MCP SERVER | NO CODE | CLOUD HOSTED

Button Sizing Engine MCP for AI Agents

Accurate Measurements and Pattern Grading for Garment Production

The Button Sizing Engine helps garment manufacturers solve complex sizing issues quickly. It converts button dimensions between Ligne, millimeters (mm), and inches, calculates necessary buttonhole lengths for pattern making, and suggests ideal spacing based on fabric type. This MCP lets your AI agents perform precise calculations essential for consistent apparel production.

A+ Quality Score 100/100

button-sizing | garment-production | conversion | textile-industry | haberdashery

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/button-sizing-engine

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.

Button Sizing Engine MCP

4 tools available
Cloud-hosted on Vinkius

Designing clothing requires more than just good taste; it demands math. The Button Sizing Engine solves the constant problem of unit inconsistency in garment manufacturing. Instead of juggling different measurement systems, this MCP handles all conversions—Ligne, millimeters, and inches—in one place.

When you connect this to your preferred AI client, you can ask for standard sizing recommendations based on where the button is placed, like a shirt cuff or coat closure. It also calculates exactly how long the buttonhole needs to be, making pattern grading simple. You can even get suggestions for optimal spacing between buttons depending on the fabric weight. It's essential utility data that moves your focus from measurement charts back to design.

Core Capabilities

01 — Convert Button Units

Takes a button dimension and changes its unit from Ligne, millimeters, or inches.

02 — Find Standard Sizes by Garment Part

Determines the correct standard button size for specific areas of clothing like cuffs or closures.

03 — Calculate Buttonhole Lengths

Figures out the minimum necessary length and required margin for a pattern's buttonhole.

04 — Estimate Optimal Spacing

Suggests ideal distance measurements between buttons based on the specific fabric weight used in production.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/button-sizing-engine** — connect your AI agent in three steps.

01 Tell your AI client what measurement you need to change (e.g., 'Convert 20L to mm') or what garment part needs sizing.

02 The MCP runs the specific calculation, checking standard industry guides for sizes and optimal spacing based on parameters like fabric weight.

03 Your agent receives a precise, actionable number, giving you dimensions needed for pattern making and production.

The bottom line is that it provides immediate, industry-standard measurements so your team doesn't waste time looking up conversion charts or sizing guides.

Built For

Anyone involved in the physical process of making clothes needs this. If you spend time adjusting patterns, grading sizes, or dealing with inconsistent unit measurements across different departments, this MCP saves you hours of tedious cross-referencing.

Pattern Maker

Uses the tool to calculate precise buttonhole dimensions and adjust pattern pieces for consistency.

Apparel Designer

Determines standard sizing recommendations for specific closures, like cuffs or collars, ensuring design integrity.

Production Manager

Checks optimal button placement and spacing across large runs of fabric based on weight estimates.

What Changes When You Connect

01 Stop guessing sizes. Use get_button_by_application to instantly find the correct standard button size needed for specific parts, like shirt cuffs.

02 Save pattern drafting time by using calculate_buttonhole_dimensions to get the exact minimum hole length, including necessary thread margins.

03 Avoid costly material waste. estimate_button_spacing suggests ideal button placement based on fabric weight, optimizing your layout before cutting begins.

04 Handle unit conversion without errors. convert_button_unit handles Ligne, mm, and inches so you never have to worry about conflicting measurement systems again.

05 Increase consistency across product lines. By getting standardized sizing for closures, every garment maintains a professional, uniform finish.

Real-World Applications

I need button sizes for my new blazer line.

Instead of checking multiple style guides, I ask the agent what size is best for coat closures. It uses get_button_by_application to give me the standard sizing needed across all models.

The pattern piece buttonhole length keeps changing.

I run into issues where my physical measurements don't match the required hole size. I use calculate_buttonhole_dimensions, and it gives me the minimum necessary length with a safety margin for stitching.

How far apart should buttons be on heavy denim?

The spacing seems arbitrary in my current process. I run estimate_button_spacing, feeding it the fabric weight, and it suggests an optimal distance to keep the garment looking balanced.

Patterns to Avoid

Mixing up unit systems

✗ AVOID

Manually trying to convert a measurement from Ligne directly to inches using a simple online calculator, only to find it's missing the correct conversion factor.

✓ INSTEAD

Use the convert_button_unit tool. It handles all three major units—Ligne, mm, and inches—in one go, guaranteeing accurate results every time.

Ignoring garment specifics

✗ AVOID

Choosing a generic button size for a shirt cuff without knowing the specific standard required by that garment part.

✓ INSTEAD

Always use get_button_by_application. It looks up the recommended sizing based on the actual component, ensuring industry compliance.

Underestimating hole length

✗ AVOID

Cutting buttonholes too short because I forgot to add a thread safety margin to my pattern pieces.

✓ INSTEAD

Run calculate_buttonhole_dimensions. This tool calculates the full, necessary length for you, including the vital thread margin.

The Right Fit

Use this MCP if your workflow relies on accurate measurements across multiple units (Ligne, mm, inches) or requires standardized sizing data specific to garment parts like cuffs and closures. It's perfect for pattern makers and design teams who need consistent technical specs.

Don't use it if you are only making a simple product that never has buttons or closures, as the conversion tools won't apply. Also, don't rely on it for creative decisions—it gives measurements; your team still needs to decide *where* those buttons go. If all you need is basic unit conversion without context (i.e., not related to garments), a general conversion utility might suffice, but this one provides the industry-specific rules.

Button Sizing Engine for Accurate Pattern Grading in Garment Production

Right now, sizing button components is an absolute mess. You're constantly flipping between spreadsheets and manuals, cross-referencing whether a cuff needs 16L or something else, then having to manually recalculate that measurement into the other two systems (mm or inches) just to plot it on your pattern piece.

With this MCP, you ask for sizing by application, like 'shirt cuff'. It spits out the standard size, and if you need it in millimeters, it converts it instantly. You get precise, ready-to-use measurements without lifting a finger or opening another tab.

Button Sizing Engine for Optimal Spacing Calculations in Haberdashery

Before, determining button spacing was guesswork. You'd measure arbitrary distances and hope the fabric weight supported it. If you got it wrong, your whole piece looked off-kilter or structurally weak.

Now, simply tell your agent about the fabric weight. It runs an estimate_button_spacing check and gives you a suggested ideal distance. It makes your pattern grading predictive instead of reactive.

4 Tools for Button Sizing, Conversions, and Pattern Grading

These tools let your AI agents find standard sizes, calculate hole dimensions, convert units, and suggest optimal button spacing for apparel manufacturing.

#	TOOL	DESCRIPTION
01	`get_button_by_application`	Provides the standard recommended size for a specific garment part (e.g., cuff or closure).
02	`calculate_buttonhole_dimensions`	Calculates the minimum required length and margin for a buttonhole in a pattern.
03	`convert_button_unit`	Changes any given button dimension from one unit (Ligne, mm, or inches) to another.
04	`estimate_button_spacing`	Suggests the ideal distance between buttons based on the weight of the fabric being used.

See It in Action

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

U I need to convert 32L button size into millimeters.

Conversion Result:
- **Input Unit:** Ligne (L)
- **Target Unit:** Millimeters (mm)
- **Result:** 8.13 mm

This is the standard conversion for that measurement.

U What's the right button size for a coat closure?

Standard Sizing Guide: Coat Closures
- For standard adult coats:
* Recommended Size: 20L
* Approximate Metric: 5.1 mm
* Industry Standard Range: 4.8mm - 5.3mm

Always check the specific garment type, but 20L is a reliable starting point.

U My fabric is medium-weight denim. How far should I space my buttons?

Button Spacing Estimate
Based on your input (Medium Denim Weight):
- Optimal Distance: 18 mm
- Acceptable Range: 16mm - 20mm

Keep the spacing consistent. This distance ensures both aesthetic balance and structural integrity for denim.

Frequently Asked Questions

01 **How does the Button Sizing Engine handle unit conversions?**

It handles all three units—Ligne, millimeters (mm), and inches. You just tell it what to convert, and you get an accurate result without having to check multiple conversion charts.

02 **Can the Button Sizing Engine help me choose a size for my cuffs?**

Yes. It uses standard industry knowledge to recommend the correct button size based on whether it's for a cuff, collar, or other specific garment part. This ensures your design is compliant.

03 **What do I need to provide to calculate the buttonhole length?**

You primarily need the diameter of the button itself. The engine then calculates the minimum required hole length, adding a necessary safety margin for stitching and pattern grading.

04 **Does this MCP help with spacing buttons on different fabrics?**

Absolutely. You input the fabric weight or type, and it suggests an optimal distance between buttons to keep your garment looking balanced and structurally sound.

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 `"button-sizing-engine": { "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

Button Sizing Engine 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 Button Sizing Engine. 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	July 2026
MCP Server	Button Sizing Engine MCP
Server ID	`019f2470-3ded-72e0-877f-ba0143801248`
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/button-sizing-engine.