

WordPress 7.0 Brings AI Into Its Core: A Practical Leap, and a Question of Sovereignty

□ Bernard Sfez - 2026-07-21 13:50



With its 7.0 release, "Armstrong," WordPress brings artificial intelligence right into its core. The promise is appealing: connect an AI to your content in a few clicks. But that default path sends your knowledge off to models hosted by third parties.

That convenience carries a cost few users stop to weigh. Internal content, client files, strategic notes. Anything you hand to a hosted AI can be logged, reused, or placed under a foreign jurisdiction. There is another way, though, sovereign by design, and we will come to it by the end of this article.

What WordPress 7.0 Actually Changes?

On 20 May 2026, WordPress shipped its major 7.0 release, codenamed "Armstrong." Two maintenance releases followed, 7.0.1 on 9 July 2026, then 7.0.2 on 17 July, the latter a forced security update patching a critical flaw. Beyond the refreshed dashboard, the Command Palette now available everywhere, and per-block custom CSS, the real headline of this release comes down to two letters, AI.

For the first time, WordPress brings artificial intelligence in not through a third-party plugin, but straight into its core.

Three building blocks give this shape:

- the Abilities API, which exposes administrative operations in a controlled, structured way
- the AI Services Registry, which lets you connect hosted model providers
- the AI Client, the unified interface plugins now standardize around

In practice, this means a site owner can now wire an AI to their content in a few clicks, where before they had to assemble a patchwork of plugins themselves. That is a real gain in simplicity. But every simplicity has a price, and it is worth knowing what it is.

Under the Hood: What the AI in WordPress Core Really Does?

To grasp what is at stake, you have to understand what these blocks actually do.

The Abilities API does not "think." It exposes site actions (create a draft, categorize content, query data) in a structured form that a model can trigger within set bounds. It is a gateway, not a brain. The brain is still the language model, and that is where everything is decided. A model on its own does not know your content; it knows only what it was trained on. For it to answer correctly about your documentation, you have to feed it the right passages at the right moment. That is the job of RAG (Retrieval-Augmented Generation). Your content is split up, turned into vectors (numerical fingerprints of meaning), and stored; then, for each question, the relevant passages are retrieved and handed to the model before it answers.

A chatbot's whole quality rides on this retrieval step. A poorly tuned RAG returns off-topic passages, and the model "improvises" to paper over the gaps: that is hallucination. A rigorous RAG never lets the model invent, it forces it to lean on real sources. WordPress 7.0 provides the plumbing to connect a model; it does not settle, on your behalf, the question of which model, where it runs, and how your content is presented to it.

The Connectors Screen: Choosing and Centralizing Your Providers

Next comes what WordPress calls the AI Services Registry, though the administrator meets it under a more down-to-earth name in the menu, Settings > Connectors. This is where you decide which AI services your site is allowed to use. WordPress ships three connectors by default, OpenAI (the GPT models), Anthropic (Claude), and Google (Gemini), and the administrator enters each API key just once. Every compatible plugin then reuses that connection without ever asking for the key again, where before you had to paste the same key into three different plugins. The list is not closed. Any other provider can be added, as soon as its plugin registers with the system, and nothing technically prevents you from connecting a model you host yourself, on your own server. That is worth keeping in mind for what follows, because it shows that the choice of model, and therefore of where your content goes, stays entirely in your hands.

From this single screen, the administrator switches a provider on or off and sees what is connected. It is a way of taking back control of management, not yet of confidentiality, because content sent to an external provider still leaves for its servers. We will come back to this.

The AI Client: One Socket for Every Plugin

That leaves the AI Client, the shared interface through which plugins talk to the providers declared in Connectors. For the administrator, the benefit is very concrete. A plugin written for the AI Client works with any registered provider, which means you can switch models without reinstalling anything. You can move from Claude to GPT from the Connectors screen, and your plugins keep working unchanged. Where you once had to juggle bespoke connections specific to each plugin, each with its own key and its own logic, you now have a single, interchangeable socket.

This progress does come with a trade-off the shrewd administrator will keep in mind. Since the key is registered at the site level, any installed plugin can use it, with no partitioning and no per-

plugin approval. And these keys, stored in the database, are not encrypted, only masked on screen. A malicious or compromised plugin therefore has access to a paid connection and to the data flowing through it. The convenience of a single socket is real; so is its exposure surface.

How Much Does It Cost?

Connecting a hosted model means starting a meter. Every provider has its own pricing, but they all bill by the token, the unit that chops text into fragments. Unlike a plugin you install once, a hosted model is paid for by use: as long as it answers, it charges. Every question, every answer, every snippet of content sent to give context eats up tokens, and therefore money.

WordPress 7.0 does deserve credit for exposing that meter. The Connectors screen tracks token consumption, latency, and error rate per provider, letting the administrator watch spending without an extra plugin. But watching is not controlling. A badly tuned plugin, a loop between two plugins, a sudden surge of traffic, and the bill can climb within hours. With no spending cap set on the provider side, the budget runs away. And so, into a platform you thought had predictable costs, a recurring and variable cost model quietly takes up residence, one you have to learn to steer.

I Connected a Provider... Now What?

WordPress 7.0 gave you the tools, you took out a plan with OpenAI or Anthropic, you pasted your key into the Connectors screen. And now, a fair question: what does it actually do?

Does it put a chatbot in front of your visitors? No, not on its own. Does it help you write a new article? Partly, through the editor's AI features, provided a plugin makes use of them.

Does it improve existing content, rewrite, summarize, translate, suggest titles, flag weak pages? Only if a tool is wired in to do so. Does it answer customers, triage requests, feed an internal knowledge base?

None of that happens automatically.

Because this is the heart of the misunderstanding. Saving an API key switches on no feature. It opens a tap; you still have to connect something to it. Connectors and the AI Client are a power outlet, not an appliance. What the business really wants, an assistant that writes in its voice, improves its pages, answers its customers correctly, still has to be assembled, tuned, and grounded in its own content. WordPress delivered the electricity. Not the light.

What Is Left for You to Do?

Once the outlet is in place, everything still has to be plugged in. You will need to find the right plugins and check they are trustworthy, subscribe to and pay the model providers, tune content retrieval to avoid made-up answers, partition access, set spending caps, test answer quality before exposing it to your customers, secure keys that, let us remember, are not encrypted, then keep all of it up to date as models and plugins evolve. Nothing insurmountable. But nothing automatic either, and above all, nothing that fits into an afternoon of tinkering.

Let us give credit where it is due, WordPress 7.0 is a fine step forward. AI enters the CMS foundation with clean, unified plumbing: the Connectors screen to manage providers, the AI Client as a common socket, the Abilities API to frame actions, and built-in usage tracking. The whole thing is open, extensible, and built to last. Hats off to the community, because it is a considerable body of work.

Set against that progress, a few questions remain. Without putting anyone on trial, they deserve to be asked.

Cost, first. You leave the reassuring flat-fee model for usage-based billing, variable, rising with traffic and liable to spiral if nothing reins it in. On top of that comes the price of the plugins that actually put the AI to work, and of maintaining them over time.

Security, next. API keys are stored in clear text in the database and accessible to any installed plugin. Every added plugin widens the exposure surface, and the Abilities API, which lets the AI trigger actions on the site, introduces a new risk: a malicious piece of text, slipped for instance into a comment, could try to trigger an unintended operation.

Confidentiality, above all. By default, your content leaves for a third-party provider, potentially subject to a foreign jurisdiction. And even internally, a real question arises: do you know exactly who has access to what? Which data is sent, to which model, readable by which plugins, and seen by which users? Without clear partitioning, the answer is rarely sharp.

Knock-on risks, finally. Every block added, every plugin connected, every provider linked is one more link to monitor, update, and secure. AI does not only add possibilities, it adds responsibilities.

All legitimate questions. They all have an answer, but that answer takes time, skill, and method. All well and good... but who is going to take care of it?



The Need for a Dual Skill Set

These questions do have answers: you now need, at your side or on your team, a dual competence, WordPress and AI. And these are two distinct trades.

The person who maintains your site's appearance, applies your updates, and keeps it in good shape does valuable work, but designing a reliable assistant, choosing and tuning a model, securing access, and guaranteeing correct answers calls for a different expertise. If you do not have it in-house, or if your usual provider reaches the edge of their field here, the question stays open.

Some will point out that their host already offers an AI. True, and those tools are useful, but they answer a different need. The assistant from a Hostinger or an IONOS mainly helps you build your site or navigate your hosting panel; it is not an assistant trained on your content to answer your customers. And even when hosted in Europe and GDPR-compliant, these services run on the provider's infrastructure, with their model and inside their ecosystem. The day you change hosts, the assistant does not follow you.

So the real question is not just "where is my data," but "who keeps control, and can I leave with it?"

Who to Turn To?

Several profiles can help you handle these: WordPress consultants, integrators, AI specialists. The essential thing is to find someone who genuinely masters both worlds, and who takes security

and confidentiality seriously rather than treating them as options.

For our part, this is the ground we have worked on for years. Data sovereignty is not a recent talking point for us, it is a core conviction, forged by long habit of demanding environments and of cybersecurity in hostile settings, where there is no room for approximation. That same rigor is what we bring to your assistant.

What we offer is simple to state. A private assistant, grounded in your own content, tested before going live on questions whose answers you already know. Your data stays on sovereign infrastructure, in France or in Europe, and never leaves for a third-party provider. A predictable managed-service cost, rather than a meter that runs wild. Clear partitioning of what each audience is allowed to see. And because the stack rests on open-source software, you are captive to no one: you can change model, hosting, or provider, and leave with your data whenever you wish.

All of it stays compatible with the CMS you already run, WordPress like any other, and answers in your users' language, French, English, German, or otherwise. WordPress delivered the electricity. We help you turn it into light.

Discover our sovereign AI chatbot - Let's talk about your project

Sources

Introducing the Connectors API in WordPress 7.0 - Make WordPress Core

Introducing the AI Client in WordPress 7.0 - Make WordPress Core

WordPress News - the 7.0 "Armstrong" release announcement

WordPress Developer Resources - Abilities API