

Connectors

The on-device MCP layer. Cephable plugs into your tools both ways — and every model runs on the client.



INBOUND · MCP CLIENT

Register any MCP server

- Launch set: M365, HubSpot, Salesforce, GitHub, Atlassian, Canva + more
- Transports: stdio (command-line) and HTTP + SSE / streaming
- Auth: OAuth (client ID + secret), API key, or none
- Start from samples & templates, or build custom



INBOUND · MCP CLIENT

Hit from the client. No proxy.

- Every Cephable model runs 100% on device
- Servers are called from the client, never a Cephable proxy
- Internal firewalled servers work over your VPN
- Returned data hits the local model. Never sent out or trained on.



INBOUND · MCP CLIENT

Context & token control

- Per-connector tool selection — load only what you need
- A local non-LLM model ranks tools per request
- Tools load dynamically, so many servers run on low memory
- Big responses filtered & summarized before reasoning



GOVERNANCE

Governed at the endpoint

- Allow / block public connectors org-wide, per-user exceptions
- Pre-provision once — the user just adds and signs in
- Approve each remote tool call; see data in and out
- Monitor connector & skill usage across users and devices

OUTBOUND · MCP
SERVER

Cephable is an MCP server

- Its own remote MCP server issues work down to your devices
- Core tool: “run a workflow on this device,” streams feedback back
- Agent-to-agent: hand work from M365 Copilot to the machine
- Also via API (OpenAPI/Swagger), mobile app & scheduled runs



UNDER THE HOOD

The on-device engine

- Cephable Gemma 4 fine-tunes: S / M / L / XL / XXL (XL/XXL MoE)
- 4-bit quantized, silicon-optimized builds, auto-selected
- Uses CPU, GPU & NPU, right silicon per workload
- ISO 27001 · SOC 2 · HIPAA (self-attested)