



picasso

long-term plan · issue #85

picasso on its own domain

Where `site/` lives, how it updates itself on every push to `main`, and what the one official link is for LinkedIn, the README and the GitHub About box. No server, no runtime: a static folder, a Porkbun domain, and the Porkbun Static Hosting plan already paid for.



01 the answer

decided · 12 october 2026

Now: GitHub Pages, free. A workflow in picasso publishes `site/` to `h0ffmann.github.io/picasso` on every push to `main` that touches it. No account changes beyond turning Pages on, no DNS work, no cost. This PDF is linked from the site's main page.

Next: your own domain. Point the Porkbun domain at the same Pages site (section 06): the deploys stay exactly as they are, and the Porkbun hosting plan can lapse at renewal. Porkbun Static Hosting with GitHub Connect (section 05) stays the alternative if you would rather keep the files at Porkbun.

The official link is `https://h0ffmann.github.io/picasso/` until the domain is pointed at it, then `https://<your-domain>` (GitHub redirects the old address). It goes in the GitHub About box (website field), the top of the README, `site/README.md`, and LinkedIn.

02 what the site needs from a host

`site/` is seven files and a drawings folder: `index.html`, `app.js`, `style.css`, `history.json` and six SVGs. There is no build step (`site/README.md`).

Live GitHub API calls. `app.js` fetches `api.github.com` from the visitor's browser (`site/app.js:55`, `:135`), and the logos load from `raw.githubusercontent.com` at pinned commits. GitHub sends CORS headers that allow any origin, so these calls behave the same on every static host. The limit is 60 unauthenticated calls an hour *per visitor IP*, not per host, and the page caches each answer for ten minutes in `localStorage`.

The one thing that goes stale is `history.json`: it is generated by `python3 site/history.py` from a full clone and committed by hand after merges. Any host serves the committed file; phase 2 below makes the deploy refresh it.

So the host needs only: serve static files over HTTPS on your domain, and take new files without a manual upload.



03 the options

option	cost	how a change goes live	live api calls	dns work
Porkbun Static Hosting, drag and drop	already paid: Starter \$30/yr (2 GB), Pro \$60/yr	manual : zip <code>site/</code> and upload after every change	work	none: Porkbun points the domain itself
Porkbun Static Hosting + GitHub Connect on a <code>site-pages</code> branch	already paid; GitHub Connect is on every paid plan	every push to <code>main</code> touching <code>site/</code> , through the workflow	work	none
GitHub Pages, custom domain, DNS at Porkbun	free for a public repo	every push , from <code>site/actions/deploy-pages</code> directly through	work	4 A + 4 AAAA records, a <code>www</code> CNAME, domain verification
Cloudflare Pages	free	every push , Cloudflare's GitHub app, output dir <code>site</code>	work	heavy : the apex needs the nameservers moved to Cloudflare

Why not drag and drop: every merge would need you to remember an upload, and the site would drift from `main` the first time you forget. That defeats a page whose whole content is "the repo's history".

Why not Cloudflare: it is good, but it only earns its keep if you move the domain's DNS to Cloudflare, which is more change than this site needs. Revisit it if you ever want a CDN in front of several sites.

Porkbun vs GitHub Pages is close. Porkbun wins today because it is paid for, the domain and the certificate are managed in one place, and no DNS records need typing. GitHub Pages wins on cost from the next renewal and on documentation: GitHub publishes exactly how a custom domain and HTTPS work, while Porkbun's GitHub Connect article names only a repository and a branch.

04 the open question about porkbun

check first

Porkbun's guide (kb.porkbun.com/article/145) lets you choose a repository and a branch, and says commits to it update the site. It does not say whether a *subfolder* can be the site root. `picasso` keeps the site in `site/`, beside the labs, so the plan does not rely on that: the `site-pages` branch carries `index.html` at its root. If Porkbun turns out to accept a folder, the branch can go and GitHub Connect can follow `main` with folder `site`.

If GitHub Connect cannot follow a non-default branch, or does not update on push, switch to the GitHub Pages fallback (section 06). Nothing in phase 1 is wasted: the domain, the link and the README changes are the same.

05 your steps: go live on porkbun

An agent does none of these: they change your Porkbun account, GitHub settings and DNS. The only repo change, the workflow, comes as a PR for you to merge.

- 1 **Merge the publish PR** (opened from issue #85 once you say go). It adds `.github/workflows/site-publish.yml`: on a push to `main` touching `site/**`, it runs `site/check.py`, copies `site/` to the root of the `site-pages` branch and pushes it. Run it once by hand from the Actions tab so the branch exists.
- 2 **Open Static Hosting**. porkbun.com → Account → Domain Management → your domain → the Static Hosting icon under "Website".
- 3 **Connect GitHub**. In "GitHub Connect" click *connect*. On GitHub choose *Only select repositories* → `hoffmann/picasso` → Install, then confirm with your password or 2FA and click Verify.
- 4 **Pick what it serves**. Back on Porkbun: Repository `picasso`, Branch `site-pages`.
- 5 **Check the domain points at hosting**. In the domain's DNS records, remove any leftover parking record (the default ALIAS/CNAME to Porkbun's parked page) only if Static Hosting says the domain is not connected; Static Hosting normally writes its own records.
- 6 **Check HTTPS**. The Static Hosting page shows the SSL certificate; wait until it is issued, then open `https://<your-domain>`.
- 7 **Check the page**. The history renders, a drawing appears, the ouroboric-dev standings load (they come from your browser, so a "rate limit" note means you reloaded a lot, not that hosting is broken), and both logos show.
- 8 **Prove the loop**. Merge any small change under `site/`; within a few minutes it shows on the domain with no upload.
- 9 **Publish the link**. GitHub → `picasso` → About (gear) → Website: `https://<your-domain>`. Then LinkedIn. The README and `site/README.md` lines come as a PR.

06 fallback: github pages

- 1 **Pages from Actions** (done with the first deploy). `picasso` → Settings → Pages → Source: *GitHub Actions*; `.github/workflows/pages.yml` uploads `site/` with `actions/upload-pages-artifact` and `actions/deploy-pages`: no extra branch.
- 2 **Verify the domain first** (stops someone else claiming it): github.com → your Settings → Pages → Add a domain, then add the TXT record it shows in Porkbun DNS.
- 3 **Porkbun DNS**. Delete the default parking ALIAS/CNAME. Add four A records on the apex (host blank): `185.199.108.153`, `.109.153`, `.110.153`, `.111.153`; four AAAA: `2606:50c0:8000::153` to `8003::153`; and `www` CNAME `hoffmann.github.io`. No wildcard record.

- 4 **Custom domain.** Settings → Pages → Custom domain: your domain → Save; when the DNS check passes, tick *Enforce HTTPS*.
- 5 **Turn off Porkbun hosting at renewal** so you stop paying for both.

07 the long run

phase 0 • plan

now

- Issue #85 and this PDF; the site on GitHub Pages.
- You name the domain, and whether the site is the apex or a subdomain such as `picasso.<domain>`.

phase 1 • live

one pr + your steps

- Now: `pages.yml` publishes `site/` to GitHub Pages.
- Then: your domain on it, section 06, done by you.
- The link in the README, `site/README.md`, About, LinkedIn. Issue #85 closes.

phase 2 • never stale

one pr

- The publish job checks out with full history and runs `site/history.py` before copying, so the history updates on every deploy instead of after a manual commit.
- It then runs on every push to `main`, not only on `site/` changes, since every merge is new history.
- `site/check.py` stays the gate: no deploy without both logos and the credit.

phase 3 • robust

when traffic grows

- A scheduled job writes `standings.json` with an Actions token; the page reads it first and calls the API live only when it is older than a day. Visitors stop sharing the 60/hour limit.
- A weekly check that the domain answers 200 with a certificate valid for 21+ days, alongside the existing `profile-ping`.

phase 4 • findable

small prs

- Open Graph and Twitter card tags using the existing social preview image, a favicon from the logo, and a canonical URL.
- The domain in `.zenodo.json` as a related identifier, so the citation points at the site too (`CITATION.cff` regenerates from it).

yearly • housekeeping

at renewal

- Renew the domain (Porkbun, at its TLD's rate) and the hosting (\$30/yr Starter is plenty: the site is under 1 MB).
- Re-decide Porkbun vs Pages then: if GitHub Connect has been flaky, move; the switch is section 06.
- Bump the pinned logo commits when the projects change their logos.

08 cost

item	porkbun path	github pages path
domain	already owned; renews at Porkbun's rate for its TLD on either path	
hosting	Starter \$30/yr (already paid this year)	\$0, hosting lapses at renewal
certificate	included	included
CI minutes	a few seconds per deploy; free on a public repository	

09 risks

risk	what happens	answer
GitHub Connect cannot follow <code>site-pages</code>	no automatic deploys	section 06, same day
Porkbun GitHub app sees the repo	read access to a repo that is public anyway	grant <i>only select repositories</i>
Standings hit the rate limit	"try again in a few minutes" on heavy reloaders	phase 3 snapshot
The domain lapses	the official link dies everywhere at once	auto-renew on at Porkbun
A wildcard or dangling DNS record	someone else serves pages on your domain	no wildcards; verify the domain on GitHub if Pages is used

Sources (fetched 12 October 2026):

- porkbun.com/products/webhosting/staticHosting: Starter \$30/yr, Pro \$60/yr, GitHub Connect on paid plans, free SSL.
- kb.porkbun.com/article/145-how-to-connect-static-hosting-to-github: the connect steps; repository and branch, no folder setting described.
- docs.github.com: managing a custom domain for your GitHub Pages site: A/AAAA addresses, no wildcards.
- picasso `site/app.js`, `site/README.md`, `site/index.html` at `b703bb9`.

Drawings: original one-line drawings from `site/drawings/`, made for picasso in homage; not reproductions of any Picasso work.