

MCP SERVER

NO CODE

CLOUD HOSTED

Dext MCP for AI Agents

Manage expenses, process invoices, and audit client accounts.

Dext equips your AI agent to handle core bookkeeping tasks like processing receipts, tracking invoices, and monitoring detailed accounting data. It lets you manage a digital inbox of financial documents and instantly classify transactions against established account categories, all through natural conversation.

A+ Quality Score 100/100

bookkeeping

receipt-processing

invoice-automation

expense-management

data-extraction

accounting-integration



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.

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.

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.

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

Encryption at rest

Ed25519

PKI vault signatures

24h TTL

Ephemeral session keys

V8 Isolate

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.

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.

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.

06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

FinOps

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

Firewall & DLP

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

Agent Activity

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

Tool Health

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

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.

Dext MCP

10 tools available

Cloud-hosted on Vinkius

Managing expenses and reconciling accounts usually means jumping between dozens of tabs: the bank portal, the physical receipt pile, the client ledger, and the spreadsheet. This MCP changes that. It lets you talk to your finances instead of clicking through them.

Connect Dext directly into your AI workflow to manage every piece of accounting data from a single source. Your agent can check the status of items in your inbox or pull technical details about supplier extractions, giving you instant clarity on what's pending review. You can also list all associated clients and audit available bank accounts for statement extraction.

With Vinkius, connecting Dext gives your AI client a full view of your financial operations. Whether you need to track specific expense patterns or classify transactions against the chart of accounts, your agent handles the heavy lifting, letting you focus on advising clients instead of data entry.

Core Capabilities

01 — Audit and review pending expenses

The agent can list all items in your inbox—whether they are receipts or invoices—and identify which ones require human review.

03 — Classify and monitor accounts

The agent reads available accounting categories or lists all associated clients in your partner account so you can keep data organized.

05 — Check system status and users

The agent monitors your account's usage limits or lists all team members who have access to the Dext account.

02 — Search financial records by supplier name

You can ask the AI to search through your entire document history for specific transactions, like finding every item from 'Staples' last month.

04 — Inspect extracted financial metadata

Get granular details from documents, including tax breakdowns and specific supplier information that was pulled automatically.

One Click on Vinkius — From Prompt to Execution

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

- 01 First, connect the Dext MCP to your preferred AI client and authorize it using your Dext API Key.
- 02 Next, prompt your agent with a financial query—for example, 'What receipts are pending review?' or 'List all clients.'
- 03 Your agent executes the necessary calls, retrieving structured data on expenses, accounts, or invoices that you can then use directly in conversation.

The bottom line is: your AI client treats Dext like a live data source, letting you query complex accounting records using simple chat prompts.

Built For

This MCP is essential for accountants and bookkeepers who spend too much time cross-referencing documents. It's perfect for the finance team member who needs to audit multiple client accounts before month-end closing, or the business owner who just wants a quick summary of monthly spending patterns without digging through portals.

Bookkeeper

You use this MCP to quickly list all items in an inbox and check which receipts are stuck waiting for classification.

Accountant

You audit the chart of accounts, monitoring available accounting categories or listing connected bank accounts before running reconciliation reports.

Business Owner

You monitor overall expense patterns and supplier extractions via chat to understand where money is going without manual data exports.

What Changes When You Connect

-
- 01 Instantly check processing statuses: Use the `list_items_pending_review` tool to immediately see which receipts are stuck or need human attention, saving you time on status checks.

 - 02 Deep data access: The `get_item_details` tool pulls granular technical metadata for any document, giving you tax and supplier info without manual form filling.

 - 03 Client visibility: You can use `list_accounting_clients` to review all associated clients and their specific integration settings in a single conversation thread.

 - 04 Targeted searching: Never waste time scrolling. The `search_inbox_by_keyword` tool lets you find every record from 'Amazon' or 'Office Depot' instantly by name.

 - 05 Financial mapping: Use `list_accounting_categories` to audit the full chart of accounts, ensuring that new transactions are classified correctly before closing books.
-

Real-World Applications

Auditing tax compliance for a client

A bookkeeper needs to verify if all required sales taxes were captured across multiple documents. They ask their agent, 'Check the tax details for all items from Q2.' The agent uses ``get_item_details`` and provides a summary of tax amounts versus expected rates.

Preparing for period-end closing

A finance team member needs a full picture of who can see client data. They ask the agent to 'List all users.' The agent uses ``list_account_users`` and confirms necessary permissions are in place across the team.

Finding lost expense reports

A business owner can't find a receipt from last month. They ask their agent to 'Search for anything related to travel supplies.' The agent uses ``search_inbox_by_keyword`` and returns the exact transaction details, including date and amount.

Handling bulk document intake

A bookkeeper just uploaded 50 new receipts. They ask, 'What's the status of these uploads?' The agent uses ``list_inbox_items`` and provides a count of processed versus pending items.

Patterns to Avoid

Asking for generic data dumps

✗ AVOID

The user says, 'Give me all the accounting info.' This results in unmanageable lists that require further filtering and analysis.

✓ INSTEAD

Instead, use ``list_items_pending_review`` to narrow the focus. Ask: 'What items are pending review?' This forces the agent to filter down to exactly what you need.

Mixing up account types

✗ AVOID

A user tries to query bank statement data when they only have receipts uploaded, leading to an error or incomplete result.

✓ INSTEAD

Always confirm the scope first. Use ``list_connected_bank_accounts`` to verify if a bank connection is active before asking for reconciliation data.

Forgetting client context

✗ AVOID

The agent lists all items but doesn't specify which company they belong to, making the output useless.

✓ INSTEAD

Always narrow your query using ``list_accounting_clients`` first. Then ask: 'Show me pending items for Client ABC.'

The Right Fit

Use this MCP if your primary bottleneck is transforming raw financial documents (receipts, invoices) into structured, auditable data points within conversation. You need to know *what* was spent and *where*. Don't use it if you just need basic document storage; that's a file system tool. Also, don't rely on it for tax law advice—it reads the numbers, but the agent won't tell you if an expense is deductible in your jurisdiction. However, if you are managing multiple client books and need to check statuses using `list_inbox_items` or confirm classification rules with `list_accounting_categories`, this MCP is essential.

Dext MCP for AI Agents: Streamlining Bookkeeping Status Checks

Right now, checking on client receipts means logging into the Dext portal, navigating through status dashboards, and clicking 'View Details' dozens of times to check if an invoice has been processed or is stuck in review. It's tedious, slow, and you risk missing a crucial delay.

With this MCP, your agent handles that entire process instantly. You simply ask, 'Which client items are pending review?' Your agent uses tools like `list_items_pending_review` to pull the exact status report directly into your chat window. You get immediate clarity on document flow without opening a single web tab.

Dext MCP for AI Agents: Mastering Accounting Data Classification

Without this integration, classifying an expense requires manual review against your internal chart of accounts. You have to remember the rules and manually map every single transaction type to the correct category in a spreadsheet.

This MCP fixes that by giving your agent access to `list_accounting_categories` . Your agent knows your established classification rules and can use them instantly, telling you which categories are available or confirming where a specific expense should land. It makes audit-ready data retrieval automatic.

Dext: 10 Tools for Accounting Data Classification

Use these functions to list clients, audit categories, retrieve item details, and manage your overall accounting records.

#	TOOL	DESCRIPTION
01	<code>get_account_metadata</code>	Retrieves operational metadata and usage limits for your Dext account.
02	<code>get_item_details</code>	Gathers detailed information about a single, specific item in the inbox.
03	<code>list_connected_bank_accounts</code>	Lists all bank accounts that are configured for statement extraction.
04	<code>list_accounting_categories</code>	Retrieves the chart of accounts or expense categories available for classification purposes.
05	<code>list_accounting_clients</code>	Lists all clients associated with your Dext partner account.
06	<code>list_inbox_items</code>	Retrieves a list of all pending and processed items, such as receipts or invoices, in the inbox.
07	<code>list_extracted_suppliers</code>	Provides a list of all suppliers that were automatically extracted from your documents.
08	<code>list_items_pending_review</code>	Identifies and lists only the inbox items that are currently waiting for manual review status.
09	<code>list_account_users</code>	Lists every user who has access to the current Dext account.
10	<code>search_inbox_by_keyword</code>	Searches the inbox for receipts or invoices using a keyword, like a supplier name or description.

See It in Action

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

U Show me all items in the inbox that I need to look at right now.



I found 7 items needing review. Here's a summary:

- **Staples:** Receipt, \$45.00 (Category Pending)
- **AWS:** Invoice, \$120.00 (Tax Details Missing)
- **Shell:** Receipt, \$89.50 (Status: Processing)

Would you like me to pull the tax breakdown for the AWS invoice?

U What was my total spending on office supplies last month?



I searched your records and found 3 items matching 'office supplies' from two different vendors. The total spend came out to **\$215.00**.

- Office Depot: \$90.00 (Date: Oct 5)
- Amazon: \$75.00 (Date: Oct 12)
- Staples: \$50.00 (Date: Oct 28)

Should I list the individual dates and amounts?

U List all bank accounts connected for statements.



You currently have three active bank connections configured:

1. **Chase Bank:** (Primary operating account, Active)
2. **Capital One:** (Savings/Investment, Active)
3. **Local Credit Union:** (Secondary Source, Inactive)

You can use these for statement extraction right away.

Frequently Asked Questions

01 How does the Dext MCP help me categorize invoices?

The Dext MCP allows your agent to access and audit your full chart of accounts. Instead of guessing, you can ask the agent which accounting categories are available for classification, ensuring every expense is logged correctly.

02 Can I use the Dext MCP to find specific receipts by vendor name?

Yes. You can tell your agent to search your entire inbox using a keyword like 'Shell' or 'Starbucks.' It finds all matching items and gives you details on total spend and dates.

03 What if I need to check which clients are in my account?

The MCP lists every client associated with your Dext partner account. This is helpful for ensuring you have visibility into all the books you manage before running reports.

04 Does using the Dext MCP require me to manually update everything?

No. The agent automates data retrieval. You tell it what you need—like listing pending items—and it pulls that structured information for you, eliminating manual dashboard navigation.

05 Is the Dext MCP good for small business expense tracking?

Absolutely. It lets you monitor expense patterns and supplier data via chat, making complex accounting tasks feel as simple as asking a colleague for a quick summary.

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 `"dext": { "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

Dext 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](https://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 Dext. 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	Dext MCP
Server ID	019d7585-188b-7095-bd27-73162010ce11
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/dext.