

From stdio to Anywhere: Securely Exposing OpenClaw as a Remote MCP Server

Nick Taylor · @nickytonline · MCP Dev Summit Toronto 2026

What is OpenClaw?

Open source, self-hosted personal AI assistant. One Gateway process, on your own hardware.

Chat, agents, channels, approvals, all under your control.



OpenClaw Auth Modes Pre-Mid February

Gateway auth modes: `none` , `password` , `token` .

Needed it reachable beyond your machine? Common pattern:
Tailscale Serve (tailnet-only), or just stay on loopback, local only.



feat(gateway): add trusted-proxy auth mode #15940

Merged

by [steipete](#) on Feb 14

main



[nickytonline:feat/trusted-proxy-auth](#)



v2026.2.14

 Conversation 7

 Commits 35

 Checks 24

 Files changed 28



[nickytonline](#) on Feb 13 • edited

Contributor



Summary

Adds a new `trusted-proxy` auth mode that delegates authentication to a reverse proxy (Pomerium, Caddy, nginx + OAuth, etc.). This allows Clawdbot to run behind identity-aware proxies or reverse proxies without requiring token auth in WebSocket payloads.

Closes [#1560](#)

Relates to [#1710](#)

Changes

Types & Schema

- Add `trusted-proxy` to `GatewayAuthMode` union
- Add `GatewayTrustedProxyConfig` type with `userHeader`, `requiredHeaders`, `allowUsers`
- Update zod schema with validation

Reviewers – [review](#)

1 more reviewer



greptile-app

Assignees



steipete

app: macos app

docs

gateway

Development

 [Feature]: Allow

Notifications



None

You're receiving no



Trusted-proxy auth mode

A generic abstraction: the OpenClaw Gateway trusts identity headers from a proxy you configure, checking, among other things:

1. the source IP is in `gateway.trustedProxies`
2. the required headers are present
3. the user header names an allowed user

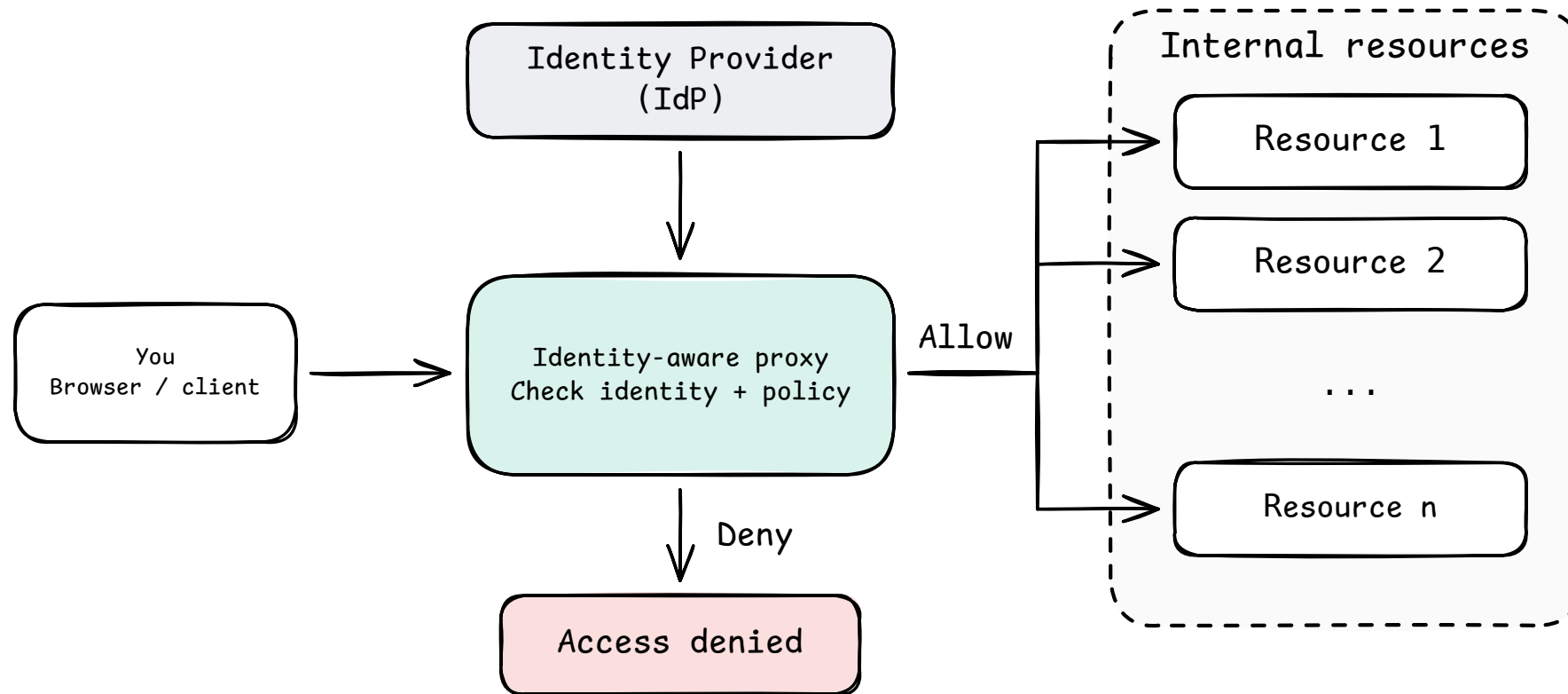


What's an identity-aware proxy (IAP)?

- Checks identity and policy on **every request**, not just once at login
- No VPN, no trust based on which network you're on
- Great at securing internal resources



What's an identity-aware proxy (IAP)?



Trusted-proxy auth mode: not just IAPs

Any proxy that authenticates and forwards identity headers qualifies.

Some concrete implementations:

- Caddy + OAuth plugin
- Cloudflare Access
- nginx + oauth2-proxy
- Pomerium
- Traefik forward auth



My Homelab OpenClaw



claw.maisonlab.dev



Wait... isn't this an MCP conference?



OpenClaw already has an MCP server

`openclaw mcp serve` exposes conversations, messages, events and approvals as tools over stdio.

docs.openclaw.ai/cli/mcp/serve



OpenClaw already has an MCP server

Example stdio client config, straight from OpenClaw's own docs. Generic, works with most MCP clients:

```
{  
  "mcpServers": {  
    "openclaw": {  
      "command": "openclaw",  
      "args": ["mcp", "serve"]  
    }  
  }  
}
```



Gap 1: only if your harness can spawn it

stdio means the client has to **spawn** `openclaw mcp serve` **itself**, as a local subprocess, wherever the client runs.

ChatGPT.com, Claude.ai, and other hosted clients **have nowhere to spawn anything.**



OpenClaw remote Gateway != remote MCP

`openclaw mcp serve --url wss://...` still runs MCP over stdio, locally. Only the hop to the Gateway is remote, over OpenClaw's own WebSocket protocol.

It's still stdio MCP, just more complex.



Software and secrets on every machine

- install and update the OpenClaw CLI
- distribute and rotate a Gateway secret, the same one for everyone

We want it secured, per user. Remote MCP + OAuth is built for that.



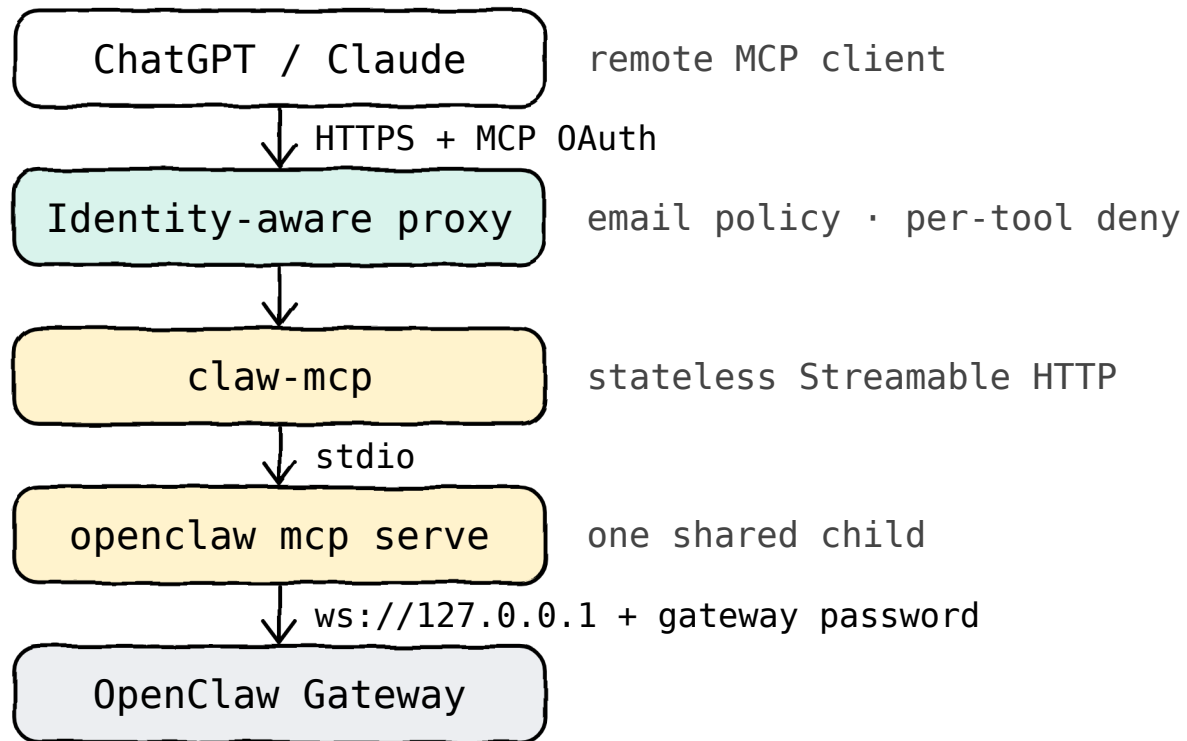
Don't fork OpenClaw. Wrap it.

claw-mcp: a small Streamable HTTP server that lives *outside* OpenClaw.

- Spawns the unmodified `openclaw mcp serve` and proxies its tools
- Adds read-only `workspace_*` tools for the agent's files
- Sits behind an identity-aware proxy, which does all the OAuth



Demo: the architecture



Wrapping a stdio tool call

```
async callTool(params: CallToolRequest["params"]) {  
  // Exclusion is enforced here too, not just hidden from tools/list.  
  const tool = (await this.listTools()).find((t) => t.name === params.name);  
  if (!tool) throw new ProtocolError(...);  
  ...  
  const client = await this.#connect();  
  return client.callTool(  
    { name: params.name, arguments: params.arguments },  
    // events_wait may block for up to 300s upstream.  
    { timeout: this.#callTimeoutMs, toolDefinition: tool },  
  );  
}
```



Policy at the edge

```
- from: https://clawmcp.demo.iamdeveloper.com
  to: http://openclaw-gateway:3000
  mcp:
    server: {}
  policy:
    - allow:
      and:
        - email: { in: [nick@nickyt.co, ntaylor@pomerium.com] }
    - deny:
      or:
        - mcp_tool: { in: [permissions_respond] }
```

The tool still shows up in the list, but **every call to it is denied**.



Demo



Gap 2: identity stops at the wrapper

In trusted-proxy mode the Gateway accepts:

- identity forwarded by a trusted proxy, **or**
- a password from **loopback**

`mcp serve` can't send proxy headers, so the bridge sits on loopback with the password.

The proxy verifies every user. The Gateway sees one operator.



The fix belongs in OpenClaw

openclaw/openclaw#161168: let `mcp serve` send identity on connect, like the TUI and `gateway call` already do.

```
openclaw mcp serve --url wss://gateway.internal \  
  --header x-forwarded-user=<verified user> \  
  --header x-forwarded-for=<client ip>
```

I opened the issue, then a draft PR. With it, the wrapper can run **one bridge per user**.



Caveats, for now

- **I have a POC draft PR.**
- **Device pairing:** each new user's bridge needs one admin approval. Trusted-proxy auto-approval only covers browser and native UI clients.
- So I opened a follow-up, linked from #161168: **opt-in auto-approval for mcp serve bridges**, with the same scope cap as browsers.



Where this could go

OpenClaw's Gateway could be a remote MCP server itself, with no stdio wrapping.



Open source, start to finish

1. **Trusted-proxy auth mode** ([#15940](#)). The community made it production-grade.
2. `mcp serve` couldn't pass identity: [#161168](#) and a draft PR.
3. Building this found the next gap: **auto-approval** for bridges.



Resources

- [Trusted-proxy auth mode PR \(#15940\)](#)
- [mcp serve identity issue \(#161168\)](#)
- [OpenClaw trusted-proxy auth docs](#)
- [OpenClaw mcp serve docs](#)
- [OpenClaw on GitHub](#)



Resources

- [claw-mcp](#): the wrapper from the demo
- [MCP TypeScript template](#): the starting point
- ["Claws Out" at AI Engineer Europe 2026](#)

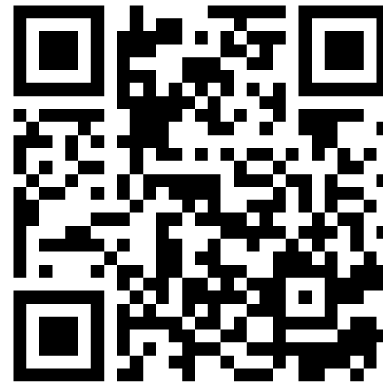


Thank You!

Nick Taylor · @nickytonline



nickyt.co



[slides](#)