

YOUR SAAS BILL IS RIDICULOUS

\$9 \$19 \$29 \$49 \$89 ...

A Skeptical Guide to Open-Source Tools
That Can Replace Expensive SaaS

Anton Biletskyi-Volokh

SaaS subscriptions pile up so gradually that nobody notices the crime scene until the invoices have formed a committee. One tool is nothing. Five are manageable. Fifteen, some per seat and some suddenly “AI-enhanced,” is how you end up paying real money for software nobody even likes.

Open source can fix part of that. It can also move the bill into hosting, APIs, GPUs, maintenance, and your own attention span. That is the trade.

So this is not a formal book and not a grand theory of software freedom. It is a fast field guide to interesting repos worth knowing about. Some are sharp replacements. Some are cautionary experiments. Some are here because the right reaction is still, very calmly, just pay for the SaaS.

The standard is simple: is this interesting enough that you would plausibly send it to someone with the message, look what I found?

**THE SUBSCRIPTION INVOICE MAY
DISAPPEAR. AND BE REPLACED BY
HOSTING.**

AGENT OUTREACH

Agent-Reach

Agent-Reach is built around a very specific complaint: your shiny AI agent can write code, but the moment you ask it to read X, Reddit, YouTube, GitHub, or a random hostile website, it becomes a philosopher instead of a worker.

The project's pitch is blunt. Give your agent "internet capability" through one install path, a set of maintained backends, and a **doctor** command that tells you what works and what does not. The repository frames this as relief from per-platform pain: paid APIs, broken scraping paths, login walls, cookies, blocked server IPs, and the endless little rituals required to make agents read the modern web. Its installation guide is one of the more revealing pieces of evidence here. Agent-Reach is not just another scraper. It is a wrapper around a messy collection of upstream tools, configuration rules, local cookie handling, and safe-by-default setup boundaries.

Part of the value is obvious. If you would otherwise pay for data access or build an ugly in-house workaround, this gives you a head start. But "zero API fees" is not the same as zero cost. Platform fragility remains. Browser-session dependence remains. Proxies remain. Cookies remain. Somebody still gets to notice when a platform changes and the happy path quietly dies.

Still, there is something respectable here. The project separates check-only installation from system-changing installation, keeps files out of the user workspace, and treats elevated permissions as an actual decision. That is the sort of boring engineering detail which usually matters more than whatever adjective the homepage chose.

If your agent genuinely needs web reach, Agent-Reach may save you from building a bad private copy of the same idea. If it does not, you are mostly installing trouble in advance.

REPLACES

Some paid scraping/search/API access and a lot of custom glue code

COST

Mostly local, but cookies, proxies, breakage and upkeep remain your bill

SETUP

Medium

BEST FOR

Agent-heavy users who need repeatable web reach without building the plumbing from scratch

URL

github.com/Panniantong/Agent-Reach

VERDICT

TRY IT FIRST

OpenHands

OpenHands used to be easy to explain: an open-source attempt to chase Devin. It is now a more ambitious control centre called Agent Canvas, which is a better product idea and a more dangerous sentence.

The repo currently describes itself as a self-hosted developer control centre for coding agents and automations. It can run OpenHands itself, but also Claude Code, Codex, Gemini, or any ACP-compatible agent across local, remote, and cloud backends. There is a real product shift here. This is no longer just “the open one.” It is trying to become the dashboard from which you operate a small colony of software interns.

That makes the replacement claim broader and fuzzier. OpenHands still overlaps with hosted coding-agent products, but it is also drifting into always-on automation and backend orchestration. Useful? Potentially. Simple? Not remotely. The README is honest enough to mention local stacks, Docker, VMs, cloud backends, and the security implications of running an agent with full filesystem access. Which is refreshing. A lot of AI tooling prefers to introduce these matters several invoices later.

This is what you buy with open source here: control, backend flexibility, and the right to build the exact kind of agent environment that will later require its own maintenance plan. That may be a fine trade if you are already serious about agents. It is an absurd trade if you merely wanted help writing unit tests.

OpenHands is interesting because it is no longer pretending to be small. That earns respect. It also earns caution.

REPLACES

Part of the hosted coding-agent workflow, not the full reliability envelope

COST

Model/API spend plus real supervision, runtime, and security responsibility

SETUP

High

BEST FOR

Teams already committed to agent-heavy engineering work

URL

github.com/OpenHands/OpenHands

VERDICT	TRY IT FIRST
---------	--------------

AGENT INFRASTRUCTURE

Composio

Composio is what you get when someone looks at agent integrations and decides the real problem is not intelligence. It is plumbing.

The project gives agents access to a very large catalogue of pre-authenticated toolkits, user-scoped sessions, triggers, and hosted or local access surfaces. In plain English: it is trying to save you from building the same miserable pile of app integrations over and over again. Email here, calendars there, support tools somewhere else, and authentication pain draped over all of it like mould.

That is a real category. And unlike many “AI infrastructure” products, Composio is at least solving something concrete. If your agent needs to act across external systems rather than merely talk about them, a reusable integration substrate is worth far more than another prompt wrapper with tasteful gradients.

The catch is hiding in the quickstart. There is a dashboard. There are API keys. There are provider adapters. There are hosted MCP endpoints. There is a CLI. There is a local tool surface. None of this is inherently bad. It just means the word “open” does not magically convert the integration layer into a maintenance-free public utility. Authentication remains work. Provider boundaries remain work. External apps continue to behave like external apps.

So the honest verdict is not “free alternative to all integration platforms.” It is narrower and better. Composio is a credible way to avoid writing your own agent-action spaghetti. Whether that is cheaper than paying someone else depends on how many times you were planning to repeat the mistake.

REPLACES

A lot of bespoke agent-integration glue, not every workflow platform above it

COST

API keys, provider dependencies, and integration maintenance still apply

SETUP

Medium

BEST FOR

Teams building agents that must take actions across many external systems

URL

github.com/ComposioHQ/composio

VERDICT

TRY IT FIRST

Langfuse

LLM observability is the sort of thing people dismiss right until a prompt quietly breaks revenue, quality, or their remaining patience.

Langfuse is one of the cleaner examples of a serious open-source AI engineering platform. It positions itself around development, monitoring, evaluation, and debugging for AI applications, with both cloud and self-hosted paths. That matters because the product is not pretending to be one tiny utility. It knows perfectly well it wants to sit in the middle of a modern LLM stack and watch everything.

If you are using prompts casually, this will look excessive. If you are shipping anything real, it starts to look uncomfortably sensible. Trace data, evaluations, prompt versions, and failure analysis are not glamorous until you lose track of them. Then they become the difference between engineering and séance.

The reason not to treat Langfuse as automatic salvation is the same reason not to treat any observability tool as salvation. You are still choosing where the complexity sits. Self-hosting a tracing and eval platform does not abolish infrastructure, storage, or organisational discipline. It simply stops billing you for the privilege of needing them. The hosted version exists for a reason, and that reason is that somebody eventually discovers they preferred dashboards to maintaining dashboards.

Still, this is one of the better arguments for open source in the entire booklet. Observability software is exactly where teams resent lock-in, cost creep, and opaque hosted pricing. Langfuse earns its place by being both serious and plainly useful.

REPLACES

A meaningful share of hosted LLM observability and eval tooling

COST

Hosting, storage, trace volume, and workflow discipline still cost money

SETUP

Medium

BEST FOR

Teams already shipping AI products instead of merely discussing them

URL

github.com/langfuse/langfuse

VERDICT

SELF-HOST IT

ABVX Agent Skills

Most “AI productivity” systems eventually produce the same quiet disaster: the model can do many things, so nobody bothers to define how it should do them, what counts as proof, or when it should stop claiming success. ABVX Agent Skills exists for that exact mess.

The repository is a small skillpack for coding agents. Its stated purpose is not to add more raw capability, but to make existing capability less sloppy: smaller diffs, evidence-first debugging, shell-output compaction, verification gates, safer browser and release claims, bounded loops, and explicit handoff discipline. The important sentence in the README is the one that says these skills are not replacements for MCP or CLI tools. MCP connects services. CLI executes work. Skills provide the discipline layer that tells an agent which route to use, which checks are mandatory, and when repeated work deserves a reusable gate instead of another prompt.

This is not replacing a glamorous SaaS line item. It is replacing prompt mush, hidden team rituals, and the recurring expense of letting agents invent their own standard of evidence every time they touch a repo. That problem is less visible than a CRM invoice, but for teams already leaning on coding agents it is often more destructive.

I built this after watching agents repeatedly do the expensive part with complete confidence and then improvise the boring part badly. That context matters. It does not make the project automatically good. It merely explains why it exists.

If a team is not serious enough to care about verification, review, and smaller diffs, this will feel like paperwork. If the team is serious, it may feel like the first adult in the room.

REPLACES

Ad-hoc prompt packs, private agent playbooks and some manual review checklists

COST

Very low runtime cost; the real cost is maintaining standards and using them consistently

SETUP

Low

BEST FOR

Teams already doing serious coding-agent work who need smaller diffs and stronger proof loops

URL

github.com/markoblogo/abvx-agent-skills

VERDICT

TRY IT FIRST

AGENTS.md Generator

There is a particular form of engineering embarrassment in which everyone agrees repo instructions matter and nobody wants to maintain them.

AGENTS.md Generator, or **agentsgen**, attacks exactly that. The project generates, updates, and checks agent-facing repo contracts such as **AGENTS.md**, **RUNBOOK.md**, command manifests, machine-readable endpoints, and related context bundles. The promise is not abstract intelligence. It is drift reduction. Repos change. Commands change. Docs rot. Humans forget. Agents guess. This tool tries to keep the repo contract readable without requiring somebody to lovingly hand-edit the same operational guidance forever.

That is a much more honest product than calling it a replacement for some giant SaaS category. It is closer to a cure for repeated setup and maintenance waste inside AI-ready repos. Useful? Yes. Mainstream? Not remotely. If your repos are not being touched by coding agents, this may look like an answer in search of a problem. Once agents are in the loop, the problem becomes obvious rather quickly.

The pleasant part is that the cost profile is sane. It is a Python CLI with a clear check/init workflow and explicit machine-readable outputs. The less pleasant part is philosophical: if your team cannot agree on what belongs in a repo contract, a generator will not save you from yourselves. It will simply help you fail more consistently.

Still, consistency is not nothing. There are worse uses of open source than automating the dull part of discipline.

REPLACES

Manual repo-instruction drafting and some repeated agent setup work

COST

Low runtime cost; value depends on sustained agent usage across repos

SETUP

Low

BEST FOR

Teams running agents across multiple repos who are tired of context drift

URL

github.com/markoblogo/AGENTS.md_generator

VERDICT	SELF-HOST IT
---------	--------------

ANSWER ENGINE

Vane

The funniest thing about Vane is that it is here partly because the original source corpus was already stale about what the repo was called.

That is not a criticism. It is a useful warning. Categories like “open-source Perplexity alternative” mutate very quickly because everyone wants the same promise: search the web, synthesise the results, and sound like you did not just feed the internet through a leaf blower. Vane lives in that zone. It is a self-hosted answering engine over search and sources, which is already a strong pitch if your default instinct is to pay a polished commercial product to do the same thing.

But this is one of those products where the replacement story gets messy very fast. Search quality rarely comes free. Good answers depend on upstream search, models, ranking, and interface choices. Even when the repo is public, the real costs may sneak back in through APIs, external providers, or sheer tuning effort. Self-hosting the shell does not automatically self-host the result.

So Vane is interesting and legitimate, but not cleanly emancipatory. It is best understood as a way to control more of the stack and possibly lower cost, not as a guaranteed escape from commercial answer products.

Which is still useful. Just not magical.

REPLACES

Some Perplexity-style answer workflows, not the whole polished product experience

COST

Search/model dependencies, tuning effort, and hosting

SETUP

Medium

BEST FOR

Users who care about controlling the answer stack more than they care about convenience

URL

github.com/ItzCrazyKns/Vane

VERDICT	TRY IT FIRST
---------	--------------

MEETING TOOLS

Meetily

Meetily is what happens when somebody looks at meeting software and decides the real sin is not bad summaries. It is sending sensitive conversations somewhere else.

The repo describes a privacy-first AI meeting assistant that runs entirely on your infrastructure. Capture the meeting, transcribe it, summarise it, and keep the whole affair off other people's servers. That is a strong and honest proposition, especially for organisations that really do have confidentiality reasons rather than fashionable paranoia.

It also carries a familiar smell: "enterprise-ready" plus local privacy plus AI features. Those words always deserve a second look because they tend to hide the sentence "you are now responsible for everything." In Meetily's case, the trade is at least straightforward. You get control, local processing, and the chance to avoid cloud retention anxiety. In return, you accept that audio capture, hardware reliability, local model performance, updates, and every awkward edge case are your problem.

There are buyers for whom that is exactly the point. There are also many people who say they want local-first meeting tooling and actually want a button they never have to think about again. Those are different species.

Meetily is credible. It is also a reminder that privacy-first usually translates to operator-first.

REPLACES

Parts of cloud meeting capture and summarisation workflows

COST

Local hardware, setup time, model performance trade-offs, and maintenance

SETUP

Medium

BEST FOR

Privacy-sensitive users who genuinely want the workload that control implies

URL

github.com/Zackriya-Solutions/meetily

VERDICT

TRY IT FIRST

VOICE

Chatterbox TTS

Chatterbox TTS has exactly the kind of discovery value this booklet wants: it is easy to understand in one sentence and instantly makes you recalibrate what “voice SaaS” is supposed to cost.

The project is an open-source text-to-speech and voice-cloning model from Resemble AI. That does not mean it is a frictionless ElevenLabs replacement. It means the category is less closed than many people assume. If your paid plan is mostly buying synthetic voice output and experimentation, a repo like this changes the conversation immediately.

The bill, naturally, does not vanish. Voice quality is still a taste problem as much as a model problem. Local compute still matters. Workflow glue still matters. And the minute you want dependable production output rather than interesting demos, you start paying again in evaluation time, hardware, or operator patience.

Still, it earns a page because it has that useful, shareable quality: not a giant platform, not a vague “AI suite,” just a concrete repo that makes an expensive category feel less inevitable.

REPLACES

Part of ElevenLabs-style TTS and voice generation workflows

COST

Local compute, output QA, and integration work still land on you

SETUP

Medium

BEST FOR

Teams experimenting with voice generation who want control before committing to a hosted stack

URL

github.com/resemble-ai/chatterbox

VERDICT

TRY IT FIRST

Presenton

Presenton is a perfectly sensible answer to a perfectly suspicious market. Yes, it generates presentations. Yes, it calls itself an alternative to Gamma, Canva, Beautiful.ai and their cousins. And yes, that immediately raises the question of whether you are solving a real business need or industrialising beige slides.

To its credit, the project is clear about what it offers: self-hosted deployment, document-to-deck generation, editable exports, model choice, and the ability to bring your own provider or API key. That is better than many AI presentation products, which tend to speak in uplifting tones about storytelling while quietly producing something that looks like a procurement deck from 2019.

Presenton is attractive if you hate SaaS lock-in and want the pipeline under your control. It is less attractive if what you really needed was taste, judgment, and an editor who can say “this slide exists because nobody had the courage to delete it.” Open source can help you generate slides. It cannot guarantee you deserved them.

This is why the verdict lands where it does. For teams with repeatable internal deck production, there is something here. For everyone else, the commercial tools may be the cheaper mistake.

REPLACES

Some prompt-to-presentation SaaS, not editorial judgment or design taste

COST

Model/API costs, operator cleanup, and hosting

SETUP

Medium

BEST FOR

Teams with repeatable deck workflows and a reason to control the stack

URL

github.com/presenton/presenton

VERDICT

JUST PAY FOR THE SAAS

MoneyPrinterTurbo

A product named MoneyPrinterTurbo is at least doing you the courtesy of announcing the genre immediately.

This is an AI short-video generation pipeline. Script, voice, subtitles, rendering, model integrations, the whole fever dream. The repo makes clear that it supports a large range of model providers and that GPU is not strictly required, which is the sort of sentence that should always be read with an eyebrow raised. “Not required” and “pleasant” are not synonyms.

The honest attraction here is obvious. Short-form media SaaS can become absurdly expensive, particularly once automation enters the chat and everyone starts pretending they need five variations of the same idea in vertical format by lunch. Running your own stack sounds appealing.

Then reality shows up. Models cost money. Voice systems cost money. Media generation and rendering cost time. Quality control costs sanity. The project itself effectively admits there is a user-experience barrier by pointing people toward easier online services if they do not want to deal with deployment and usage complexity. That is not a flaw in the repo. It is a clue about the category.

MoneyPrinterTurbo is real, useful, and entirely capable of replacing part of the subscription spend for determined operators. It is also exactly the kind of thing that convinces people they have saved money while quietly moving the bill into infrastructure, cleanup, and regret.

REPLACES

Parts of AI short-video SaaS for people willing to run the workflow themselves

COST

Model/API bills, rendering time, QA labour, and possible GPU appetite

SETUP

Medium

BEST FOR

Operators with repeated video needs and a high tolerance for pipeline maintenance

URL

github.com/harry0703/MoneyPrinterTurbo

VERDICT

JUST PAY FOR THE SAAS

MEDIA UTILITY

yt-dlp

Sometimes the correct open-source alternative is not a platform. It is a command-line tool that has quietly outlived the marketing departments of an entire product category.

yt-dlp downloads video, audio, subtitles and playlists. That is the job. It does the job with extraordinary seriousness. Installation is well documented. The tooling is blunt. The license is permissive. There is almost nothing to romanticise, which is precisely why it works.

This is not replacing a collaboration suite or a strategic media platform. It is replacing the need to pay for half-baked downloader products that exist mainly because some people would rather install a subscription than a binary. Where legal or policy boundaries allow its use, yt-dlp is the kind of software that makes a whole paid niche look faintly embarrassing.

There is very little else to say. No soothing AI promises. No synthetic “workflow revolution.” Just a tool, maintained well enough that people keep relying on it.

You do not always need a platform. Sometimes you need a wrench.

REPLACES

Paid video-downloader utilities far more than whole media systems

COST

Very low, aside from usage context and your own responsibility

SETUP

Low

BEST FOR

People who need reliable media extraction instead of another subscription

URL

github.com/yt-dlp/yt-dlp

VERDICT	SELF-HOST IT
----------------	---------------------

AUTOMATION

Activepieces

Activepieces earns promotion because once you remove the obvious heavyweight from this category, it is the automation repo most likely to make someone say, wait, this is serious.

The project is a self-hostable automation platform with AI, MCP and agent-centric workflow language, openly positioned against Zapier and Make. That overlap is precisely why it was previously treated as a Rabbit Hole next to n8n. But if the goal is discovery value rather than cataloguing the most famous defaults, Activepieces is actually the sharper inclusion. It feels newer, more pointed, and easier to file mentally as a practical find rather than a platform everybody already knows exists.

That does not make it magically cheaper. Workflow ownership still means credentials, breakpoints, upgrade anxiety, and the grim little moment when somebody realises a “simple automation” has quietly become part of business operations. Hosted products charge partly because they absorb some of that mess. Self-hosting brings it home.

Still, this is exactly the sort of repo a useful booklet should surface: not theoretical, not gimmicky, and just unfamiliar enough to reward attention.

REPLACES

A large share of Zapier/Make-style automation for teams willing to own the flows

COST

Hosting, integrations, debugging, and process ownership remain your problem

SETUP

Medium

BEST FOR

Operators who want serious automation without paying forever for every connection

URL

github.com/activepieces/activepieces

VERDICT

TRY IT FIRST

CRM

NocoDB

NocoDB is a useful corrective to the idea that every open tool must become a grand platform. Sometimes a pleasant interface over a database is already enough.

The project positions itself in Airtable territory, but with a database-first posture and easy installation paths through Docker and related setups. That makes it immediately more interesting than the usual no-code theatre. If you already have structured data, NocoDB promises to make it more usable without demanding that you rebuild the company in a proprietary spreadsheet cosplay environment.

That does not mean it is free in the childish sense. The interface may be the visible part, but the hard part is still your data model, your permissions, and the sober realisation that once people enjoy a nice grid they will immediately ask it to behave like an application platform. This is not NocoDB's fault. It is a recurring human condition.

If you know what data you have and why, NocoDB starts to make a lot of sense. If you are looking for a magical system that will compensate for undefined operations, it will not save you. Nothing will.

REPLACES

Airtable-style interfaces, especially when SQL already exists underneath

COST

Data design, hosting, permissions, and operational ownership

SETUP

Medium

BEST FOR

Teams that already have structured data and want a better surface over it

URL

github.com/nocodb/nocodb

VERDICT

TRY IT FIRST

CRM

Twenty

Salesforce is a cathedral. Twenty is what happens when someone decides a CRM should behave more like a developer product than a religion.

Twenty describes itself as “The #1 Open-Source CRM” and, more specifically, as a system that gives technical teams the building blocks for a custom CRM they can “build, ship, and version like the rest of your stack.” That wording matters. This is not a humble contact manager. It is a fairly ambitious attempt to turn CRM into something closer to application infrastructure: objects, fields, views, workflows, agents, a cloud quickstart, a CLI, and a self-hosted path through Docker Compose.

Salesforce, however, is not merely a CRM database with a polite user interface. It is an empire built from procurement, process, consultants, integrations, and institutional habit. Twenty is far more believable as a replacement for teams that want a modern CRM core without buying into that entire civilisation. If you are technical, want a programmable data model, and dislike being trapped inside somebody else’s worldview, Twenty becomes interesting very quickly.

It is still a CRM platform, though. The stack is explicit: PostgreSQL, Redis, BullMQ, NestJS, React, TypeScript, Nx. There is a hosted path if you want convenience and a self-hosted path if you prefer responsibility. That is a respectable product choice. It is also a reminder that you are not escaping complexity. You are moving it onto your side of the fence.

This is a real product, with real ambition, and a real self-hosting story. Just do not pretend that makes it a magic eraser for every Salesforce-shaped problem.

REPLACES

Salesforce only partially; more credibly replaces lighter CRM stacks for technical teams

COST

Cloud fee or self-hosted infra, plus maintenance and migration effort

SETUP

High

BEST FOR

Technical teams that want a programmable CRM instead of a boxed one

URL

github.com/twentyhq/twenty

VERDICT

TRY IT FIRST

CRM

Chatwoot

Support software is one of those categories where the invoice becomes offensive before the product becomes good.

Chatwoot is a modern open-source and self-hosted customer support platform, openly positioned as an alternative to Intercom, Zendesk, and related systems. That is a strong and reasonable claim. An inbox is an inbox for longer than vendors like to admit, and once your support stack becomes expensive enough, the temptation to own it yourself starts looking less ideological and more grown-up.

The project helps its own case by being direct. Scale, flexibility, Docker distribution, control over data, modern support surface. Fine. None of this requires a TED Talk. You need channels, messages, agents, and something your team can actually use.

The complication is that support is not just software. It is process. It is response discipline. It is routing. It is history. It is the deeply human tradition of building one ticket workflow and then immediately ignoring it in Slack. Self-hosting Chatwoot does not protect you from any of that. It merely stops you paying a premium while you do it.

For organisations with enough inbound support to justify a proper system, Chatwoot looks quite sensible. For tiny teams who mainly need a contact form and self-respect, it is probably more platform than they need.

REPLACES

A large share of Intercom/Zendesk-style support inbox tooling

COST

Hosting, updates, channel configuration, and team process

SETUP

Medium

BEST FOR

Teams with real support traffic that no longer want premium-hosted pricing

URL

github.com/chatwoot/chatwoot

VERDICT

TRY IT FIRST

CRM

Listmonk

Email software is an excellent way to discover how many companies are happy to charge rent on a mailing list you already own.

Listmonk is a self-hosted newsletter and mailing-list manager with a refreshingly plain proposition: one binary, PostgreSQL, and a proper admin surface for newsletters and subscriber management. It is not trying to become your growth guru. That immediately makes it more trustworthy than half the market.

The value here is clear. If you have an audience, need owned distribution, and are tired of paying escalating SaaS prices for what is essentially structured email with a database attached, Listmonk is attractive. It looks serious, it is fast, and it does not appear ashamed of being practical.

The cost, naturally, does not vanish. Deliverability remains a real skill. Mail infrastructure remains work. Bad list hygiene remains your fault. And once you stop paying a hosted service, you also stop outsourcing the comforting fiction that someone else was responsible for whether your messages arrived.

Still, this is one of the cleaner replacement stories in the booklet. The trade is easy to understand and, for many operators, worth taking.

REPLACES

A lot of Mailchimp-style newsletter and mailing-list management

COST

Hosting, email infrastructure, deliverability, and list hygiene

SETUP

Medium

BEST FOR

Teams that value owned email distribution and can manage the operational basics

URL

github.com/knadh/listmonk

VERDICT

SELF-HOST IT

ANALYTICS

Plausible Analytics

Plausible is one of the most convincing arguments in this booklet because it fixes a category that genuinely deserves to be fixed.

Analytics should tell you what happened, not force you into an emotional relationship with a dashboard labyrinth and an advertising empire. Plausible is open source, privacy-first, cookie-free, and explicit about being an alternative to Google Analytics. It offers both managed cloud and self-hosted paths, which is exactly the sort of adult behaviour more software categories could imitate.

The pleasant surprise is that this is not one of those cases where open source feels like a noble compromise. For many sites, Plausible is simply a better product. Smaller, clearer, easier to trust, and substantially less irritating. The cloud version is paid, and the README explains why: not because your attention is free money, but because the business model is to sell software rather than harvest users.

That also means self-hosting is optional rather than ideological. If you want the open product but not the maintenance, paying for the hosted version is still consistent with the thesis here. The point is not to eliminate every invoice. The point is to stop paying for bad ones.

Plausible is what a healthy category looks like after somebody removes the nonsense.

REPLACES

Google Analytics for a great many sane use cases

COST

Low if hosted sensibly; self-hosting adds normal ops overhead

SETUP

Low

BEST FOR

Operators who want clean analytics without surveillance bloat

URL

github.com/plausible/analytics

VERDICT

SELF-HOST IT

PHOTO STORAGE

Immich

Everyone loves the idea of self-hosting photos right up to the moment somebody mentions backups.

Immich is a high-performance self-hosted photo and video management system, and it is exactly the sort of product that tempts competent people into heroic optimism. The repo is polished. The feature set is real. The experience is clearly aiming at the Google Photos class of convenience, but with control and privacy placed back in the user's hands.

Then the README does the decent thing and reminds you about the 3-2-1 backup rule. Good. Because that is the whole argument. Self-hosting media is not just about features. It is about accepting that your memories now live inside a system whose reliability is your responsibility. Storage grows. Disks fail. Migrations happen. Mobile uploads behave strangely at the worst possible moment. None of this is a moral issue. It is the work.

If you genuinely care about ownership, Immich looks excellent. If you mostly care about not thinking about the infrastructure behind your family archive, the commercial cloud begins to look less like a scam and more like a service.

This is a very good product. It is also one of the easiest places to confuse admiration with suitability.

REPLACES

Much of the Google Photos-style personal archive experience

COST

Storage, backup discipline, maintenance, and migration responsibility

SETUP

Medium

BEST FOR

People who truly want control over their media archive and mean it

URL

github.com/immich-app/immich

VERDICT

TRY IT FIRST

Rabbit Holes

free-for-dev

This is not a product. It is a directory of free tiers and service offers, which is useful in the same way a map of cheap petrol stations is useful if you were already going to drive somewhere foolish. Its value is obvious: plenty of developers genuinely need a catalogue of free services. The danger is equally obvious: you can lose an afternoon building a theoretical stack of zero-dollar components and still end up paying later in time, migration pain, or arbitrary plan limits. Useful, yes. A replacement for SaaS? Not remotely.

Awesome MCP Servers

This is a catalogue, not a product, and the difference matters. It is a genuinely useful directory if you are exploring the MCP ecosystem and want to understand what kinds of local and cloud tool surfaces exist. It is also the sort of resource that can turn a practical engineering afternoon into six hours of highly educated wandering. Proceed with curiosity and a timer.

Official MCP Servers

The official MCP servers repository is not a shop window. It is a reference shelf. That makes it valuable for developers who want to understand the protocol and inspect canonical examples. It also makes it a poor substitute for production-ready products. The repo itself says as much. Read it for orientation, not because you are hoping the examples will run your company by teatime.

GO GET LOST →

free-for-dev

github.com/ripienaar/free-for-dev

Awesome MCP Servers

github.com/punkpeye/awesome-mcp-servers

Official MCP Servers

github.com/modelcontextprotocol/servers

The Bill Moves

Open source is valuable when it gives you leverage you actually need. It becomes expensive when it hands you chores and persuades you to call them sovereignty.

That is the recurring pattern here. Some tools clearly earn the operational burden. Plausible does. yt-dlp does. Listmonk often does. Others are better treated as experiments first. Twenty lives there. Agent-Reach lives there. Activepieces may live there unless your automation volume is real enough to justify adopting the machinery.

And some categories still deserve a mature reaction: just pay for the SaaS.

Not because open source failed. Because the bill never disappeared in the first place. It moved into hosting, APIs, GPUs, maintenance, migration risk, support discipline, or your own attention.

Repository information for this booklet was reviewed in August 2026.