major labs

RESEARCH REPORT · AGENTIC WEB INFRASTRUCTURE

The State of MCP

What the protocol unlocked, how far it spread, and the trust layer it still lacks.

A field scan of 2,348 Model Context Protocol servers. May 2026

Editorially independent of Major Matters, operationally independent of Mastercard.

MCP, BY THE NUMBERS

What eighteen months built, and what it skipped

>10,000 [A] active public servers	97M+ [A] monthly SDK downloads	18 mo zero to standard
2,348 servers we scanned	~1,200 an engineer could evaluate	48% no dependency manifest
953 expose a remote surface	210 remote and abandoned	72.8% [A] peak attack success

THE ONE-PARAGRAPH VERSION

MCP solved a real problem. The trust layer never shipped with it.

Before MCP, every connection between a model and a tool was bespoke. MCP turned that into write-once, and adoption followed fast: more than 10,000 active public servers and 97M-plus monthly SDK downloads in about eighteen months.

But the layer that lets an enterprise trust a given server never shipped alongside the layer that made servers easy to build. About a third of repos have no documentation, nearly half no manifest, four in ten expose a remote surface, and the honest count an engineer could evaluate is ~1,200, not the tens of thousands the directories list.

The protocol has won. The trust layer underneath it is the next thing someone has to build.

PART 1 · WHAT MCP UNLOCKED

It turned M × N into M + N

BEFORE · TO NOV 2024

M × N

Every model-to-tool pairing a bespoke integration. Cost grew with the product.

WITH MCP

M + N

Write one server, reach every compatible client. Add a client once, reach every server.

Anthropic introduced MCP on 25 November 2024. Its own metaphor has stuck: MCP is the USB-C of AI integration. OpenAI, Google, and Microsoft all adopted it within five months.

PART 1 · WHAT MCP UNLOCKED

Things an agent could not do cleanly before

→ Reach any tool from any client

One server works in Claude, ChatGPT, Gemini, Copilot, Cursor, VS Code.

→ Run long jobs and ask mid-task

Async tasks plus secure browser prompts, so credentials skip the chat.

→ Chain tools across applications

Read an issue, open a PR, post to chat, in one run, one interface.

→ Return interactive surfaces

MCP Apps send back dashboards and forms, not just text. (Jan 2026)

→ Find servers programmatically

The official registry lets agents discover tools, not hard-code them.

→ Reach inside a private network

Tunnels: outbound-only, no inbound port opened. (Anthropic + OpenAI, May 2026)

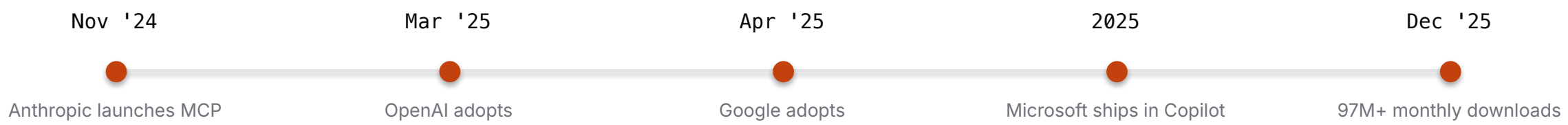
PART 2 · HOW BIG IT GOT

A cross-vendor standard, measured by use

>10,000 [A]
active public servers

97M+ [A]
monthly SDK downloads

4 of 4 [A]
major labs adopted it



Downloads overcount (CI re-pulls), but the trajectory is not in doubt: the Python SDK alone cleared ~250M/month by spring 2026. [B]

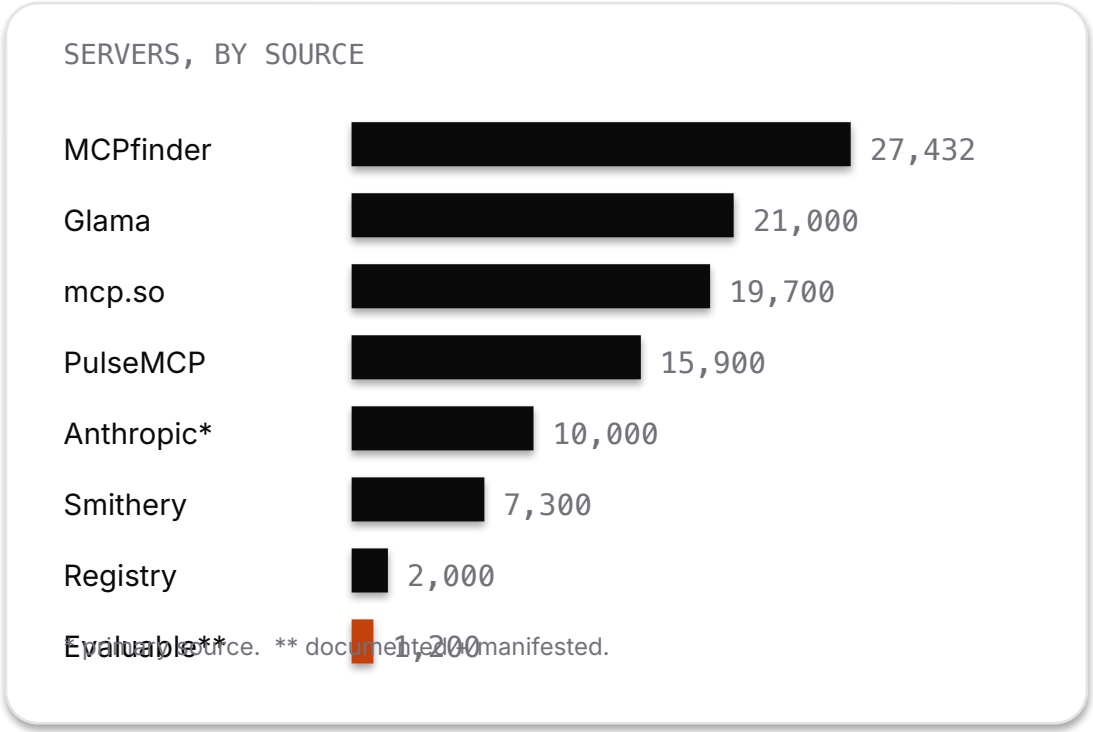
PART 2 · HOW BIG IT GOT

Nobody can count MCP servers cleanly

Directories disagree by 4x and don't measure the same thing: forks, thin wrappers, and re-publishes inflate every total. [C]

The one figure with a primary source is Anthropic's own: more than 10,000 active public servers.

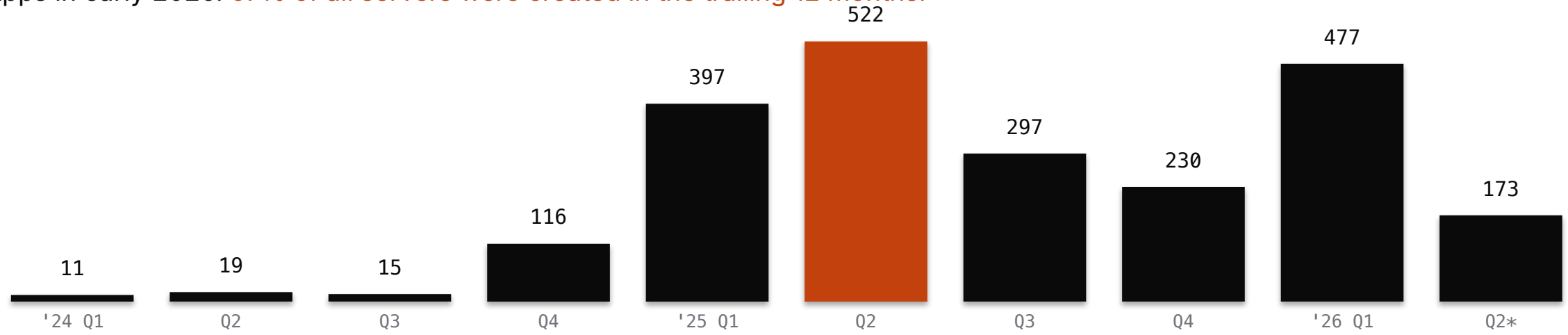
So we stopped counting and asked a better question: how many could an engineer actually adopt? That answer is ~1,200.



PART 2 · HOW BIG IT GOT

From near-zero to 2,300 in eighteen months

Two waves: the first after launch and the OpenAI and Google adoptions in spring 2025, the second after the new spec and MCP Apps in early 2026. 57% of all servers were created in the trailing 12 months.



* 2026 Q2 is a partial quarter (two months at time of scan).

PART 2 · HOW BIG IT GOT

Two ecosystems wearing one name

39%
of stars in the top 10 repos

67%
of stars in the top 50

74
median stars per server

9 in 10
owners have one repo

At the top, a healthy, vendor-backed protocol; below it, a long tail of individual experiments. Nothing in MCP tells the two apart. An agent told to find a server samples the whole distribution.

PART 3 · WHAT WE FOUND

A census of the public surface

2,413	2,348	2,348
discovered	active	deep-analyzed

MCP-related repos across five GitHub queries, deduplicated; each deep-analyzed for README, manifests, transport, license, and maintenance. A census of the public surface, not a security audit of any single server. Scanner: the Major Labs bench agent we call Sentinel.

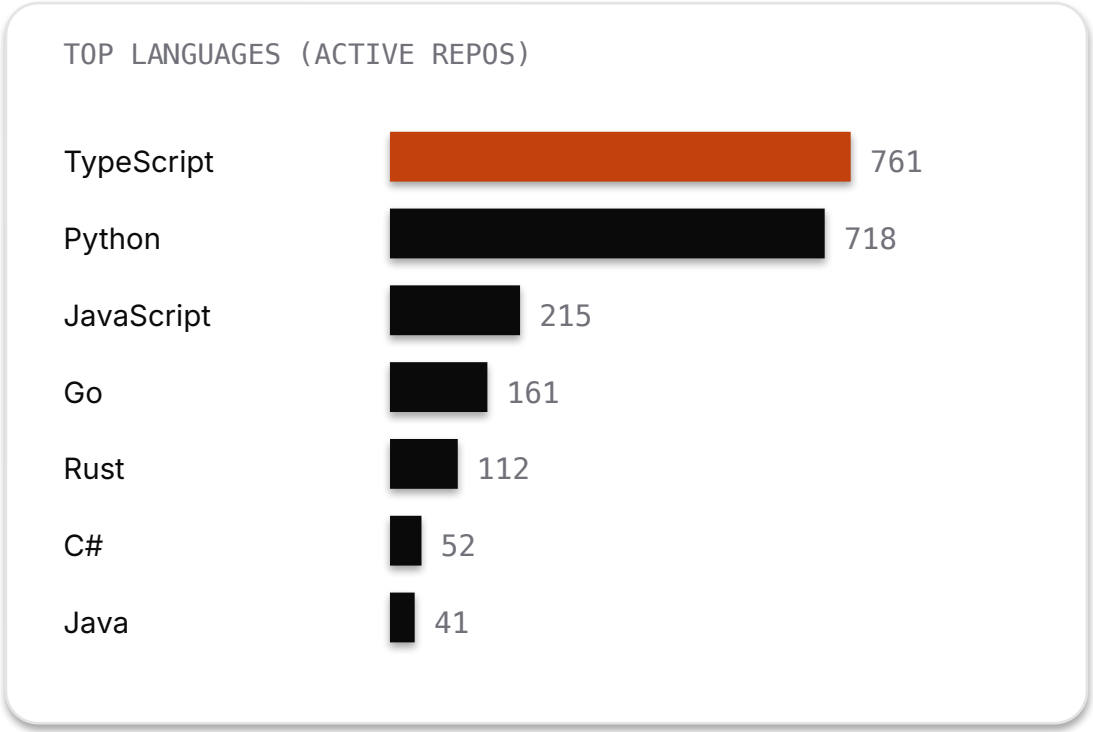
FINDING 1 · THE PROTOCOL IS WINNING

Healthy at the top, and built by app developers

— GitHub, AWS, Microsoft, Sentry, Cloudflare, Stripe and Atlassian all ship maintained servers; the official org ships the SDKs, inspector and registry.

— Attention is winner-take-most: the top 50 repos hold 67% of all 1.8M stars.

— TypeScript and Python are 63% of repos: the builders are app developers. 1,171 (50%) were pushed in the last 30 days.



FINDING 2 · THE QUALITY LONG TAIL

Half of it isn't deployable software

35%
have no README at all

48%
have no dependency manifest

~1,200
truly evaluable servers

823 repos have no documentation; 1,129 ship no manifest. Two independent signals converge: the honest count of functional servers is around 1,200, not the tens of thousands the directories list.

FINDING 3 · THE SECURITY SURFACE

Four in ten expose a remote surface. Some are abandoned.

953
HTTP-capable servers

210
abandoned and exposed

30/13/10%
dual / stdio / HTTP

HTTP transport is the remote attack surface: auth, input handling, SSRF. 210 of those servers have had no push in 180+ days.

An abandoned, network-exposed endpoint is what you find out about after it is exploited. In any other supply chain, the 182 would already be someone's incident.

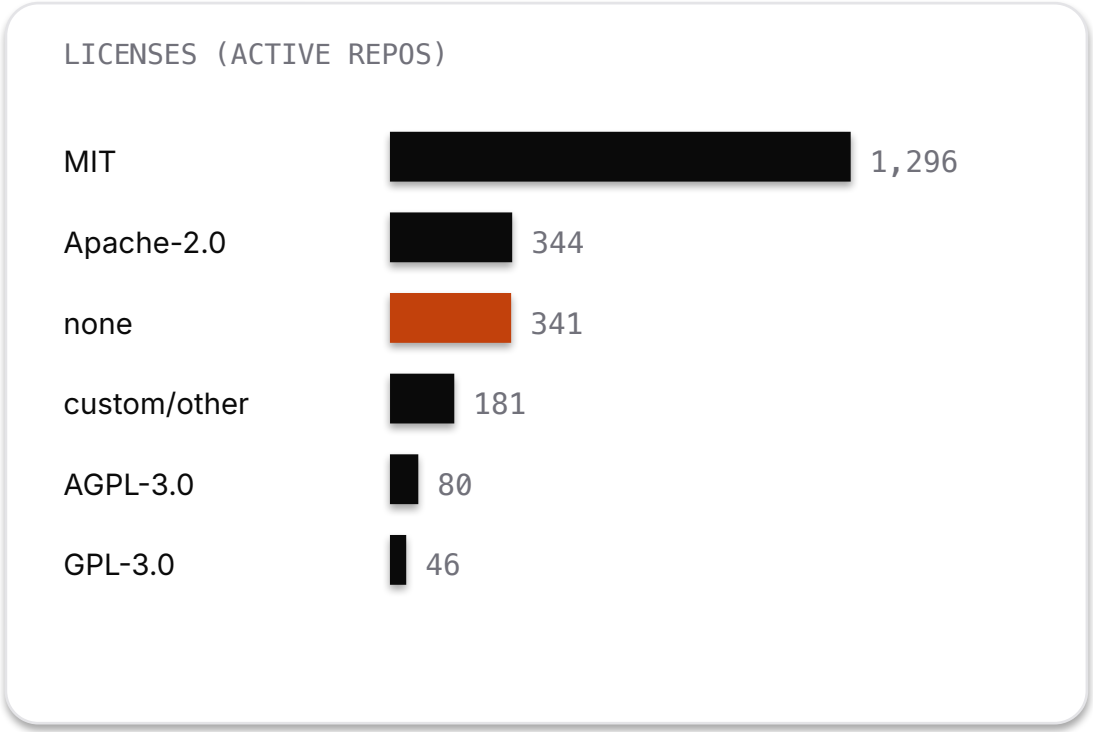
FINDING 4 · GOVERNANCE BLACK HOLES

One in five can't be adopted. Some can't be understood.

522
no clear license

102
no license + no README

341 carry no license and 181 an unrecognized one: over 1 in 5, blocked pending legal review. 102 have neither a license nor a README. Selection has to be gated, not searched.



PART 4 · THE SECURITY REALITY

The worst weaknesses are not bugs. They are the trust model.

MCP treats a tool's description as authoritative. It enters the agent's context as trusted content before any code runs. Hide an instruction in it and you have an attack with no execution and no runtime trace. A patch cannot fully remove a model that executes instructions from the content it consumes.

PART 4 · TOOL POISONING

The peer-reviewed core: hide the attack in the description

72.8% [A]
peak attack success (o1-mini)

<3% [A]
best refusal rate (Claude-3.7)

20 [A]
leading agents tested

MCPTox (AAAI 2026) ran 1,312 malicious cases across 45 live servers and 353 real tools. More capable models were often more vulnerable, not less.

Standard safety alignment does not catch this. The best-defended model still complied with the malicious metadata 97% of the time.

PART 4 · THE BASE RATE, AND THE RECORD

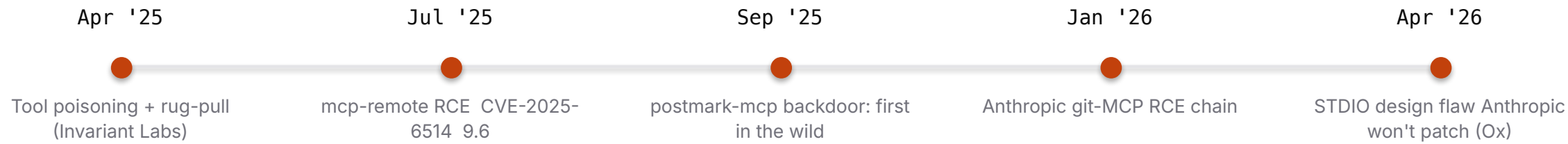
The dangerous patterns are everywhere, and exploited

82% [B]
path-traversal-prone file ops

67% [B]
code-injection APIs

34% [B]
command-injection APIs

Endor Labs static analysis of 2,614 implementations. Capability findings, not confirmed exploits: the code can do the dangerous thing. [B]



PART 5 · THE GOVERNANCE RESPONSE

The trust layer, under construction

Sep '25	Official MCP Registry launches in preview
Nov '25	Largest spec revision: OAuth 2.1 + client security rules
Dec '25	Donated to the Agentic AI Foundation (Linux Foundation)
Jan '26	MCP Apps, the first official extension
Feb '26	NIST AI Agent Standards Initiative
Mar '26	2026 roadmap names enterprise readiness the top priority
May '26	MCP tunnels (Anthropic + OpenAI); Opus 4.8 approval gating

The common thread across every vendor's guidance: defense in depth. No single control is sufficient. [A]

PART 6 · THE FORWARD READ

Growth first. Incident second. Registry third.

Every prior package ecosystem ran this sequence; MCP is mid-stride between the first and the second. For eighteen months capability outran governance: every spec revision added power faster than the trust layer added safety.

Major Labs is building the third stage: an ongoing MCP quality and provenance index, with this scan as its seed, so you can tell whether a server is maintained, scoped, documented and attributable before an agent calls it.

WHAT THIS MEANS FOR ANY REGULATED ENTERPRISE

The layer that lets you trust a server is missing

1. **Agent-reachable tools are a new third-party-risk category.**
   Same supply-chain exposure as any library, almost none of the registry, scanning or provenance tooling npm and PyPI eventually grew. Treat it like one.

2. **“Find an MCP server for X” is an unsafe default.**
   39% undocumented, half un-manifested, 1 in 5 unlicensed. Automated selection can route an agent into an abandoned or unattributable server.

3. **Govern remote transport first.**
   The 910 HTTP-capable servers, especially the 182 abandoned ones, are where risk concentrates. Treat them like any externally reachable endpoint.

4. **The missing control is a registry with a quality bar.**
   Not a marketplace. A gate: maintained, scoped, documented, attributable. Whoever runs that gate owns the trust layer.

METHOD, LIMITS AND SOURCES

How the scan works, and what it does not claim

Source	Public GitHub, 5 queries, deduplicated by canonical URL. Discovery completed 2026-05-28.
Sample	2,413 discovered, 2,348 active (not archived, not forks), 2,348 deep-analyzed.
Transport	Inferred from README and manifests, not from running servers. HTTP-capable means a server advertises a network surface, not a confirmed vulnerability.
Functional count	~1,200 is an inference, corroborated by two independent signals: missing documentation and missing dependency manifest.
Not in scope	Active probing, auth testing, SSRF detection against live servers. That is the next pass, and where a measured vulnerability rate will come from.
Source strength	[A] primary/peer-reviewed/official · [B] vendor research, method disclosed · [C] secondary/estimate. Scan figures are firsthand; soft numbers are never headlined.