

Grok Bot and Alternatives

Last modified: 2026-09-13 00:11:34 (PDT)

Coding agents edit a checkout and return a branch or pull request ([coding-agent platforms](#)). *Grok Bot-style* products are a different shape: a named, persistent teammate with its own computer, so it can click through apps and websites the way a person would and keep going after you close your laptop. This chapter reviews [Grok Bot](#) and the alternatives we could verify from primary sources, including [Rakazo](#).

The question for the lab is not “is the demo impressive?” but “what does it add to Claude Code plus [ai-config](#), and what data or approval boundary does it move?”

 A fast-moving, opinionated snapshot

As of August 2026, this category is days to weeks old in public form. Grok Bot launched as an early beta on 11 August 2026 (xAI 2026d). Product names, plan gates, and security wording will drift. The claims below are taken from vendor docs and project READMEs as they stood on 26 August 2026. Re-check those sources before acting on any of them.

1 What This Category Is

A coding agent lives in a repository. A Grok Bot-style teammate lives on a *computer*: a browser, a filesystem, a terminal, and often a graphical desktop, with logins and files that survive from one task to the next.

That architecture buys three things coding agents do not optimize for:

- Work in tools that have no clean API or MCP server, by driving the user interface.
- Unattended runs after the operator’s laptop is closed.
- Named roles that keep memory, routines, and preferences instead of starting from a fresh sandbox every issue.

It also moves a different risk: the agent holds app sessions, not just a git checkout. A login placed on a shared computer is available to every teammate on that computer. Vendor docs for Grok Bot are explicit that **named Bots are not a security boundary** (xAI 2026a).

This chapter is not a catalog of every desktop chat app. It covers products that either ship that “teammate with a computer” shape or are the closest vendor and self-hosted substitutes a lab member would actually reach for.

2 What We Already Use

The lab already runs agents for repository work, documented in the [coding-agents](#) chapter and the [orchestration](#) chapter. The pieces that matter as a yardstick here are:

- **Claude Code** (terminal, IDE, and cloud), with skills, hooks, permissions, and subagents.
- **Portable agent config** in [Morrison-Lab/ai-config](#): skills, hooks, and memories that we install once and expect a session to load ([how the config reaches a machine](#); [customizing an agent](#)).
- **GitHub as the review surface**: work returns as a branch or pull request, with human approval before merge.
- **Cursor** as an interactive IDE, not as a second cloud computer that signs into our apps.

A new teammate product is useful to us only if it does something that stack does not already do well, without putting lab credentials on a computer we do not control.

3 Grok Bot

[Grok Bot](#) (xAI 2026c) is xAI’s early-beta product for named AI teammates. Each Bot is a persistent agent you message like a colleague. The Bots on one account share **one user-scoped cloud computer** with a browser, filesystem, and terminal (xAI 2026f). They can use connectors (shown as Plugins in the app) where those exist, and computer use for apps and websites without a clean API. Several Bots can run in parallel, each with its own *screen* on that shared computer. The screens are work surfaces, not isolation.

It is a separate product from the consumer Grok chatbot. You install a desktop app (macOS or Windows), sign in with a **Cursor** account, and can continue the same Bot from iOS (xAI 2026b). There is no Linux desktop app as of this survey. Eligible plans listed in the getting-started docs are:

- SuperGrok Plus and SuperGrok Heavy
- Cursor Pro+ and Cursor Ultra
- Cursor Teams Standard and Premium

Grok Bot requires cloud data storage. Accounts using Cursor Legacy Privacy Mode must change that setting before it will start (xAI 2026b). Privacy, training opt-out, and account deletion follow [Cursor’s privacy](#) and [security](#) documentation, not a separate Grok-only data plane (xAI 2026a).

Skills, routines, and demonstration

Grok Bot has its own in-app skill and routine objects (xAI 2026e):

- A **skill** is a reusable instruction pack (when to use it, inputs, steps, validation, deliverable, approvals).
- A **routine** tells one Bot when to run a workflow, on a schedule or, where supported, after a Cursor-account event such as a Slack message or GitHub notification.
- **Teach a task**, when the control is visible, records a browser demonstration (up to ten minutes, no microphone) and drafts a skill from it.

A Bot can own up to 50 routines. Background routines can run while the laptop is closed. A test run performs *real* work.

Those objects are Grok Bot’s product surface. They are not a substitute for the lab **ai-config** corpus. Because Grok Bot signs in with a Cursor account, a session can still load that corpus as a **Cursor plugin** (skills, user-global rules, and commands from [.cursor-plugin/plugin.json](#)): the same plugin path this lab’s Cursor sessions already use. That is not Claude Code’s install path. Grok Bot does not load the `~/.claude` symlink install, `CLAUDE.md @imports`, or `Claude hooks/hooks.json`. Cursor Cloud uses `.cursor/hooks.json` instead ([how the config reaches a machine](#)).

Approvals and the shared-computer boundary

The security docs describe operator controls worth taking at face value (xAI 2026a):

- Per-action **Allow once**, **Deny**, and **Always allow**.
- **Auto Review** rules: Require Approval always wins over Always Allow when both match.
- **Take control** of the Agent Computer for passwords, passkeys, two-factor codes, CAPTCHAs, and payments. Do not paste those into chat.
- Local-computer execution is a *separate* switch (default: ask every time) and does not stop the Bot from using its cloud computer.
- Deleting a Bot does not wipe shared-computer files or browser sessions.

The docs’ own least-privilege advice matches lab practice: connect only the tools a workflow needs, start with drafts, and keep sending, publishing, purchasing, deletion, and production changes behind approval.

i Useful to us? Not as the agent-config workflow

Grok Bot is a capable product in a category we do not currently run: unattended work inside signed-in web apps. It does not replace Claude Code. Plugin skills from [Morrison-Lab/ai-config](#) do load when the Cursor plugin is installed (`.cursor-plugin/plugin.json`: skills, user-global rules, commands). The Claude-only install path does **not**: no `~/.claude` symlink, no `CLAUDE.md @imports`, no Claude `hooks.json`. Cursor Cloud uses `.cursor/hooks.json` instead. It has no Linux desktop app. Every Bot on an account shares logins and files. Using it for lab GitHub, email, or data systems would put those sessions on an xAI-managed computer under Cursor's data settings. If someone already has an eligible Cursor or SuperGrok plan, a narrow trial on *non-sensitive* operational chores (public-web research, drafting from files you attach) is reasonable. Do not treat named Bots as isolation, and do not enable local-computer execution unless there is a specific reason.

4 Rakazo

[Rakazo](#) (Rakazo contributors 2026b) is an Apache-2.0, self-hosted platform that describes itself as an “open-source Grok Bot alternative.” The complete core product is in that repository: a web app, an Electron desktop client, and an Expo mobile app talking to one API. You bring model credentials (through the [Pi](#) harness) and choose where the computer runs.

As of 26 August 2026 the GitHub listing showed on the order of 1,300 stars. The README marks the product as beta.

Each bot has a thread, memory, routines, and history. Bots can delegate to peer bots or to short-lived subagents. Integrations can come from Composio or Pipedream Connect, or from a user-installed MCP server, Treg endpoint, or OpenAPI document.

Computers you host

The important design split is the same one Grok Bot has, except Rakazo lets you choose the provider (Rakazo contributors 2026a):

- **Team Computer** (default): bots share browser sessions and tools. Per-bot folders organize work; they are **not** a security boundary.
- **Private Computer**: the whole workspace is that bot's home.
- **Providers**: Docker (local default), E2B, Daytona, and Box for remote desktops, plus a trusted “this machine” desktop provider.

Pi runs in the Rakazo API/worker process, not inside the sandbox. Screen operation needs a model that can use image tool results. The Electron app is a client of the same API; on first launch it asks whether bots should keep using Docker or run on this Mac as you. That local-computer provider is the least isolated option and is not for a public or shared server (Rakazo contributors 2026b).

Self-hosting is a long-running API, a Graphile Worker, Postgres, and a computer provider — not a static site ([self-hosting guide](#)).

💡 Useful to us? Yes, as the open implementation to watch

Rakazo is the closest thing we found to Grok Bot that we could legally inspect, self-host, and point at our own models. That matches lab priorities (local or lab-hosted compute, model choice, no extra vendor computer) better than Grok Bot itself.

It is still a beta product with a real operations burden (Postgres, a worker, sandbox images, encrypted connector secrets). Do not adopt it as a dependency of lab workflow today. If we ever need persistent sandboxed teammates for browser-and-shell chores that should not run on an xAI VM, this is the codebase to evaluate — and its Team-versus-Private computer split is the right mental model even if we never run the app.

5 OpenClaw

[OpenClaw](#) (OpenClaw Foundation 2026a) is a MIT-licensed, self-hosted **gateway** for a personal assistant that meets you in messaging apps. You run one Gateway process on your own machine or a server. The Control UI, CLI, and TUI talk to that Gateway. Channels include WhatsApp, Telegram, Slack, Discord, Google Chat, Signal, iMessage, and others. Companion apps can add voice, canvas, camera, and device-local actions.

It works with hosted and local model providers. Skills, tools, and plugins extend the assistant. The project is built for a **single operator**.

This is not a Grok Bot clone. There is no vendor cloud VM per teammate. The Gateway stays on the host. **Sandboxing is off by default**: tool execution for the main session runs on the host unless you set `agents.defaults.sandbox` (OpenClaw Foundation 2026b). The docs warn that inbound messages are untrusted input, and that you should read the security and sandboxing guides before exposing the Gateway or connecting other users.

i Useful to us? Only with sandboxing, and not as a Grok Bot stand-in

OpenClaw is relevant if we want an always-on assistant reachable from Slack or Telegram on hardware we already operate. That is a messaging-and-gateway problem, not a “bot with its own computer” problem.

Default host-side tool execution is incompatible with how we treat untrusted prompts and shared channels. Anyone trying it should turn sandboxing on (`non-main` or `all`) before connecting a group channel. It does not replace Claude Code, and it will not automatically load `ai-config`.

6 Claude Cowork

[Claude Cowork](#) (Anthropic 2026) uses the same agentic architecture as Claude Code, inside Claude Desktop (with web and mobile access described as rolling out). You describe an outcome; Claude works across local files and connected tools and returns documents, spreadsheets, or organized folders. [Claude in Chrome](#) is the browser path. Sub-agents split parallel workstreams.

Cowork is on paid Claude plans. Connectors, skills, and plugins load from **Customize** on the claude.ai account at session start. Cowork **does not read** the Claude Code CLI's `~/.claude` directory. A skill that exists only in the lab's symlink install must be added again under Customize (Anthropic 2026).

That last point is the lab-config trap: Cowork is the same vendor family as our daily coding agent, but it is not the same install path.

💡 Useful to us? Yes, for knowledge work beside the terminal

Cowork is the lowest-friction vendor option if the task is files, slides, or Chrome automation and the operator already pays for Claude. It is not a persistent cloud computer of its own in the Grok Bot sense, and it will not pick up `ai-config` for free. Use it as a desktop knowledge-work surface, not as a second copy of our coding-agent stack.

7 ChatGPT Work

[ChatGPT Work](#) (OpenAI 2026) is OpenAI's surface for delegating a task with a clear outcome (a brief, deck, analysis, recurring update, or file). Chat remains for short answers. On the desktop app, Work can use local files, apps, and the browser when those tools are available. A **Work locally** versus **Cloud** control chooses whether the run needs your computer or should continue after you close the app (OpenAI 2026).

[Computer Use](#) is a plugin on the ChatGPT desktop app for macOS and Windows with Work and Codex: screen recording and accessibility permissions, plus per-app approval. Codex stays the software-development view in the same desktop app; Work is the everyday-work view of similar agent machinery (OpenAI 2026).

i Useful to us? Only inside an existing ChatGPT workspace

ChatGPT Work is the OpenAI analog of “hand it a job and come back.” Cloud Work is the closest OpenAI match to Grok Bot’s laptop-closed computer. Local Work and Codex Computer Use are closer to Cowork: they operate *your* machine under approvals. None of this loads **ai-config**. For repository work we already document Codex separately ([Codex pull-request reviews](#)). Do not add ChatGPT Work as a lab-wide tool unless a project is already inside that workspace and the data-handling rules for that workspace are acceptable.

8 Comparison

“Relevance to us” is the bottom line and follows from the rows above it.

Dimension	Claude Code + ai-config (ours)	Grok Bot	Rakazo	OpenClaw	Claude Cowork	ChatGPT Work
What it is	Repository coding- agent stack	Named teammates on a vendor cloud computer	Self-hosted Grok Bot-style platform	Self-hosted messaging gateway	Desktop knowledge- work agent	Outcome- oriented ChatGPT agent
Computer	Local checkout or cloud coding en- vironment	One user- scoped xAI VM, shared by all Bots	Docker / E2B / Daytona / Box / this machine	Host Gateway; optional tool sandbox	Local files; Chrome; cloud sessions on some plans	Local or Cloud Work; optional Computer Use
Isolation of named agents	Worktrees, permis- sions, hooks	Not a security boundary	Team Computer is not; Private Computer is	Per-agent workspace; sandbox off by default	Session and folder access you grant	Workspace and plugin permis- sions

Dimension	Claude Code + ai-config (ours)	Grok Bot	Rakazo	OpenClaw	Claude Cowork	ChatGPT Work
Model choice	Whatever that harness is configured for	xAI / Cursor product path	Bring your own via Pi	Hosted or local providers	Anthropic	OpenAI
License / lock-in	Reusable across our repos	Closed; Cursor account required	Apache-2.0	MIT	Anthropic product	OpenAI product
Loads ai-config / <code>~/.claude</code>	Yes, when installed	Cursor plugin yes; <code>~/.claude</code> / Claude hooks no	No	No	No (Customize on <code>claude.ai</code> only)	No
Linux desktop app	Yes (CLI)	No	Web client; Electron is a client of that API	Yes	Claude Desktop	Desktop app; Computer Use is macOS/Windows
Relevance to us	The baseline	Pass for lab workflow; optional narrow trial	Evaluate if we need this shape	Gateway only, sandboxed	Knowledge work beside the terminal	Only if the project already lives in ChatGPT

9 Recommendation

For the lab, as of August 2026:

- **Keep repository work in Claude Code plus ai-config.** Grok Bot can load the Cursor plugin from that repo (skills, user-global rules, and commands in `.cursor-plugin/plugin.json`). It still does not load Claude Code's `~/.claude` symlink install, `CLAUDE.md @imports`, or `Claude hooks.json`; Cursor Cloud uses `.cursor/hooks.json` instead. Rakazo, OpenClaw, Cowork, and ChatGPT Work do not load that config. None of these is a substitute for a reviewable pull request.

- **Do not adopt Grok Bot as lab infrastructure.** It is a vendor cloud computer that shares logins across Bots, requires Cursor data storage, and has no Linux desktop app. A personal trial on non-sensitive chores is the most it should be.
- **Treat Rakazo as the open reference implementation** of the Grok Bot shape. Evaluate it if we need persistent sandboxed teammates under our keys; do not deploy it as production lab tooling while it is beta.
- **Do not confuse OpenClaw with Grok Bot.** It is a self-hosted messaging gateway. Sandbox it before any shared channel, or skip it.
- **Use Claude Cowork** when the job is local files or Chrome and you already pay for Claude, knowing you must re-add skills under Customize.
- **Leave ChatGPT Work** to projects that already use ChatGPT; keep coding in Codex.

None of these replaces the coding-agent platforms in [Coding-Agent Platforms](#). They are teammates-with-computers and knowledge-work desktops. The portable lab config still lives in `ai-config`, and the review surface is still GitHub.

10 Collaborative AI Workspaces: Claude Cowork and Gemini Spark

The 2026 AI ecosystem has expanded beyond reactive chat windows and command-line coding orchestrators into collaborative workspace agents (measured 2026-09-01). These systems operate directly on multi-file workspaces, desktop applications, and cloud productivity suites to automate complex, multi-step analytical and administrative workflows.

Claude Cowork

[Claude Cowork](#) (Anthropic 2026) (also discussed in Section 6) is Anthropic’s desktop-native agent architecture designed to collaborate directly within local folders and desktop application environments:

- **Local Sandboxed Execution:** Cowork runs in a sandboxed virtual environment on the host machine, reading, modifying, and creating local files (spreadsheets, markdown manuscripts, datasets) without requiring manual file uploads or cloud synchronizations.
- **Desktop Application and Browser Interaction:** Beyond static file manipulation, Cowork interfaces with local applications and browser sessions (via [Claude in Chrome](#)) to execute multi-application workflows.
- **Persistent Project Context:** Cowork organizes workspaces into persistent Projects that retain custom instructions, reference documentation, and project memory across sessions.
- **Best Use Cases:** Knowledge work with heavy local file management, local data hygiene, and multi-document manuscript preparation where code and data are managed directly on the workstation.

Gemini Spark

Gemini Spark (Google) represents a cloud-native, always-on agent paradigm deeply integrated into the Google Workspace ecosystem:

- **Asynchronous 24/7 Cloud Agency:** Unlike desktop agents that terminate when the terminal closes or the laptop sleeps, Gemini Spark executes continuously on Google Cloud infrastructure. It continues running scheduled workflows, monitoring inputs, and synthesizing outputs even when the user is disconnected.
- **Google Workspace Ecosystem Integration:** Spark natively interfaces with Gmail, Google Drive, Google Docs, Google Sheets, and Google Calendar to execute automated organizational workflows (e.g., aggregating weekly updates, auditing spreadsheets, and triggering reminder notifications).
- **Proactive Event Triggers:** Spark initiates actions based on calendar milestones, email arrivals, or document modifications rather than waiting for an explicit interactive prompt.
- **Best Use Cases:** Asynchronous operational monitoring, automated team coordination, and cross-document synthesis across cloud-hosted Google Workspace environments.

Related Collaborative Workspace Offerings

Similar collaborative workspace paradigms have emerged across other frontier ecosystems:

- **ChatGPT Work and OpenAI Canvas:** [ChatGPT Work](#) (OpenAI 2026) (see Section 7) and Canvas provide side-by-side document and code editing with inline line-level revisions, interactive targeted edits, and multi-file artifact tracking.
- **Cursor and Google Antigravity Agent Workspaces:** Developer-centric workspace agents providing multi-agent delegation, worktree isolation, and structured planning workflows (such as Conductor extension spec-driven development, [Conductor extension](#)).
- **Notion AI and Microsoft Copilot Studio:** Enterprise knowledge graph agents designed for querying organizational wikis and automating business process workflows.

Comparative Taxonomy of Workspace Agents

Dimension	Claude Cowork	Gemini Spark	OpenAI Canvas / ChatGPT Work	Developer Workspace Agents (Antigravity / Cursor)
Primary Architecture	Desktop-native (sandboxed)	Cloud-native (always-on)	Cloud web interactive	Desktop IDE / CLI workspace
Execution	Active session /	24/7 continuous	Interactive	Session-scoped
Lifetime	local machine	cloud	turn-by-turn	multi-agent

Dimension	Claude Cowork	Gemini Spark	OpenAI Canvas / ChatGPT Work	Developer Workspace Agents (Antigravity / Cursor)
Data Locality	Local files (cloud sessions on some plans)	Google Workspace cloud	OpenAI cloud sandbox	Local repository worktrees
Primary Domain	File and document workflows	Cloud suite coordination	Document and text drafting	Codebase refactoring and review
Key Strength	Local file privacy and tool use	Autonomous background monitoring	Real-time targeted line edits	Deterministic builds, tests, git workflows

Practical Guidance for Research Teams

When selecting a collaborative workspace agent for academic and computational research:

1. **Evaluate Data Confidentiality and Cloud Transmission:** For private research datasets and unpublished manuscripts with confidentiality constraints, desktop-native environments operate on local files under user supervision, though any integration that calls cloud APIs or browser extensions still transmits prompt context to provider infrastructure.
2. **Use Cloud Agents for Asynchronous Administration:** For non-sensitive scheduling, lab meeting summaries, and routine administrative reminders, always-on cloud agents (like Gemini Spark) reduce cognitive overhead by automating background tasks.
3. **Keep Code and Statistical Analyses in Version Control:** While workspace agents are effective for document drafting, statistical pipelines and computational scripts should remain in dedicated git-managed repositories with automated CI validation and code review.

11 Hosted AI Agent Offices: Vestra and Similar

[Vestra](#) (Vestra AI 2026d) sells what it calls an “AI agent office”: a hosted, always-on general-purpose agent that you reach from the tools you already use and that runs on a cloud machine the vendor operates. This section reviews Vestra and four products in the same category (measured 2026-09-09):

- [Manus](#) (Manus AI 2026)
- [Lindy](#) (Lindy 2026b)

- [Zapier Agents](#) (Zapier 2026a)
- [Relevance AI](#) (Relevance AI 2026a)

The category sits between two things this site already covers. [Make](#) and Zapier’s core product are visual automation builders: you draw the workflow, and the AI parts are modules inside it. The collaborative workspace agents in Section 10 (Claude Cowork, Gemini Spark) work on files and documents you own. An agent office is closer to a hired assistant than to either: you describe an outcome in chat, the vendor’s agent plans and executes it on the vendor’s infrastructure, connects to your accounts through OAuth, and keeps a memory of your organization between tasks. Section 5 describes the self-hosted, open-source version of the same idea.

11.1 Vestra

Vestra’s marketing describes two named agents. “Bash” is the executor: you describe “outcomes, not instructions”, it “plans it, in the open, before anything runs”, and it can split work across specialist sub-agents, running up to ten tasks in parallel. “Nami” is the memory layer, which the handbook says builds a company memory by “watching everything: every task Bash finishes, every document, every decision” plus whatever is in the connected tools (Vestra AI 2026a). Each task runs on “a real cloud machine with files, a browser, and a terminal”, and the platform routes each task to a model it picks from several providers unless you pin one.

- **Target user.** The use-case pages address:

- founders
- sales and marketing
- support
- finance
- recruiting
- operations
- research and strategy teams

Press coverage aims it at non-technical domain experts and small teams. The site describes a private beta.

- **Pricing** (Vestra AI 2026b). Four self-serve tiers, all “entire office, all features”:

- Lite at USD 19/month for 2,000 credits on a 4 GB machine
- Pro at USD 97 for 10,000 credits
- Max at USD 197 for 20,000 credits
- Ultra at USD 497 for 50,000 credits on a 32 GB machine

A credit is “the unit of work” and the cost per task varies. The first 1,000 credits are free, and custom engagements with forward-deployed engineers start at USD 4,000/month.

- **Data handling** (Vestra AI 2026c). The privacy policy (effective 2026-09-04) says content is not used for training by default, but task content is sent to whichever model provider handles the task, and one listed model tier (Meta’s Muse Spark 1.3) permits that provider to train on it. OAuth flows and access tokens are handled by a subprocessor, Composio, Inc. Deleting the account removes stored data within 30 days. User-created agents may be indexed by search engines by default, with opt-out by email.
- **Agent and API access.** Chat channels are Slack, Telegram, WhatsApp, iMessage, Gmail, and a desktop overlay on the Mac app. “Abilities” package skills, specialist agents, connectors, and guardrails into installable bundles. We found no public API documentation. The memory store can be exported as plain Markdown.

11.2 Manus

Manus (Manus AI 2026) is the best-known product in the category. Its home page lists a browser operator, a web app builder, “Wide Research” for parallel research tasks, slide and image generators, email (“Mail Manus”) and Slack integrations, a team plan with SSO, and an API for developers. Its pricing page renders client-side and its privacy policy returned only a title when we fetched them, so we could not verify plan prices or the training policy from the vendor (measured 2026-09-09); check `manus.im/pricing` and `trust.manus.im` directly before relying on either.

11.3 Lindy

Lindy (Lindy 2026b) is a “Slack-native teammate”: everyone on a team gets a private assistant in Slack DMs and a shared one in channels. It runs scheduled routines, connects to over 1,000 apps, supports MCP and computer use, and ships with “40+ skills: research, data analysis, decks, dashboards” that a team can extend by saving a completed task as a new skill.

- **Pricing** (Lindy 2026a). Plus at USD 29.99/month for 3,000 credits per user, Pro at USD 99.99 for 15,000, Max at USD 199.99 for 35,000, and an Enterprise tier with HIPAA, a BAA, SSO, and audit logs. Credits do not roll over.
- **Data handling.** “Your data is encrypted, never sold, and never used to train models”, with SOC 2 Type II, GDPR, HIPAA, and PIPEDA compliance claimed. “Nothing irreversible happens without your approval.”
- **Agent and API access.** MCP support is the extension route; the pricing page lists no public API.

11.4 Zapier Agents

Zapier Agents (Zapier 2026a) are “superhuman teammates” layered on Zapier’s 9,000-app integration catalog. You describe the agent, attach knowledge sources (FAQs, documentation, public links), and it acts across those apps, with real-time activity monitoring. Templates cover lead enrichment, content, support, candidate ranking, and expense classification.

- **Pricing** (Zapier 2026b). Agents bill in *activities*, separate from Zap tasks: “any billable action an agent takes: using a trigger, answering from a knowledge source, running an action, web browsing or scraping, a web search, or a message sent via the Chrome extension.” The Free plan includes 400 activities/month with a 10-activity cap per run; paid plans start at about 1,500 activities/month with a 40-activity cap. Zapier’s paid plans start at USD 19.99/month billed annually; nonprofits get 15% off.
- **Data handling.** Neither the Agents page nor the pricing page states how AI features handle data, so read Zapier’s separate privacy and security documents.
- **Agent and API access.** Agents inherit Zapier’s triggers, webhooks, and Chrome extension; this is the most “automation builder” of the five.

11.5 Relevance AI

Relevance AI (Relevance AI 2026a) calls itself “the home of the AI Workforce”. Agents are built three ways: a drag-and-drop canvas, plain-language description, or programmatically through MCP and an API. Each agent “owns one narrow task”, and a workforce nests them with parallel streams, human-in-the-loop approvals, evaluation pass-rate tracking, audit trails, and triggers from CRM events, email, calendar, webhooks, and cron.

- **Pricing.** The public pricing page lists only an Enterprise tier with custom pricing (Relevance AI 2026c); a case study on the agents page quotes USD 0.01 to 0.11 per task.
- **Data handling** (Relevance AI 2026b). The data security policy covers AES-256 encryption in transit, need-to-know staff access, and background checks, but does not address model training on customer data, data residency regions, or certifications.
- **Agent and API access.** The strongest developer surface of the five: API, MCP, webhooks, and cron.

11.6 Useful to us? Not for research work

For an epidemiology lab whose work is code, data, and manuscripts, these products solve a problem we mostly do not have. Their value is in gluing SaaS business tools together (CRM, inbox, calendar, Slack) for people who do not write code. Our work lives in Git repositories, R and Python environments, and Quarto documents, where the coding agents in [the agent catalog](#) and the collaborative workspace agents in Section 10 already reach the files directly.

Three concerns weigh against adopting any of them for lab work:

- **Data.** An agent office works by connecting to your accounts and remembering what it sees. Vestra’s memory layer is designed to watch “every document, every decision”; its policy names one model tier that trains on task content; and Manus’s policy could not be read at all. Human-subjects data, unpublished results, and grant drafts should not flow through a vendor whose training and residency terms we cannot verify. Lindy’s explicit no-training statement and HIPAA/BAA tier is the exception, and it costs enterprise pricing.
- **Reproducibility.** A task that ran on a vendor’s cloud machine with an auto-selected model leaves no script we can re-run. For analysis, that is disqualifying.
- **Cost.** Credit-metered plans at USD 19 to 497 per person per month buy less than a coding-agent subscription for our use, because the expensive part (a hosted VM with a browser) is what we least need.

Where they could earn a place is in lab administration: a scheduled routine that drafts a weekly summary from a shared inbox, triages a calendar, or files reminders. Zapier Agents is the cheapest way to try that, because the free tier covers 400 activities a month and the integration catalog is the largest. If a member wants to try Vestra or Manus, use a throwaway account connected only to non-sensitive tools, and treat Section 5 as the option that keeps the data on our own machine.

12 AI-Enabled Knowledge Workspaces: Notion and Alternatives

Notion and its competitors sell a hosted wiki, database, and document editor with an AI layer on top. This section reviews Notion’s AI features and the closest alternatives from the perspective of a lab that already keeps its notes in git-backed Quarto and Markdown (this site, and the [Benchbook](#) pattern). The question is not whether these products are good — most are polished — but whether any of them beats a plain-text wiki that a coding agent can already read, write, and version. Prices and feature lists below were read from each vendor’s own pages and are volatile (measured 2026-09-09).

Notion

Notion’s AI surface has three layers (Notion Labs 2026b):

- **Notion AI** in the editor: drafting, rewriting, summarizing, and auto-filling database properties.
- **Notion Agent** and **Custom Agents**: a chat agent that runs multi-step tasks across the workspace and connected apps (Slack, Google Drive, GitHub), plus scheduled or triggered agents that run without a user present. Custom Agents were free through 2026-05-03 and now bill by credit, at \$10 per 1,000 credits, on Business and Enterprise plans.

- **AI Meeting Notes** and **Enterprise Search** across connected sources.

For agent access from outside Notion there are two routes. The public REST API exposes pages, databases, blocks, users, and search under internal, OAuth, or personal access tokens (API version 2026-03-11) (Notion Labs 2026c). The Notion MCP server is a remote server hosted by Notion, authorized by OAuth and scoped to what the user can already see, that Claude Code, Cursor, and Codex can connect to (Notion Labs 2026e). It exposes tools for search, fetch, page and database creation and update, comments, file uploads, and custom-agent sessions, rate-limited to 180 requests per minute per user (30 for keyword search), with some tools gated behind Business or Enterprise plans (Notion Labs 2026f).

Pricing: Free, Plus at \$10 per member per month, Business at \$20, Enterprise on request. Full Notion Agent, Custom Agents, and Meeting Notes require Business or above; Free and Plus get a limited AI trial (Notion Labs 2026g). Students and educators at accredited universities get the Plus plan free for a one-member workspace, re-verified annually (Notion Labs 2026d). That is a personal plan: a shared lab workspace with AI would be Business at \$20 per member per month.

Export: pages export as Markdown, databases as CSV plus Markdown subpages, and whole pages or workspaces as HTML or PDF. Comments survive only in HTML; only the current database view exports; “include subpages” for full exports is a Business or Enterprise feature; a workspace export can take up to 30 hours; and workspace owners can disable export entirely (Notion Labs 2026a).

Obsidian

Obsidian is the opposite design: a local Markdown folder with an editor on top. Notes are plain files on disk, so any coding agent can already read and write them with no API at all, and a vault can live in a git repository. Obsidian has no first-party AI features (measured 2026-09-09); the AI plugins are community-maintained (Obsidian 2026c). The official Obsidian CLI (Obsidian 1.12 or later, enabled in Settings, with the app running) exposes search, note creation, tasks, and developer commands, and the documentation names “agentic coding tools” as an intended user (Obsidian 2026a). MCP servers for Obsidian are community projects, several of which wrap that CLI.

Pricing: the app is free for personal and commercial use; a \$50 per user per year commercial license is optional. Sync (\$4 per month annual) and Publish (\$8 per month annual) are optional subscriptions, with a 40% discount for students, faculty, and nonprofit staff (Obsidian 2026b). Export is a non-question: the vault is the export.

Coda, now Superhuman Docs

Coda has been renamed Superhuman Docs; coda.io/pricing now redirects to superhuman.com/plans/docs (Superhuman 2026). The vendor pages are rendered

client-side and could not be read in full for this review, so the numbers below come from a third-party summary (Breeze 2026): Pro at \$15 per Doc Maker per month (\$12 annual), Business at \$40 (\$33 annual), Free and Enterprise unchanged, with a rename date of 2026-07-08. The billing unit is the “Doc Maker”, so viewers and editors of existing docs are free. Treat the product as in transition: a rename and a price increase in one quarter is a lock-in signal in itself.

Google Workspace with Gemini

Gemini is bundled into every Google Workspace plan rather than sold as an add-on: Starter (\$7 per user per month list) has limited Gemini in Gmail, Standard (\$14) and above have Gemini across Docs, Sheets, Meet, and Gemini Notebook (Google 2026c). For universities, Education Fundamentals is free and includes Gemini for Education and Gemini Notebook, though in-app Gemini in Docs and Sheets needs Education Plus (\$6 per user per year); the Google AI Pro for Education add-on is \$15 per user per month on an annual commitment (Google 2026a). Agent access is through the Docs, Sheets, and Drive APIs, which let a program create and edit document structure (Google 2026b); Google’s pricing and API pages reviewed here mention no hosted MCP server for Docs (measured 2026-09-09). Google’s always-on Gemini Spark agent is covered in Section 10. Lock-in is moderate: a Doc exports as a file, but comments, suggestions, and version history stay in Google’s copy. Where the campus already licenses Workspace, the marginal cost to the lab is whatever that contract allows.

Microsoft Loop and OneNote with Copilot

Copilot in Loop offers preset prompts (Create, Brainstorm, Blueprint, Describe) and page summaries inside Loop workspaces (Microsoft 2026b), and Microsoft positions Loop alongside Copilot Pages (canvases for saving AI output) and Copilot Notebooks (grounded research spaces) (Microsoft 2026a). All of it requires a Microsoft 365 Copilot license on top of a qualifying base plan. Copilot Business is \$18 per user per month (annual, reduced from \$21) for small-business plans; the enterprise add-on is listed separately (Microsoft 2026d). Academic Copilot licenses exist for A1, A3, and A5 plans through volume licensing (Microsoft 2026c). None of the Microsoft pages reviewed here offers an MCP server for Loop or OneNote, and neither page describes a Markdown export, which makes this the weakest export story in this table. The .docx route for reviewer edits, described earlier in this chapter, remains the practical bridge to Word users.

Confluence with Roovo

Confluence Cloud is free for up to 10 users; Standard and Premium are per-user and include Roovo (search, chat, and agents) with 25 and 70 Roovo credits per user per month respectively (Atlassian 2026e, 2026d). Credits are pooled at the organization level, reset monthly, and overage billing at \$0.01 per credit starts 2026-12-03 (Atlassian 2026d). Atlassian’s academic

licensing gives qualifying institutions 50% off cloud list price (Atlassian 2026a). The Atlassian Rovo MCP Server is hosted at `mcp.atlassian.com`, authorizes by OAuth 2.1 or API token, and lets Claude, ChatGPT, GitHub Copilot CLI, and Gemini search, summarize, create, and update Confluence pages and Jira work items within the user’s existing permissions (Atlassian 2026c). Export is per page to Word or PDF, and per space to PDF, CSV, HTML, or XML (Atlassian 2026b); there is no Markdown export, so leaving Confluence means converting HTML.

Anytype

Anytype is a local-first, source-available (Any Source Available License 1.0) knowledge base with zero-knowledge encrypted peer-to-peer sync (Any Association 2026b). It has no built-in AI. Instead it ships a Local API on `localhost`, authenticated by an API key generated from a challenge in the desktop app (Any Association 2026d), and an “Agents’ Skill” that lets Claude Code, Cursor, Gemini CLI, or GitHub Copilot run sandboxed JavaScript against that API to read, search, create, and bulk-edit objects (Any Association 2026a). Community MCP servers wrap the same API. Membership is a free tier plus paid storage tiers, with a 50% student discount (Any Association 2026e). Export is Markdown, PDF, HTML, or Anytype’s own Any-Block format, in JSON or Protobuf (Any Association 2026c).

Craft

Craft is a native document editor (macOS, iOS, Windows, web) with a credit-metered AI assistant: 15 credits on Free, 50 per month on Plus (\$6.40 per month annual), with a Team plan at \$50 per month for up to 10 seats (Craft Docs 2026a). It offers a hosted MCP endpoint per connection with user-defined scope, which Claude Desktop, Claude Code, ChatGPT, Cursor, VS Code, and other clients can attach to, and which since version 3.3.5 can create and update documents, not only read them (Craft Docs 2026c). Export is Markdown, `TextBundle`, PDF, and image on every platform, plus `.docx` on macOS, iOS, and web (Craft Docs 2026b). No education discount is listed.

Comparison

Product	AI features	Agent / MCP access	Export and lock-in	Academic pricing	Verdict
Notion	Notion AI, Notion Agent, Custom Agents (credits), meeting notes, enterprise search	Hosted MCP server (OAuth); REST API	Mark-down/CSV/HTML/PDF; comments and views partly lost; export can be disabled by admins	Plus free for student/educator; shared AI workspace is \$20 per member per month	Best hosted option; still a walled garden
Obsidian	None first-party; community plugins	Local files; official CLI; community MCP servers	Vault is plain Markdown on disk	Free app; 40% off Sync/Publish	Closest to what we already do
Superhuman Docs (Coda)	Coda AI, in transition	Packs and API	Per-doc export; rebrand and repricing in 2026	None listed	Avoid until it settles
Google Workspace + Gemini	Gemini in Docs/Sheets/Slides; no Gemini Notebook, Spark agent	Docs/Drive APIs; no hosted MCP documented	File export; comments and history stay behind	Fundamentals free; Plus \$6 per user per year; AI Pro add-on \$15 per month	Use what the campus already licenses
Microsoft Loop / OneNote + Copilot	Copilot prompts and summaries; Copilot Pages and Notebooks	No MCP server documented	No Markdown export documented; weakest export	Academic Copilot via A3/A5 volume licensing	Only if the collaborators live in Teams
Confluence + Rovo	Rovo search, chat, agents (credit-metered, overage from 2026-12)	Hosted MCP server (OAuth 2.1)	Word/PDF per page, HTML/XML per space; no Markdown	50% off cloud list	Fine for Jira shops; poor for prose

Product	AI features	Agent / MCP access	Export and lock-in	Academic pricing	Verdict
Anytype	None built in	Local API with key; agent skill; community MCP	Mark- down/HTML/PDF/ Block; local-first, encrypted	50% student Discount	Interesting, niche
Craft	Credit- metered assistant	Hosted MCP with scoped connections	Mark- down/TextBundle/ PDF/.docx	None listed	Nice editor, small ecosystem

Useful to us?

Mostly no, and for a structural reason rather than a feature gap. Every product above adds AI by putting a model *inside* the vendor’s editor and then, more recently, adding an MCP server so outside agents can reach in. A git-backed Quarto or Markdown wiki inverts that: the agent is already a first-class editor of the files, every change is a reviewable diff, CI renders and spell-checks the result, and the wiki is the export. [Benchbook](#) is the same idea with a written contract for the agent; this site is the same idea with a publishing pipeline. Neither needs an MCP server to let Claude Code read the lab’s notes, and neither can be locked, metered, or renamed by a vendor.

What the hosted products do better, and a plain repository does not:

- **Non-technical collaborators.** Notion, Google Docs, and Loop have editors that a clinician or a first-year student can use without learning git. The .docx round-trip described earlier in this chapter covers review comments, but not live co-editing.
- **Databases with views.** Notion and Coda tables with filters, rollups, and forms have no clean Markdown equivalent; a CSV in a repo is the honest substitute.
- **Meeting capture.** Notion’s AI Meeting Notes and Copilot’s transcription are convenient; the output is text and can be committed afterward.
- **Always-on agents.** Notion Custom Agents, Gemini Spark, and Rovo agents run on a schedule without a machine of ours awake. A GitHub Actions cron running a coding agent does the same for a repository.

Recommendations (measured 2026-09-09):

- Keep lab knowledge in git-backed Markdown or Quarto, and let coding agents edit it directly.
- If a hosted editor is needed for a specific collaborator group, prefer the one the campus already licenses (Google Workspace with Gemini) over a new per-seat subscription, and treat its content as a draft that gets committed.

- Among the standalone products, Obsidian is the only one whose data model is already ours, so a lab member who wants a graphical editor over the same repository should start there.
- Notion is the best of the hosted walled gardens, and the Notion MCP server makes it reachable from Claude Code; but the AI features that distinguish it are Business-plan and credit-metered, and export loses structure, so budget for the exit before entering.
- Re-check prices and plan boundaries before acting on any row of the table; several changed within the last quarter.

References

- Anthropic. 2026. *Claude Cowork Overview*. Documentation. <https://claude.com/docs/cowork/overview>.
- Any Association. 2026a. *Anytype Agents' Skill*. Documentation. <https://doc.anytype.io/anytype/features/integrations/anytype-agents-skill>.
- Any Association. 2026b. *Anytype Desktop (Anytype-Ts)*. Documentation. <https://github.com/anyproto/anytype-ts>.
- Any Association. 2026c. *Anytype Import and Export*. Documentation. <https://doc.anytype.io/anytype/data/import-and-export>.
- Any Association. 2026d. *Anytype Local API*. Documentation. <https://doc.anytype.io/anytype/features/integrations/local-api>.
- Any Association. 2026e. *Anytype Memberships*. Documentation. <https://doc.anytype.io/anytype/resources/memberships>.
- Atlassian. 2026a. *Atlassian Cloud Licensing*. Documentation. <https://www.atlassian.com/licensing/cloud>.
- Atlassian. 2026b. *Export Content to Word, PDF, HTML and XML*. Documentation. <https://support.atlassian.com/confluence-cloud/docs/export-content-to-word-pdf-html-and-xml/>.
- Atlassian. 2026c. *Get Started with the Atlassian Rovo MCP Server*. Documentation. <https://support.atlassian.com/atlassian-rovo-mcp-server/docs/getting-started-with-the-atlassian-remote-mcp-server/>.
- Atlassian. 2026d. *How Rovo Credits Work*. Documentation. <https://support.atlassian.com/rovo/docs/rovo-usage-limits/>.

- Atlassian. 2026e. *Learn about the Features of Confluence Cloud Plans*. Documentation. <https://support.atlassian.com/confluence-cloud/docs/learn-about-confluence-cloud-plans/>.
- Breeze. 2026. *Coda Is Now Superhuman Docs: What Changed for Users*. Documentation. <https://www.breeze.pm/articles/coda-is-now-superhuman-docs>.
- Craft Docs. 2026a. *Craft Pricing*. Documentation. <https://www.craft.do/pricing>.
- Craft Docs. 2026b. *Export to PDF, Word, and Markdown*. Documentation. <https://support.craft.do/en/import-and-export/export/document>.
- Craft Docs. 2026c. *MCP (Craft Help Center)*. Documentation. <https://support.craft.do/en/integrate/mcp>.
- Google. 2026a. *Compare Google Workspace for Education Editions*. Documentation. https://edu.google.com/intl/ALL_us/workspace-for-education/editions/compare-editions/.
- Google. 2026b. *Google Docs API Overview*. Documentation. <https://developers.google.com/workspace/docs/api/how-tos/overview>.
- Google. 2026c. *Google Workspace Pricing*. Documentation. <https://workspace.google.com/pricing>.
- Lindy. 2026a. *Lindy Pricing*. Product website. <https://www.lindy.ai/pricing>.
- Lindy. 2026b. *Lindy: The AI Teammate That Will 3x Your Output*. Product website. <https://www.lindy.ai/>.
- Manus AI. 2026. *Manus: Hands on AI*. Product website. <https://manus.im/>.
- Microsoft. 2026a. *Compare Microsoft Loop, Copilot Pages, and Copilot Notebooks*. Documentation. <https://support.microsoft.com/en-us/microsoft-365-copilot/compare-microsoft-loop-copilot-pages-and-copilot-notebooks>.
- Microsoft. 2026b. *Frequently Asked Questions about Copilot in Loop*. Documentation. <https://support.microsoft.com/en-us/loop/frequently-asked-questions-about-copilot-in-loop>.
- Microsoft. 2026c. *License Options for Microsoft Copilot*. Documentation. <https://learn.microsoft.com/en-us/copilot/microsoft-365/microsoft-365-copilot-licensing>.
- Microsoft. 2026d. *Microsoft 365 Copilot Plans and Pricing*. Documentation. <https://www.microsoft.com/en-us/microsoft-365-copilot/pricing>.

Notion Labs. 2026a. *Export Your Content*. Documentation. <https://www.notion.com/help/export-your-content>.

Notion Labs. 2026b. *Notion AI*. Documentation. <https://www.notion.com/product/ai>.

Notion Labs. 2026c. *Notion API Introduction*. Documentation. <https://developers.notion.com/reference/intro>.

Notion Labs. 2026d. *Notion for Education*. Documentation. <https://www.notion.com/help/notion-for-education>.

Notion Labs. 2026e. *Notion MCP*. Documentation. <https://developers.notion.com/docs/mcp>.

Notion Labs. 2026f. *Notion MCP Supported Tools*. Documentation. <https://developers.notion.com/guides/mcp/mcp-supported-tools>.

Notion Labs. 2026g. *Notion Pricing*. Documentation. <https://www.notion.com/pricing>.

Obsidian. 2026a. *Obsidian CLI*. Documentation. <https://obsidian.md/help/cli>.

Obsidian. 2026b. *Obsidian Pricing*. Documentation. <https://obsidian.md/pricing>.

Obsidian. 2026c. *Obsidian: Sharpen Your Thinking*. Documentation. <https://obsidian.md/>.

OpenAI. 2026. *Get Started with ChatGPT Work*. Documentation. <https://learn.chatgpt.com/docs/get-started-with-work>.

OpenClaw Foundation. 2026a. *OpenClaw*. Software. <https://github.com/openclaw/openclaw>.

OpenClaw Foundation. 2026b. *OpenClaw Sandboxing*. Documentation. <https://docs.openclaw.ai/gateway/sandboxing>.

Rakazo contributors. 2026a. *Rakazo Computer Runtime*. Documentation. <https://github.com/elie222/rakazo/blob/main/docs/computer-runtime.md>.

Rakazo contributors. 2026b. *Rakazo: Open-Source Grok Bot Alternative*. Software. <https://github.com/elie222/rakazo>.

Relevance AI. 2026a. *Relevance AI Agents*. Product website. <https://relevanceai.com/agents>.

Relevance AI. 2026b. *Relevance AI Data Security Policy*. Policy document. <https://relevanceai.com/data-security-policy>.

- Relevance AI. 2026c. *Relevance AI Pricing*. Product website. <https://relevanceai.com/pricing>.
- Superhuman. 2026. *Superhuman Docs Pricing and Plans*. Documentation. <https://superhuman.com/plans/docs>.
- Vestra AI. 2026a. *Vestra Handbook*. Documentation. <https://vestra.ai/handbook>.
- Vestra AI. 2026b. *Vestra Pricing*. Product website. <https://vestra.ai/pricing>.
- Vestra AI. 2026c. *Vestra Privacy Policy*. Policy document, effective 2026-09-04. <https://vestra.ai/policies/privacy>.
- Vestra AI. 2026d. *Vestra: Your AI Agent Office*. Product website. <https://vestra.ai/>.
- xAI. 2026a. *Approvals, Security, and Privacy*. Documentation. <https://docs.x.ai/grok-bot/approvals-security-and-privacy>.
- xAI. 2026b. *Get Started with Grok Bot*. Documentation. <https://docs.x.ai/grok-bot/get-started>.
- xAI. 2026c. *Grok Bot Overview*. Documentation. <https://docs.x.ai/grok-bot/overview>.
- xAI. 2026d. *Introducing Grok Bot*. Product announcement. <https://x.ai/news/introducing-grok-bot>.
- xAI. 2026e. *Skills and Routines*. Documentation. <https://docs.x.ai/grok-bot/skills-routines-and-automations>.
- xAI. 2026f. *Use the Computer and Apps*. Documentation. <https://docs.x.ai/grok-bot/computer-and-apps>.
- Zapier. 2026a. *Zapier Agents*. Product website. <https://zapier.com/agents>.
- Zapier. 2026b. *Zapier Plans and Pricing*. Product website. <https://zapier.com/pricing>.