willow vs obot

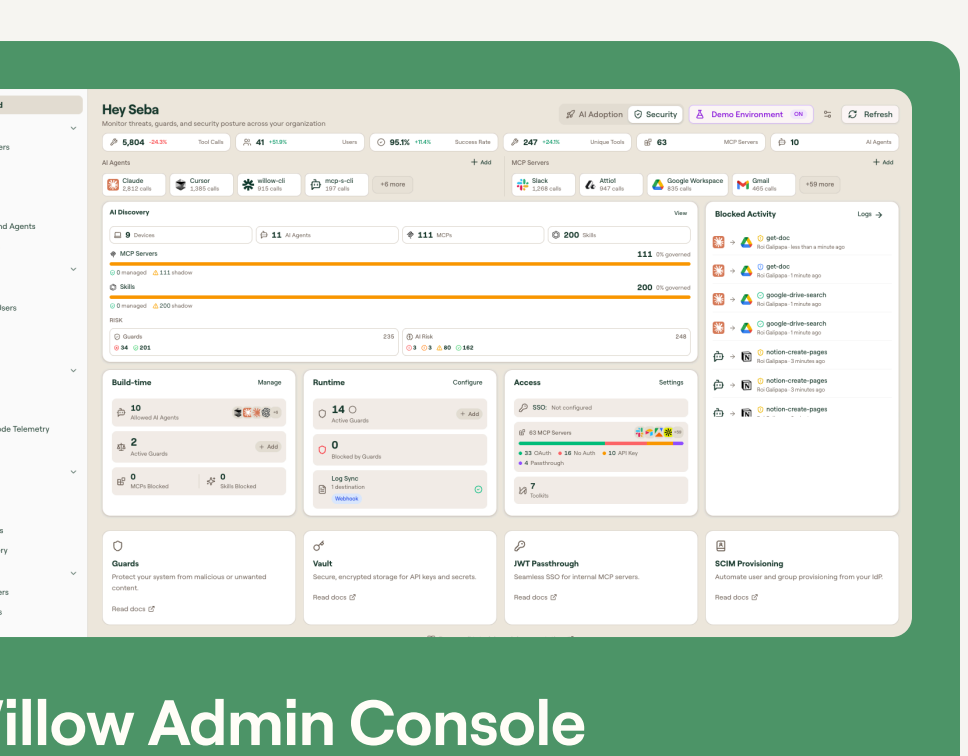
Willow vs. Obot: managed governance, or an open-source MCP gateway you run yourself

Obot is an open-source MCP gateway you host on your own infrastructure. Willow is a managed, certified governance platform your IT and security teams run without standing up Kubernetes. Compare both on openness, compliance, discovery, and deployment.

Get Started | Book a Demo

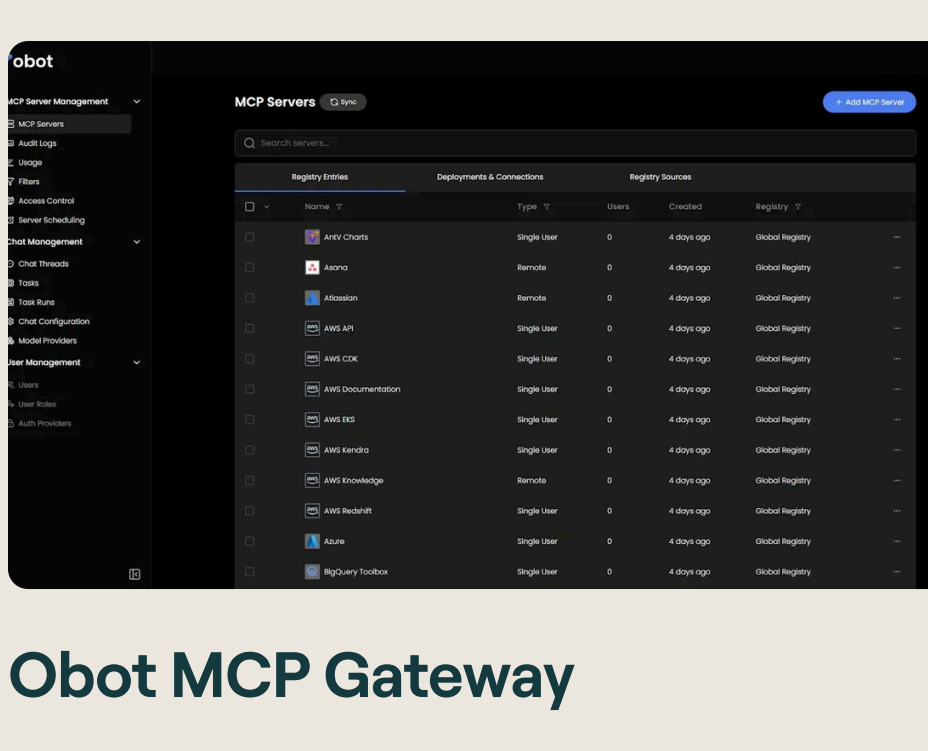
Trusted by

Wix · agora · papaya global · innovid



Willow Admin Console

A managed access and enablement control plane for the whole organization. It provides employee self-service, machine users, fine-grained per-agent and per-action policy, shadow-AI discovery, and IT-run identity and lifecycle, delivered as a certified managed service. It is proven at enterprises like Wix with roughly 5,000 weekly active users.



Obot MCP Gateway

An open-source-first MCP gateway and control plane, built by the team behind Rancher and Cloud.com. The MIT core is free to self-host on your own infrastructure with data that never leaves your environment, and a commercial Enterprise Edition and managed Obot Cloud sit on top. It is a credible, well-funded challenger with a young product.

The Bottom Line

Obot is open-source with K8s deployment and GitOps config for DevOps teams. Willow provides shadow AI detection, unified build & runtime guards, infrastructure-as-code governance, and a plugin marketplace at enterprise scale.

Willow is the better fit when

You want certified, managed governance without running infrastructure yourself. It gives you SOC 2 Type II compliance out of the box, employee self-service, a curated and enterprise-vetted catalog, endpoint discovery of shadow AI across web and local usage, enterprise identity providers included rather than paywalled, and deployment anywhere including on-premises, air-gapped, hybrid, and the EU, delivered as a supported service.

Obot is the better fit when

You want a free, open-source, self-host-forever gateway where data never leaves your environment, you have the DevOps capacity to run it on your own Kubernetes or cloud, and you value an open agent framework and an auditable MIT codebase.

Pick Willow if

Your goal is to operationalize AI across real teams with governance your organization can own and adapt, backed by a vendor certification and a proven enterprise-scale rollout, rather than assembling and maintaining the gateway yourself.

Willow vs Obot, side by side

Dimension	Willow	Obot
Architecture	Managed access and enablement control plane for the organization	Open-source-first MCP gateway and control plane, self-hostable
Deployment options	SaaS, dedicated cloud, on-prem on AWS, GCP, or Azure, air-gapped, hybrid, EU residency, delivered as a managed service	Any Kubernetes cluster, Docker, any cloud, or on-prem, self-managed, with Obot Cloud as the managed option
Licensing and pricing	Commercial, managed	Open-source core under MIT, free to self-host; Enterprise Edition and Obot Cloud priced privately
Authentication	OAuth 2.0, OIDC, SAML, JWT, and SSO with Okta, Entra, JumpCloud, Google, and more, included	OAuth 2.1 built in, with Google and GitHub free; Okta, Entra, SAML, and OIDC in the paid Enterprise Edition
Identity & Access	RBAC, groups, SCIM, and fine-grained per-agent and per-action policy	Multi-role RBAC, per-tool permissions, and policy-as-code per user, group, and agent
Machine Users	Generally available service accounts, using an API key or OAuth2 client credentials	API-key auth for programmatic access; no separately documented machine-user product
Background agents	Governed autonomous agents, each with its own scoped identity, least-privilege capabilities, and full audit	Nanobot, an open-source agent framework that turns MCP servers into agents, in alpha (as of July 2026)
Connector and MCP catalog	1,000+ curated, enterprise-vetted connectors with risk scoring	Built-in curated MCP catalog with schema visibility and composite bundled servers
API-to-MCP conversion	Wrap any REST API as a governed MCP	Not a highlighted capability
Employee self-service portal	Connect Panel and marketplace, one-click connection for any employee	Publish-to-employees, instant MCP connection by URL, and a built-in chat client
IT approval workflows	Approve the catalog once, per-user MCP policy of none, needs-approval, or allow, and governance as code through GitHub	Access policies per user and group; no Git-based pull-request workflow marketed for the full configuration
Shadow AI detection	Endpoint scan agent finds MCPs, skills, and agents on web and local, with risk scoring	Governs what flows through the gateway; no endpoint discovery agent evidenced
Vibe app monitoring	Tracks employee use of vibe-coding app builders like Lovable and Base44, and flags when those apps connect to internal systems	Not offered
Governed Chrome	Browser extension for visibility into web AI usage and OAuth flows through managed Chrome	Not offered
Observability and audit logs	Full per-call logs, forwarded to Splunk, Coralogix, Loki, CrowdStrike, or a webhook, with configurable retention	Every tool call logged with user, agent, server, arguments, and outcome, with usage dashboards, encrypted at rest and in transit
AI security	Built-in runtime and buildtime guards, response PII masking, plus third-party integrations	Gateway-level MCP filtering of requests and responses, with PII redaction and payload inspection
Token optimization	Token and context optimization to cut cost per call	Usage dashboards; token-cost optimization not a highlighted capability
Compliance	SOC 2 Type II, GDPR, EU residency	No vendor certification; self-hosting means you own compliance in your own environment
Reference customers	Wix at roughly 5,000 weekly active users, Agora, Riskified, Innovid, Lansweeper	No named customers on the site at capture

Get Started | Book a Demo

Fit Check

Governed Background Agents

Willow approach

- Every background agent has its own identity and access key, so it authenticates as itself and its capabilities define its blast radius.
- Scope each agent to the exact tools, skills, and rules it needs, and nothing more.
- Every call passes through the gateway under the same guards, policy, and logging as any other traffic.
- A risk-scored inventory answers which agents can touch a given system.

Obot approach

- Obot's agent story is Nanobot, an open-source framework, Apache-2.0 licensed with real community traction, that turns MCP servers into agents with reasoning and tool orchestration.
- It is a genuine open framework for building agents, and it is in alpha (as of July 2026), with its own README warning to expect significant breaking changes.

Bottom line

Both let you work with agents, but they optimize for different things. Obot gives builders an open framework to create agents. Willow optimizes for governing autonomous agents, giving each one its own scoped identity, least-privilege access, and full audit, so security can say yes to unattended work.

Infrastructure as Code

Manage your whole AI configuration the way engineering manages everything else, as code in Git.

Willow approach

- Sync your toolkits, skills, commands, MCP servers, and clients to a GitHub repository as version-controlled files.
- Review every change through pull requests, and keep a full history of who changed what.
- Two-way sync keeps Willow and the repository in step, so a change in either place flows to the other.
- Authentication secrets are never written to the repository, so the configuration is safe to store.

Obot approach

- Obot is open-source and supports policy-as-code, so a technical team can manage rules declaratively, and it manages configuration in its own control plane.
- It does not market a two-way, pull-request-reviewed GitHub workflow for the full toolkit, skill, and MCP configuration.

Bottom line

Both respect that platform teams want configuration in code. Willow adds a two-way GitHub workflow for the full configuration, with pull-request review and history, so AI governance is managed the same way as the rest of the stack.

Self-Service and IT Approval Workflows

Employees self-serve the tools they need, and IT approves once instead of drowning in tickets.

Willow approach

- Employees browse approved tools in the Connect Panel and marketplace and connect any AI client in one click, with no IT ticket.
- IT approves the catalog once, and sets a per-user policy of none, needs-approval, or allow for user-added MCPs.
- When your identity provider deprovisions a user through SCIM, Willow removes that user, and because access flows through their groups, their access to the MCPs and tools they had goes with it.

Obot approach

- Obot publishes MCP servers to employees for self-service connection, and admins set access policies per user and group.
- The free tier authenticates through Google and GitHub, and enterprise identity providers (Okta, Microsoft Entra, JumpCloud, Auth0) sit in the paid Enterprise Edition. Obot does not document SCIM provisioning.

Bottom line

Both publish to employees and keep admins in control. Willow includes enterprise identity providers rather than behind an edition upgrade, and adds SCIM-driven lifecycle that Obot does not document, so provisioning and offboarding follow your identity provider from day one.

Guards and DLP

A configurable content layer that redacts, blocks, warns, or requires approval on prompts and actions.

Willow approach

- Guards inspect every tool call and response at runtime, and inspect content before it is published at build-time.
- Built-in detectors cover secrets and PII, and you can add regex, JSONata, LLM, or custom function checks.
- When a guard fires it can redact the sensitive part, warn, require approval, or block, and the most restrictive outcome wins.

Obot approach

- Obot filters MCP traffic at the gateway, inspecting, modifying, and blocking requests and responses, with PII redaction and payload inspection.

Bottom line

Both do gateway-level request and response filtering with PII handling, and neither should be sold as a proven defense against encoded prompt injection. Willow's distinction is the configurable policy actions, redact, warn, require approval, or block, applied at both build-time and runtime, as a DLP and policy-as-code layer.

ROI and Cost Optimization

See where tokens go and cut them, not just report the spend.

Willow approach

- Token Usage Analytics shows where tokens come from across MCP servers, toolkits, skills, and tool responses, by team, tool, and use case.
- Reduce cost by tightening toolkits, moving long skill content into references, and optimizing the tool-response output format, including JSON Compact, CSV, YAML, or TOON.
- The result is fewer tokens per call, not only a report of what was spent.

Obot approach

- Obot provides real-time usage dashboards that report tool calls and activity across users, agents, and servers.

Bottom line

Both give visibility. Willow goes a step further and actively reduces the tokens each call consumes, so cost control is built into how the platform runs, alongside analytics by team and tool.

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