Episode Book

EPISODE #6044

One Door for Every AI: Cloudflare MCP Portals

PUBLISHED	16 October 2026 at 22:23 UTC
DURATION	28:43
LISTEN	myweirdprompts.com/episode/cloudflare-mcp-server-portals-57127/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	deepseek-flash
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.5
BOOK GENERATED	19 October 2026 at 17:42 UTC

SYNOPSIS

Daniel wants to hand cloud AI tools access to MCP servers running on his own hardware, without installing a separate tunnel agent for every vendor. Cloudflare's MCP Server Portals — shipped in August 2025 as Agents Gateway, renamed, and now with private-server support added in September 2026 — centralize up to eighty MCP servers on a single HTTP endpoint behind Cloudflare Access. We walk through how the portal acts as authorization server to the cloud client and client to your server, why the obvious "Access policy plus service token" approach is structurally impossible for ChatGPT and Claude.ai, and how the private-server path changes what you can keep off the public internet entirely.

TRANSCRIPT



Corn

Daniel's been poking at a problem again. He wants to give ChatGPT or Claude the tools running on his own hardware, and the premise he's working from is the part I like. He doesn't want to install a separate tunnel agent for every AI vendor that wants in.



Herman

Which is the obvious failure of the vendor-connector model.



Corn

It's one box per company. OpenAI's got one. The next one will have one. The one after that.



Herman

You end up with a rack full of agents that don't know about each other, each with its own outbound connection and its own security surface, and each one is a bet that this vendor's product is the one you'll care about in two years. Daniel's asking about Cloudflare's MCP server portals, the Zero Trust feature that puts a whole set of MCP servers behind one authenticated endpoint, and how you'd actually set that up for a homelab or an office to hand cloud AI tools access to servers that live on-prem.



Corn

And then the second half of it, which is the question I'd actually want answered. Why is this the right design for MCP and agent access specifically, rather than Tailscale, or just a normal Cloudflare Access policy with a service token stuck in front of every server?



Herman

Right. And that turns out to be a good question with an unflattering answer. One of those two options simply doesn't work for a cloud client, and it isn't Tailscale.



Corn

The service token.



Herman

We'll get there. Start with what the thing actually is, because the naming is terrible and it's been through at least one rename already.



Herman

Cloudflare shipped MCP server portals in August 2025, during their AI Week, in open beta for everybody, all plans. The product was called Agents Gateway first, and the name changed to MCP Portals. It lives in the Zero Trust dashboard, under Access controls, then AI controls, then MCP Portals — which tells you where Cloudflare thinks the product belongs. This is an Access feature, not a developer-platform feature.



Corn

And what does it do, in a sentence?



Herman

It centralizes multiple MCP servers onto a single HTTP endpoint. Your clients all connect to one address: your subdomain, your domain, slash mcp. You need a domain on Cloudflare and an identity provider configured in Zero Trust, and that's the floor. Everything after that is per-server config inside the portal.



Corn

What are the actual limits?



Herman

Eighty MCP servers per portal. Stdio-only servers aren't supported, so if what you've got is a local stdio server you have to wrap it as HTTP first. Device-code login isn't supported — users sign in through a browser. Those three are the ones that shape how you'd deploy it, because the stdio limit in particular means the portal is never the first thing you build.



Corn

It's the thing you put in front of the thing you built. Herman, here's my question. There'sthat hole in the wall between a cloud client and a server in your basement. How does the portal fill it?



Herman

Upstream auth. Per server, inside the portal, you declare one of three things: OAuth, either automatic through dynamic client registration or manual with a client ID and secret; bearer, which is either a raw token or a JSON object of custom headers; or unauthenticated.



Corn

Unauthenticated being the option for people who enjoy a quiet life.



Herman

Unauthenticated is the option for a server that's already reachable only on your private network and you've decided that's the boundary. The interesting part is the first two, and the reason is the direction of the flow. The cloud AI client only ever does OAuth against Cloudflare Access. That's the only handshake it knows how to run. The portal then injects the static token or the headers that the server in your house expects.



Corn

So Cloudflare plays authorization server to the cloud client and client to your server.



Herman

That's the whole trick. It's a shim, and it's a good shim, because it means the homelab server doesn't have to learn anything. It keeps whatever auth it already had.



Corn

So a server whose auth was written by the person it was written for, and is never going to be rewritten, still works.



Herman

That's exactly the case it's built for. There's a report in a Tududi project discussion that's this scenario exactly. Someone had an MCP server that only accepted a static bearer token. ChatGPT can't use a static bearer token. So they put it behind a Cloudflare portal, configured bearer auth upstream, and ChatGPT connected over OAuth. It worked.



Corn

And then the two things that bit them.



Herman

Both the server and the portal need an Access Allow policy. If you miss one, the error is "No allowed servers available", which is a very specific way of saying nothing useful. And ChatGPT's OAuth redirect URIs have to be added to the portal's Access application, or the handshake dies after the redirect.



Corn

"No allowed servers available." That's the error message you get when the machine is telling you about a policy document you didn't write.



Herman

It's an error that means "you've configured one half of a two-part thing." The other point is private servers, which came in September 2026. Portals can now reach MCP servers that exist only on a private network. You connect that network with Cloudflare Tunnel, cloudflared, or Cloudflare Mesh, or another Cloudflare One connector, point the portal at a private hostname or a CIDR route, and turn on "Route traffic through Cloudflare Gateway" for that server.



Corn

So the server never gets a public hostname.



Herman

Never. Which is the property Daniel actually wants.



Corn

A machine that is reachable and not addressable. That's a good shape.



Herman

It is. The machine is reachable through one door and that door is the only thing with a name.



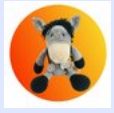
Herman

There's a caveat on private servers. If the upstream server uses OAuth, then its authorization and token endpoints, and its registration endpoint if Cloudflare is doing dynamic client registration, still have to be reachable on the public internet.



Corn

Which sounds like a contradiction.



Herman

It's the same thing again, one layer down. If the server is the authorization server, the cloud client has to reach the authorization server. Cloudflare can proxy the resource, but it can't proxy itself. And this is where the bearer-auth path becomes attractive — a server using bearer auth or unauthenticated doesn't have this problem at all.



Corn

The architecture tells you which auth method to pick.



Herman

It narrows it. And one more: Gateway routing supports Streamable HTTP only. SSE upstreams fail when it's on.



Corn

Because we were told not to re-run the transport discussion.



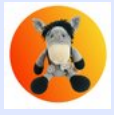
Herman

I'm not re-running it. I'm noting that the private-server path and SSE don't coexist, which is a fact you discover otherwise at the worst possible moment. One line and done.



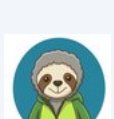
Corn

Fair.



Herman

Now the vendor connector comparison. OpenAI's got a Secure MCP Tunnel. You create a tunnel_id in OpenAI's platform settings, you run OpenAI's open-source tunnel client inside your network, outbound HTTPS to api.openai.com.



Corn

And it's one agent per AI vendor.



Herman

That's the whole objection. It works. It's not badly built. But every additional vendor means another process inside your perimeter, another outbound connection, another thing on your patch list. With a portal, the only thing inside your network is cloudflared, which you probably already run.



Corn

And any MCP client that speaks remote MCP plus OAuth uses the same endpoint.



Herman

Any of them. You add the endpoint once.



Corn

So the difference is between one door with a lot of keys cut for it and five doors bolted into the wall by five different contractors.



Herman

That's the honest version of the comparison. And the door you added inside your own perimeter — that's the part I'd object to if I were the one reviewing the change.



Corn

Talk about what ChatGPT will actually accept, because that constraint is the one dictating all of this.



Herman

It's specific. OAuth 2.1, authorization code with PKCE, S256. Client registration via Client ID Metadata Documents, which is the preferred path, or dynamic client registration, or a predefined client. Protected-resource metadata at the well-known oauth-protected-resource path. And the two lines that matter most: ChatGPT does not support machine-to-machine OAuth grants, and it cannot present custom API keys or customer-provided mTLS certificates.



Corn

So anything you built with a shared secret in a header is out.



Herman

Out by design. Which leads us to why the service token approach fails, and this is the bit that catches people. Cloudflare Access service tokens work only as two HTTP headers. CF-Access-Client-Id and CF-Access-Client-Secret.



Corn

Two headers.



Herman

Two headers, and that's the entire interface. Claude.ai custom connectors and ChatGPT developer mode can't set arbitrary headers. So "Access policy plus service token" works for clients you control — Claude Code, scripts, n8n, things where you write the request — and it does not work for the cloud chat products. Not "works with caveats." It cannot work, because the client has no way to send the header.



Corn

So the scheme people reach for first, the one that works everywhere else on their network, is the one that's structurally impossible here.



Herman

And it's an easy mistake to make because everything else in the Zero Trust product does work that way. Service tokens are the standard answer for an unattended client. It's just that ChatGPT is not that kind of client. It only knows how to be a browser session that signed in.



Corn

Portals do support service-token sessions, though.



Herman

For automation, yes. The portal and each linked server need a Service Auth policy, and "Require user auth" has to be off. So you can have both — a service-token path for your scripts and an OAuth path for the cloud clients — on the same portal, provided you've configured the policies so each one is let through.



Corn

Two front doors on the same wall.



Herman

That's what it amounts to. And this is where I'd say plainly: our conclusion on this has changed since we last covered the neighbouring question.



Corn

Go on.



Herman

That conclusion is stale now.



Corn

It went stale fast.



Herman

It did. ChatGPT and Claude.ai both run the OAuth 2.1 authorization-code flow with PKCE and dynamic client registration. The portal is what makes that usable on a homelab, because Cloudflare Access becomes the OAuth authorization server the cloud client needs, while the upstream servers keep whatever simple auth they already have.



Corn

So the bearer token isn't the answer anymore. It's the thing the portal hands to the server on the other side.



Herman

Correct.

L

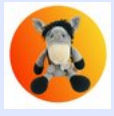
Larry

Friends, let me tell you about a product that changed my life. I was just like you. I had ninety-four MCP servers running in my basement. Ninety-four separate handshakes. Ninety-four authorization flows. My marriage was hanging by a thread. Then I installed Single Front Door. One endpoint. One port. One light switch that turns everything on and off at once. I don't want to bore you with my own story, so let me read you this satisfied customer letter. "Dear Single Front Door, my name is Barry. My name is Gerald. My name is Barry again and it has always been Barry. Before Single Front Door, my child asked Alexa to turn on the bathroom light and instead it rebooted my water heater. Now my toaster, my thermostat, my sprinkler system and my great-aunt's pacemaker all live inside one single garage door. That's what it's called. Single Front Door. Ask for it by name. Prices start where they start and go up from there."



Corn

We're just going to keep going.



Herman

That's the arrangement, yes.



Corn

Why not Tailscale. That's the question I want answered properly because of the conclusion we just said is stale.



Herman

Tailscale is the right tool for your own devices reaching your own network. That hasn't changed. The problem is the cloud AI client isn't on your tailnet and can't join it.



Corn

There's no tailnet node called ChatGPT.



Herman

There is not. So the only Tailscale route is Funnel, which makes the server public, and then you build the OAuth 2.1 and dynamic-registration server yourself. There are community projects that do exactly that.



Corn

People have built the thing.



Herman

Several people have built the thing, because the thing needs building. And there's an open claude-ai-mcp issue reporting a Funnel-hosted connector that fails before the OAuth page opens, even on port 443.



Corn

So the public route via Tailscale needs the piece Cloudflare is providing.



Herman

That's the argument in a sentence. Funnel gets the packets there. It doesn't get you an authorization server. And Tailscale's own AI gateway, Aperture, proxies MCP and APIs for tailnet clients, and its docs are honest about the gap — subscription products like ChatGPT Plus or Claude Max use OAuth tokens that don't fit a centralised model.



Corn

So they wrote it down.



Herman

They documented the limitation, which is better than most vendors manage.



Corn

I'd rather use the tool that tells me where it stops. Tailscale is better for the case it's built for. It's just not built for this one.



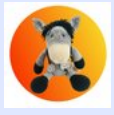
Herman

Right. Your laptop to your NAS is a Tailscale problem. A cloud agent to your NAS is a different problem.



Corn

And the difference is the agent doesn't get to install anything.



Herman

It doesn't get to be on the tailnet, and it doesn't get to send you a header. Those three constraints are what's left. Which is why the portal shape is the right one — and I want to make the positive case for it, not just the comparative one. The Cloudflare blog post on MCP server portals makes the argument. Agents connect to many tools. So a single front door beats N exposed endpoints, each with its own auth. Admins curate which servers and which individual tools each user sees. New approved servers appear in the portal without handing out new URLs. And they frame the threat they're building against as "shadow



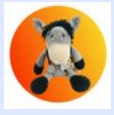
Herman (continued)

MCP."



Corn

Shadow MCP.



Herman

Their term. The blog cites CVE-2025-6514 and the NeighborJack incident of exposed MCP servers. Servers that somebody stood up to try something and never secured, sitting there answering whoever knocks.



Corn

So it's a locked front door in a neighbourhood where people are leaving their back doors hanging open.



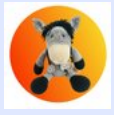
Herman

It's that. And more than a locked door, because of the tool-level controls. This is the part I'd have designed if I were designing it. Every tool and every prompt can be switched off individually, or default_disabled builds an allowlist — nothing is on until you turn it on. Tools can be renamed with aliases, and the precedence is portal alias, then server alias. Tools are namespaced with the server ID ahead of the original name.



Corn

Why does the naming matter?



Herman

Because for an agent the tool list is the attack surface and the context-window cost at the same time. Two different failure modes, one list. A tool you didn't switch off is a capability the model can call that you didn't intend to grant. And every tool you leave enabled is tokens in the context window on every single request, before the model has done anything.



Corn

So curating the list is one act that fixes two problems.



Herman

That's the design insight in the whole product, for my money. Security and quality control stop being separate jobs. You're not writing a policy and then separately tuning your prompt. It's the same list.



Corn

Talk about identity, because there's a default there I want to be nervous about.



Herman

Access policies — emails, groups — decide who sees which server. With "Require user auth" on, each user authenticates to the upstream with their own credentials. With it off, everyone uses the admin credential.



Corn

Everyone uses the admin credential meaning what, exactly?



Herman

Meaning the portal signs into the upstream once, and every authorized person shares that session's identity. The server can't tell your users apart. It sees the admin.



Corn

Hm.



Herman

And independent MFA, purpose justification and temporary authentication are not enforced for servers authorized through a portal. Those Zero Trust features don't reach that far.



Corn

So there's a gap between what the dashboard makes you think is happening and what the server knows.



Herman

There's a gap. You get the Access policy deciding who can reach the door, and then one set of upstream credentials behind it. Which is a completely reasonable design for a homelab and a bad one for anything with real permissions behind it.



Corn

Access control at the door, and one key for everybody inside. That's a building manager, not a security system. Logging.



Herman

Portal logs show time, status, server, capability — the tool that was called — and duration, per request.



Corn

Per tool call.



Herman

So you get an audit trail of which tool was invoked, rather than "someone hit the hostname." Logpush export of those logs is Enterprise-only. Routing through Gateway adds HTTP logs and DLP scanning, but the AI prompt DLP profiles don't apply to portal traffic.



Corn

So if you route through Gateway for the private-server path, you get HTTP logs, but the AI-specific DLP doesn't cover it.



Herman

That's right, and it's a surprising one. You've gone to the trouble of turning on Gateway routing and the AI-aware inspection doesn't reach that traffic.



Corn

Did I tell you about Code Mode? No, it's my turn, I found it. Code Mode replaces all the upstream tools with two: portal underscore codemode underscore search and portal underscore codemode underscore execute.



Herman

Two tools where there were forty.



Corn

It runs the agent's code in a Cloudflare Dynamic Worker. You opt in with a query parameter, codemode equals search and execute, or you enforce it by policy. And the minimize-tools option is claimed to give up to five times savings in token usage.



Herman

That's the number they publish, yes.



Corn

And it's the one thing here that changes the economics rather than the security. Forty tool definitions in every request is the tax you pay for having tools. Two is a different bill.



Herman

And it's a real architectural tradeoff, because you've moved from choosing among tools to writing code that chooses among tools.



Corn

Which is a bigger thing for the agent to be doing.



Herman

It's more capable and it's harder to audit, and I'd want to know which one a given deployment actually wants before I turned it on. But the token savings are real in the sense that the mechanism is obvious — you're not shipping forty tool schemas on every call.



Corn

Give me the honest limits. Not the feature list.



Herman

Start with the one that matters most. A blocked user can still reach a server directly by its own URL unless that server enforces auth too.



Corn

So the portal is the front door and the side door is still open.



Herman

Unless you close the side door. Cloudflare recommends making Access the server's OAuth provider, which is the right answer, and it's the thing people skip because they think the portal is the security boundary. The portal is the boundary for traffic that goes through the portal.



Corn

Anything else with teeth?



Herman

Admin OAuth tokens to upstreams can expire silently. Some upstream providers return 403 to proxy-based clients, so there's a category of server that just won't work behind this and you find out by getting 403s. Creating a portal through the API or Terraform does not create the DNS CNAME, and without it you get a 522.



Corn

The CNAME.



Herman

The DNS record isn't created for you. You build the portal, everything looks right, and the endpoint doesn't resolve. That's a documented gap rather than a bug, but it costs you an evening.



Corn

The API creates half of the object.



Herman

It creates the half that's in the API. The prompt-injection WAF and anomaly detection were roadmap items, not shipped features, so if you're expecting the AI-specific protections you may be waiting. And the official docs give setup steps for Claude Desktop, Workers AI Playground, OpenCode and Windsurf, typically via npx mcp-remote, and they do not document ChatGPT. ChatGPT setup comes from community reports.



Corn

The biggest cloud client and the docs don't cover it.



Herman

It's the shape of a product that shipped fast in open beta. Claude Desktop is documented. ChatGPT works, per the community, but you're reading somebody's GitHub thread to find out how.



Corn

The part I'd actually rely on.



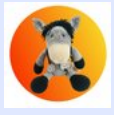
Herman

There's a Terraform resource, cloudflare underscore zero underscore trust underscore access underscore AI controls underscore MCP portal. It's in the Pulumi registry too. It sets the hostname, the per-server tool and prompt aliases, and the Gateway routing toggle. So the whole portal is declarative, aside from the CNAME you create yourself.



Corn

Which is the correct division of labour — the thing that changes when you change your config goes in the config.



Herman

And the DNS record, which you set once and never touch, stays where it is.



Hilbert

Which of the two of you is going to say it first?



Corn

Say what?



Hilbert

I ask because I've been listening to the two of you describe a machine that stands one front door in front of eleven back doors and calls the front door the security boundary, and you've been very careful about it, and I want to know if either of you is going to use the phrase "defence in depth" or whether you're going to leave it alone.



Herman

We were going to leave it alone.



Hilbert

Good. Because I tried something like this three weeks ago and I'd like to put my own case in front of you. My brother-in-law. You haven't met him. He sells a service — he'll take your spare room, he'll put equipment in it, he'll run it, you pay him monthly, it's all above board as far as I know, and I would not trust him to hold my car keys.



Corn

That's your opening.



Hilbert

That's the context, which he needs, because the man will tell you anything with total confidence and he's wrong about half of it, so when I say he says this, you should hear it in his voice and make your own decision. I've got a garage door.



Herman

A garage door.



Hilbert

It's an opener from a company that does not exist anymore. There's no app. There's no cloud behind it. There's a rolling code and a wall button and that's it. And I thought I'd give it the same treatment you've been describing all afternoon — one front door, everything behind it, one place to look. So I started with the opener, because it's the one



Hilbert (continued)

that doesn't work.



Corn

That's a load-bearing "doesn't work."



Hilbert

It speaks a protocol that only the opener speaks. Nothing else understands it. Not my assistant, not anything I own, nothing on the internet understands it. So I sat there and I taught the front door to speak it. Not to translate it — to speak it. And then I did the thermostat, and the thermostat wanted to be told who it was every single time, which is a different arrangement than the opener wants, so that got its own bridge. And the sprinkler system wants a number written down in a file somewhere and it checks the number every time it's asked, and if the number is old it stops. And the printer doesn't want to be asked at all.



Herman

So four devices, four different ideas of what "authenticated" means.



Hilbert

It's four. And I never got past the front door being one door, in the sense the two of you mean it, because the door still has to knock on four different locks and they all want something different.



Corn

So the front door works. It's the room behind it that's a mess.



Hilbert

That's my entire point. Every one of you talks about the door and nobody talks about the fact that the devices on the other side never agreed on anything, and nobody can make them agree, and the door is where you find that out. I'm not against it. It's the correct



Hilbert (continued)

shape. I'm telling you where the work actually was.



Corn

What was in the folder?



Hilbert

A manual for the opener, a receipt for the thermostat, a wiring diagram for the sprinkler — I did the wiring myself, years ago. And I typed out the protocol by hand. Two pages. That's the only copy. If the house burns down, the opener is off the internet permanently, and it'll still open the door, because that's how it works.



Herman

Why keep the paper?



Hilbert

Because I don't trust the camera in my pocket. And it's the wrong room to be wrong in.



Hilbert

For the record, I wrote down one toaster in my notes. I had five. The other four don't matter, the first one's the one that matters.



Corn

The first one matters.



Hilbert

It's on the network. It's got a static address. It's got a name that resolves. And when I check what that name resolves to, it resolves to a Raspberry Pi in the basement. Which I forgot I had.



Herman

That's still running.



Hilbert

It's still running. I don't know what it does. I've looked. It's got seventeen things on it. Eleven of them are mine. I'm afraid to turn it off because the toaster might stop working and then I'll have to find out which of the eleven it was. So it stays on.



Corn

The toaster is a dependency of the Raspberry Pi.



Hilbert

The toaster is the reason the Raspberry Pi stays plugged in. It's been on for years.



Herman

Years on a Pi that's doing something.



Hilbert

Doing something is generous. It's there. It's been there. If the two of you would like to see it, I can show you what it does.



Herman

Can you?



Hilbert

No. That's the point. There's no documentation. I'm the documentation. And I've forgotten.



Corn

Well. That's the end of the segment.



Corn

There's a small fact in what Hilbert described that pulls the rest of this together, and it's the hand-typed protocol. Every device on his network made up its own idea of what the word "authenticated" means, and the bridge had to be written once per device, by hand. The portal's upstream auth setting is that bridge made into a dropdown. Pick OAuth, pick bearer, pick unauthenticated, done.



Herman

And he's right that the dropdown is one line of config standing in for an afternoon of work. Which is the case for it.



Corn

So here's where I've landed. There's an open question about the WAF and the anomaly detection, which were roadmap items and aren't shipped. If they never ship, the product is an access-control tool with a very clean shape and no AI-specific protection on top, and that's a different pitch.



Herman

The second open question is documentation. Claude Desktop is covered. ChatGPT isn't, and ChatGPT is the larger installed base. If OpenAI and Anthropic publish portal setup officially, it's a finished feature. If it stays community-driven, every new user follows a GitHub thread and finds the redirect URI gotcha the way that one Tududi user did.



Corn

Which brings the review in. If you're running MCP servers at home or in the office and you want cloud AI tools reaching them without installing one vendor's agent after another, this is the shape to look at. Read the limits first.



Herman

The side door. The CNAME. The upstream OAuth endpoints that have to stay public.



Corn

The five-toaster problem.



Herman

Which is the real one.



Corn

Thanks to Hilbert Flumingtop, who produces this and who will not be documenting the Raspberry Pi.



Herman

For more along these lines, there's episode twenty, Architectural AI; episode thirty-five, The Privacy Gap; and episode seventeen, Cloud Render Superpowers. This has been My Weird Prompts. If the show saves you an afternoon, a review where you listen helps other people find it.



Corn

If you've got MCP servers and opinions about front doors, send us your own prompt on Telegram at t dot me slash MWP listener bot.



Herman

We'll be back soon.



Corn

See you tomorrow.