Christopher Ross
thisismyurl.com

EPISODE 4 · SITES I'VE NEVER SEEN

The headless WordPress four-question test

FOR	anyone evaluating whether headless WordPress is the right answer for their site
FROM	*Sites I've Never Seen*, Episode 4 — "Headless WordPress in 2026: When It Helps and When It Hurts"
TIME REQUIRED	one meeting, before any contract is signed

> The idea behind this four-question test is Owen's rule from the episode: ***if the team can't name the front-end framework they're committing to before the meeting ends, they're not ready for headless.*** Symptom-led headless decisions are the most expensive kind of decision you can make in WordPress in 2026. The cost runs about two years and the back half of an editorial team's morale.

Four questions, asked in order. All four need crisp answers before the project signs.

Question 1 — What is the bottleneck, in one sentence?

Not "the site is slow." Not "we need to modernize." Not "our developers want React."

The answer is one sentence that names a specific technical bottleneck and the layer where it lives:

- *"Our category templates have twelve server-side queries that can't be cached because each user sees a different feed. That's a bottleneck on the front end."*
- *"Our checkout flow has a 600ms time-to-interactive that's hurting conversion, and we've optimized server-side as far as we can."*
- *"Our editorial team needs to publish to web, mobile app, and signage simultaneously from one CMS, and our current theme can't render to three targets."*

Symptom answers ("the site is slow," "we want to be modern") fail this question. The right next step on a symptom is the thirty-minute audit (https://thisismyurl.com/downloads/30-minute-audit/) from episode 3 (https://thisismyurl.com/podcast/enterprise-wordpress/), not a headless project.

Question 2 — Which framework are we committing to, and who owns it after launch?

The answer names a specific framework AND the person or team that will maintain it after the launch agency goes home:

- *"Next.js with Faust.js. Our internal dev team owns it after launch — we have two React engineers who already work on the marketing app."*
- *"Astro with WPGraphQL. Our marketing agency owns front-end maintenance for the first year; we'll transition it in-house at month 12."*
- *"Bespoke React front-end. We have a five-person front-end team and headless WordPress is their primary platform."*

If the answer is "we'll figure that out during implementation," the project is not ready. Front-end framework selection is the single decision that determines whether the project is maintainable or a six-month rewrite-then-abandon.

The three flavors that dominate in 2026:

STACK	BEST FOR	TRADE-OFF
Faust.js + Next.js (WP Engine)	Fastest path from "we have WordPress" to "we have headless"	Tied to Next render model + WP Engine hosting recommendations
WPGraphQL + Astro / Eleventy	Content-driven sites with steady publishing cadence	Harder to set up dynamic personalisation
WPGraphQL + bespoke React/Vue/Svelte	Maximum flexibility	Maximum maintenance; requires dedicated front-end team

If anyone in the meeting is pitching Frontity, the proposal is out of date — Frontity has been deprecated since 2022.

Question 3 — What is the editorial workflow on day one of launch?

The answer is a concrete description of what an editor does, in what order, to publish a post on the new system:

- Where do they write the draft?
- How do they preview it as it will render to readers?
- Where do they upload images and how do those flow to the front-end?
- How do they handle scheduled publishing?
- What happens to existing scheduled posts and revisions?

The preview problem. In a standard WordPress site, the preview button shows the editor the post as it will render to a reader. In a headless site, preview lives in a separate front-end environment that has to be wired up deliberately. Until it's done — and it almost always gets deferred to a phase two that takes six months — the editors are working blind. Find out when the preview pipeline will work, before the project signs.

The page-builder problem. If the site uses Elementor, Beaver Builder, Divi, or any custom Gutenberg blocks that render via PHP, those rendering paths break in headless. Each one

needs a corresponding front-end component. Find out which blocks are in use and who's building the front-end equivalents.

Question 4 — What does the schema and canonical strategy look like for the new front-end?

The answer names the SEO consequences and the team that owns them:

- Who's rendering structured data (`schema.org` JSON-LD) in the new front-end?
- Who's handling canonical tags, meta descriptions, OpenGraph, Twitter cards?
- How does the sitemap get generated and submitted?
- What happens to existing redirects from Yoast / Rank Math?

The two-weeks-before-launch failure mode. Most teams discover the schema problem when somebody runs the rich-results test on the staging URL and watches half the markup disappear. By that point nobody has the bandwidth to rebuild the structured-data layer, so the launch ships with broken schema and the rankings move accordingly.

None of the WordPress SEO plugins (Yoast, Rank Math, SEOPress) render their schema in headless without explicit additional work. That work needs to be scoped before the project signs, not after.

Decision rule

If all four questions have crisp answers, headless is probably the right call for the site.

If any of them are hedged, the right next step isn't headless. It's the conversation that figures out the missing answer — which is often a much smaller engagement than the full headless project.

A bad headless decision costs about two years and the back half of an editorial team's morale. The front end gets faster. The content team gets slower. And content teams beat front-end speed for revenue impact every time.

Built to go with episode 4 (https://thisismyurl.com/podcast/headless-wordpress/) of Sites I've Never Seen, and the source essay Headless WordPress in 2026: When It Helps and When It Hurts (https://thisismyurl.com/headless-wordpress/).

LISTEN TO THE EPISODE

Headless WordPress in 2026: When It Helps and When It Hurts

Built to go with episode four of *Sites I've Never Seen*. The episode is where the argument lands; this is what you take into the next meeting.

Listen to the episode **Read the source essay** **Send me a note**