MCP SERVER | NO CODE | CLOUD HOSTED

Targetprocess MCP

Audit your entire agile portfolio from your chat client.

Targetprocess MCP connects your AI client to Apptio Targetprocess for enterprise Agile planning. It lets you query detailed product backlogs, track user stories and active bugs, and map out global project features directly from the terminal. Stop switching between web dashboards; get constant programmatic awareness of your organization's roadmap execution.

A+ Quality Score 100/100

agile-planning | scrum-master | portfolio-management | bug-tracking | sprint-planning | visual-management

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/targetprocess

SCAN TO VIEW

The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

Targetprocess MCP

6 tools available
Cloud-hosted on Vinkius

Running an agile portfolio is complex enough without constantly fighting clunky management panels. This MCP gives your AI client direct access to Apptio Targetprocess data, letting you treat your entire product backlog like a text document. You can ask your agent to pull together the current sprint schedule and cross-reference it against high-priority bugs. Need to map out what features are currently being developed? It's there. This integration lets you bypass the graphical interface friction that slows down deep work, providing programmatic awareness of everything from global project scopes to individual user story requirements. Because this MCP lives in Vinkius, you connect once and gain access to all your specialized tools across multiple systems.

Core Capabilities

01 — Audit current technical debt

List reported bugs and defects against active projects without leaving your coding environment.

02 — Map project scopes and features

Get structured lists of all defined global products, capabilities, and parent projects in the system.

03 — Track sprint timing and commitments

Query active sprints and time-bound iterations to see what the team is currently focused on.

04 — Read requirement specifications

Pull detailed product backlogs by listing specific user stories assigned to a project.

05 — Identify all system users

Retrieve a list of every registered user within the Targetprocess account for auditing purposes.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/targetprocess** — connect your AI agent in three steps.

01 First, you connect your AI client to this MCP via Vinkius and provide your organizational host ID and access token.

02 Next, you prompt your agent with a complex request, such as asking for the top three unassigned user stories while also listing all open bugs.

03 Finally, your agent executes the necessary functions in sequence, returning structured data that pinpoints exactly what's happening across your entire product roadmap.

The bottom line is you use natural language to run complex portfolio reports that usually require five different web clicks and three hours of context switching.

Built For

This MCP is essential for Engineering Team Leads, Agile Product Owners, and Release Management Specialists who get frustrated by the sheer number of dashboards. If your job requires correlating bug reports with current sprint commitments, this tool saves hours.

Agile Product Owner

Uses this to validate requirements textually, pulling lists of features and user stories directly into their prompt instead of navigating the graphical interface.

Engineering Team Lead

Runs automated checks on defect queues using list_bugs and syncs sprint scopes with list_iterations during code reviews to keep development aligned.

Release Management Specialist

Invokes deep structural readouts that analyze user story progression, mapping global projects cleanly over the terminal for formal sign-offs.

What Changes When You Connect

01 You get instant visibility into technical debt. Instead of manually navigating a bug dashboard, you can query list_bugs to find all high-priority defects right in your prompt.

02 Stop guessing about team focus. By calling list_iterations, your agent tells you exactly which sprint the development cycle is currently operating within.

03 Mapping requirements becomes simple. You don't need to click through menus; asking for list_user_stories pulls detailed specs instantly, validating needs against the actual product backlog.

04 Understand the big picture scope by calling list_features and list_projects. This lets you see the entire global hierarchy of products without leaving your terminal.

05 Streamline audits with list_account_users. You can pull a roster of every registered user directly into your workflow for compliance checks.

Real-World Applications

Diagnosing sprint blockers

A team lead notices delays during a review. Instead of asking three people, they prompt their agent: 'What are the active issues tracked in our main project and what is the current iteration?' The MCP runs list_bugs and list_iterations together, immediately highlighting the confluence of technical debt and limited time.

Validating feature scope

An owner needs to prove that a new capability (feature) supports three specific user stories. They prompt their agent to run list_features first, then use list_user_stories to confirm the exact requirements against the global product map.

Pre-release compliance audit

A release manager needs a snapshot of all technical assets involved in the deployment. They ask their agent to list_projects, then immediately run list_account_users and list_features to ensure every user has access to the necessary components.

Quickly getting context on new hires

A manager needs to understand who is working in a specific product line. They simply ask their agent to run list_account_users, receiving a structured roster without logging into any user management portal.

Patterns to Avoid

Over-relying on dashboards

✗ AVOID

Manually clicking through the project dashboard, then opening the bug tracker tab, and finally downloading a CSV of user stories. This takes 15 minutes and involves copy-pasting three separate files.

✓ INSTEAD

Use this MCP to combine all data points in one prompt. Ask your agent to 'List all projects, identify active bugs, and pull associated user stories.' The MCP handles the orchestration for you.

Forgetting global context

✗ AVOID

Only focusing on a single project's immediate backlog while missing the parent product limitations. This leads to scope creep because the team doesn't know which global feature is required.

✓ INSTEAD

Always start by asking for list_features before listing_user_stories. This ensures your agent grounds the requirements in the correct high-level capability.

Treating data as static

✗ AVOID

Running a single report on bugs and assuming it's accurate today. The system might have changed the sprint or closed multiple issues since the initial run.

✓ INSTEAD

Always combine list_bugs with list_iterations in one query. This guarantees you see current technical debt relative to the active development cycle.

The Right Fit

Use this MCP if your job requires correlating data from different agile views—like linking a specific user story (list_user_stories) to an active bug (list_bugs) while simultaneously knowing which sprint it belongs to (list_iterations). You need programmatic, multi-source reporting that doesn't require manual dashboard navigation. Don't use this if you just need to read documentation or update status; for those tasks, your local IDE is fine. If you only care about project structure without defects, a simple directory listing tool will suffice. But when the complexity involves 'who,' 'what feature,' and 'when,' this MCP handles the entire process.

The Pain of Context Switching

Today, to get a full picture of project health, you open your primary dashboard. You see a list of projects. Then you have to click into the 'Bugs' tab to check for critical defects. Next, you navigate to the 'Backlog' section to pull user story details. Finally, you switch tabs again just to confirm which development sprint everyone is assigned to. It’s a painful cycle of clicking, opening new windows, and manually cross-referencing dates.

With this MCP, that entire sequence disappears. You tell your agent what you need—for example, 'Show me all open bugs related to the current sprint.' The system runs list_bugs and list_iterations in one go, feeding you a single, cohesive report without you touching a dashboard.

Targetprocess MCP: Full Portfolio Visibility

The specific manual steps that vanish are the multi-step data correlation process. You no longer have to manually cross-reference feature lists against project scopes or track down which users own a certain bug report across separate tabs.

Now, you ask your agent for the full scope—'List all global features and map them to their active projects.' The answer arrives immediately, structured and ready for action. It’s about having continuous knowledge of your product roadmap.

Targetprocess: 6 Tools for Agile Planning

These six tools give you complete programmatic control over reading project scopes, bug reports, user stories, and feature hierarchies within Targetprocess.

#	TOOL	DESCRIPTION
01	`list_bugs`	Reads a list of currently reported technical defects or anomalies in the system.
02	`list_features`	Generates a structured list of high-level product capabilities and features.
03	`list_iterations`	Retrieves active time containers, showing which sprints or cycles are currently running.
04	`list_projects`	Lists every defined project within the Targetprocess account.
05	`list_user_stories`	Fetches detailed records of specific user stories, capturing requirement specifications.
06	`list_account_users`	Provides a complete list of all registered users in the Targetprocess account.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U **Retrieve the current active Sprint iterations and pull the details of the top 3 unassigned bugs logged under our primary development project.**

Gathering structural records ('list_iterations')...
Evaluating active Sprint: 'Iteration Q4-B'.
Now querying current technical debts via 'list_bugs'...

Found Top Defects:
- Bug-145: 'Memory leak in auth parser' (Critical)
- Bug-199: 'Missing CSS button radius' (Medium)
- Bug-231: 'Gateway timeout during 2FA' (High)

U **Extract the details for user story #4552 in the current sprint.**

Interrogating agile entities ('get_entity_details')...
User Story #4552: 'Implement SSO Auth'
- State: In Progress
- Assigned to: Alice B.
- Story Points: 5
- Iteration: Sprint 14

U **List all high priority bugs that are currently 'Open'.**

Querying Targetprocess data ('search_entities')...
I found 2 'Open' bugs tagged with High Priority:
1. Bug #5120: 'Payment gateway timeout in production'
2. Bug #5125: 'Null pointer exception on invoice generation'

Frequently Asked Questions

01 **How do I use Targetprocess MCP to find out what bugs are open?**

You call the list_bugs tool. This function queries and provides you with a structured array listing all currently reported defects, allowing you to assess technical debt immediately.

02 **Can Targetprocess MCP show me which sprint we're in?**

Yes, use the list_iterations tool. It fetches time-bound records showing the active sprints and development cycles for your team.

03 **Does Targetprocess MCP help with project mapping?**

Absolutely. You can run list_projects to see all defined projects, and then use list_features to understand the high-level capabilities associated with those projects.

04 **How do I find out who works in my Targetprocess account?**

Run the list_account_users tool. This provides a full roster of every registered user within your organization's instance.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"targetprocess": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

Targetprocess is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by Targetprocess. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Targetprocess MCP
Server ID	`019d7610-87cd-7324-b7e0-e418df470ddd`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/targetprocess.