MCP SERVER | NO CODE | CLOUD HOSTED

Atlassian MCP for AI Agents

Manage Jira Issues and Confluence Documentation Context

The Atlassian Jira & Confluence MCP lets your AI agent operate across your entire project ecosystem. Instead of jumping between dashboards, you can ask natural language questions to audit complex Jira issues, review active agile sprints, and retrieve deep knowledge from enterprise Confluence wikis—all in one conversation.

F Quality Score 3.6/100

agile-boards | issue-tracking | wiki | sprint-management | bug-tracking | team-collaboration

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/atlassian-jira-confluence

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.

Atlassian (Jira & Confluence) MCP

9 tools available
Cloud-hosted on Vinkius

Managing projects means juggling tasks, bugs, requirements, and documentation. These pieces live across different systems: your development board in Jira, and your spec sheets in Confluence. Before this MCP, you'd spend time copying ticket numbers into search queries or manually cross-referencing dates between two separate dashboards.

With this integration, your AI agent acts like a direct navigator across both platforms. You can ask it to find all high-priority tickets related to a specific feature and then immediately pull the associated architectural decision document from Confluence—all without leaving your chat window. It turns complex, siloed workflows into simple, conversational commands. This entire catalog of tools is hosted on Vinkius, meaning you connect once and get access to this deep project intelligence alongside thousands of other services.

Core Capabilities

01 — Search Jira issues using advanced criteria

Find specific tickets or groups of issues across your organization using complex JQL queries.

02 — Review project status and active sprints

Get an overview of all available agile boards, list current sprints, and track which tasks are due in the near future.

03 — Access deep documentation from Confluence

Search across your entire wiki knowledge base using CQL and extract the full text content of specific pages or spaces.

04 — Understand user identities within Atlassian

Automatically pull current authenticated user information to ensure actions are assigned correctly.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/atlassian-jira-confluence** — connect your AI agent in three steps.

01 Subscribe to this secure MCP and provide your Atlassian Domain, Email, and API Token.

02 Connect the credentials to your preferred AI client (Claude, Cursor, Windsurf, etc.).

03 Use natural language prompts to ask questions that cross-reference tasks in Jira with documentation in Confluence.

The bottom line is you use your agent to talk to both Atlassian services simultaneously from one chat interface.

Built For

This MCP is built for people who spend their day jumping between Jira and Confluence. If your job involves understanding the full context of a bug—from its initial spec sheet to its final deployment status—you need this.

Product Manager

Pulls complete summaries of entire epics, pulls specific criteria from tickets, and drafts release notes based on project progress.

Scrum Master

Audits sprint health across multiple boards, aggregates blockers, and builds real-time views of overall project status without manual dashboard hopping.

Engineering Team Lead

Checks current bugs in an active sprint, retrieves specific specification documents from Confluence, and traces ticket lifecycles directly within their IDE.

What Changes When You Connect

01 Get instant visibility into project status. Use list_boards and list_sprints to quickly see which agile boards have active sprints without logging into the dashboard.

02 Tackle complex debugging cycles. Combine search_issues with get_page to find a bug ticket, then immediately pull the original specification document that describes the required behavior.

03 Stop manual data gathering. Instead of copying criteria, your agent can summarize complete epics or audit sprints across multiple boards using list_sprints and search_issues.

04 Know exactly who's working on what. The get_myself tool confirms current user identity, ensuring that all ticket actions are attributed correctly from the start.

05 Rapid knowledge retrieval. Use search_content to find technical guides or architectural patterns buried deep in Confluence wikis, then use get_page to stream the full text for context.

Real-World Applications

The requirement changes mid-sprint

A developer realizes a bug ticket (ENG-402) needs more info. They ask their agent, which uses get_page to pull the detailed architectural design from Confluence and feeds it back into the conversation alongside the issue details.

Auditing project scope before a meeting

A Product Manager asks the agent for all active boards (list_boards) and then uses search_issues to aggregate every open story related to 'checkout flow' across those boards, giving them one cohesive report.

Finding outdated technical guides

A support specialist needs a guide for an old API. They prompt the agent, which uses search_content on Confluence and then pulls the full text of the most relevant page (get_page), allowing them to answer the customer instantly.

Getting status across multiple teams

A Scrum Master asks for all active sprints in a department. The agent uses list_sprints and reports back which boards are running, allowing the master to see overall project health at a glance.

Patterns to Avoid

Jumping between tabs

✗ AVOID

Manually opening Jira, searching by JQL. Then going to Confluence, pasting keywords into CQL search. Finally, cross-referencing the dates and statuses in a spreadsheet.

✓ INSTEAD

Ask your agent to combine these steps: 'Find all high-priority bugs assigned to me that mention 'API' and pull the corresponding requirements document from Confluence.' This uses search_issues and search_content together.

Assuming documentation location

✗ AVOID

A developer knows a spec exists but can’t remember if it’s in Space A or Space B, forcing them to click through multiple spaces.

✓ INSTEAD

Use the agent's search_content tool first. It searches across all Confluence spaces and provides the exact page link and content summary you need.

Missing context for a ticket

✗ AVOID

A team member gets an issue key but has to manually go find the original design document, wasting 15 minutes of research time.

✓ INSTEAD

Use get_issue and ask your agent to cross-reference it with Confluence. The agent can use both tools simultaneously to retrieve context from both systems.

The Right Fit

You should connect this MCP if your day involves frequently asking 'Why?' or 'How?' about a task, issue, or bug. Specifically, if you need the connection between *what* was built (Jira) and *why* it was built (Confluence), this is essential. Don't use this if you only ever work within Jira or only ever write documentation in Confluence; then, a dedicated single-tool MCP will do fine. If your primary need is simply to track user roles and permissions, the get_myself tool handles that, but for full workflow visibility, connecting all tools is best.

Atlassian Jira & Confluence: Auditing Project Backlogs via AI Agents

Right now, checking the status of a project requires logging into multiple portals. You jump to Jira to check if the ticket is 'Ready,' then open Confluence to find the most recent spec document. You copy key identifiers and paste them in different places just to piece together a timeline.

With this MCP, you simply ask your agent: 'Show me all story tickets for Feature X that reference documentation from the Payment Gateway v2 page.' The system uses both search_issues and get_page, giving you one concise answer with direct links. You stop manually hopping tabs.

Atlassian Jira & Confluence: Managing Enterprise Documentation Context

Finding the right documentation is a massive waste of time. Instead of remembering which space (list_spaces) holds the answer, you have to run through dozens of search results manually, hoping one has the full text you need.

Now, simply ask your agent: 'What's our policy for handling PII data?' It runs a powerful search_content query across every relevant Confluence space and presents the most accurate, up-to-date page immediately. You get precision without the guesswork.

9 Tools in the Atlassian Jira & Confluence MCP for Project Management

Use these tools to manage projects by searching tickets, listing boards, getting page content, and tracking user identities across your entire Atlassian ecosystem.

#	TOOL	DESCRIPTION
01	`list_projects`	Lists every project key available in your Jira instance so you know which domain to query.
02	`search_issues`	Searches all Jira issues using complex JQL queries based on criteria like assignee or status.
03	`get_issue`	Fetches the complete details for a single Jira issue when you know its exact key.
04	`list_boards`	Shows all active agile boards, helping you decide where to look for backlogs or running sprints.
05	`list_sprints`	Retrieves a list of historical and current sprints tied to a specific Jira board.
06	`get_myself`	Pulls information about the user who is currently authenticated to Atlassian, confirming permissions and identity.
07	`list_spaces`	Lists all available spaces in Confluence so you know which knowledge area to search.
08	`search_content`	Searches across your entire Confluence wiki using CQL, finding pages related to a topic or keyword.
09	`get_page`	Extracts the full rich text content from a specific Confluence page for detailed reading.

See It in Action

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

U What are all high-priority tickets assigned to me related to login errors?

🔎 **Sprint Audit: Login Errors**

I found 4 critical issues currently in progress:

- `ENG-501` : Password Reset flow failure (Priority: Blocker)
- `ENG-489` : MFA enrollment bug (Priority: High)
- `ENG-320` : Session timeout error (Priority: Medium)

These tickets are linked to the 'Authentication Flow' Confluence page. Should I pull the full text of that spec for context?

U Find documentation on our new billing API structure.

I searched across all available spaces and found two pages: **Payment Gateway v2** (most recent) and *Legacy Stripe Migration Docs*.

The first one seems most relevant. It details rate limits, required fields, and the full JSON schema for endpoints like `/v2/charge`. Would you like me to read out the text from this page now?

U List all active boards and see what's due next week.

Board Overview

You have 5 active Jira Boards. Here's a quick look at immediate deadlines:

- `CORE-BACKEND` : Sprint 44 has 12 tickets due next Tuesday.
- `MOBILE-REVAMP` : Sprint 12 is clear, but there are 8 stories flagged for review by end of week.

The `CORE-BACKEND` board looks like the priority right now. Want me to pull a list of those specific tickets?

Frequently Asked Questions

01 **How does the Atlassian Jira & Confluence MCP help with project planning?**

This MCP gives your agent real-time access to both task boards and documentation. You can ask it to audit a sprint's scope by checking related tickets in Jira against the official requirements documented in Confluence, giving you full context before starting.

02 **Can I use this MCP to find API details for old features?**

Yes. If the documentation exists in Confluence, your agent can search across all spaces and pull up the specific page content using tools like get_page. It's perfect for finding archived or legacy technical specs.

03 **Does the Atlassian MCP support different types of Jira boards?**

It supports multiple board types, allowing your agent to list and view active sprints across various project domains. This means you get a comprehensive overview of all work happening in one place.

04 **What if I need to cross-reference an issue key with a document?**

You just ask the agent directly. It can use your issue key to find the ticket details and then automatically search Confluence for related documentation or specifications, linking all the information together.

05 **Is this Atlassian MCP better than using a separate Jira integration?**

Because it handles both Jira and Confluence in one flow, you avoid having to use two different tools. You keep your conversation focused on solving problems rather than managing multiple connections.

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 `"atlassian-jira-confluence": { "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

Atlassian (Jira & Confluence) 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 Atlassian (Jira & Confluence). 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	Atlassian (Jira & Confluence) MCP
Server ID	`019d7554-7404-736e-9c86-d10ac66b4d01`
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/atlassian-jira-confluence.