MCP SERVER | NO CODE | CLOUD HOSTED

Deterministic Array Operations MCP for AI Agents

Guaranteed Data Quality and Large Dataset Processing

Deterministic Array Operations provides high-precision data engineering capabilities for AI agents. It lets your agent process huge lists of records—chunking them safely, eliminating duplicates by a specific ID, or finding common items between two datasets—all while keeping all the math local and reliable.

A+ Quality Score 100/100

data-processing | javascript | array-manipulation | data-deduplication | performance-optimization | data-transformation

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/deterministic-array-operations

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.

Deterministic Array Operations MCP

3 tools available
Cloud-hosted on Vinkius

LLMs struggle with big data. When you give an agent a massive JSON array to manipulate, it often hits context limits or, worse, skips records entirely. This MCP fixes that problem by moving heavy collection transformations outside of the AI's core processing loop. You feed your raw data into this connector, and we handle the math using a pure V8 JavaScript engine, guaranteeing absolute mathematical precision every time. Need to split 10,000 items for an API call? Use chunking. Got two massive lists and need to know what overlaps? Intersect them instantly. This is essential infrastructure for any agent dealing with data quality or large payloads. Accessing this kind of specialized function via the Vinkius catalog means your AI client can run these powerful, local calculations without ever needing to talk to an external API.

Core Capabilities

01 — Split massive lists into batches

The agent takes a large array and safely splits it into smaller, predictable chunks of a specified size.

02 — Filter out duplicate records

It removes redundant entries from an array, allowing you to specify which unique identifier (like an email or ID) defines a duplicate object.

03 — Find common items between datasets

The agent compares two distinct arrays and returns only the records that appear in both lists.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/deterministic-array-operations** — connect your AI agent in three steps.

01 You provide your raw data—a large JSON array or two separate arrays—to your AI client.

02 The agent calls the appropriate tool (like deduplicate_array) through this MCP, sending the data to the deterministic engine for processing.

03 The MCP returns a clean, mathematically verified result: a smaller chunked list, a unique filtered array, or the final intersecting set of records.

The bottom line is that your AI agent delegates complex, memory-intensive math problems to a dedicated, reliable engine instead of trying to solve them itself.

Built For

Data Engineers and BI Analysts who spend too much time cleaning up data sets after an initial LLM pass. If your workflow involves processing payloads larger than 10k records or requires guaranteed uniqueness checks, this MCP is for you.

Data Engineer

They use the MCP to reliably clean and structure raw JSON inputs before loading them into a database, ensuring data integrity that simple LLM prompts can't guarantee.

BI Analyst

They run comparisons between two large source datasets (e.g., sales leads vs. current customers) to find common records or missing overlaps without manual spreadsheet merging.

Backend Developer

They integrate the MCP into agent workflows that require breaking massive data streams into smaller, rate-limited batches for external API calls.

What Changes When You Connect

01 You eliminate the risk of data hallucination. When running `array_deduplicate`, you get perfectly clean records, not just an approximation.

02 Avoid API failures due to payload size. Using `array_chunk` ensures your massive datasets are always broken into predictable, rate-limit-safe batches.

03 Instantly identify common elements across disparate data sources. The `array_intersect` tool finds overlaps between two complex lists in milliseconds.

04 Keep everything local and private. All array transformations run within your secure infrastructure; zero API calls mean no data leaves your system.

05 Process structured objects, not just simple strings. You can specify a unique key when running deduplication on object arrays.

Real-World Applications

Cleaning up customer contact lists

A marketing team needs to merge two spreadsheets of leads, but both contain duplicate entries based on email. Instead of spending hours cleaning in Excel, they run the deduplicate_array tool using 'email' as the key, getting a perfectly unique list instantly.

Sending large datasets to external APIs

A data scientist has 15,000 records that need to be sent to a third-party API with a strict 100-item limit. They use the array_chunk tool to reliably segment the whole payload into exactly 150 chunks for sequential processing.

Finding mutual clients between sales teams

Two different regional sales teams provide lists of client IDs that need reconciliation. Using the array_intersect tool quickly finds every single client ID present in both reports, saving hours of manual comparison.

Patterns to Avoid

Relying solely on LLM context

✗ AVOID

Asking an agent to deduplicate a 700-item list of user objects by running the prompt: 'Deduplicate this array and only keep unique users.' The AI might skip records or misinterpret object structures.

✓ INSTEAD

Instead, use the dedicated `array_deduplicate` tool. Pass the data and specify the key (e.g., 'user_id'). This guarantees mathematical precision regardless of list size.

Overloading API calls

✗ AVOID

Passing a single 1,000-item payload directly to an external API that has a strict 500-record limit, causing the entire process to fail and requiring manual segmentation.

✓ INSTEAD

Use the `array_chunk` tool first. Segment your data into guaranteed batches of size 500, then loop through those chunks for reliable processing.

Comparing raw strings

✗ AVOID

Trying to find overlaps between two lists of unique product identifiers that are formatted differently (e.g., one has dashes, the other doesn't). The agent might fail to match them.

✓ INSTEAD

Use `array_intersect` on structured data or ensure your inputs are pre-cleaned with a consistent format before running the tool.

The Right Fit

You should use this MCP if your core problem involves guaranteed mathematical operations on massive collections of structured data. Specifically, when you need to deduplicate by a specific key (use `array_deduplicate`), or when rate-limiting an external API requires precise batching (use `array_chunk`). If your workflow simply needs basic filtering or simple text comparisons, a general LLM prompt might suffice. However, if you are dealing with object arrays, complex JSON payloads, or need to guarantee that every single item is accounted for across two lists, this MCP provides the necessary deterministic control that vanilla AI reasoning lacks.

Deterministic Array Operations: Solving Data Duplication in JSON Payloads

Today, when you ingest raw data—say, a list of user profiles from multiple sources—you end up with massive JSON arrays. You then spend ages manually writing prompts to ask the agent to 'clean this up' or 'find duplicates.' This often leaves you with incomplete or mathematically inaccurate results because LLMs aren't built for pure set theory.

With this MCP, your agent handles data cleaning reliably. Instead of vague instructions, you call `array_deduplicate` and specify the key (like 'user_id'). The result is a perfectly filtered array containing only unique records, giving you total confidence in your source data.

Deterministic Array Operations: Managing Data Segmentation for API Calls

When integrating with external services, the biggest headache is hitting strict payload size limits. You can’t just paste 10,000 records into a single API call; it fails, and you start over. Manually calculating how many batches of 500 items you need is tedious.

This MCP solves that with `array_chunk`. It takes your entire data set and mathematically guarantees the correct number of perfectly sized segments, giving your agent a reliable pipeline for high-volume API interaction.

Deterministic Array Operations: 3 Tools for Data Transformation

These tools allow your agent to perform complex array operations like filtering duplicates or splitting large payloads into manageable batches.

#	TOOL	DESCRIPTION
01	`array_chunk`	Splits a large JSON array into smaller sections of a size you specify.
02	`array_deduplicate`	Removes duplicate items from an array, or filters complex object arrays based on a specific unique key.
03	`array_intersect`	Compares two JSON arrays and returns only the items they have in common.

See It in Action

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

U I have two lists of product IDs. Can you tell me which ones are in both?

Intersection Results:

The following 12 unique Product IDs appeared in both the 'Q3 Inventory' and 'Client Wishlist' datasets:

- PID-4092 (Widget Alpha)
- PID-7811 (Mega Sensor)
- PID-9001 (Power Cell X)
- ...
- PID-1123 (Mini Adapter)

U I need to send 5,000 user records to our billing system, but it only accepts batches of 200. Please chunk this data for me.

✅ **Chunking Complete.**

Your 5,000-record dataset has been successfully segmented into the following chunks:

- **Chunk 1:** (Records 1 - 200)
- **Chunk 2:** (Records 201 - 400)
- ...
- **Chunk 25:** (Records 9801 - 10000)

Total chunks generated: 25. You can now process these batches sequentially.

U I have a list of user objects, but some IDs are duplicated. Can you deduplicate this based on the 'email' field?

Deduplication Report:

- **Original Count:** 850 records
- **Unique Records Found:** 792 records
- **Duplicates Removed:** 58 records (via `array_deduplicate`)

The filtered list is ready for the next stage of your pipeline.

Frequently Asked Questions

01 **What problem does Deterministic Array Operations solve regarding large data sets?**

It solves the issue where standard AI models lose context or hallucinate when processing massive lists. This MCP guarantees mathematical precision, allowing your agent to work with datasets that are too big for the model's core memory.

02 **Do I need this if my data sets aren't in JSON format?**

The tool requires structured data (like JSON arrays). If your data is unstructured text, you must clean it first. This MCP works best when handling lists of records or objects.

03 **Can I use Deterministic Array Operations to find common items between two different databases?**

Yes, if you can export the relevant data from those databases into two separate arrays, this MCP can run `array_intersect` on them. It reliably finds all shared identifiers or records.

04 **Is Deterministic Array Operations better than just pasting the list into my AI agent?**

Absolutely. Pasting a list directly risks data loss and is non-deterministic. Using this MCP guarantees that every single item you want to process will be handled correctly by the dedicated engine.

05 **How does it help with API calls? Do I have to code the chunking myself?**

No, you don't need custom code. You simply ask your agent to use the chunking functionality; it handles splitting the data into perfect batches for safe processing.

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 `"deterministic-array-operations": { "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

Deterministic Array Operations 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

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	Deterministic Array Operations MCP
Server ID	`019e3867-746a-703c-8d9a-da5998b28aba`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE