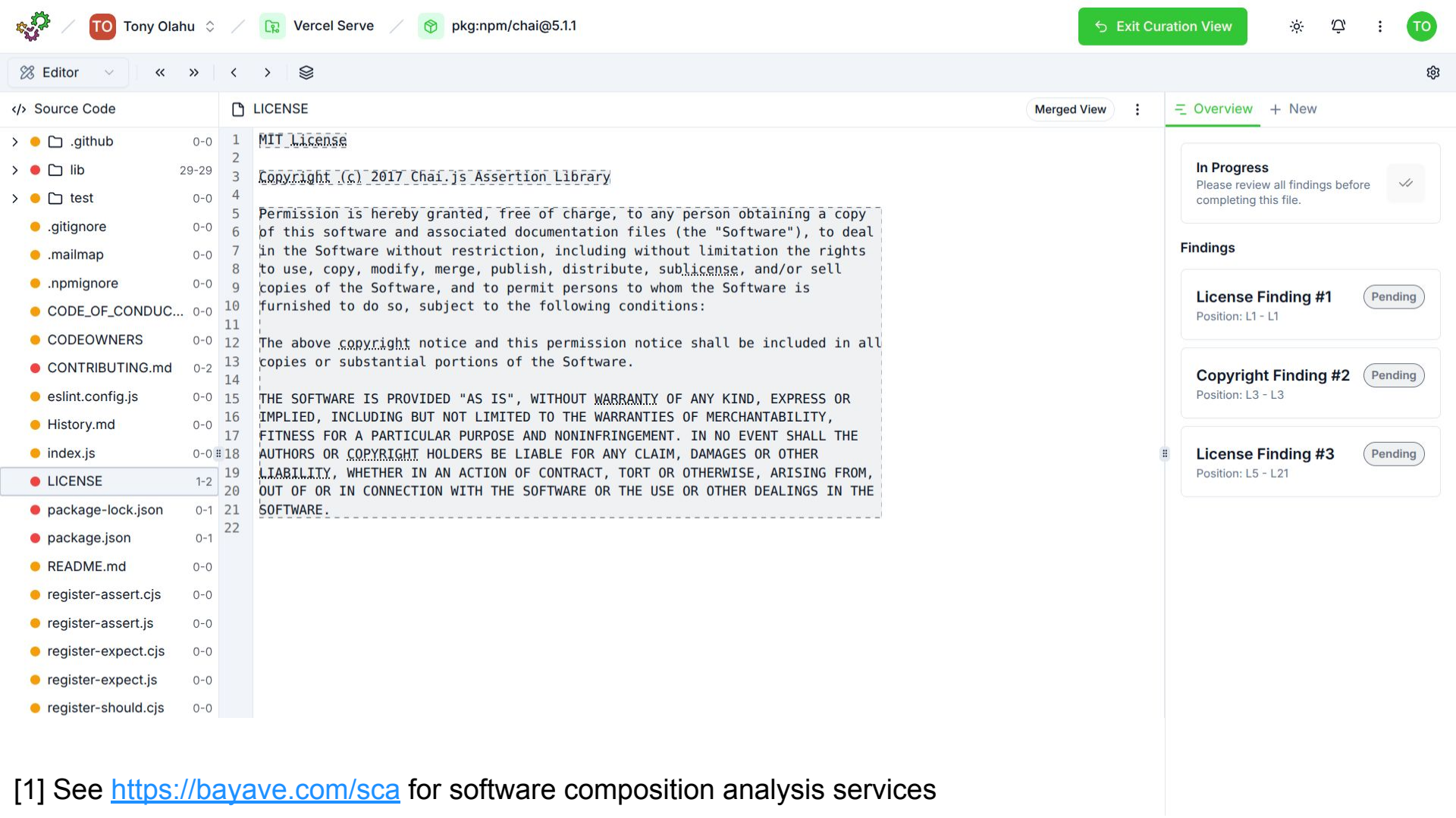
August 19, 2025

Würzburg, August 19, 2025 – The law firm JUN Legal GmbH, headed by IT attorney Chan-jo Jun, has filed a lawsuit with the Munich I Regional Court against SAIC Motor Deutschland GmbH, the German importer of the MG Motor brand. The core of the allegation is the lack of the necessary copyright usage rights to the software installed in the vehicle due to a violation of the open source conditions in an MG-4 Electric leased by the firm.

https://youtu.be/gtqIfv_pXrI

The lawsuit alleges that the Chinese automobile manufacturer SAIC relies extensively on open source components for the software used in its vehicles without fulfilling the associated legal obligations. Specifically, it alleges that neither the required copyright notices and license texts are included, nor is the associated source code made accessible, as is sometimes required by the licenses. Even after requests and reminders, no materials were provided. When the license terms are so blatantly violated, a vehicle owner must fear being sued by a rights holder or a subsequent buyer. The missing licenses constitute a legal defect.

[1] See https://jun.legal/en/2025/08/19/pressemitteilung-klage-gegen-mg-importeur-wegen-verletzung-von-open-source-bedingungen-eingereicht/



[1] See https://bayave.com/sca for software composition analysis services

Overview
Version Management
SBOM Management
Develop
Deliver
Release Management
SBOM Export
Third-Party Notices
Regulatory Documentation
Monitor
Support
Feedback
Collapse

License Compliance

HTML Download

Third Party Notices

Preview: Showing details for 5 of 9 components
Want to see everything? Download the complete Third Party Notices report.

Table of Contents

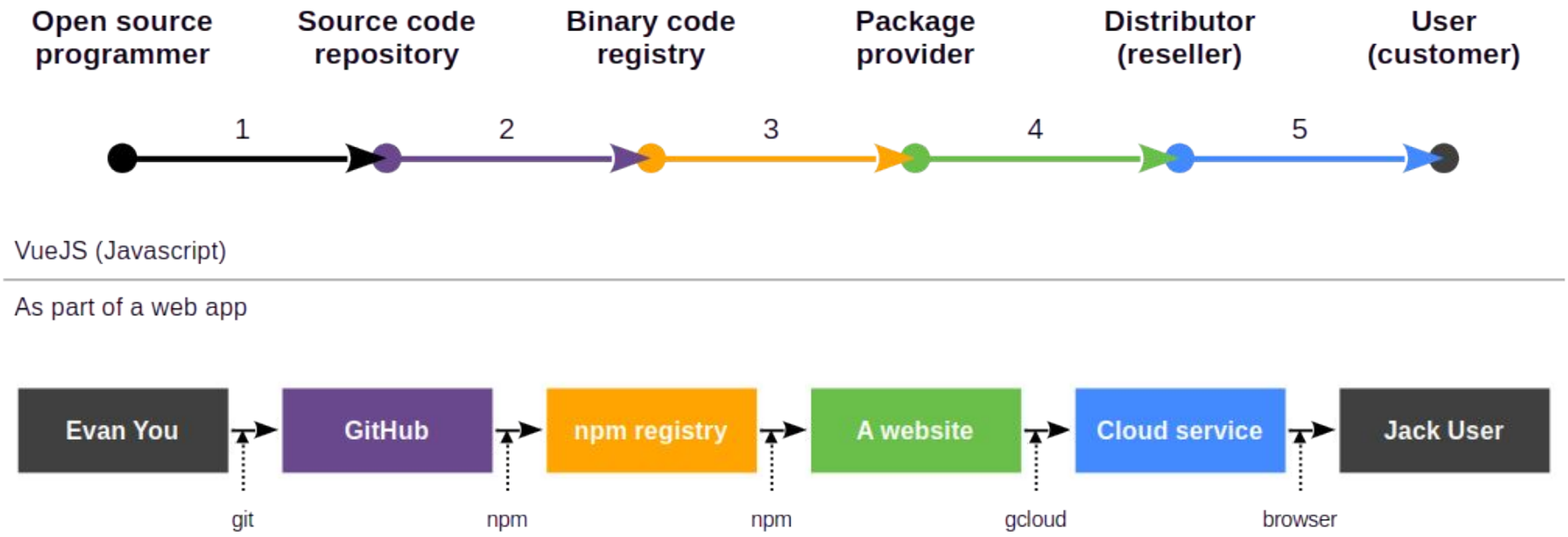
This software depends on external packages and source code. All applicable license and copyright information is listed below.

pkg:npm/asynckit@0.4.0

This package contains the following copyright statements:

4. Vulnerability Management

The Software Supply Chain [1]



[1] The software supply chain is different from the dependency graph!

Example Open Source Vulnerabilities [1] [2] [3]

Open-source software is an obvious target for supply chain attacks

[1] See https://heartbleed.com/
[2] See https://en.wikipedia.org/wiki/Log4Shell
[3] See https://news.apache.org/foundation/entry/apache-struts-statement-on-equifax

Overview
Packages
Develop
Deliver
SBOM Export
Third-Party Notices
Monitor
Vulnerabilities
Support
Feedback
Collapse

Vulnerabilities

Package	Risk	Security Advisory	Summary	Detected
lodash 4.17.21	HIGH 8.1	GHSA-r5fr-rjxr-66jc	lodash vulnerable to Code Injection via `_.template` im…	5/16/2026
glob 10.4.5	HIGH 7.5	GHSA-5j98-mcp5-4vw2	glob CLI: Command injection via -c/--cmd executes m…	5/16/2026
minimatch 3.1.2, minimatch 9.0.5	HIGH 7.5	GHSA-7r86-cg39-jmmj	minimatch has ReDoS: matchOne() combinatorial bac…	5/16/2026
cross-spawn 7.0.3	HIGH 7.5	GHSA-3xgq-45jj-v275	Regular Expression Denial of Service (ReDoS) in cros…	5/16/2026
flatted 3.3.1	HIGH 7.5	GHSA-25h7-pfq9-p65f	flatted vulnerable to unbounded recursion DoS in pars…	5/16/2026
fast-uri 3.1.0	HIGH 7.5	GHSA-q3j6-qgpj-74h6	fast-uri vulnerable to path traversal via percent-encod…	5/16/2026
minimatch 9.0.5, minimatch 3.1.2	HIGH 7.5	GHSA-23c5-xmqv-rm74	minimatch ReDoS: nested *() extglobs generate catast…	5/16/2026
picomatch 4.0.2, picomatch 2.3.1	HIGH 7.5	GHSA-c2c7-rcm5-vvqj	Picomatch has a ReDoS vulnerability via extglob quan…	5/16/2026
fast-uri 3.1.0	HIGH 7.5	GHSA-v39h-62p7-jpjc	fast-uri vulnerable to host confusion via percent-encod…	5/16/2026
vite 5.4.9	MEDIUM 6.5	GHSA-vg6x-rcgg-rjx6	Websites were able to send any requests to the devel…	5/16/2026

Rows per page: 10 11–20 of 36

5. Regulatory Compliance

Regulation of Software and Services

Almost all regulation is about improving cyber resilience and security [1]

- NIS2, DORA, … **and now the CRA**

Common requirements across different regulations and legislation

- Improving your own security standards and processes
- Managing the security of the software supply chain

[1] For the EU, see https://digital-strategy.ec.europa.eu/en/policies/cybersecurity-policies

The European Union's Cyber Resilience Act (CRA) [1] [2]

The CRA regulates products "with digital elements"; vendors must

- Ensure products are secure by design and default
- Have an effective vulnerability management process
- Provide ongoing support for a product's lifetime
- Prepare technical documentation demonstrating compliance
- Draw up an EU declaration of conformity (self or third parties)
- Report actively exploited vulnerabilities within 24 hours
- Provide users with clear information on security, support, and updates

Adopted in 2024 with the compliance (implementation) due by 2027

Incompliance comes with hefty fines (up to €15M or 2.5% of worldwide sales)

[1] See https://digital-strategy.ec.europa.eu/en/policies/cyber-resilience-act
[2] See https://eur-lex.europa.eu/eli/reg/2024/2847/oj

Applicability (Product Categories)

The CRA is broadly applicable with only a small set of exceptions [1]

Regulated products can be classified into four categories

1. Class I important products
    a. Product is critical to the cybersecurity of other products or
    b. Product disruption has significant adverse effects
2. Class II important products
    a. Infrastructure products like hypervisors, firewalls, etc.
3. Critical products
    a. Product is critically depended upon by essential entities (see NIS2)
    b. Product disruption has significant impact on critical supply chains
4. All other regulated products

[1] Exceptions are primarily products already covered by (even stronger) regulation

Obligations (Essential Requirements)

1. Product security requirements (“secure by default”)
2. Vulnerability management requirements

Product Security / Essential Requirements 1

1. **Has no known exploitable vulnerabilities (when put on the market)**
2. Has secure configuration by default (when put on the market)
3. Can be updated, automatically if applicable, with vulnerabilities fixes
4. Is protected from unauthorized access by appropriate control mechanisms
5. Protects the confidentiality of stored, transmitted, or otherwise processed data
6. Protects the integrity of stored, transmitted, or otherwise processed data
7. Processes only data that is adequate, relevant, and necessary for the task
8. Protects the available of essential and basic functions, even after an attack
9. Minimizes its potential negative impact on itself or connected devices
10. Has been designed, developed, and produced to limit attack surfaces
11. Has been designed, develope, and produced to reduce the impact of an attack
12. Monitors and records data about any security-relevant internal activity
13. Provides the option to securely, easily, and permanently remove all data

Vulnerability Management / Essential Requirements 2

1. **Identifies and documents components and related vulnerabilities**
2. Addresses and remediates vulnerabilities without delay, including updates
3. Applies effective and regular reviews and tests of product security
4. Shares and publicly discloses relevant information about fixed vulnerabilities
5. Has a policy on coordinated vulnerability disclosures (CVD)
6. Takes measures to facilitate information sharing about product vulnerabilities
7. Has mechanisms for securely distributing product updates
8. Ensures that security updates are distributed without delay

From the Perspective of an Open Source Developer

Open-source software does not fall under CRA regulation if and only if

- It is not part of a commercial activity by the distributor

Hence, a community open source project is not affected

- But an open source distributor firm like Suse or Red Hat is

Involved parties may have to be “open-source software stewards”

- Stewards have to fulfill a subset of vendors’ obligations

A vendor, which distributes the open-source software, remains responsible

Technical Product Documentation

Any product with digital elements must come with a technical documentation that

1. Has a general description of purpose, etc.
2. Describes the product's engineering and vulnerability management processes
3. Assesses relevant cybersecurity risks against the product
4. Provides all information used to determine the product support period
5. Provides a list of standards, specifications, and certification schemes applied
6. Reports the tests performed to verify conformity with the CRA
7. **Provides a copy of the EU declaration of conformity**
8. **Provides its software bill of materials**

This documentation must be kept for at least 10 years and the support period

Startups and SMEs can provide a simplified technical documentation

Continuous Product Security Management

Vendors must assess the security risks of their software supply chain

- They must then use this information to manage their processes to
  - Minimize security risks
  - Prevent incidents, and
  - Minimize their impact

This applies to the whole product lifecycle, not just to market introduction

This also applies to all third-party components incl. open-source software

Third-Party Software Due Diligence

For each and every component from a product's dependency graph

- Assess the risks intrinsic to the third-party component and
- Do not use the component if it does not pass muster

If a component is used in the product, the risk assessment must be

- Included in the technical documentation and
- Updated throughout the product support period

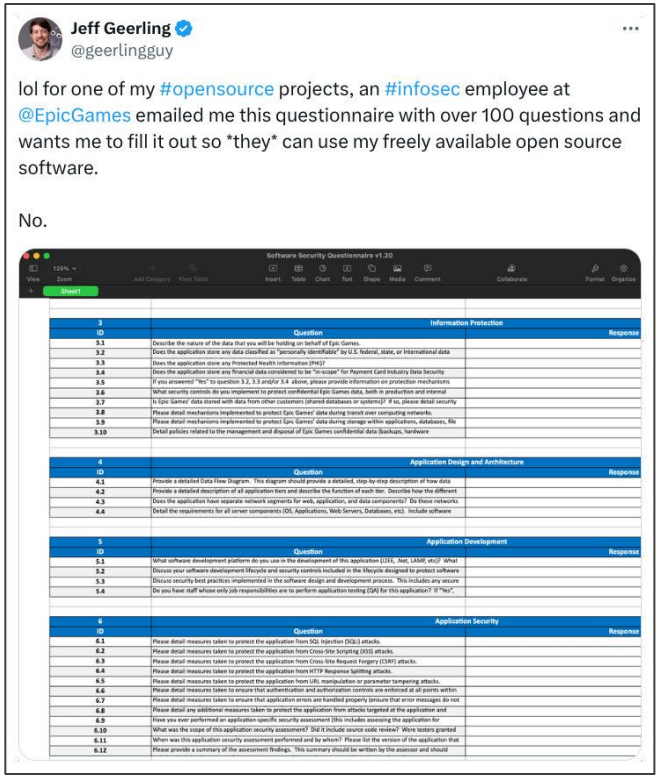
Risk Assessment of an Open Source Component

Open-source software is a third-party software

But open source community ≠ supplier

- Ask nicely, and you may be helped
- Ask unsmartly, and you may be ridiculed

Some communities provide the needed information



Options for Open Source Risk Assessment

Current options

1. Ask the open source project community for help
2. Perform the risk assessment yourself
3. Rely on an open source steward to provide assurances [1]
4. Pay a third party for appropriate assurances

Future options

5. Rely on European Union certification schemes

[1] The EU is working on an “attestation” scheme, yet unfinished

Practical Consequences for Managing Vulnerabilities

Ensure security for the whole product’s lifecycle

- Ongoing security assessments
- Incident reporting within 24 hours
- Expedient vulnerability fixes
- Beyond service existing contracts

Practical Consequences for Managing the Supply Chain

Manage all **incoming components**

- Create and maintain a correct SBOM
- Assess the risk of all component suppliers
- Manage new vulnerabilities expediently

Track all **outgoing product versions**

- Track SBOM of each delivered version
- Manage vulnerabilities accordingly
- Provide necessary updates / fixes

Summary

1. SBOM management
2. Open source governance
3. License compliance
4. Vulnerability management
5. Regulatory compliance

Thank you! Any questions?

dirk.riehle@fau.de – https://oss.cs.fau.de

dirk.riehle@bayave.com – https://bayave.com

dirk@riehle.org – https://dirkriehle.com – @dirkriehle

Advertisement: The AMOS Project

A student Scrum team

- Nine students, mostly Master computer science

Develops open-source software

- Every semester, 300h work each student

According to industry partner requirements

- Which incurs a fee

In a given semester

- October through February, April through July

A. Legal Notices

Legal Notices 1 / 3 (Copyright Notice, Other Notices)

Copyright

Notice to users

This document provides pragmatic technical guidelines on how to work with open-source software and its communities and has been created with appropriate diligence. The user is solely responsible for the use of this document. Bayave GmbH is not a law firm and the author(s) are not lawyers. This document does not provide legal advice. The recipient of this document should consult their own legal counsel when applying this information. Best practices change over time and the reader should check whether the described practices are still considered valid.

Legal Notices 2 / 3 (Disclaimer, Limitation)

Disclaimer of warranty

Unless required by applicable law or agreed to in writing, Bayave GmbH provides this document without warranties or conditions of any kind, either express or implied, including, without limitation, any warranties or conditions of fitness for a particular purpose. The recipient is solely responsible for determining the appropriateness of using this document and assumes any risks associated with it.

Limitation of warranty

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall Bayave GmbH be liable to the recipient of this document for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of using this document or out of the inability to use this document (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if Bayave GmbH has been advised of the possibility of such damages.

Legal Notices 3 / 3 (Confidentiality)

Confidentiality notice

Bayave GmbH considers all information contained in this document to represent trade secrets and confidential and proprietary business information. No part of this document may be reproduced, copied, distributed, or transmitted in any form or by any means, including photocopying, without the prior written permission of Bayave GmbH. All requests should be sent to attention: Geschäftsführer, Bayave GmbH, Juvenellstr. 3, 90408 Nürnberg, Germany.

Other company and product names mentioned in this document are the intellectual property of their respective companies and as such shall remain the sole property of those respective companies.