willow vs

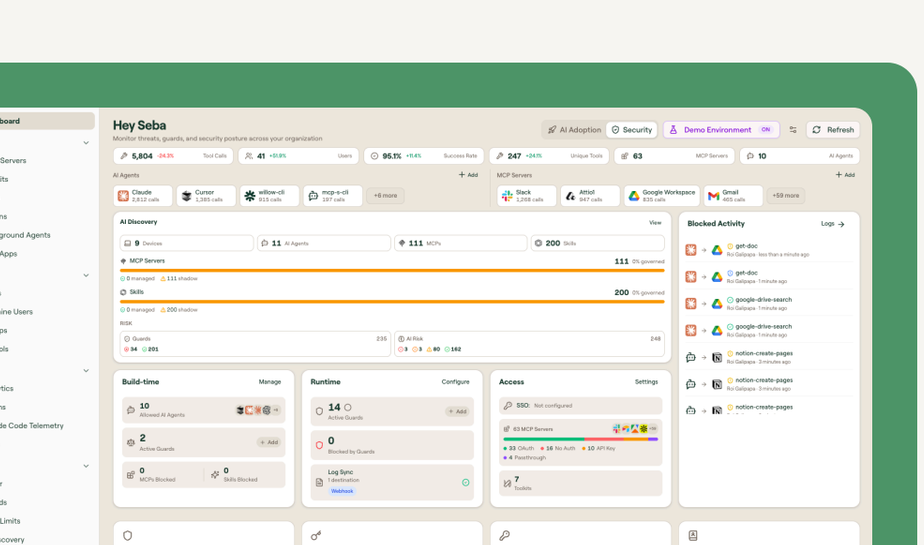
Willow vs. MetaMCP: a governed platform, or a proxy you run yourself

MetaMCP is a free, MIT-licensed MCP proxy that aggregates the MCP servers you already run behind one endpoint. Willow is a managed governance and enablement platform for the whole organization, with a curated catalog, endpoint discovery, and employee self-service. Compare what each one gives you, and what each one asks you to build.

Get Started | Book a Demo

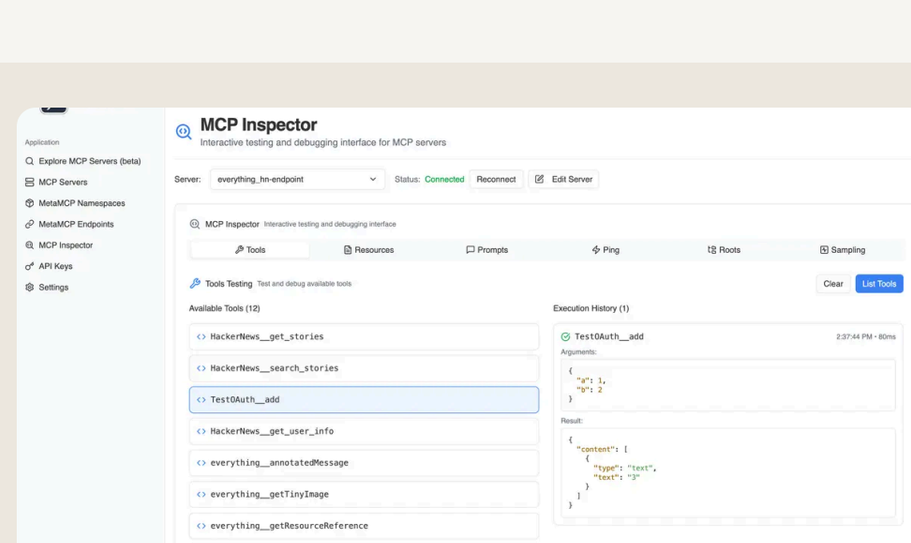
Trusted by

WiX agora papayaglobal innovi



Willow Admin Console

A managed access and enablement control plane for the whole organization. It provides a curated catalog with per-connector risk assessment, employee self-service, machine users, fine-grained per-agent and per-action policy, endpoint discovery of shadow AI, and IT-run identity and lifecycle. It deploys as SaaS, dedicated cloud, on-premises, hybrid, air-gapped, or with EU residency, it holds SOC 2 Type II and GDPR, and it is proven at Wix across roughly 5,000 weekly active users.



MetaMCP Management Interface

In its own words, "a MCP proxy that lets you dynamically aggregate MCP servers into a unified MCP server, and apply middlewares." You group the MCP servers you already run into namespaces, expose each namespace as an endpoint over SSE, Streamable HTTP, or OpenAPI, protect it with an API key or OAuth, and optionally filter and rename the tools inside it. It is MIT licensed, free, and self-hosted with Docker Compose.
Its README describes who it is for: "Generally developers can use MetaMCP as infrastructure to host dynamically composed MCP servers through a unified endpoint, and build agents on top of it."

It is free, and free is a real answer

What it does that Willow either does not do, or does not do better.

MetaMCP wins

It is free, and free is a real answer

MIT licensed, no paid tier, no commercial offering. For a team willing to build the rest, the software costs nothing, and there is no vendor to negotiate with, no procurement cycle, and no lock-in. Willow is a commercial platform and does not compete on that.

MetaMCP wins

It runs entirely on your infrastructure

The recommended install is Docker Compose, and nothing leaves your environment. Because it is a container, it runs in a disconnected environment as readily as a connected one, so a sovereignty or air-gap requirement does not rule it out.

MetaMCP wins

Its tool-level control is unusually granular

Enable or disable at the individual tool level, override tool names, titles, and descriptions per namespace, and attach custom MCP annotations, with overrides merged into whatever the upstream server returns so provider metadata survives. If three servers expose ninety tools and your agent needs eleven of them with clearer names, MetaMCP solves that cleanly.

MetaMCP wins

It has MCP rate limiting, and Willow does not

MetaMCP can cap the requests an endpoint accepts in a time window, either across all clients sharing one counter or per individual user. If you need to throttle request volume at the gateway, MetaMCP does it and Willow does not.

MetaMCP wins

It has real community adoption

More than 2,500 stars, hundreds of forks, and a steady flow of community pull requests. This is a well-regarded tool that a lot of engineers have chosen deliberately.

MetaMCP wins

It plugs in wherever MCP already works

MetaMCP is itself an MCP server, so it drops into any MCP client without special support. Each namespace can be exposed over SSE, Streamable HTTP, or as an OpenAPI endpoint for clients like Open WebUI, which makes it easy to put in front of tooling you have already built.

Where Willow is the right choice

What it does that MetaMCP does not set out to do.

Willow wins

You get a catalog instead of a supply problem

MetaMCP's unit of configuration is a server you provide, written out as a command and its arguments. That is the correct design for a proxy, and it means connector supply is entirely yours: every server in your namespace is one you found, vetted, hosted, and now maintain. Willow ships a curated catalog with per-connector authentication and a risk assessment on each entry, and converts any REST API into a governed MCP when the connector you need does not exist. The difference is not a feature, it is who does the sourcing.

Willow wins

You see the AI nobody registered

MetaMCP governs precisely what you put into it, which is what a proxy should do and also its boundary: it cannot tell you what it does not know about. Willow's endpoint discovery finds unmanaged MCP servers, skills, AI agents, devices, and OAuth flows on employee machines, including local servers and skill files that never cross the network, and scores each discovered MCP server and skill 0 to 10 with a risk band, so shadow findings are triaged rather than just listed. This extends to vibe app monitoring, tracking use of app builders like Lovable and Base44 and flagging when those apps are wired into internal systems.

Willow wins

Governance is built rather than assembled

MetaMCP's middleware layer is the right architectural home for policy, and MetaMCP says what is in it today: one built-in example, "Filter inactive tools," with "tool logging, error traces, validation, scanning" listed as future ideas, and their documentation calling the middleware system "still under active development." Willow ships request and response guards with PII masking, per-user and per-agent policy, and approval workflows today. Willow's guards block plaintext injection but not encoded, skill-based, or image-based attempts, so the difference here is built-in guards against middleware you write, rather than a solved prompt-injection problem.

Willow wins

Employees can use it without you

MetaMCP is administered by whoever runs it, and clients are connected by pasting endpoint URLs into configuration files. Willow gives employees a Connect Panel and a self-service catalog with IT approval workflows behind it, which is what makes a rollout reach the roughly 5,000 weekly active users Wix runs it across, most of whom are not engineers.

Willow wins

Configuration is reviewed code

Willow manages MCP servers, skills, and plugins as code in GitHub, with pull-request review and two-way sync, so a change to who can reach what is a diff someone approved. MetaMCP is configured through its web interface and database, with environment-variable references keeping secrets out of config. Headless admin API access is the first item on its roadmap, so programmatic configuration is coming rather than current.

Willow wins

Compliance arrives with the product

Willow holds SOC 2 Type II and GDPR, with EU residency available. MetaMCP holds no certifications and does not claim any, which is the ordinary and correct position for a self-hosted open-source project: when you run it, the compliance boundary is yours, and the evidence you show an auditor is evidence you produce.

What you take on when you run it yourself

MetaMCP is maintained by its author, who has been open about what that means. The repository's status note reads, in his words, "development on this project will be slow, but the project is still maintained." He has said he would rather keep merging community pull requests than ship new features that introduce bugs he cannot resolve, and community pull requests were merged as recently as June 2026.
Operationally, that leaves three paths. The most recent tagged release is v2.4.22 from December 2025, and the published image tagged latest resolves to that same build, so the recommended Docker Compose install gives you that code. Newer work sits on a branch named ai-dev, which he describes as "the forward ongoing dev branch" with the note "Please test before you build the image based on this branch." There is also a community-maintained fork, which he links to himself.
All three are legitimate choices that teams make every day. None of them is a vendor telling you which version is supported, so the upgrade path is yours to own. That is the real trade for the zero licence cost. With Willow the same question has a shorter answer: a company on the other end, a support agreement, a SOC 2 Type II report, and a version that is current because keeping it current is someone's job.

The Bottom Line

MetaMCP is open-source for aggregating existing MCP servers with namespace management. Willow provides shadow AI detection, unified build & runtime guards, infrastructure-as-code governance, and a plugin marketplace at enterprise scale.

MetaMCP is the better fit when

You are a technical team that already runs its own MCP servers and wants them composed behind one namespaced endpoint, for free, on your own infrastructure, with no vendor in the path. It is genuinely good at that job, its tool-level remixing is more granular than most commercial products offer, and it has MCP rate limiting that Willow does not.

Willow is the better fit when

Governing MCP access is one part of rolling AI out to an organization rather than a piece of plumbing for one team. It ships a curated catalog with risk scoring instead of asking you to supply every server, converts any REST API into a governed MCP, discovers the AI your employees run outside any gateway, gives non-technical employees self-service, manages configuration as reviewed code in GitHub, and arrives with SOC 2 Type II and a vendor operating it.

The real question

It is not which is better. These are different weight classes doing different jobs. MetaMCP is infrastructure you assemble and operate. Willow is a platform you are given. Pick on whether you want to own that work or receive it.

What each one is

	Willow	MetaMCP
What it is	A managed access and enablement control plane for the organization	An MCP proxy that aggregates the MCP servers you already run into one endpoint
Who it is built for	The organization, including employees who will never open a terminal	Developers, in its own words, using it as infrastructure
Licence and cost	Commercial, managed	MIT, free
Who operates it	Willow	You, with Docker Compose or Kubernetes, on your own infrastructure
Who stands behind it	Willow, with support and SOC 2 Type II	An open-source project maintained by its author, with a Discord community and community contributors

Deployment and infrastructure

Simple to stand up is not the same as operated.

Willow approach

- Deploy as SaaS, dedicated cloud, on-premises on AWS, GCP, or Azure, hybrid, or fully air-gapped.
- EU data residency is available.
- Every option is delivered as a managed service, and it typically goes live in about ten days.

MetaMCP approach

- Docker Compose is the recommended path, with Dev Containers and local development documented as alternatives.
- It runs on your own hardware, including disconnected.
- You size it, patch it, scale it, back it up, and upgrade it.

Bottom line

The difference here is not where each one can run, but who runs it. MetaMCP goes anywhere a container goes, including into a disconnected environment. What it does not come with is anyone operating it.

Security and access control

Coherent for a team, against built for an organization.

Willow approach

- Enterprise identity across OAuth 2.0, OIDC, SAML, and JWT, with SSO to Okta, Entra, JumpCloud, and Google.
- SCIM lifecycle, so access ends when employment does.
- Admin roles, plus per-user and per-agent policy of allow, needs-approval, or deny.
- Guards with PII masking, and machine users as a managed identity rather than a shared API key.

MetaMCP approach

- OIDC single sign-on with auto-discovery and PKCE.
- API-key or OAuth authentication on each endpoint, per MCP Spec 2025-06-18.
- Namespace and endpoint scoping, plus registration controls for who can create an account.
- No role model, approval workflow, or per-user policy on individual servers.

Bottom line

MetaMCP's access model is coherent and it is enough for a technical team that trusts everyone holding a key. It is not an organizational access model, and it does not set out to be one. If your requirement is that access ends at offboarding without anyone remembering to do it, that is the gap.

Connectors and the MCP catalog

The difference most teams underestimate.

Willow approach

- A curated catalog where each MCP server is scored 0 to 10 for risk, broken into authentication, tools, and guards.
- API-to-MCP conversion turns any REST API into a governed MCP server.
- Skills, plugins, toolkits, and commands on top, publishable to teams.

MetaMCP approach

- You register each MCP server yourself, as a command and its arguments.
- There is no catalog, marketplace, or connector library.
- Aggregated tools can be exposed outward as OpenAPI, but arbitrary REST APIs are not converted inward into MCP servers.

Bottom line

MetaMCP aggregates the servers you bring. Willow supplies them, risk-scores them, and converts your REST APIs into more of them. Both are defensible designs, and the cost of the first one lands on whoever has to find and maintain forty MCP servers.

Employee experience

A tool your platform team uses, or a capability your organization has.

Willow approach

- Employees browse an approved, risk-scored catalog in the Connect Panel and connect any AI client in one click.
- IT approves the catalog once and sets per-user policy behind it.
- A single command installs the gateway into an IDE.

MetaMCP approach

- There is no end-user surface, and MetaMCP does not claim one.
- Integration documentation covers Cursor and Claude Desktop, both by editing client configuration.
- Endpoint URLs are distributed by whoever administers the deployment.

Bottom line

This is a scope choice rather than a shortcoming. MetaMCP is infrastructure, and infrastructure does not need a front door for the finance team. If your rollout has to reach people who will never edit a JSON file, it needs one.

Management and observability

What you can prove after the fact.

Willow approach

- Per-call logs with configurable retention, forwarded to Splunk, Coralogix, Loki, CrowdStrike, or a webhook.
- Token and context optimization, with usage analytics by team and tool.
- Logs are append-only rather than immutable.

MetaMCP approach

- Application log levels are documented and configurable.
- The Inspector is an interactive debugging tool with saved server configs, not an audit surface.
- Tool logging and error traces are listed among future middleware ideas.

Bottom line

MetaMCP produces logs. What it does not yet have is an audit product, and its own roadmap is where that sits. If someone will one day ask you which employee's agent called which tool with what arguments, decide now which system answers that.

Why enterprises pick Willow

> We are six to ten months ahead of most companies in AI adoption. More code to production, fewer incidents, real outcomes. Willow is what made it possible to move that fast without slowing down our security posture.

Asaf Yonay
Head of AI Core, Wix

WIX

Wix needed a secure, governed way to connect employees and agents to internal tools, documentation, and workflows. With Willow, the AI Core team built the enterprise MCP infrastructure that now supports nearly 600 tools and 300,000+ weekly tool calls across engineering, product, design, HR, finance, legal, and business teams.

60%
of enterprise AI rollouts stall on security review
Source: Gartner, 2025

84% / 91%
of developers use AI coding tools, with up to 91% of these tools unmanaged or unapproved by IT and security teams
Source: European Cyber Security Organization

Most
enterprises have no visibility into shadow AI usage
Source: IDC, 2025