

Why and how we updated our license policy for use of open source

Introduction



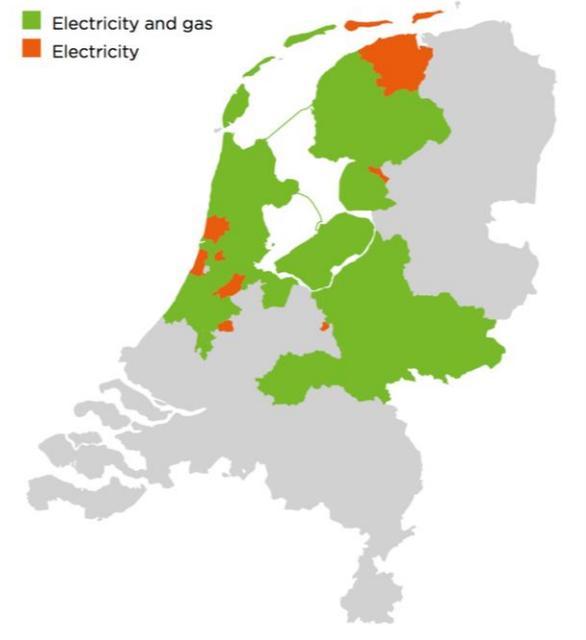
Nico Rikken
Open Source Advocate &
Solution Architect
@Alliander

Electricity grid length
92,000 km
91,000 km in 2019

Gas grid length
42,000 km
42,000 km in 2019

Number of customer connections
5.8 million
5.8 million in 2019

Number of employees
5,881 FTEs
5,703 FTEs in 2019



Why a License Policy?

Open Source Drives Innovation — But It Comes With Responsibilities

Innovation Through Open Source



Open Source is a catalyst for collaboration and technological innovation.

Legal & Compliance Challenges



Licensing, IP rights, and compliance risks must be managed.

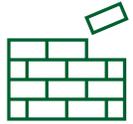
The Role of the OSPO



Open Source Program Office (OSPO) serves as a knowledge hub for open source.

The First Version of the Policy

How we got started



- **Starting Point** It began with the Linux Foundation's training *Fundamentals of Professional Open Source Management (LFC110)*. The insights and best practices gained from this training laid the groundwork for shaping Alliander's initial license policy.



- **Collaboration** The initial license policy for use of open source was created in close collaboration with the Legal department.



- **Content** The policy included an overview of commonly used license types—permissive, copyleft, non-commercial, and more—along with their status within Alliander.



- **Reviewed open source licenses list** An internal list of reviewed licenses was published to guide developers on what was permitted, restricted, or required further review.



- **Process for New Licenses** A clear process was agreed with Legal to evaluate and classify newly encountered licenses, ensuring consistent and compliant decision-making.



- **Impact** It quickly became the most visited page on the OSPO wiki, highlighting the demand for clear, accessible guidance on the compliant use of open source software.

Why a Revision Was Needed

Things became challenging

alliander



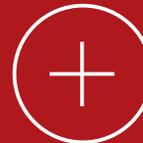
Challenges



Growing Complexity



Compliance-Driven Adoption



Developer Feedback



Limited Automation



Nuanced Cases

The Revision Process

Things to address

alllander



Things to address



Different situations



Clarify Typical Principles



Provide a Summary to Educate



Provide Examples of Typical Software and Typical Licenses



More Explicit on License Types



Guidelines for Exceptions

The New Structure of the Policy

How we achieved it

User-Friendly Design



- Highlight underlying principles
- Policy Matrix
- Collaborative Review
- Communication & Adoption

Future-Proofing



- Scalable structure
- Living document

The New Structure of the Policy

Policy in greater detail

- Describe relevance of software license
- Describe process
- Describe scope
- Opportunity for new licenses

Introduction

All code and software are subject to copyright and cannot be copied or used without permission. Software is typically accompanied by a license, outlining the conditions for its use. You should know, understand and comply with the license of the software you use. Always review the license of any open source component you intend to use to ensure compliance with its terms and the Alliander License Policy for Use of Open Source (this document).

This policy contains reviews of the commonly used open source licenses. It is developed in consultation with the Alliander Legal team and maintained by the Alliander OSPO. It is intended for evaluating licenses of open source software and hardware that Alliander uses.

The policy includes an overview of frequently used open source licenses, showing whether they are approved for use within Alliander and highlighting important considerations. It provides a framework for the use of open source licenses, based on the practical operational context of Alliander and its strategic goals.

Some open source licenses contain terms of reciprocity known as 'copyleft' that require publication of source code depending on the use case. Some common licenses are not open source licenses but are still included in this policy for clarity.

 Do you have a question? Is there a license missing? Do you want to use a license that's not allowed or requires permission? We are happy to help.

Contact us by email at OSPO@alliander.com with a brief description of the open source software and license in question, as well as your project. We'll help help you further.

The New Structure of the Policy

Guiding principles

- Review of typical situation at Alliander
- Consider change of use

Guiding principles for use of open source

This license policy takes into account Alliander's situation and needs when using open source software:

- At Alliander, most applications developed in-house with open source components are intended for internal use, not offered to external users.
- Alliander doesn't want to be required to publish its own code because of the use of third party software with copyleft licenses.
- Alliander should be able to enter into agreements with companies that have different license policies, without changing many software components.
- Alliander should be able to take an application designed for internal use and make it available to external users, without first having to change many software components to avoid triggering copyleft obligations.

The New Structure of the Policy

Disclaimers

- Not for publishing
- License is only one aspect

Publishing open source

This license policy covers the use of open source software, not publishing or distributing. The process of publishing open source software is explained at [Start developing open source](#)

Licenses are just one of the aspects of using open source software

While it may seem obvious, the license is not the only factor to consider when choosing an open source library. Evaluation criteria for selecting an open source library include governance, security practices, and community activity. These and other considerations can be found on the Alliander OSPO wiki page: [Policy for Evaluating and Selecting Open Source Software](#)

The New Structure of the Policy

Policy overview



Use-case →	Code snippet	Library	Tooling for development, build, test, deploy	Standalone software application
↓ License type	A few copied lines of code	Examples: component, framework, JavaScript component, protocol library, database driver	Examples: compiler, linter, testing	Examples: database, file storage, webserver, other software consumed as a service (SaaS)
Permissive license <i>Few obligations other than a copyright and license notice</i> Examples: MIT, BSD-3-clause, Apache-2.0, PostgreSQL, CC0, ISC, BlueOak-1.0.0, Unlicense	Allowed, more obligations apply when distributing Example: A few lines copied from a permissively licensed library	Allowed, fewer obligations Examples: AngularJS, FastAPI, OpenSSH, GDAL	Allowed, fewer obligations Examples: Maven	Allowed, fewer obligations Examples: PostgreSQL, Kafka, Kubernetes, Nginx
Weak copyleft <i>Obligations when changing the source code of the module</i> Examples: LGPL-2.0, MPL-2.0, EPL-2.0	Not allowed	Allowed, more obligations apply when distributing Examples: Power-Grid-Model, exifreader, pgJDBC, Graphviz	Allowed, more obligations apply when distributing Examples: Sonarqube-scan-action	Allowed, more obligations apply when distributing
Strong copyleft <i>Obligations when changing the source code of the entire application</i> Examples: GPL-2.0, GPL-3.0, CC-BY-SA	Not allowed Examples: Stack Overflow, Stack Exchange (CC-BY-SA license)	Not allowed, requires permission from OSPO Examples: CKEditor, Rpy2	Allowed, more obligations apply when distributing Examples: Pylint	Allowed, more obligations apply when distributing Examples: Linux kernel
Network copyleft <i>Obligations when changing the source code of the entire application including networked services</i> Examples: AGPL-3.0	Not allowed	Not allowed under open source license, contact your contract manager Examples: Pyreadr, Ultralytics YOLO, iText	Not allowed, requires permission from OSPO Examples: RStudio, Renovate	Not allowed, requires permission from OSPO Examples: Grafana , MinIO, OpenRemote, Nextcloud
Source available <i>Various limitations (not open source)</i> Examples: SSPL-1.0, BUSL-1.1, SAP developer license, Unity Companion License 1.3, Esri Master License Agreement	Not allowed	Not allowed under open source license, contact your contract manager Examples: Unity 3D libraries, SAP libraries, ESRI libraries, NVIDIA CuBLAS	Not allowed under open source license, contact your contract manager Examples: Terraform	Not allowed under open source license, contact your contract manager Examples: MongoDB , Hashicorp Vault , Redis
Non-commercial <i>Not for commercial use (not open source)</i> Examples: CC-BY-NC	Not allowed	Not allowed	Not allowed	Not allowed

The New Structure of the Policy

License types

License types	
Permissive license	Examples: MIT, BSD-3-clause, Apache-2.0, PostgreSQL, CC0, ISC, BlueOak-1.0.0, Unlicense
Weak copyleft	Examples: LGPL-2.0, MPL-2.0, EPL-2.0
Strong copyleft	Examples: GPL-2.0, GPL-3.0, CC-BY-SA
Network copyleft	Examples: AGPL-3.0
Source available	Examples: SSPL-1.0, BUSL-1.1, SAP developer license, Unity Companion License 1.3, Esri Master License Agreement
Non-commercial	Examples: CC-BY-NC

Addendum

License categories in ScanCode LicenseDB

- **CLA** Agreement defining contribution rules and how contributions are licensed.
- **Commercial** Proprietary software licensed directly from a vendor; license details require verification.
- **Copyleft** Open source requiring redistributed or modified versions to use the same license terms.
- **Copyleft Limited** Requires sharing source and changes, but obligations for linked proprietary code are limited.
- **Free Restricted** Permissive-style license with usage or redistribution restrictions (e.g., no commercial use).
- **Patent License** License granting rights to use specific patents; may accompany a software license.
- **Permissive** Non-copyleft open source requiring attribution and minimal obligations.
- **Proprietary Free** Free to use but under proprietary terms that must be followed.
- **Public Domain** Software with no explicit obligations except keeping the notice per policy.
- **Source-available** Source can be viewed (and sometimes modified) but not fully open-source.
- **Unstated License** Software with copyright but no license; must contact owner to clarify usage rights.

<https://scancode-licensedb.aboutcode.org/help.html#license-categories>

The New Structure of the Policy

Use cases

alliander

Use-cases

Code snippet

A few copied lines of code

Library

Examples: component, framework, JavaScript component, protocol library, database driver

Tooling for development, build, test, deploy

Examples: compiler, linter, testing

Standalone software application

Examples: database, file storage, webserver, other software consumed as a service (SaaS)

Addendum

Use-case categories

- **Use cases** - The way you use open source licensed software determines which obligations you trigger.
 - **library** - you use the licensed component as a library (creating a combined work, a derivative work)
 - **compiler** - you use the licensed component as a compiler taking input and producing output (e.g. GCC, Autoconf)
 - **snippet** - you use a part of the licensed component's source code from another project, book, web page
 - **test** - you use the licensed component for testing your software (e.g. linter, cyclomatic analysis)
- **Provisioning** - The way you provide the software to your user.
 - **source-code-distribution** - you distribute the licensed component, in source code form
 - **binary-distribution** - you distribute the licensed component, in binary (non source code) form
 - **local-use** - you used the licensed component locally (e.g. and editor you use to write software)
 - **provide-service** - you provide a service over a REST api. No source is distributed.
 - **provide-webui** - you provide a webui which is distributed to the user's browser.
- **Modification** - whether or not you have modified the licensed component.
 - **modified** - you have made modifications to the licensed component
 - **Unmodified** - you have not made any modifications to the licensed component

The New Structure of the Policy

Policy

Green	Preferred license types (permissive licenses).
Yellow	Allowed, but less preferred licenses. These require more in-depth knowledge of the particular license when applying.
Orange	Licenses that are by default not allowed, but for which an exception is occasionally granted.
Red	Licenses that are not allowed, or which cannot be used under the open source license. A contract must be procured.

The New Structure of the Policy

Examples

Not allowed, requires permission from OSPO

Examples: CKEditor, Rpy2

Allowed, more obligations apply when distributing

Examples: Pylint

Not allowed under open source license, contact your contract manager

Examples: Pyreadr, Ultralytics
YOLO, iText

Not allowed, requires permission from OSPO

Examples: RStudio, Renovate

The New Structure of the Policy

Detailed license list



Overview of reviewed open source licenses

Entire list of reviewed licenses.

Tip: For projects on GitHub an Action is available to check your dependencies. Note that this is an approximation of the policy, so not all cases might be detected: <https://github.com/Alliander/check-project-dependencies/>

Use-case →	Notes and details	Licence type	Code snippet	Library	Tooling for development, build, test, deploy	Standalone software application
↓ License			A few copied lines of code	Examples: component, framework, JavaScript component, protocol library, database driver	Examples: compiler, linter, testing	Examples: database, file storage, webserver, other software consumed as a service

Permissive licenses

Few obligations other than a copyright and license notice

Academic Free License (AFL) AFL-1.1 AFL-1.2 AFL-2.0 AFL-2.1 AFL-3.0	<input type="button" value="Quick Summary: Academic Free License 3.0 (AFL)"/>	Permissive license <i>Few obligations other than a copyright and license notice</i>	Allowed, more obligations apply when distributing	Allowed, fewer obligations	Allowed, fewer obligations	Allowed, fewer obligations
Apache License 2.0 Apache-2.0	<input type="button" value="Quick Summary: Apache License 2.0 (Apache-2.0)"/>	Permissive license <i>Few obligations other than a copyright and license notice</i>	Allowed, more obligations apply when distributing	Allowed, fewer obligations	Allowed, fewer obligations	Allowed, fewer obligations

Lessons Learned

Insights from our journey toward policy for use of open source



Collaboration is key

Legal, developers, and software asset managers must work together.

Policy is never finished

Keep iterating based on real-world experience.

Clarity pays off

A clear policy is used more often—and followed more closely.



Practical Takeaways



Start Simple, Scale Smart

Start with a simple policy, but be ready to scale.



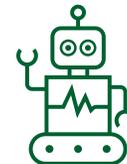
Involve Users Early & Often

Involve users early and often



Document Exceptions Explicitly

Document exceptions and edge cases explicitly



Automate Where Possible

e.g., via code scanners

Addendum

Coding LLMs

- Extensive code-snippets of questionable origin
- **Theoretical risk**
 - Copyright and license violations during training
 - A similarity check before producing result isn't a cleanroom implementation
 - Should we include notices of all code the LLM is trained on?
- **Practical risk**
 - Will original authors even find out and be able to prove a license violation?
 - What insight do organizations expect if you get into an agreement?
- **Policy ideas**
 - Use LLMs that are trained in a legal way
 - Keep track of LLM-generated code and the models and services used to produce it



Check out our website @ alliander.com/en/open-source/



Visit our Github community @ [Github.com/alliander-opensource](https://github.com/alliander-opensource)



Learn more about [LFENERGY](https://www.lfenergy.org/) @ lfenergy.org/



Mail us @ ospo@Alliander.com