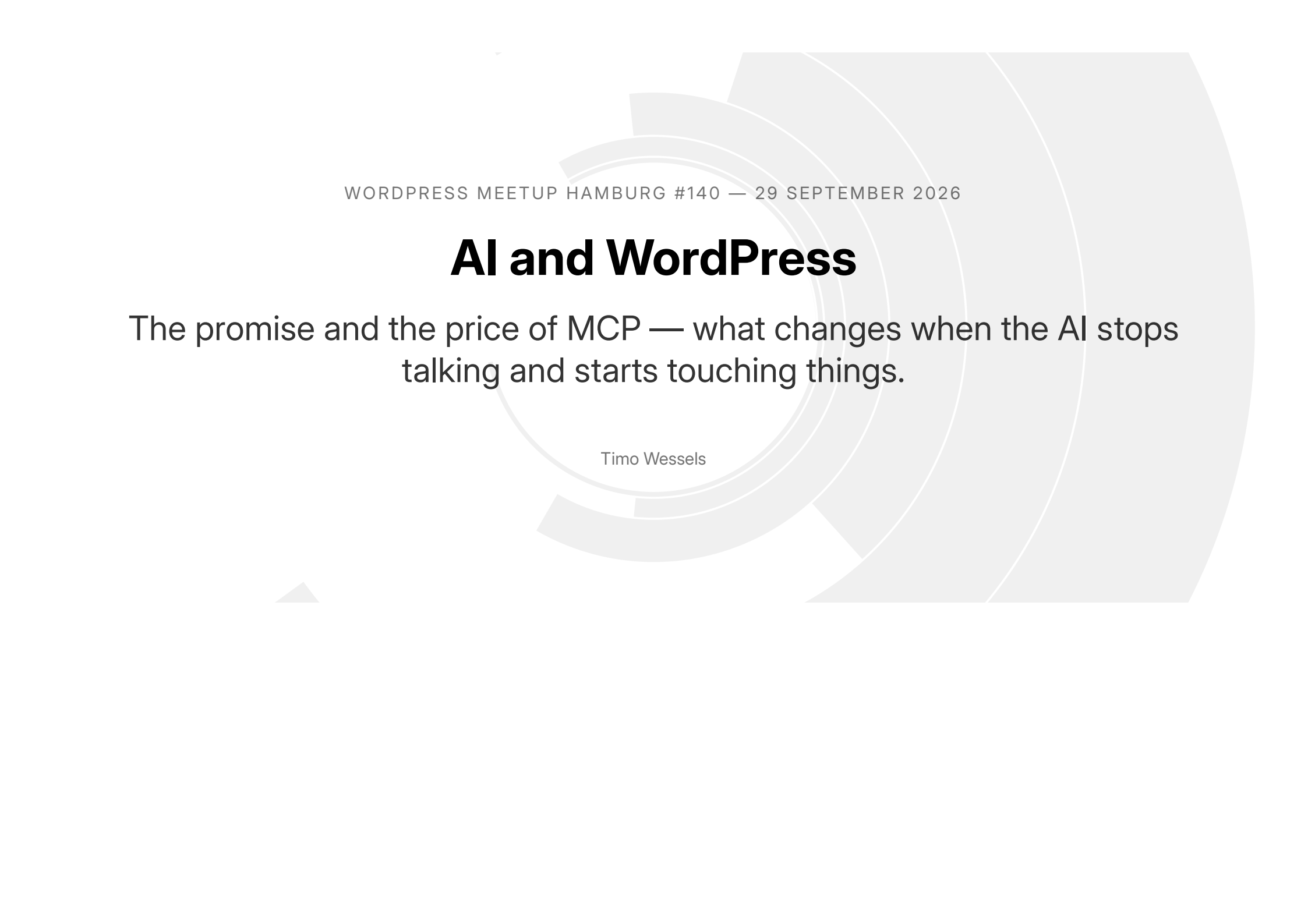




WORDPRESS MEETUP HAMBURG #140 — 29 SEPTEMBER 2026

AI and WordPress

The promise and the price of MCP — what changes when the AI stops talking and starts touching things.

Timo Wessels

The evening in fifteen steps

01 The AI no longer just sits in the chat window

02 All of this already exists

03 What suddenly works

04 I am not a developer

05 And then you want to change something

06 Every connection you make is a risk you make

07 "Goodbye WordPress" — the other side

08 Away from WordPress, and back again

09 Static or generated

10 The wrong funeral

11 The middle ground

12 So what does that mean for us?

13 For Monday morning

14 Your turn

15 Sources

The AI no longer just sits in the chat window

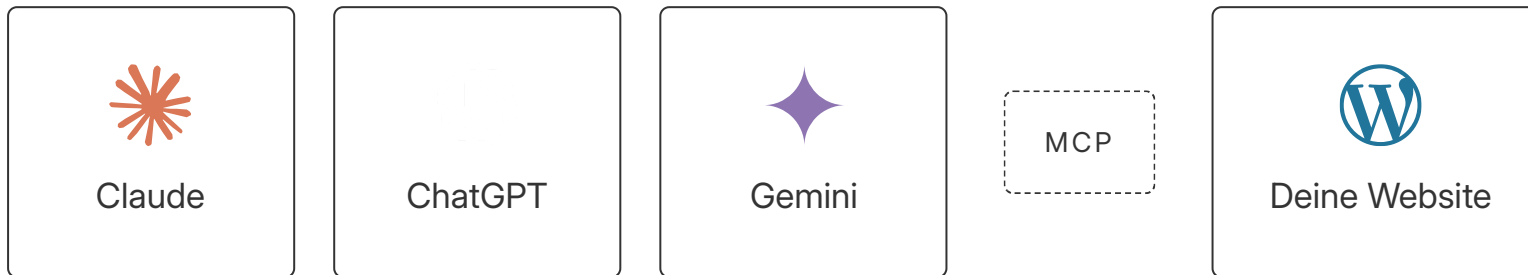
Until recently the AI talked. Now it is allowed to touch things.

- Yesterday: you ask, it answers, you paste the result in yourself.
- Today: you connect it once — and it does that part too.
- MCP is the standard plug for it. USB-C for AI.
- Not a product, not a company, nothing to buy — an agreement.
- Open to all: Claude, ChatGPT and Gemini use the same plug.

| The difference is not the speed. The difference is the access.

One plug, many models

That is the whole trick of MCP: not every AI needs its own connection to your website. There is one agreement they all keep.



Implement the agreement once and you can change the model without starting again. That is precisely why the standard spread so quickly — it saves work for everyone involved.

All of this already exists

Not a forecast, and not something cobbled together in a GitHub basement. It is in WordPress itself.

- WordPress 7.0: one place for AI credentials, under Settings → Connectors.
- WordPress does no thinking. It manages the connection outward, nothing more.
- A plugin may only use that connection once you have approved it.
- The MCP Adapter is what makes your own site reachable by an agent.
- Meta Box builds fields with it, Etch builds in the builder — the vendors have moved.

| You used to have to talk WordPress into this. Now the door is built in.

The four pieces that already exist

None of this has to be invented. All of it can be installed today.

Connectors API — in core since WordPress 7.0

One place for the keys to AI services, under Settings → Connectors. Every plugin used to build its own field for that. And a plugin may only use the key once you have approved it.

MCP Adapter — the door outward

Makes your own WordPress site reachable by an AI agent. This is the official route from the WordPress project itself; the older Automattic plugin that started it has been retired.

Meta Box — the AI builds the structure

Through the new Abilities an agent can create and fill post types, taxonomies, field groups and field values. Not just writing text — building the frame the content later sits in.

Etch — the AI works inside the builder

The connector attaches an agent to your open builder tab: create pages, set styles, build components. What you see on screen changes while the agent works.

What suddenly works

What used to be an afternoon is now a coffee break.

- An image or a piece of HTML in — five to ten minutes later the page stands.
- Good for a landing page, a small project, a first draft.
- Very good for grind work across hundreds of posts.
- The repeatable half of our work is becoming cheap.
- And it was rarely the half we enjoyed.

| "It stands" is not the same as "it is finished".

I am not a developer

The most interesting evidence tonight is not on YouTube.

- No PHP, no Python, no degree — and around thirty tools of my own.
- I did not learn to program. I learned to describe, and to check.
- Checking only comes with years. Here, experience is an advantage.
- It grew over months. And plenty of it was thrown away again.
- Every tool is a promise: maintenance, documentation, understanding.

| If I can do this, the line is not where you think it is.

What came of it

It started with a search across my own notes. By now it is a loop that checks itself.

A search across everything I own

Where it began: full-text search over everything I have ever written down. The AI does not have to guess how my shop works — it can look it up. That alone changed more than any cleverer model did.

The rules live in the file itself

Every file carries a marker at the top: what it belongs to, which contract governs it, how it must be written. Whoever opens it — me or an AI — knows in the first paragraph what is allowed here. Rules kept somewhere else do not get read.

Tools that bring knowledge in

Download websites, strip the content out of the layout, make code searchable, read PDFs. Around two dozen of them by now. None of them is clever — they only make sure there is something to look up.

One fact, one place, one checker

A fact is written in exactly one place. Generators distribute it where it is needed, and check scripts recompute it and complain when a copy disagrees. That is the part nobody sees, and the reason it still fits together two years later.

And then you want to change something

The first eighty percent take ten minutes. The last twenty still take a day.

- The AI knows WordPress. It does not know your website.
- Framework, class names, nesting, style guide — all of it open questions.
- Without rules it builds differently every time. And each result works on its own.
- A craftsman who has lost his memory every morning.
- One page always works. Ten sites over two years — who holds that together?

| Speed moves the work. It does not remove it.

A real example, from this week

Alexander Schell had his agency's website built entirely by AI — design system, Figma library, components in Drupal — and wrote publicly about how it went. He is the best witness, because he is in favour.

What he reports

Eight times faster than comparable projects, four times cheaper, because one person covered frontend, backend, UX and testing. That is not a small number, and I believe him.

Where he was needed

By his own account not while building, but while approving and correcting. His examples: "No, the colour comes from the token." "No, that is the third card, the others go." "Compare it to the Figma layout, the spacing is off."

What is in the delivered markup

The home page carries 792 values written straight into the classes, 129 distinct: twelve font sizes from 10 to 52 pixels and twelve colour values, typed by hand instead of taken from a token. The headings look healthy until you read the order: an h2 before the h1, then it jumps to h4. And the gold on the dark blue reaches 3.34:1 while being set at 10 to 13 pixels in 47 places, where 4.5:1 is required. Counted on 8 September 2026; he says himself the site is live and not yet finished.

Why this is not an accusation

It is the confirmation. Exactly this work — deciding that one colour applies and the other three do not — stays with a human. It does not shrink when building gets faster. It becomes visible.

This is not a bug in the prompt. That is the nature of things: AI produces plausible variants. A design system needs the opposite — one decision that applies everywhere.

Alexander Schell, Activate & TAXINIO

Every connection you make is a risk you make

The measure is not what the AI does today. It is what it could do.

- What counts is what a system is capable of — not what you happened to type.
- Phoning the tradesman, or handing him the key: that is the difference.
- Customer data involved? Then a data processing agreement under Article 28 GDPR.
- The line runs between consumer and business plans, not between free and paid.
- And every new connection is also a new way in.
- Then the jurisdiction question: a US provider, a European alternative, or a model on your own machine.

| Contracts first, then the plug. Not the other way round.



"We didn't expose anything yet" isn't a privacy policy argument. Privacy policies are about what a system is capable of doing with data and who can access it, not just your prompting.

Kathy Zant

"Goodbye WordPress" — the other side

If I do not make the opposing case myself, it sits in the room anyway. Just unspoken.

- Over a thousand websites, ten years of WordPress and Elementor — untouched since February 2026.
- Ninety rental units: weeks of work planned, standing in twenty minutes.
- He buys no plugins any more. He builds them to fit, without the thousand settings.
- No plugins, no login, no attack surface — his security argument, turned around.
- But: never one prompt, more like two hundred. And he does not let the AI design.

| That is not a spectator. That is one of us who left.

Most small businesses never really needed WordPress. WordPress was easy for us, the developer, to set it up for them.

Emanuel, "I Create Your Site"

Away from WordPress, and back again

The move rarely fails on the technology. It fails on the question of who maintains the site in six months.

- A single page needs no CMS. Granted.
- The question is not what builds faster, but who keeps it afterwards.
- The client wants to change the opening hours: a text file? Git? A branch? A prompt?
- The compromise everyone lands on in 2026: WordPress behind, Astro in front.
- What gets kept is the back end — the part everyone calls old-fashioned.
- And you need not choose: headless, or shipped from a Git repo through Vercel.

| You are not throwing WordPress away. You are throwing the front end away.

Static or generated

Two ways a page comes into existence — and the difference decides how reliably an AI can work on it.

- Static: the HTML already exists. It is the result.
- Generated: the HTML appears on request, out of database, theme, plugins, filters and hooks.
- In the second case you do not write the result. You write the thing that produces it.
- For an AI that means: in one case it can check what it wrote. In the other it cannot.
- Same model, very different reliability. Not because of the AI, but because of the distance to the output.

| What you cannot look at before it exists, you cannot check either.

What each side costs

Neither camp wins. Both charge you, just in different places — and you decide per project which bill you would rather pay.

Static — in favour

Nothing runs, so nothing can crash. No PHP at the front, no database on the network, no weekend updates. What you see is what shipped — you can read it, check it, version it. And it is absurdly cheap to host.

Static — against

Every change needs somebody with access to the files. No roles, no approvals, no editor. And it scales badly: see the staircase on the next slide.

Generated — in favour

Roles, permissions, approvals, search, forms, translations — all there, without anyone building them. The client changes things whenever they like. This is what a CMS was invented for, and at that it is unbeatable.

Generated — against

You never write the result, you write the machine behind it. One update, one plugin, one filter, and the output changes without you touching anything. More surface, more maintenance, more that can break.

The staircase

It always starts the same way: one page, static, done. And then the client comes back with the next wish.

One page.

Perfectly fine. Static is exactly right.

Add a contact page.

Now you need a form.

A form.

Now you need a service that sends mail.

A mail service.

Now you need a webhook you do not entirely understand.

Embed a video.

Now you need consent handling.

Consent handling.

Now you need a cookie banner.

Five steps from "static is enough" to a CMS you built yourself — only worse.

The wrong funeral

Somebody has died, yes. But the whole industry is standing at the wrong grave.

- Creating content and managing content are two entirely different things.
- Who may change it? Who approved it? Which version is the real one?
- Where is it used — and in how many languages?
- And who is answerable for what the AI published?
- The cheaper creating gets, the more managing matters. Not less.

| The photocopier did not abolish the filing room. It made one necessary.

For some websites, you don't need a CMS. But creating and managing content are two very different things.

Mathias Bolt Lesniak

The middle ground

The question is not "AI or WordPress". It is: who may change this site in two years, and with what?

- He does not abolish the CMS. He rebuilds it, for one client, alone.
- WordPress runs anywhere: if a host gets too expensive, you move. With everything.
- And if you are gone tomorrow, your client finds somebody who knows it.
- The normal case is the middle: the client does text and images, we do the rest.
- Bricks components as blocks, Etch, PHP blocks in core — templates with guardrails.

| The AI builds the pieces. WordPress holds the frame, the permissions and the handover.

It is not about the framework

Which tool wins is the wrong question. There are three that count — and they are best asked before you commit to anything.

Maintainability: can somebody else change it?

Not "can I change it". Can somebody who was not there change it, a year from now, without phoning me? Names in the markup and values in the token system are not a matter of taste — they are the answer to exactly that question.

Scalability: does the fiftieth page cost what the first did?

If every new page needs fresh decisions about spacing, colour and variants, it does not scale — it gets more expensive with every page. If the pieces are settled, a new page is only: pour in the content.

Access: who can even get in?

Not accessibility in the WCAG sense, but the plain question: who may publish, and can they? One person with a Git account and a deploy process — or anyone on the team who can log in? That decides whether a site stays alive or freezes.

And that is exactly the offer

Set it up properly once: templates, components, roles, permissions. Then write and publish as often as you like, without anybody touching code. The effort sits at the start rather than in every single post — that is the whole bargain, and it is twenty years old.

Two ways to write the same card

On the left is what actually ships. On the right is the same element, with names for its parts. Both work. Only one of them can still be read in a year.

WITHOUT NAMES — WHAT THE AI BUILT

```
<div class="flex h-full flex-col justify-start gap-3.5">
  <div class="flex size-10 shrink-0 items-center
    justify-center rounded-full bg-gold/[0.18] text-gold">
    <span class="flex w-full flex-col before:absolute
      before:inset-0 before:content-['']">
      <h3 class="font-sans text-[19px] leading-6
        font-semibold text-white">Bessere Qualität</h3>
      <span class="mt-3.5 font-body text-sm
        leading-[23px] text-white/70">...</span>
```

WITH NAMES — THE SAME PIECE

```
<article class="feature-card">
  <div class="feature-card__icon">...</div>
  <h3 class="feature-card__title">Bessere Qualität</h3>
  <p class="feature-card__text">...</p>
</article>
```

A framework like ACSS takes the decision off your hands: consistent spacing, consistent colour, checked contrast — at the price of every site resembling the next, down to the identical icons. Utility classes take nothing off your hands: total freedom, and afterwards nobody knows what an element is. The middle is unspectacular: your own names for your own pieces, values from the token system. Then the markup says what a thing is, and the values say how it looks.

So what does that mean for us?

The uncomfortable question is not whether the AI can do our work.

- Eight hours against one — the client sees the result, not the road.
- If building becomes cheap, what is actually being paid for?
- Probably for what comes after: liability, maintenance, handover, structure.
- Not learning every tool. But understanding what each one does to you.
- Knowing how things go wrong over years is not something you can prompt.

| Is the website still our product — or by now the easiest part of it?

What it costs — and what it hangs on

Everyone talks about speed. Almost nobody talks about the invoice and the dependency, and those decide whether this still holds in three years.

The price is known today, not tomorrow

I pay around 200 euros a month for the subscription I work with. For me that pays off easily. But I have no contract saying it will still do so next year: prices rise, plans get restructured, quotas get cut. Anyone basing their costing on it is costing on somebody else's ground.

The model under you gets swapped

What works reliably today can behave differently after a model change. That is why my knowledge lives in a system rather than in a chat history — a system survives the swap, a conversation does not. It is also the answer to "which AI should I use": the less that answer matters, the better off you are.

And it hangs on your machine

Tools, credentials and configuration sit locally. If the disk dies, it is all gone for a while. So: backups — and not only of the website, but of what you built yourself. Precisely the part nobody thinks of, because it looks like tooling rather than work.

Without guardrails an AI takes a site apart

Write access plus one misread instruction is a deleted post type. Test system, its own user, few permissions, a backup first. That is not distrust — it is the same reflex as before a big plugin update.

For Monday morning

Concrete, harmless first steps — including for everyone who has never done this.

- Test system first. Never a client's live site.
- Never your admin account: a separate user, few permissions, deleted afterwards.
- No real customer data. Test content is enough for everything you want to learn.
- Write down what you connected. In a year nobody will remember otherwise.
- At a client: contracts first, then the plug.

| The measure is what the access would be capable of — not what you are doing with it.

Your turn

No conclusion. Nobody has finished answers right now.

- Who here has let an AI touch a real website?
- And who checked afterwards what it changed?
- Would you give an AI the credentials to a client's site?
- Where exactly is your line — and why is it there?
- What would you never hand over?

| I am deep in this. That is precisely my blind spot — which is why I am asking you.

Sources

You do not have to take my word for any of this. Here is where it comes from.

- The standard itself, and both official WordPress announcements.
- The tools we talked about — Meta Box, Etch, Bricks.
- The video of the one who left, and the site that was built with AI.
- The article on creating versus managing.
- And the two addresses on security and data protection.

| All clickable. If you want to see something live, say so and we will open it.

Sources

Everything that came up tonight, with its address. If you want to see any of it live, say so and we will open it.

Model Context Protocol — the official documentation

The standard itself, including the USB-C analogy I borrowed.

The Connectors API in WordPress 7.0

The core team's announcement: one place for every AI credential.

The MCP Adapter for WordPress

How your own site becomes reachable — the official route.

Meta Box Abilities

The AI creates post types, field groups and field values.

Etch Connector

The agent works inside your open builder tab.

"Goodbye WordPress." — the video

Eighteen minutes of an agency owner explaining why he left.

Activate Media — built with AI, documented publicly

The site from chapter 5. Right click, view source, and you can see what we talked about.

Mathias Bolt Lesniak — "The Website Builder Is Dead, Long Live Content Management"

The article behind the wrong funeral.

Content Security Policy: the report-only mode

Report instead of block — the only sensible way in.

AI and data protection: processing under Article 28 GDPR

What a processing agreement requires, and why consumer plans do not qualify. German.

Anchor Hosting — Austin Ginder's blog

Security written up transparently, including how he works with AI.

Bricks: components as blocks

Design in the builder, edit in the block editor — with guardrails.

PHP-only blocks since WordPress 7.0

Our meetup topic from June, here as a reference.

"WordPress is a prison" — the argument on LinkedIn

This exact argument, in public, this week. The post is one side; the comments are the other, and the better half.

This presentation

It goes up with the recap after the evening.
