

- **Git-based Editing:** The simplest “CMS” is just editing the Markdown/HTML files directly. Since you only do major updates a couple times a year, you might be comfortable doing this in a code editor or via GitHub’s web editor. If so, you may not need any admin UI at all. You could structure your content as Markdown files for each project (with frontmatter for metadata) so it’s easy to add/edit. Many static generators support data files (JSON/YAML) or collections of Markdown for content – making it quite straightforward to update once you know the format.
- **Other Flat-File CMS:** Since you’re familiar with Kirby (used on kgrz.design), you know the convenience of flat-file CMS. If you wanted something similar for kev.studio, you could consider a lighter flat-file CMS like **Elefant, Grav, or even Kirby itself**. However, using those would mean not using GH Pages (you’d need a PHP server if it’s not static). Another approach: use Kirby to periodically generate a static export for kev.studio (Kirby has a plugin for static site export). But this might be circuitous given static site generators are purpose-built for this. Overall, for maintainability, sticking to a pure static workflow (plus optional Netlify CMS) is likely easiest and has fewer moving parts.
- **CloudCannon or Forestry:** These are services that provide a UI to edit static site content (CloudCannon is commercial, Forestry was an editor that tied to Git – it’s now retired/absorbed into other tools). These can be handy if multiple people edit the site or if you really want a CMS dashboard. In your case, they might be overkill, but it’s good to know such options exist. CloudCannon, for example, allows in-browser editing and visualizing of Jekyll/Ellevy sites, acting as a lightweight CMS layer.

In summary, you can absolutely run a static site with *no* CMS and just do code commits for changes (this keeps things ultra-simple and you have full control). But if you anticipate wanting to tweak text or add projects without pulling out a code editor, Netlify CMS is a fitting solution – it’s free and “developer-friendly” since it just works with Git. The integration can be done gradually (you can always add it later once the site is up).

[Optional]: Since you mentioned a separate Kirby site (kgrz.design) you want to keep, ensure your new static site cleanly links to or distinguishes that content. For example, if kgrz.design is for your *design journal or blog*, and kev.studio is portfolio, provide cross-links (“See my design research at kgrz.design”) and keep the branding consistent but not confusing. Maintaining two sites is fine; just make their purposes clear to users.

5. Cargo vs. Custom Static Site – Pros & Cons

Finally, to decide whether to stay on Cargo or go fully custom, consider the following comparison:

- **Performance:** A custom static site will be extremely optimized – you ship only the code you write, with no unnecessary scripts. This often translates to faster load times, especially on mobile, compared to generic site builders. Cargo is reasonably fast for simple sites, but any platform adds some overhead (analytics scripts, built-in libraries, etc.). With static hosting, you can push your Google PageSpeed Insights scores very high (90+ easily) by fine-tuning everything. Also, Cargo’s optional mobile view is essentially a separate simplified version of your site; in a custom build you can have one responsive design, avoiding any clunky transitions. Cargo’s own docs note that their “*mobile optimized design*” doesn’t reflect certain custom styles ²⁵, implying some performance/UX