

Migrating from GPLv3 to Pimcore Open Core License (POCL)

Whitepaper – April 2025

Important note: This licensing change only affects users of the free and open-source Pimcore Community Edition previously licensed under GPLv3. **If you're already using a Pimcore Professional Edition or Enterprise/PaaS Edition** under a commercial agreement, **nothing changes for you.** Your existing contracts, rights, and usage terms remain fully intact – POCL has no impact on your deployment or support.

Executive Summary

With the release of Pimcore Platform Version 2025.1, Pimcore has transitioned its Community Edition from the GNU General Public License v3 (GPLv3) to the new Pimcore Open Core License (POCL). Version 2024.4 marks the final release under GPLv3.

This shift is more than a licensing update – it's a strategic evolution. POCL was designed to meet the realities of today's regulatory and operational landscape, where compliance, scalability, and commercial flexibility are critical. While GPLv3 played a key role in Pimcore's growth, its strict copyleft provisions increasingly introduced friction for enterprises, especially those navigating risk assessments, procurement policies, and legal audits.

POCL replaces this friction with clarity. It enables companies to build confidently on Pimcore – without the legal overhead or uncertainty of GPLv3. The license maintains access to the source code, supports internal and commercial use, and aligns seamlessly with frameworks like NIS2 and modern software governance practices.

This white paper is a practical guide for legal teams, developers, product managers, and decision-makers. It outlines:

- The key differences between GPLv3 and POCL – and why they matter
- How the transition affects existing GPLv3-based extensions and projects
- A clear, low-friction path to becoming POCL-compliant
- Licensing strategies for supporting both legacy and future Pimcore environments



Own the Digital World

Whether you're an enterprise scaling fast, or a small team looking to stay compliant and flexible, this guide will help you navigate the transition smoothly – and strategically.

Contents

Executive Summary	1
1 Understanding the Licensing Shift.....	3
1.1 GPLv3 in Enterprise Environments: Benefits and Barriers	3
1.2 POCL: A License Designed for the Modern Enterprise	3
1.3 Side-by-Side Comparison: GPLv3 vs. POCL	4
Summary	4
2 Implications for Existing GPLv3 Extensions	4
2.1 Why GPLv3 and POCL Cannot Coexist	4
2.2 Redistribution and Derivative Controls under POCL	5
2.3 Compliance Action Plan for Legacy GPLv3 Code	6
2.4 Summary: Clear Boundaries = Confident Development	7
3 Migration Path: From GPLv3 to POCL Compliance.....	7
3.1 When Migration Is Easy – and Why It Usually Is	7
3.2 The 5-Step POCL Migration Plan	8
3.3 What If You Can't Relicense?	8
3.4 Summary: Simple, Documented, and Done	9
4 What This Means for Your Extensions	9
4.1 You Own It? You Control It.....	9
4.2 Supporting Old and New: Maintain Two Versions.....	10
4.3 What You Can Do Under POCL	10
4.4 What You Cannot Do Under POCL	10
4.5 Clarifying the Boundary: GPLv3 vs. POCL Extensions.....	11
4.6 Summary: Full Control, Clear Choices	11
5 Common Questions, Honest Answers.....	11
Final Summary: Clarity, Control, and Confidence.....	13

1 Understanding the Licensing Shift

1.1 GPLv3 in Enterprise Environments: Benefits and Barriers

The GNU General Public License v3 (GPLv3) was created to protect software freedom through strong copyleft principles. It has been a cornerstone of open-source innovation, ensuring that derivative works remain open and accessible to all.

However, in enterprise settings, GPLv3 increasingly introduces friction. Its legal implications can slow down projects, introduce ambiguity, and raise red flags during legal and compliance reviews. Key concerns include:

- **Copyleft Obligations**
Any software that incorporates or builds upon GPLv3-licensed components must itself be released under GPLv3. This can force organizations to expose proprietary code, even unintentionally.
- **Audit and Approval Complexity**
Legal and procurement teams often flag GPLv3 components as high risk, due to potential license “contamination” and unclear boundaries between open and proprietary code.
- **Ongoing Compliance Overhead**
Organizations must maintain detailed audit trails for GPLv3 usage, adding delays and costs to vendor onboarding, release planning, and compliance reporting.

1.2 POCL: A License Designed for the Modern Enterprise

In response to these challenges, Pimcore introduced the **Pimcore Open Core License (POCL)** – a license designed to enable open development without exposing organizations to the pitfalls of copyleft obligations.

POCL provides the same level of access to the Pimcore source code, but under a framework better suited to enterprise use, productization, and commercial scalability.

Key features of POCL include:

- **Defined Redistribution Boundaries**
Prevents redistribution of the Pimcore core in hosted or competing offerings – preserving commercial fairness.
- **Revenue-Based Licensing Triggers**
Enables free use for startups and smaller organizations (under €5M annual



Own the Digital World

global revenue), with a seamless path to commercial licensing as you grow.

- **Derivative Use Compliance**

If you distribute software based on POCL, it must be clearly marked and also licensed under POCL – ensuring that obligations are transparent and traceable.

1.3 Side-by-Side Comparison: GPLv3 vs. POCL

Attribute	GPLv3	POCL
License Model	Strong Copyleft	Controlled Open Core
Redistribution Rights	Unrestricted, viral	Restricted; redistribution requires POCL
Use in Proprietary Systems	Risk of contamination	Permitted, within scope
SaaS/Managed Services Use	Allowed	Explicitly restricted
Revenue-Based Licensing	No	Yes – free below €5M
Legal Audit Provisions	None inherent	Pimcore retains audit rights

Summary

Where GPLv3 prioritizes openness at all costs, POCL balances openness with control – enabling innovation without compromising legal safety. For enterprises, this shift means less ambiguity, fewer barriers, and more confidence in scaling Pimcore for real-world use cases.

2 Implications for Existing GPLv3 Extensions

2.1 Why GPLv3 and POCL Cannot Coexist

At a legal level, GPLv3 is not compatible with POCL:

- **GPLv3** requires that any software derived from or linked with GPL code must also be distributed under GPL. This “viral” copyleft ensures continued



Own the Digital World

openness—but can unintentionally impose licensing obligations on unrelated proprietary code.

- **POCL** imposes a clear prohibition on combining its code with GPLv3-licensed components. Section 3.4 of the POCL expressly forbids any use of GPLv3 code within a POCL-based system, even internally.

What this means in practice:

- You cannot run, test, deploy, or distribute GPLv3 extensions in a Pimcore 2025.1+ environment governed by POCL.
- You cannot relicense POCL-based code under GPLv3 or any other license.
- Attempting to combine GPLv3 and POCL code could violate both licenses simultaneously, exposing your organization to compliance risk.

Key takeaway: There must be a clear boundary between Pimcore versions using GPLv3 (≤ 2024.4) and those using POCL (≥ 2025.1).

2.2 Redistribution and Derivative Controls under POCL

POCL introduces a structured, business-safe approach to software distribution. If you're building or distributing code based on POCL-licensed Pimcore, you must follow these requirements:

Derivative Licensing Rules:

- Any code that qualifies as a derivative of the POCL core must be released under POCL – you may not relicense it under GPL or other open-source licenses.
- The derivative must include clear labels indicating it's governed by POCL.
- You must provide a copy of the POCL license alongside your distribution.

Internal Deployment Restrictions:

Even internal use of GPLv3-based extensions within POCL systems is prohibited. The concern is not redistribution alone, but the legal contamination that can occur when incompatible licensing terms apply to the same codebase or runtime environment.

Why This Matters for Enterprises:

These controls give organizations:

- Confidence that their proprietary extensions won't trigger open-source obligations



Own the Digital World

- A legally defensible position when submitting software for security or regulatory audits
- A consistent compliance standard that can be documented and enforced internally

2.3 Compliance Action Plan for Legacy GPLv3 Code

If your team has created or inherited GPLv3-based Pimcore extensions, here's how to handle the transition in a way that's clean, documented, and compliant:

Step 1: Conduct a Code Audit

- Identify all Pimcore modules, extensions, and integrations.
- Note which components are GPLv3-licensed.
- Flag any third-party libraries or plugins included in your stack.

Step 2: Verify Intellectual Property (IP) Ownership

- Confirm that your organization authored the code or received it under a contributor license agreement (CLA) that permits relicensing.
- If any components were developed by outside contributors, determine whether relicensing is legally viable.

Step 3: Isolate or Reclassify Legacy Code

- Maintain legacy GPLv3 extensions exclusively in Pimcore environments up to version 2024.4.
- Tag these versions clearly and enforce boundaries to avoid accidental deployment on POCL-based systems.

Step 4: Refactor When Needed

- If an extension cannot be relicensed but is essential for future projects, consider rebuilding it using internal code – under POCL terms.
- Alternatively, explore available POCL-compatible extensions in the Pimcore ecosystem or marketplace.

Step 5: Document Your Decisions

- Create an internal record that details which code is GPLv3 vs. POCL, how relicensing decisions were made, and what legacy support policies are in place.
- This documentation is critical for future audits, vendor reviews, or legal inquiries.

2.4 Summary: Clear Boundaries = Confident Development

POCL doesn't break your past work – but it does demand a clear legal boundary around it. **Most GPLv3 extensions can be relicensed if you own the code.** For everything else, smart segmentation and clear documentation will ensure compliance.

This isn't about rewriting everything. It's about knowing what you've built, understanding your rights, and making informed decisions moving forward.

3 Migration Path: From GPLv3 to POCL Compliance

Moving from GPLv3 to the Pimcore Open Core License (POCL) may sound complex – but for most developers and teams, it's remarkably straightforward. If your organization **owns the code**, and that code doesn't embed GPLv3-licensed third-party components, **you can relicense it**. That's the core principle driving this transition.

This section offers a clear roadmap for navigating the switch: one that minimizes effort and maximizes compliance confidence.

3.1 When Migration Is Easy – and Why It Usually Is

Most Pimcore extensions and modules are developed in-house or under contract. That means:

- You own the code
- You control how it's licensed
- You don't need to rewrite anything

Common scenarios where migration is simple:

- A digital agency built custom integrations for a client project
- A product company developed internal modules for its CMS
- A freelance developer created and maintained a GPLv3 plugin and has retained all rights

Key principle: If you wrote it, you can relicense it.

3.2 The 5-Step POCL Migration Plan

Step 1: Inventory and Categorize Your Code

- List all extensions, modules, integrations, and supporting libraries.
- Identify the current license of each (GPLv3, MIT, custom, proprietary, etc.).
- Highlight any third-party code not written by your team.

Step 2: Confirm You Own It

- Check contributor agreements and employment contracts to verify that your organization holds full copyright.
- Exclude any GPLv3 third-party code, unless you've received permission to relicense.

Step 3: Choose a Licensing Path

- POCL-Only: For new deployments based on Pimcore 2025.1+
- Version-specific license (GPLv3 & POCL): To support both legacy and modern systems
- Isolate/Refactor: When code includes GPL components you cannot relicense

Step 4: Update Your Licensing

- Add a LICENSE.md file to your project
- Replace GPLv3 headers with POCL language in source files
- If you choose version-specific licensing, clearly state the version boundary (e.g., "Use GPLv3 for Pimcore ≤ 2024.4, POCL for ≥ 2025.1")

Step 5: Communicate and Document

- Add comments or banners in your codebase and documentation
- Log the licensing decision internally – including when, why, and by whom the transition was made
- Notify clients, partners, or downstream users if your plugin is publicly shared

3.3 What If You Can't Relicense?

If you identify code that:

- Includes GPLv3 code from third-party sources
- Has contributions from outside developers without clear agreements

Then you have three options:

- **Refactor** – Rebuild the component in-house under POCL
- **Isolate** – Restrict that code to GPLv3-only Pimcore environments
- **Negotiate** – Contact contributors and request relicensing rights

These situations are relatively rare – and typically solvable with planning.

3.4 Summary: Simple, Documented, and Done

For most teams, migrating to POCL is a low-friction, high-impact step. It doesn't require rewriting your work – just organizing, clarifying, and relabeling it where needed.

And it brings real benefits:

- Reduced compliance uncertainty
- Fewer audit bottlenecks
- A modern license that grows with your business

Start with what you own. Separate what you can't relicense. And document the rest.
That's the whole path.

4 What This Means for Your Extensions

The transition to the Pimcore Open Core License (POCL) may raise questions about existing extensions and custom developments. The good news is: if your organization built its extensions in-house – without relying on third-party GPLv3 code – you are fully in control. You can continue developing and distributing those extensions under POCL, with minimal effort.

This section explains exactly what you can (and cannot) do with your extensions going forward.

4.1 You Own It? You Control It.

Under copyright law, the original author of a software component holds full licensing authority – unless that right was transferred or shared. If your team wrote the code, and no incompatible third-party GPLv3 code was included, you are free to relicense it under POCL.

This applies to:

- In-house development teams building solutions on Pimcore
- Agencies creating client-specific plugins
- Freelancers maintaining GPLv3-based plugins under their own IP

If you own the code, you can relicense it under POCL. No rewrite needed.

4.2 Supporting Old and New: Maintain Two Versions

If your extension needs to work with both older and newer versions of Pimcore, the cleanest approach is to **maintain two separate branches**

- A **GPLv3 branch** for compatibility with Pimcore CE \leq 2024.4
- A **POCL-licensed branch** for Pimcore CE \geq 2025.1

These should be managed as distinct releases to avoid violating either license. POCL explicitly prohibits sublicensing under GPLv3, so the same codebase cannot legally carry both licenses.

Important: This is not "dual licensing" in the traditional sense. It's version-specific licensing – and it keeps you fully compliant with both POCL and GPLv3 terms.

If you're publishing the extension, clearly label which branch applies to which Pimcore version, and include separate LICENSE.md files in each branch.

4.3 What You Can Do Under POCL

POCL is designed to offer broad flexibility, especially for internal teams, integrators, and extension developers.

You can:

- Build proprietary plugins or modules
- Keep extensions private or share them selectively
- Offer commercial versions or premium support
- Use POCL-based code without open-sourcing your extensions
- Modify Pimcore core for internal use (no sharing required)

There is no obligation to publish your changes or distribute your extensions – POCL lets you control how your work is used.

4.4 What You Cannot Do Under POCL

To preserve the integrity and sustainability of the ecosystem, POCL introduces a few key restrictions:

- You may not **relicense POCL core or derivatives** under GPLv3 or any other license
- You may not **combine POCL-based systems with GPLv3 components**
- You may not **offer Pimcore core as a hosted SaaS solution** without a commercial license



Own the Digital World

- You may not **fork Pimcore** into a competing public or commercial offering

These constraints ensure that while the code remains accessible and flexible, Pimcore's business model – and your own proprietary developments – remain protected.

4.5 Clarifying the Boundary: GPLv3 vs. POCL Extensions

If your extension includes even a small part of GPLv3-licensed third-party code, **you cannot use or distribute it in a POCL environment.**

For clarity:

- You **can** license your own GPLv3 code differently for specific versions (if you own it)
- You **cannot** relicense third-party GPL code as POCL
- You **must** isolate GPLv3 extensions to legacy Pimcore deployments (up to 2024.4)

POCL Section 3.4 explicitly prohibits mixing GPLv3 and POCL components.

4.6 Summary: Full Control, Clear Choices

POCL gives you the freedom to:

- License your own work how you choose
- Keep your work private or share it on your terms
- Offer paid, commercial extensions without legal risk
- Avoid the viral effects of GPLv3 in commercial environments

As long as you've verified ownership, POCL lets you scale without legal complexity. It's the next step in turning your Pimcore development work into long-term, compliance-aligned assets.

5 Common Questions, Honest Answers

When switching to a new license model, questions come up. That's natural – and expected. This section is designed to give you practical, transparent answers to the things teams and legal departments most often ask about POCL.

Can I still use my old GPLv3-based Pimcore code?

Yes – if you're using it with Pimcore versions released under GPLv3 (2024.4 or earlier). That code remains valid and legal under the terms it was created with. You're just not allowed to use that same GPLv3 code in newer POCL-based Pimcore versions.



Own the Digital World

Do I need to rewrite my extensions?

Probably not. If you wrote the code yourself – and didn't bring in third-party GPLv3 libraries – then you can simply relicense your code under POCL. No rewrite required.

Can I offer my plugin under both GPLv3 and POCL?

Yes – and that's often the best approach. Just make sure it's clearly communicated which version of the plugin applies to which version of Pimcore. For example:

- GPLv3 for Pimcore ≤ 2024.4
- POCL for Pimcore ≥ 2025.1

This is a common practice of version-specific licensing, and it's perfectly compliant.

Is POCL considered open source?

No, it's source-available, not open source. You can view, modify, and use the code – but you can't redistribute or relicense it freely like you can with GPLv3. It's designed to offer open development freedom without the complications that come from unrestricted redistribution.

What happens if our revenue grows beyond €5 million?

Congratulations – that's a good problem to have. Once your company crosses the revenue threshold, you'll need a commercial POCL license for continued use in production. Pimcore offers flexible license options for this step, and your team won't need to refactor or remove code to comply.

Can I still keep my changes private?

Absolutely. You're not required to contribute modifications back, and there's no obligation to open-source internal developments unless you want to share them.

What if we already deployed a GPLv3 plugin in a POCL environment?

This is not compliant and should be corrected. POCL section 3.4 prohibits combining POCL and GPLv3 code. If you're unsure whether your current setup includes any such overlaps, now's the time to run a licensing audit and resolve it.

Is POCL hard to explain to stakeholders?

Not at all. It's actually easier than GPLv3 in most business conversations. You can summarize it as: "You get the source code. You can use and modify it. But you can't redistribute Pimcore core or offer it as a service unless licensed."

What about contractors or external teams using our code?

That's fine – as long as they're working under your license. Just make sure subcontractors, freelancers, or clients are aware of POCL's redistribution restrictions if they'll be modifying or deploying Pimcore-based systems on your behalf.

Is there a grace period for moving to POCL?

Older versions of Pimcore remain under GPLv3, and you can still maintain those environments. But anything built for 2025.1 or newer must follow POCL terms. If you're actively developing on the new stack, now is the time to align.



Own the Digital World

Summary: Clear Questions Deserve Clear Answers

The most powerful thing about POCL is its clarity. It's not open-ended. It's not vague. It tells you exactly what you can do – and what you can't – so your team can move forward with confidence.

Final Summary: Clarity, Control, and Confidence

The move from GPLv3 to the Pimcore Open Core License (POCL) isn't just a licensing change – it's a strategic evolution. It's about aligning the platform with the real needs of modern teams, from startups to enterprises, from engineers to legal departments.

POCL empowers you to:

- Use and extend Pimcore freely within your organization
- Build private or commercial solutions without license risk
- Grow beyond the constraints of GPLv3, without giving up access or flexibility

For most developers and companies, the migration is refreshingly straightforward:

- If you own the code, you can relicense it
- If you support legacy systems, version-specific licensing keeps everything compatible
- If you're scaling fast, POCL grows with you
- No guesswork. No forced disclosures. No ambiguity.

The goal is simple: let you build amazing things with Pimcore – safely, sustainably, and with confidence.