

The invisible backbone of Europe: will PHP survive digital sovereignty?

By Frank , 9 September 2026

The invisible backbone of Europe: will PHP survive digital sovereignty?



By Frank Dumon

When European policymakers speak about digital sovereignty, it often sounds like a story about independence, innovation, and strategic autonomy. We are investing massively in open source software to break free from the grip of American tech giants like Microsoft, Google, and Amazon. Europe wants to control its own data, its own systems, and its own digital future. But there is an uncomfortable truth that is rarely mentioned in policy documents or press conferences: this European sovereignty rests on an invisible, chronically underfunded foundation. The name of that foundation is PHP.

PHP, originally standing for "Personal Home Page" and now for "PHP: Hypertext Preprocessor", has been the silent force behind the global web for decades. More than seventy percent of all websites with a known backend language run on PHP. It is

the language that made WordPress possible, that drives Drupal and TYPO3, and that enables millions of businesses, governments, and organizations to manage their digital presence without being dependent on proprietary solutions. Yet PHP is rarely celebrated as a strategic asset. It is the workhorse of the internet: indispensable, but rarely in the spotlight.

As the administrator of this Drupal website, this reality is not an abstract concept or distant future vision, but daily practice. Every update, every security patch, every new feature is a reminder that we are building on a foundation maintained by a relatively small group of developers. The question is not whether PHP still works today – it does, excellently – but whether the current model of development and maintenance is sustainable for the critical infrastructure of an entire continent. Can we continue to rely on a technology that is so widespread, yet receives so little structural support? And what does this mean for the future of European digital independence?

The paradox of European digital sovereignty

The numbers don't lie and paint a picture of enormous dependence. Germany, the economic heart of Europe and a champion of digital sovereignty, is allocating 108 million euros for the new generation of the *Government Site Builder*. This is the platform through which the German federal government is migrating hundreds of web pages to a modern, secure, and maintainable system. The platform relies heavily on TYPO3, the PHP-based content management system that is particularly popular in the DACH region. This investment is impressive and demonstrates political will to invest in open source alternatives.



But Germany is not alone. The European

Commission itself currently manages approximately 770 websites based on Drupal. From policy documents to citizen information, from internal portals to public services – Drupal forms the backbone of a significant portion of the EU's digital presence. The system is chosen for its flexibility, security, and ability to manage complex content structures. It is a conscious choice for open source, for transparency, and for avoiding vendor lock-in.

And then there is Nextcloud, the PHP-based collaboration and file-sharing service that positions itself as the European alternative to Google Drive, Dropbox, and Microsoft OneDrive. Approximately 300,000 German federal users use Nextcloud services daily. They share documents, collaborate on projects, store sensitive government information – all on a platform that runs on PHP. Nextcloud embodies the promise of digital sovereignty: data that remains within European jurisdiction, falls under European law, and is not subject to the Cloud Act or other foreign legislation that could demand access to data.

TYPO3, Drupal, and Nextcloud share one fundamental common feature that is rarely mentioned in policy documents on digital sovereignty: they are all written in PHP. While Europe pours millions of euros into the *application* of these open source solutions, the maintenance of the *underlying language* itself remains underappreciated and underfunded. It is a classic example of short-sighted policymaking: one is willing to pay for the facade of the house, for the interior design, for the security systems, but neglects the foundation until cracks appear. And when those cracks come – and they always do – it is often too late to strengthen the foundation without having to demolish the entire building.

Digital sovereignty: more than just open source

The concept of digital sovereignty is complex and multifaceted. At its core, it is about the ability of a state or region to exercise control over its digital infrastructure, data, and technology. For Europe, this means concretely: less dependence on American and Chinese tech giants, more transparency in algorithms and data processing, and the ability to develop and manage its own digital services without external pressure or influence.

Open source software is seen as a cornerstone of this strategy. The reason is clear: when the source code is public, it can be checked for backdoors, the community can contribute to improvements, and there is no single vendor with a monopoly on the technology. You are not dependent on the goodwill of one company for security updates, new features, or even the continued existence of the product. This sounds ideal, and in many ways it is.

But open source is not a magic solution. It merely shifts dependencies; it does not eliminate them. Instead of being dependent on Microsoft or Google, you are now dependent on the open source community that maintains the software. And here's the rub: many critical open source projects are maintained by a handful of volunteers, or by developers funded by competitors with different interests. The PHP Foundation, established in 2021 to provide structural funding for PHP development, is supported by companies like JetBrains, Automattic, and Rasmus Lerdorf (the creator of PHP itself). But the budget is modest compared to the enormous impact of the language.

Moreover, there is a fundamental difference between using open source software and contributing to its sustainability. Many organizations, including governments, are excellent at the former but fall short on the latter. They install Drupal or TYPO3, adapt it to their needs, and benefit from years of prior development. But how many of these organizations contribute structurally to the development of the underlying platforms? How many budget not only for the implementation and maintenance of their own instance, but also for supporting core development?

TYPO3 versus Drupal: two titans on the same foundation

The choice of a content management system is for many organizations a strategic decision that will impact their digital operations for years, if not decades. It is not just a technical choice; it is a choice for a certain way of working, a certain philosophy

about content management, and a certain vision on scalability and maintenance. In Europe, TYPO3 and Drupal are two of the main players in this field, both built on PHP, both open source, but with different strengths and target audiences.

TYPO3: The DACH standard with a focus on structured management

TYPO3 is a robust, enterprise-grade CMS framework that is particularly strongly rooted in the DACH region (Germany, Austria, Switzerland). The system was first released in 1997 and has since built a loyal following, especially among medium to large organizations that need to manage complex content structures. TYPO3 excels through an intuitive editorial interface specially designed for non-technical users, the power of TypoScript configuration offering unparalleled flexibility, and excellent multilingual support making it ideal for organizations that need to publish content in multiple languages.

TypoScript, TYPO3's configuration language, is both a strength and a barrier. It allows developers to customize virtually any aspect of the system without altering the core code, facilitating upgrades and benefiting stability. But it also has a steep learning curve and requires specialized knowledge that is not always easily found outside the DACH region. This has led to a situation where TYPO3 is dominant in Germany and surrounding areas, but less known in other parts of Europe and the world.

TYPO3 is a system that thrives within structured, bureaucratic environments. It is particularly well-suited for governments, large corporations, and educational institutions that require strict workflows, detailed permissions, and the ability to manage complex content structures across multiple websites and domains. The German choice for TYPO3 in the Government Site Builder is therefore no surprise; it fits perfectly with the German culture of thoroughness, structure, and long-term planning.

Drupal: Global flexibility for complex challenges

The Web's Most Powerful Open Source CMS

Community-built and AI-ready, Drupal gives organizations the freedom and flexibility to create digital experiences without limits.

[Try Drupal CMS →](#)

[See what Drupal can do →](#)



Drupal, on the other hand, is globally distributed and architecturally more flexible. Where TYPO3 excels in structured content management within a clearly defined framework, Drupal shines in adapting to unique, complex requirements that do not always fit a standard pattern. Drupal has advanced, granular permissions as a core feature, making it particularly suitable for projects with high security requirements and complex user role structures.

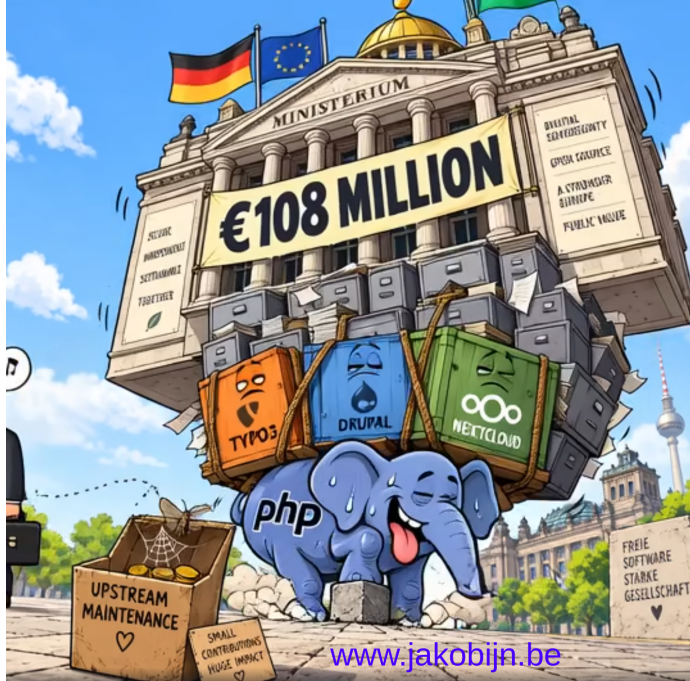
Drupal's architecture is based on the concept of "entities" and "fields", allowing developers to model virtually any form of content without limitations. This makes Drupal especially suitable for projects like this website, where managing articles, books, videos, images, archives, and related content requires a flexible yet structured approach. The system can grow from a simple website to a complex digital ecosystem with thousands of content types, integrated workflows, and advanced publishing capabilities.

Drupal's community is global and diverse, with strong presences in North America, Europe, and increasingly also in Asia and South America. This global spread means that knowledge and expertise are more widely available, but also that development is sometimes less focused on specific regional needs. Drupal is chosen by organizations willing to invest in custom development and that need a system that can grow with changing needs.

Whether a government chooses the streamlined control of TYPO3 or the flexible power of Drupal, the dependence on PHP remains absolute. Both systems stand or fall with the quality, security, and performance of the underlying PHP version. A security vulnerability in PHP itself can affect both TYPO3 and Drupal installations. Performance improvements in PHP 8.x benefit both systems. And if PHP were to

disappear or no longer be adequately maintained, both TYPO3 and Drupal would face an existential crisis.

The financial paradox: millions for projects,



There is a stark discrepancy between the

amounts European governments spend on implementing open source systems and the amounts made available for maintaining the underlying technology. Germany is allocating 108 million euros for the Government Site Builder – an impressive amount demonstrating political will and strategic insight. But how much of that budget goes towards supporting TYPO3 core development? How much does the German government contribute structurally to the PHP Foundation?

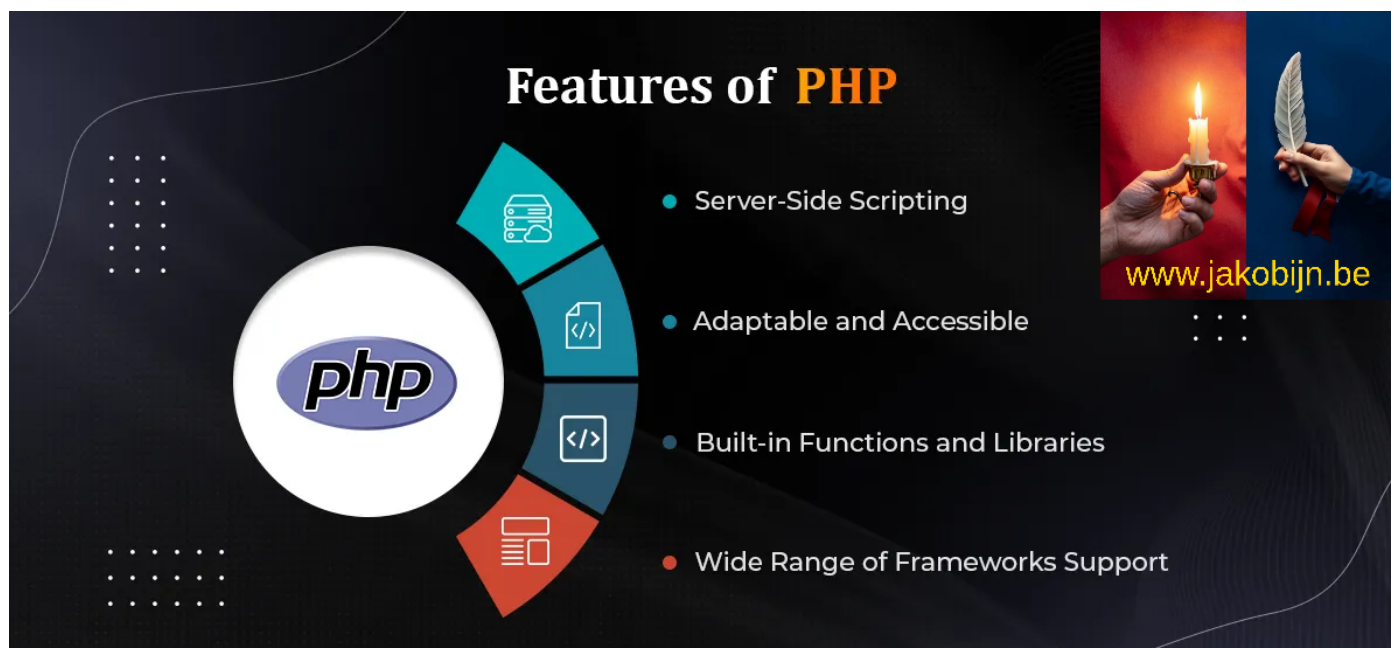
The reality is that most governments and large organizations budget for implementation, customization, and first-line maintenance of their own CMS instance. They pay developers to build modules, customize themes, and realize integrations. They invest in hosting, security, and employee training. But the budget for supporting upstream development – the work done on the core of PHP, TYPO3, or Drupal itself – is often negligible or nil.

This is a form of free riding that is unsustainable in the long run. The developers working on the core of PHP often do so as volunteers alongside their paid jobs, or are funded by commercial companies pursuing their own interests. The PHP Foundation, established to change this, has a modest budget primarily funded by a handful of tech companies and individual donors. But this pales in comparison to the economic value PHP generates for the global economy.

Suppose Europe is truly serious about digital sovereignty. Wouldn't it then be logical to reserve a small percentage of the millions spent on open source implementations to support the underlying technology? If the EU manages 770 websites on Drupal, and Germany spends 108 million euros on TYPO3, and hundreds of thousands of civil servants use Nextcloud, wouldn't a structural contribution to PHP development be the most logical insurance you could take out?

The future of PHP: survive or disappear?

The prediction that PHP would die is a recurring theme in the tech world since the rise of alternatives like Node.js, Python, Ruby, and Go. Every time a new, hip programming language emerges, articles appear with titles like "PHP is dead" or "Why we moved away from PHP". And yet, more than twenty years after the first critical obituaries, PHP still powers more than seventy percent of all websites with a known backend language. WordPress, which accounts for more than forty percent of the entire web, runs on PHP. Drupal, TYPO3, Joomla, Magento – all PHP.



Will PHP survive? **The short answer is yes**, but not without a fight, and not without fundamental changes in how we view it and how it is maintained. PHP has evolved in recent years from a simple scripting language with inconsistent function names and loose typing into a performant, strongly typed programming language. With the introduction of PHP 7 and later PHP 8.x, the language has been thoroughly modernized. JIT compilation has been added, the type system strengthened, and performance drastically improved. The technical argument against PHP – that it is slow, insecure, or amateurish – is largely outdated.

But the *strategic* argument is all the more urgent. The way PHP is developed and maintained is not in line with the critical role it plays in the global digital infrastructure. The PHP Foundation does important work, but operates with a budget that is modest compared to the impact of the language. There are about ten full-time developers working on the core of PHP, supported by a network of part-time contributors and volunteers. For a language that runs on millions of servers and supports trillions of euros in economic value, this is remarkably little.

The dependence on a small group of developers creates a vulnerability. What happens if Rasmus Lerdorf, the creator of PHP, decides to step back? What if JetBrains or Automattic decide to stop their sponsorship? What if the next generation of developers is not interested in maintaining a language that many consider "old"? These risks are real and rarely discussed in strategic plans for digital sovereignty.

Moreover, there is the challenge of knowledge transfer and talent development. Young developers are drawn to new, hip technologies. They learn React, Node.js, Python, Rust. PHP is often seen as boring, old, or only suitable for legacy systems. This perception problem is dangerous because it leads to a shortage of new developers specializing in PHP. And without new developers, a technology eventually dies a slow death, not due to technical inferiority, but due to a lack of human capital.

Recommendations: from free to fair

If Europe is truly serious about digital sovereignty, then it is time for a fundamental reconsideration of how we deal with open source software. Open source does not mean free – it means freedom. Freedom to view, modify, and share the code. But that freedom comes at a price, and that price must be paid if we want the technology to remain sustainable.

Concrete recommendations for European governments and organizations:

First: Reserve a structural percentage of the IT budget for upstream contributions. If you spend 100,000 euros on implementing a Drupal website, reserve at least 5,000 euros per year for contributions to the Drupal Association and the PHP Foundation. Make this a fixed part of your budget, just as you budget for hosting and security.

Second: Stimulate knowledge sharing and talent development. Invest not only in training your own employees in using Drupal or TYPO3, but also contribute to the broader PHP community. Sponsor students, organize meetups, write documentation, share knowledge publicly. Help attract the next generation of PHP developers.

Third: Evaluate dependencies proactively. Make an inventory of all open source components you use, from the application level to the programming language. Assess the health of these projects: how active is development? How large is the community? How sustainable is the funding? Use this information to make strategic decisions about where your own contributions are most needed.

Fourth: Collaborate with other organizations. Digital sovereignty is not an individual struggle; it is a collective effort. Governments, companies, non-profits, and educational institutions should work together in consortia that provide structural funding to critical open source projects. The combined impact of hundreds of organizations each contributing a modest amount can be enormous.

Conclusion: sovereignty requires responsibility

PHP will not disappear overnight. The inertia of enterprise systems, the scale of WordPress, Drupal, and TYPO3, and the enormous cost of migration guarantee PHP at least another decade of dominance. But survival is not the same as thriving. And for a technology so central to Europe's digital infrastructure, survival is not good enough.

True digital sovereignty begins with recognizing the chain of dependency. It is not enough to *use* open source software; one must also be willing to *invest* in the communities that maintain the language on which that software runs. As long as Europe spends 108 million euros on the superstructure, but allows the substructure to wither into a charity project, digital sovereignty remains a paper tiger.

The task for us, as users, developers, policymakers, and citizens, is to demand that the foundation is taken as seriously as the building that rests upon it. PHP is not just a programming language; it is the invisible backbone of the European digital economy. It is time to give that backbone the support and recognition it deserves.

Geopolitical hypocrisy: war financing versus digital self-reliance



It is a question that not only burns on the retina, but ruthlessly exposes the fundamental hypocrisy of current European policy. How is it that the European Union effortlessly makes tens of billions of euros available for financing a destructive conflict in Eastern Europe, while finding a fraction of that amount for building a genuine, sustainable digital sovereignty apparently proves to be an insurmountable hurdle? This glaring disproportion is no accident, but a symptom of a deeper geopolitical and moral failure.

Europe speaks loudly about strategic autonomy, but in practice the continent remains a slavish follower of Atlantic interests.

The willingness to pump immense sums into a war machine – supported by a narrative of moral superiority, despite the widespread and documented corruption in the recipient country – stands in stark contrast to the refusal to invest in its own technological backbone. Financing a war is politically 'visible'; it provides photo opportunities, reinforces traditional power structures, and meets the expectations of the established order. Maintaining open source code, structurally supporting the PHP Foundation, or securing critical Nextcloud infrastructure, on the other hand, is invisible, technical, and boring work. It yields no direct political gain, but is the only path to true independence.

This is the true price of illusion. As long as European policymakers choose the semblance of geopolitical relevance through military escalation, instead of investing in the sober, daily labour of digital independence, digital sovereignty remains an empty shell. We build castles in the air on a foundation of sand, while throwing away the money needed to solidify that sand on fuel for a conflict that only undermines European prosperity and stability in the long run. True sovereignty begins not on the battlefield, but in the code, the servers, and the moral courage to fund our own technological base fairly, instead of letting it wither in favour of a destructive foreign adventure.

Source list

1. CloudNews.tech. "The European digital sovereignty has an invisible dependency: PHP." Accessed via <https://nl.cloudnews.tech/de-europese-digitale-soevereiniteit-heeft-een-onzichtbare-afhankelijkheid-php/>
2. Arocom.de. "TYPO3 vs. Drupal: The Comprehensive CMS Comparison." Accessed via <https://www.arocom.de/en/knowledge/drupal/typo3-vs-drupal>
3. PHP Foundation. "About the PHP Foundation." <https://opencollective.com/phpfoundation>
4. European Commission. "Open Source Software Strategy 2020-2023." EU Publications, 2020.
5. Bundesregierung Deutschland. "Government Site Builder: Modernization of the Federal Web Presences." 2023.
6. Nextcloud GmbH. "Nextcloud in the Public Sector." <https://nextcloud.com/government/>
7. Drupal Association. "State of Drupal 2023." <https://www.drupal.org/state-of-drupal>
8. TYPO3 Association. "TYPO3: The Enterprise Open Source CMS." <https://typo3.org/>
9. W3Techs. "Usage Statistics of PHP for Websites." <https://w3techs.com/technologies/details/pl-php>
10. European Commission. "EU Digital Sovereignty: Strategic Framework." COM(2021) final.
11. GitHub. "The State of Open Source Security Report 2023." GitHub, 2023.
12. Lerdorf, Rasmus. "The History of PHP." Presented at various conferences, 2015-2023.
13. JetBrains. "PHP Developer Survey 2023." JetBrains s.r.o., 2023.
14. Linux Foundation. "Open Source Software and Digital Sovereignty in Europe." 2022.
15. OWASP. "PHP Security Guidelines." Open Web Application Security Project, 2023.

<https://youtu.be/rAR1QDAS7og?list=PLpVC00PAQQxGFC06mLqoPT4hHaA1Ykn2Z>