

MCP Server & Tooling Landscape, September 2026

Public sample, redacted edition. This is a real deliverable with client-identifying details removed (a redaction log is kept). The substance, tables and all 80 citations are unchanged. **Everything below is a snapshot as of 2026-09-25.** Versions, star counts, CVE status and prices will drift, so re-check anything before you rely on it.

Prepared for: [client name redacted], a small team building local-first agent tooling

Research date: Friday 2026-09-25 (all times PT unless marked). Everything here comes from live sources read on this date. Bracketed numbers like [3] point to the Sources list at the end.

Method: I read spec pages, official blogs, READMEs, docs, NVD/OSV records, and security write-ups directly. GitHub facts (license SPDX, archived flag, stars, latest commit, latest release) were pulled from each repo's public page and its `commits.atom` / `releases.atom` feeds on 2026-09-25 [78]. Commit and release dates are **UTC dates**. Star counts are a point-in-time snapshot.

0. Executive answer

Adopt now: For new work, target the **2026-07-28 spec** (it's stateless, uses Streamable HTTP + stdio, and moves Tasks into an extension) on the **Tier-1 Python (v2.x) or TypeScript (v2.x) SDK** or **FastMCP 4.x**. Run a small, pinned set of mostly vendor-maintained servers locally: **Playwright MCP** (browser), **GitHub MCP server** in `--read-only` mode with trimmed toolsets, **MCP Toolbox for Databases** / a DuckDB local server (data), and **SearXNG MCP** pointed at your own SearXNG (search). Put local memory on plain files (**basic-memory**, or **mcp-memory-service** with local ONNX embeddings). Run each server in a container through **ToolHive** or **Docker MCP Gateway**. For hosts that work with Ollama/vLLM, use **opencode**, **goose** (now an AAIF/Linux Foundation project), **Zed**, or **Codex CLI**. Codex has the strongest per-tool approval and allow-list controls I found.

Watch: Registry GA (it's still "preview"), HTTP-over-stdio transport unification, DPoP and agent-identity auth, the `tools/call` result-shape redesign, and progressive tool discovery. All of these are on the 2026-08-22 roadmap [9].

Avoid: Unmaintained or archived hosts and servers: **Continue** (repo read-only), **Roo Code** (archived), **mcpghost** (archived), **mcp-router** (archived), the self-hosted **Browserbase MCP** (archived), and the `servers-archived` reference servers (Puppeteer, SQLite, Postgres, and others). Don't install random npm MCP packages without pinning and scanning them; the malicious `postmark-mcp` is the canonical example. Don't let any tool-bearing server run with broad tokens and auto-approve on. Prompt injection through tool *output* is an architectural problem that no server patch fixes [63].

1. Top “adopt now” table

Layer	Pick	Maintainer	License	Latest release / last commit (UTC)	Self-host	Main risk / condition
Spec target	MCP 2026-07-28 (current)	MCP maintainers (AAIF / LF Projects)	Apache-2.0 for new contributions [10]	Released 2026-07-28 [3][4]	n/a	Breaking initial handshake and no Session Plan migration older servers
Server SDK (Python)	mcp Python SDK v2.2.0 or FastMCP v4.0.9	Official (Tier 1) / PrefectHQ	MIT / Apache-2.0	2026-09-07 / 2026-09-24 [78]	yes	FastMCP third-party framework on top of the protocol pin major version
Server SDK (TS)	@modelcontextprotocol/server + /client 2.1.0 (v2 line)	Official (Tier 1)	MIT badge on README; repo SPDX shows NOASSERTION during the MIT→Apache transition [18] [20]	2026-09-23 [78]	yes	v1.x gets fixes for at least 6 months after v2 release
Coding host (local models)	opencode v2.0.16	anomalyco org (redirect from sst/opencode)	MIT	2026-09-24 [78]	yes	MCP to eat con Disable agent v globs [2]
General agent host	goose	AAIF / Linux Foundation	Apache-2.0	last commit	yes	Support Ollama 15+

Layer	Pick	Maintainer	License	Latest release / last commit (UTC)	Self-host	Main risk conditions
		(formerly <code>block/goose</code>)		2026-09-25 [78]		provided by the browser [30]
Browser automation	Playwright MCP v0.0.82	Microsoft	Apache-2.0	2026-09-18 [78]	yes (local)	“Not a security boundary. Origin allow-lists aren’t security controls. Use <code>--isolate</code> ” [38]
Browser debugging	Chrome DevTools MCP v1.10.1	Google (ChromeDevTools.org)	Apache-2.0	2026-09-23 [78]	yes (local)	Usage is on by default . Pass <code>--usage-statistics</code> and <code>--performance-crux</code> [
GitHub	github-mcp-server 1.12.2 (local Docker)	GitHub	MIT	2026-09-16 [78]	yes (local binary/Docker); remote is GitHub-hosted	Toxic-flow prompt injection public issues [Use <code>--read-only</code> <code>--toolset</code> lockdown mode [
Databases	MCP Toolbox for Databases v1.13.0	Google (<code>googleapis/mcp-toolbox</code>)	Apache-2.0	2026-09-25 [78]	yes	Prebuilt executables 1 tools

Layer	Pick	Maintainer	License	Latest release / last commit (UTC)	Self-host	Main requirements
						Point it to a read-only DB role
Analytics (local)	DuckDB/MotherDuck local MCP v1.0.8	MotherDuck	MIT	2026-08-19 [78]	yes	Can read and write to the local filesystem [51]
Search	mcp-searxng v2.4.0 + your own SearXNG	Community (ihor-sokoliuk)	MIT	2026-09-22 [78]	yes (fully)	Only as trustworthy as the SearXNG instance you point it to [43]
Memory	basic-memory v0.23.2 (Markdown on disk)	Basic Machines	AGPL-3.0	2026-08-25 [78]	yes	AGPL matters if you redistribute or host others
Memory (alt)	mcp-memory-service v11.14.0	Community (doobidoo)	Apache-2.0	2026-09-25 [78]	yes	Local OEmbed [47]. Still maintained
Isolation / gateway	ToolHive v0.51.2	Stacklok	Apache-2.0	2026-09-25 [78]	yes	Pre-1.0
Isolation / gateway (alt)	Docker MCP Gateway v0.44.1	Docker	MIT	2026-09-23 [78]	yes	README lists Docker Desktop 4.59+ as a prerequisite. The CLI plugin c

Layer	Pick	Maintainer	License	Latest release / last commit (UTC)	Self-host	Main risk / condition
						also run its own
Config pinning	mcp-context-protector	Trail of Bits	Apache-2.0	last commit 2026-02-13 [78]	yes	Low rec activity. Treat it pattern a tool, not long-liv produc

2. State of the spec

2.1 Current revision

- **Current protocol version is 2026-07-28** . Revisions are dated with the date of the last backward-incompatible change [1]. The GitHub releases list shows 2026-07-28 (stable), preceded by an RC, then 2025-11-25 , 2025-06-18 , 2025-03-26 , and 2024-11-05 [4].
- The official blog announced it on **July 28, 2026**, calling out a stateless core, Multi Round-Trip Requests (MRTR), header-based routing, cacheable list results, authorization hardening, and a formal extensions framework [3].
- Adoption signal from the blog: “close to half-a-billion downloads a month” across Tier-1 SDKs, with TypeScript and Python each past 1 billion total downloads [3]. These are self-reported numbers.

2.2 What changed in 2026-07-28 (primary: changelog [2])

- **Stateless.** The `initialize / initialized` handshake is gone. Every request carries its protocol version and client capabilities in `_meta` . A mandatory `server/discover` RPC advertises versions and capabilities (SEP-2575) [2].
- **No protocol sessions.** `Mcp-Session-Id` has been removed. Servers that need state mint explicit handles and pass them back as tool arguments (SEP-2567) [2].
- **MRTR replaces server-initiated requests.** Servers return `resultType: "input_required"` , and the client retries with `inputResponses` . This replaces server-sent `elicitation/create` , `sampling/createMessage` , and `roots/list` (SEP-2322) [2].

- **Tasks** moved out of core into the `io.modelcontextprotocol/tasks` extension, now polling-based (`tasks/get` , `tasks/update`) (SEP-2663) [2].
- `subscriptions/listen` replaces the HTTP GET stream and `resources/subscribe` [2].
- **SSE resumability removed.** A broken stream means you re-issue the request [2].
- **Required `Mcp-Method` / `Mcp-Name` headers** on Streamable HTTP POSTs, so gateways can route and authorize without parsing bodies (SEP-2243) [2][3].
- **Cache hints.** `ttlMs` and `cacheScope` on list and read results. Tools should be listed in a deterministic order [2].
- **Deprecated:** Roots, Sampling, and Logging (SEP-2577). The legacy HTTP+SSE transport is now formally deprecated. OAuth Dynamic Client Registration is deprecated in favour of Client ID Metadata Documents (CIMD). Deprecated features stay for at least 12 months [1][2].

2.3 Transports

- There are two standard bindings: **stdio** (newline-delimited JSON-RPC to a client-launched subprocess) and **Streamable HTTP** (each message is a POST to a single endpoint, with replies as JSON or a request-scoped SSE stream) [5].
- Roadmap: collapse these into **one HTTP model, spoken over stdio for local servers** (“HTTP/2 over stdio”), and add ETag caching [9]. That makes stdio vs. HTTP a transport detail rather than a design fork. Good news for local-first setups.

2.4 Authorization

- Authorization is **OPTIONAL**. When it is implemented, it’s OAuth 2.1 (draft-13), servers **MUST** publish RFC 9728 Protected Resource Metadata, and clients **SHOULD** support CIMD. DCR is retained only for backward compatibility [6].
- 2026-07-28 hardening: RFC 9207 `iss` validation (mix-up defence), a required DCR `application_type` (this fixes `localhost` redirect rejections for CLI/desktop clients), and credentials bound to their issuer [2][3].
- Roadmap next: **DPoP**, workload identity federation, ID-JAG, and RFC 8693 token exchange for agent and sub-agent delegation [9].
- **For the client:** local stdio servers don’t need OAuth at all. OAuth matters for remote servers you expose (for example on a hosted workspace or at the edge). If you build one, implement CIMD and `iss` validation from day one.

2.5 Elicitation

- Two modes. **Form** is flat primitive schemas only. **URL** is out-of-band, for anything sensitive. Servers **MUST NOT** use form mode for passwords, API keys, or tokens [7].
- In 2026-07-28, elicitation is delivered inside an `InputRequiredResult` (MRTR), not as a server-initiated request [7]. URL mode is still flagged as possibly changing in future revisions [7].

2.6 Structured output

- Tools may declare `outputSchema`. Results can carry `structuredContent`, which may be any JSON value. Schemas may use any JSON Schema 2020-12 keywords (SEP-2106) [2][8].
- Tool annotations **MUST be treated as untrusted** unless they come from trusted servers [8].
- Honest caveat: the roadmap says returning both `content` and `structuredContent` “has confused server and client authors alike and produced diverging implementations,” and a redesign of `tools/call` is planned [9]. **Expect this surface to change.** Don’t over-invest in clever dual-format outputs.

2.7 Registry

- The official **MCP Registry is still in preview**: “Breaking changes or data resets may occur before general availability” [13]. The API has been frozen at **v0.1** since 2025-10-24 [14].
- It’s a **metadata** registry (it points to npm/PyPI/Docker/remote URLs). It uses reverse-DNS namespaces verified via GitHub, DNS, or HTTP challenge. It **delegates security scanning** to package registries and downstream aggregators [13]. In other words, being listed ≠ being vetted.
- The official registry codebase is “not designed for self-hosting”. Private registries should implement its OpenAPI spec instead [13].
- **Size**: I paged the live API (`/v0.1/servers?version=latest`) and counted **35,953** latest-version server entries at 2026-09-25 06:51 PT [15]. That’s my own count. The API returned an HTTP 500 on one earlier full pass, so treat the figure as approximate.
- Working-group lead is from Stacklok (the ToolHive vendor) [14]. Worth knowing when you weigh ToolHive’s registry features.

2.8 Governance

- Anthropic donated MCP to the **Agentic AI Foundation (AAIF)**, a directed fund under the Linux Foundation, on **2025-12-09**. Co-founders were Anthropic, Block, and OpenAI [11][12][80]. The LF says it “will not dictate the technical direction of MCP” [11].
- Technical governance is **BDFL-style**: two Lead Maintainers (David Soria Parra, Den Delimarsky) with final veto, six Core Maintainers, and membership that is individual rather than per-company [10].
- **Licensing is in transition from MIT to Apache-2.0**. New contributions are Apache-2.0. Older MIT contributions stay MIT until their authors consent to relicensing [10][20]. That’s why GitHub shows `NOASSERTION` for many official repos. Practical effect for you: both licenses are permissive, so this isn’t a problem.
- A formal feature lifecycle has been adopted (Active → Deprecated → Removed), with a minimum 12-month deprecation window [1][2].

3. SDKs

Official tiering: Tier 1 needs a 100% conformance pass, new spec features before release, triage within 2 business days, and critical fixes within 7 days. Tier 2 needs 80% conformance and new features within 6 months. Tier 3 is experimental [16].

SDK	Tier (per docs [17])	Latest release (UTC) [78]	Notes
TypeScript	1	@modelcontextprotocol/ * 2.1.0 (2026-09-23), v1 line 1.30.1	v2 is the stable line for 2026-07-28. It's split into client/server packages and uses Standard Schema (Zod v4, Valibot, ArkType). PRs are limited to 1 per new contributor while v2 settles [18]
Python	1	v2.2.0 (2026-09-07); v1 line 1.30.0	v2.0.0 shipped 2026-07-28 alongside the spec [78]
Go	1	v1.8.0 (2026-09-14)	"Maintained in collaboration with Google" (repo description)
C#	1	v2.2.0 (2026-08-13)	"Maintained in collaboration with Microsoft"
Rust (rmcp)	1 on docs page	rmcp-v3.4.1 (2026-09-23)	Discrepancy: the 2026-07-28 blog said Rust supports the new spec "in beta" [3], while the SDK page lists it as Tier 1 [17]
Java	2	v2.0.1 (2026-08-19)	
Ruby	2	v1.6.0 (2026-09-21)	Reached 1.0 on 2026-07-27 [19]
Swift	3	0.12.1 (2026-05-07)	Last commit 2026-04-29, so slowest-moving
PHP	3	v0.8.1 (2026-08-29)	
Kotlin	3	0.15.0 (2026-07-28)	
FastMCP (framework)	n/a (third-party)	v4.0.9 (2026-09-24)	Apache-2.0. Its maintainers say 4.0 ships support for 2026-07-28 features: background tasks, stateless interactivity, enterprise auth [3][21]

Recommendation. For a team building local-first servers with a Python/TS split: Use **FastMCP 4.x** for Python tools you write quickly, and the **official TS v2** packages for anything that ships to edge runtimes (TS v2 runs on Node, Bun, and Deno [18]). Build against the 2026-07-28 semantics, but test that hosts still on 2025-11-25 can talk to you. The spec defines backward compatibility for handshake-based versions [1].

4. Clients / hosts, with a focus on local models

Key fact first: **Ollama is not an MCP client**. A PR adding experimental MCP support to Ollama (#13700) was **closed unmerged**. The maintainer said they would “keep the surface area small and frozen” and pointed people to Pi instead [33]. So an MCP-aware host always sits between Ollama/vLLM and your servers.

Host	Maintainer / license	Last commit (UTC) [78]	Local-model path	MCP support notes
opencode	anomalyco org (redirect from sst/opencode), MIT	2026-09-25	Any OpenAI-compatible endpoint (Ollama/vLLM). Local-model setup not detailed on the MCP page I read	Local (stdio) and remote servers, OAuth with DCR, per-agent tool enable/disable via globs [23]
goose	AAIF/LF (aaif -goose/goose), Apache-2.0	2026-09-25	Ollama listed among 15+ providers [30]	“70+ extensions via MCP”. Desktop, CLI, and API [30]
Zed	Zed Industries	2026-09-25	Not verified in this pass	Supports MCP Tools and Prompts only . No sampling or elicitation yet. Per-tool permission rules (mcp:<server>:<tool>), default confirm [24]
Codex CLI / IDE / ChatGPT desktop	OpenAI, Apache-2.0 (CLI repo)	2026-09-25	Not verified in this pass	stdio + Streamable HTTP, CIMD + DCR, enabled_tools / disabled_tools , per-tool approval_mode (auto / prompt / writes / approve), per-tool

Host	Maintainer / license	Last commit (UTC) [78]	Local-model path	MCP support notes
				output-token caps [25]. Best policy controls I found
Cline	Cline, Apache-2.0	2026-09-25	Ollama, LM Studio [27]	MCP via <code>mcpServers</code> JSON. Omitting <code>type</code> defaults to legacy SSE , so set <code>"type": "streamableHTTP"</code> [29]. Gotcha: Cline’s own local-model guide recommends “compact prompt”, and that removes MCP tools [28]
LM Studio	LM Studio (closed source)	n/a	Built-in runtime (MLX/GGUF)	MCP host since 0.3.17, local and remote, Cursor-style <code>mcp.json</code> . Warns that MCP servers built for frontier models can overflow local context [26]
Open WebUI	Open WebUI, GitHub SPDX = NOASSERTION	2026-09-21	Ollama-native UI	Native MCP from v0.6.31, Streamable HTTP only , admin-only. The docs say OpenAPI remains “the preferred integration path”. Use mcpo to bridge stdio servers [22]
ollmcp (mcp-client-for-ollama)	Community, MIT	2026-09-11	Ollama-first TUI	STDIO/SSE/Streamable HTTP, human-in-the-loop, multi-server [34]
Pi (earendil-works/pi, formerly	Earendil, MIT	2026-09-25	Recommended by an Ollama maintainer [33]	MCP is an extension you build or install , not built in [77]

Host	Maintainer / license	Last commit (UTC) [78]	Local-model path	MCP support notes
badlogic/pi-mono)				
Continue	Apache-2.0	2026-07-21		“No longer actively maintained and is read-only”, final 2.0.0 release [31]
Roo-Code	Apache-2.0	2026-05-15		Repository archived [32][78]
mcp-host	MIT	2026-04-13		Repository archived [78]. Third-party 2026 guides still recommend it, which is exactly why you verify

For a single-machine home lab: run the agent host (opencode or goose) on that machine. Point it at Ollama or vLLM’s OpenAI-compatible endpoint. Run stdio servers as child processes of that host, or via ToolHive/Docker for isolation. LM Studio’s docs and Cline’s blog both warn that MCP tool schemas can swamp local-model context [26][28]. Keep your active toolset small per agent (opencode globs, Codex `enabled_tools` , Zed profiles).

5. Servers worth adopting, by category

Legend: **Official** = the vendor that owns the product or API. **Ref** = MCP steering-group reference server. **Community** = independent.

5.1 Coding / git

Server	Maint.	License	Latest rel. / last commit (UTC)	Self-host	Notes & risks
github/github-mcp-server	Official (GitHub)	MIT	1.12.2 (2026-09-16)	Local Docker/binary; remote at <code>api.githubcopilot.com/mcp/</code>	<code>--toolsets</code> to trim, <code>--read-only</code> , lockdown mode that only surfaces public-

Server	Maint.	License	Latest rel. / last commit (UTC)	Self-host	Notes & risks
					repo content from authors with push access [37]. Invariant showed a malicious public issue driving an agent to leak private repo data. They call it architectural, not a server bug [63]
mcp-server-git (reference)	Ref	MIT→Apache transition	repo release 2026.8.31	Local	Three CVEs fixed in 2025.12.17 or earlier: path-validation bypass in <code>git_init</code> (CVE-2025-68143), argument injection via <code>git_diff</code> / <code>git_checkout</code> (CVE-2025-68144), and <code>--repository</code> scope bypass (CVE-2025-68145) [69]. The servers README says reference servers are “not production-ready solutions” [20]

Server	Maint.	License	Latest rel. / last commit (UTC)	Self-host	Notes & risks
oraios/serena	Community (Oraios AI)	Changed: Serena app is now GPL-3.0-or-later ; the SolidLSP library stays MIT. Tag <code>mit-final</code> dated 2026-09- 14 [42]	v1.7.0 (2026-08- 09)	Local	LSP-based semantic code tools. License change affects anyone embedding or redistributing it commercially. Pin to <code>mit- final</code> or accept GPL
upstash/context7	Official (Upstash)	MIT (MCP server only)	<code>@upstash/cont ext7-mcp</code> 4.1.1 (2026-09-14)	No. The API backend, parser, and crawler are private [41]	CVE-2026-75130 (NVD, 2026-08- 18): prompt injection via “Custom AI Instructions”, CVSS 3.1 9.0 / CVSS 4.0 6.4, “through 2.1.2” [73]. Fix status for the hosted feature is unknown from the sources I read. The docs are community- contributed and Upstash disclaims their accuracy and security [41]

5.2 Filesystem

- **Reference filesystem server** (`@modelcontextprotocol/server-filesystem`). Symlink escape and prefix-match escape (CVE-2025-53109/53110) were fixed in 0.6.4 / 2025.7.01 [67][68]. It’s fine for scoped local use, but it’s still

a reference implementation [20]. Most coding hosts (opencode, Zed, Codex) have built-in file tools, so in a coding host you often don't need this server.

- **DesktopCommanderMCP** (community, MIT, v0.2.51, 2026-09-17 [78]) is active, but a shell-plus-filesystem server is maximal blast radius. Only use it inside a container.

5.3 Browser automation

Server	Maint.	License	Latest rel. (UTC)	Notes
microsoft/playwright-mcp	Official (Microsoft)	Apache-2.0	v0.0.82 (2026-09-18)	Accessibility-snapshot driven. The README says it is not a security boundary , -- allowed-origins / -- blocked-origins “do not serve as a security boundary”, and secrets masking is “a convenience and not a security feature”. It has -- isolated and -- sandbox flags [38]. 0.0.x versioning, so pin it
ChromeDevTools/chrome-devtools-mcp	Official (Google)	Apache-2.0	v1.10.1 (2026-09-23)	Debugging and performance traces. Google usage statistics on by default. Trace URLs may go to the CrUX API. Opt out with -- no-usage-statistics, --no-performance-crux [39]. Has a --slim mode

Server	Maint.	License	Latest rel. (UTC)	Notes
browser-use	Browser Use (company OSS)	MIT	0.13.10 (2026-09-04)	Large agent-browser library. Its MCP mode was not verified in this pass
browserbase/mcp-server-browserbase		Apache-2.0	archived	README: “archived and no longer maintained”; recommends the hosted version [40]
executeautomation/mcp-playwright, hangwin/mcp-chrome	Community	MIT	last commits 2025-12-13 / 2026-01-06	Stale. The official Playwright MCP supersedes them

5.4 Search / research

Server	Maint.	License	Status (UTC)	Self-host	Notes
ihor-sokoliuk/mcp-searxng	Community	MIT	v2.4.0 (2026-09-22)	Yes, end to end (needs your own SearXNG with JSON enabled) [43]	The best sovereign option
brave/brave-search-mcp-server	Official (Brave)	MIT	v2.1.4 (2026-09-17)	Server local, API remote	Replaced the archived reference Brave server [20]. Paid API; pricing not checked
firecrawl/firecrawl-mcp-server	Official (Firecrawl)	MIT	v3.24.1 (2026-08-26)	“Cloud and self-hosted support” [44]	26 tools in the full profile, which is heavy for local context. Keyless tier limited to

Server	Maint.	License	Status (UTC)	Self-host	Notes
					scrape/search/parse [44]
exa-labs/exa-mcp-server, tavily-ai/tavily-mcp, jina-ai/MCP, perplexityai/modelcontextprotocol	Official vendors	MIT / MIT / Apache-2.0 / MIT	commits Aug-Sep 2026 [78]	Server local, API remote	Paid APIs, so vendor-dependent. Not deep-read
Reference fetch server	Ref		active	Local	Simple fetch-to-markdown [20]. Pair it with an egress policy

5.5 Memory / knowledge graphs

Server	Maint.	License	Status (UTC)	Local-model friendly?	Notes
basic-memory	Basic Machines	AGPL-3.0	v0.23.2 (2026-08-25)	Yes. Plain Markdown on disk, cloud optional [46]	Human-readable, git-able, no lock-in
mcp-memory-service	Community (doobidoo)	Apache-2.0	v11.14.0 (2026-09-25)	Yes. “Embeddings run locally via ONNX” [47]	Knowledge graph with typed edges. Bus factor ≈ 1
Graphiti MCP (getzep/graphiti)	Zep (company OSS)	Apache-2.0	v0.30.2 (2026-09-08)	Partial. Default LLM is OpenAI; Ollama works “as an OpenAI-compatible endpoint” [45]	Temporal KG on FalkorDB (default) or Neo4j. Heaviest option; needs an LLM that handles structured output well
Reference memory server	Ref		active	Yes	JSON knowledge graph, fine for demos [20]

Server	Maint.	License	Status (UTC)	Local-model friendly?	Notes
mem0 / OpenMemory	Mem0	Apache-2.0	active (2026-09-25)	Not verified	I didn't verify OpenMemory's MCP docs (the README path returned nothing). Unknown

5.6 Databases

Server	Maint.	License	Status (UTC)	Notes
googleapis/mcp-toolbox (formerly genai-toolbox)	Official (Google)	Apache-2.0	v1.13.0 (2026-09-25)	Prebuilt tools for PostgreSQL, MySQL, MariaDB, SQL Server, Oracle, MongoDB, Redis, Elasticsearch, ClickHouse, and more, plus GCP databases [48]. Self-hosted binary
motherduckdb/mcp-server-motherduck	Official (MotherDuck)	MIT	v1.0.8 (2026-08-19)	Local DuckDB files, in-memory, S3, MotherDuck [51]. Great fit for local research data
crystaldba/postgres-mcp	Community (Crystal DBA)	MIT	last release v0.3.0 (2025-05-16), last commit 2026-08-16	Restricted (read-only, safe SQL parsing) vs. unrestricted modes [49]. Release cadence has stalled
supabase/mcp	Official (Supabase)	Apache-2.0	v0.13.0 (2026-09-17)	<code>read_only=true</code> , <code>project_ref</code> scoping. Local via Supabase CLI at

Server	Maint.	License	Status (UTC)	Notes
				localhost:54321/mcp [50]
redis/mcp-redis	Official (Redis)	MIT	0.5.1 (2026-08-05)	
Kubernetes (mcp-server-kubernetes)	Community			CVE-2026-46519: allow-list env vars were enforced only at tools/list , not at call time. Fixed in 3.6.0 [72]. This is a general lesson: <i>discovery-time filtering is not authorization</i>

5.7 Local-model bridges

- **mcpo** (Open WebUI, MIT): wraps a stdio MCP server as an OpenAPI HTTP server [36]. It’s still what Open WebUI points to for stdio [22], but the last commit was **2026-02-27** [78]. Expect lag on 2026-07-28 features.
- **ollmcp**: see §4.
- **mcpghost**: **archived** [78]. Don’t adopt it.
- Pattern: prefer a host with a native provider for Ollama/vLLM (opencode, goose, Cline, Zed) over a translation proxy.

5.8 Creative tooling

Server	Maint.	License	Status	Notes
Figma MCP server (remote mcp.figma.com/mcp , or desktop)	Official (Figma)	Proprietary service	active	Remote is available on all plans. Desktop needs a Dev or Full seat on a paid plan. Write-to-canvas is “ free during the beta ” and will become usage-based paid [53]. Budget and lock-in risk

Server	Maint.	License	Status	Notes
GLips/Figma-Context-MCP (Framelink)	Community	MIT	v0.13.2 (2026-06-18)	Self-hosted, uses your Figma API token
mcp-for-blender (formerly ahujasid/blender-mcp)	Community	MIT	last commit 2026-09-25	<code>execute_blender_code</code> runs arbitrary Python. Set <code>BLENDER_MCP_SAFE_MODE=1</code> . Telemetry is opt-in [52]
joenorton/comfyui-mcp-server	Community	Apache-2.0	v1.1.1 (2026-02-17)	Small and low-activity. Not deep-read

6. Gateways, proxies, aggregators

Project	Maint.	License	Latest rel. (UTC)	What it's for	Notes
ToolHive	Stacklok	Apache-2.0	v0.51.2 (2026-09-25)	Runs every server in an isolated container with minimal permissions. Self-hostable registry and gateway. K8s operator, OTel [55]	Best fit for sovereign isolation. Open-core (an Enterprise edition exists) [55]
Docker MCP Gateway (docker mcp)	Docker	MIT	v0.44.1 (2026-09-23)	Containerized servers, secrets via Docker Desktop, OAuth, catalog, call tracing [54]	Smoothest path is the Docker Desktop Toolkit (4.59+) [54]

Project	Maint.	License	Latest rel. (UTC)	What it's for	Notes
ContextForge	IBM	Apache-2.0	v1.0.7 (2026-09-21)	Federates MCP, A2A, and REST/gRPC. 40+ plugins, OTel [56]	Heavier and more enterprise-oriented
agentgateway	agentgateway project (governance not verified)	Apache-2.0	v1.6.0-alpha.2 (2026-09-22)	MCP + A2A + LLM gateway [57]	Good for an edge proxy in front of remote MCP
microsoft/mcp-gateway	Microsoft	MIT	no releases; last commit 2026-08-25	K8s-oriented	
TBXark/mcp-proxy	Community	MIT	v1.1.0 (2026-09-15)	Aggregate many servers behind one HTTP endpoint	
sparfenyuk/mcp-proxy	Community	MIT	v0.12.0 (2026-05-14)	studio↔HTTP bridge	Slowing
metatool-ai/metamcp	Community	MIT	2.4.22 (2025-12-19)	Aggregator	Stale releases
mcp-router			"v0.6.4 — Final release", archived [78]		Avoid

With 2026-07-28, gateways can route and authorize on the `Mcp-Method / Mcp-Name` headers without parsing JSON bodies [3]. Check that any gateway you pick has shipped 2026-07-28 support before relying on that.

7. Security

7.1 Incident / vulnerability classes, with real examples

Class	Real example	Source
Tool poisoning (hidden instructions in tool descriptions)	Invariant Labs PoC: an <code>add</code> tool whose description exfiltrates <code>~/ .cursor/mcp.json</code> and SSH keys, 2025-04-01	[62]
Rug pull (description changes after approval)	Same Invariant write-up; mitigation is pinning or hashing tool descriptions	[62]
Cross-server shadowing	A malicious server redirects a trusted <code>send_email</code> tool	[62]
Indirect prompt injection via tool output (“toxic flow”)	A malicious GitHub issue makes the agent leak private repo data into a public PR. Not fixable server-side	[63]
Prompt injection via a server’s content feature	Context7 “Custom AI Instructions”, CVE-2026-75130, CVSS 3.1 9.0	[73]
Malicious package / supply chain	<code>postmark-mcp</code> on npm turned malicious in 1.0.16 (released 2025-09-17) and BCC’d every email to an attacker. 1,643 downloads	[75][76]
Client-side RCE via OAuth metadata	<code>mcp-remote</code> OS command injection from a crafted <code>authorization_endpoint</code> , CVE-2025-6514, CVSS 3.1 9.6	[65]
Unauthenticated dev tooling	MCP Inspector < 0.14.1 RCE (no auth between client and proxy), CVE-2025-49596, CVSS 4.0 9.4	[66]
DNS rebinding on localhost HTTP servers	TS SDK < 1.24.0 didn’t enable DNS-rebinding protection by default, CVE-2025-66414	[70]
Session hijack / authz bypass in SDK transports	Python SDK < 1.27.2 routed by session ID without checking the principal, CVE-2026-52869 (CVSS 7.1)	[71]
Path traversal / sandbox escape in servers	Filesystem symlink and prefix bugs (CVE-2025-53109/53110). <code>mcp-server-git</code> trio (CVE-2025-	[67][68][69][74]

Class	Real example	Source
	68143/4/5). MCP Atlassian file exfiltration in remote mode, CVE-2026-73496 (CVSS 7.7), fixed in 0.22.0	
Discovery-time filtering ≠ authorization	mcp-server-kubernetes CVE-2026-46519 (CVSS 8.8)	[72]
Over-broad tokens / scope creep	The spec's Scope Minimization section; OWASP MCP02	[61][64]

OWASP's **MCP Top 10** is still at the beta/pilot stage, with items labelled MCP01:2025 – MCP10:2025 (token exposure, scope creep, tool poisoning, supply chain, command injection, prompt injection, authn/authz, telemetry, shadow servers, context over-sharing) [64].

7.2 What the spec requires or recommends (Security Best Practices, 2026-07-28 [61])

- **Token passthrough is forbidden.** Servers MUST NOT accept tokens not issued to them.
- Clients doing OAuth discovery must guard against **SSRF**: HTTPS only, block private and link-local ranges, validate redirects.
- **State handles are not authentication.** Bind them to the verified user.
- For **local servers**: one-click install MUST show the full command and get consent. Clients SHOULD sandbox servers. Servers should prefer stdio, or require auth or use unix sockets if they use HTTP.
- Validate authorization URLs: only http(s) , never launch them via a shell.

7.3 Scanners and guards (what they actually do)

Tool	License	Status (UTC)	Runs fully local?	Notes
Snyk Agent Scan (formerly invariantlabs mcp-scan)	Apache-2.0	snapshot build 2026-09-25	No. Sends tool names and descriptions to the Snyk analysis API and requires SNYK_TOKEN . States it does not store tool-call contents [58]	Also executes stdio server commands from your config to fetch descriptions [58]. Run it inside a sandbox. Output format marked experimental
Cisco AI Defense mcp-scanner	Apache-2.0	4.8.4 (2026-08-28)	Partially. The YARA and static/offline	Good for CI with offline mode

Tool	License	Status (UTC)	Runs fully local?	Notes
				JSON modes need no API. LLM and AI Defense engines are optional [59]
Trail of Bits mcp-context-protector	Apache-2.0	last commit 2026-02-13	Yes	Trust-on-first-use pinning of server config, blocks unapproved changes (anti-rug-pull), quarantines tool responses, strips ANSI [60]
promptfoo, NVIDIA garak	MIT / Apache-2.0	active	Yes	General LLM red-teaming. Not MCP-specific; not evaluated here

7.4 Mitigations mapped to a local-first setup

1. **Pin everything.** Exact npm/PyPI/container versions, plus tool-description pinning (context-protector) to catch rug pulls [60][62].
2. **Containerize servers** (ToolHive or Docker Gateway). Mount only the project directory and deny network by default for filesystem and git servers [54][55][61].
3. **Separate trust zones per agent.** Don't load an untrusted-content server (web fetch, GitHub issues, Context7) in the same session as a secret-bearing or write-capable server. This is the toxic-flow lesson [63].
4. **Least-privilege tokens.** GitHub fine-grained PAT scoped to specific repos, `--read-only` by default, lockdown mode on [37]. Read-only DB roles [48][49].
5. **Approval policy.** Codex `default_tools_approval_mode = "writes"`, or Zed `confirm` [24][25]. Never set a global "always allow" on write tools.
6. **Keep SDKs patched.** Python $\geq 1.27.2$ (or v2.x). TS $\geq 1.24.0$ (or v2.x). Inspector $\geq 0.14.1$ [66][70][71].
7. **Turn off vendor telemetry** where it exists (Chrome DevTools MCP) [39].

8. Mature vs. worth adopting vs. avoid

Mature (safe to build on): The spec process (dated revisions, a 12-month deprecation policy, conformance-tested tiers) [1][16]. The Tier-1 SDKs [17]. The stdio and Streamable HTTP transports [5]. Vendor-owned servers from Microsoft, Google, and GitHub that ship weekly or monthly releases [78].

Worth adopting, with conditions: FastMCP 4.x (third-party, but fast-moving and well-aligned [3][21]). ToolHive / Docker Gateway for isolation. SearXNG + basic-memory / mcp-memory-service for sovereign research and memory. MCP Toolbox and DuckDB MCP for data.

Not mature yet: The Registry (preview [13]). The tool-result shape (redesign planned [9]). URL-mode elicitation (flagged as possibly changing [7]). Agent-identity auth (roadmap [9]). Tasks (an extension, heading toward core [9]).

Avoid: - Archived or unmaintained hosts and tools: Continue [31], Roo Code [32], mcphost, mcp-router, Browserbase self-host MCP [40], `servers-archived` (Puppeteer, SQLite, Postgres, and others) [20]. - Stale community bridges or aggregators as critical-path components: metamcp (last release 2025-12-19), lastmile `mcp-agent` (last commit 2026-01-25), mcpo (last commit 2026-02-27) [78]. - Building new features on deprecated Roots, Sampling, Logging, or HTTP+SSE [2]. - Unvetted registry entries. The registry does not scan code [13].

9. Recommendations tied to the client's stated priorities

Local-first / sovereign - Default to **stdio servers launched by your host**, containerized through ToolHive. Expose remote Streamable HTTP only where you need edge access, and put agentgateway or ToolHive's gateway in front with OAuth + CIMD [6][55][57]. - Search: self-host SearXNG and use mcp-searxng [43]. Memory: basic-memory (Markdown in git) [46]. Data: DuckDB MCP [51]. - Context7 is convenient, but its backend is closed and it carries a 2026 prompt-injection CVE [41][73]. For sovereign docs retrieval, consider indexing docs yourself into basic-memory or mcp-memory-service.

No vendor lock-in - Use servers that are open source *and* self-hostable: Playwright MCP, GitHub MCP (local), MCP Toolbox, DuckDB, SearXNG, basic-memory, mcp-memory-service, Graphiti. Hosted-only pieces are Context7's backend, Figma write-to-canvas (future usage-based pricing [53]), and paid search APIs. - Watch licenses. **Serena moved to GPL-3.0-or-later** in September 2026 [42]. basic-memory and Cherry Studio are **AGPL-3.0** [78]. Fine for internal use; review before any product redistribution.

Budget-conscious - Everything in the adopt table is free to self-host. The costs are optional APIs (Brave, Exa, Tavily, Firecrawl cloud), Figma after the beta, and Docker Desktop licensing for the Toolkit UI (Docker's terms weren't checked here, so **unknown**).

Production-grade - Build on 2026-07-28 and a Tier-1 SDK. Add OTel via the `_meta` trace context conventions [2]. Write explicit handles for state [2]. Keep tool lists deterministic for prompt-cache hits [2]. Make pinning, containerization, and least privilege non-negotiable (\$7.4).

By type of work - *Agentic coding*: opencode or Codex CLI + GitHub MCP (read-only, trimmed toolsets) + reference git server (patched) or host-native git + Serena (mind the GPL) + Playwright MCP for E2E. - *Deep research*: goose or opencode + mcp-searxng + reference fetch + basic-memory / Graphiti + DuckDB MCP. Isolate this "untrusted content" agent from write-capable coding tools. - *Browser + creative*: Playwright MCP (`--isolated`), Chrome DevTools MCP (telemetry off), mcp-for-blender (safe mode), and Figma MCP or Framelink.

10. Risks, unknowns, and things I could not verify

- **Rust SDK status conflict.** The blog says “beta” for 2026-07-28 support [3]; the SDK page says Tier 1 [17].
- **Context7 CVE fix status** for the hosted Custom AI Instructions feature: **unknown**. NVD lists “through 2.1.2”, but the npm MCP package is now 4.1.1 [73][78], and the version schemes may not correspond.
- **Registry size.** 35,953 is my own API count at 06:51 PT, and one earlier full pass hit an HTTP 500 [15]. There’s no official figure on the pages I read.
- **Host-by-host support for 2026-07-28.** I didn’t verify which hosts (opencode, goose, Zed, Cline, LM Studio, Open WebUI) have implemented the new stateless protocol vs. 2025-11-25 . Open WebUI’s docs still reference `session.initialize()` timeouts [22], which suggests older-protocol clients.
- **Local-model quality with MCP.** I found no primary benchmark. Vendors (Open WebUI, LM Studio, Cline) warn about context bloat and weak tool use [22][26][28]. Model recommendations in third-party blogs weren’t used as evidence.
- **[Redacted: one client-specific hosting platform]** was not researched in this pass. Its MCP support is **unknown** here.
- **mem0/OpenMemory MCP details, browser-use MCP details, comfyui-mcp-server details, exa/tavily/perplexity/jina server specifics:** metadata only, not deep-read.
- **Pricing** for paid APIs and Docker Desktop: not checked.
- **agentgateway governance** (LF-hosted or not): not verified.
- **GitHub star counts** are snapshots from 2026-09-25 and are only a popularity signal, not quality.
- The unauthenticated GitHub API was rate-limited in the research environment. Repo metadata came from public HTML pages and Atom feeds instead [78]. License `NOASSERTION` means GitHub couldn’t classify the license file, not that there is no license.

Appendix A. Repo metadata snapshot (2026-09-25)

Repo (as fetched → final)	License (GitHub SPDX)	Archived	Stars	Last commit (UTC)	Latest release (date)
1mcp-app/agent	Apache-2.0	false	507	2026-09-22	Release v0.39.0-09-20)
agentgateway/agentgateway	Apache-2.0	false	5,039	2026-09-25	v1.6.0-alpha.2 (2026-09-25)
ahujasid/blender-mcp → ahujasid/mcp-for-blender	MIT	false	29,326	2026-09-25	—

Repo (as fetched → final)	License (GitHub SPDX)	Archived	Stars	Last commit (UTC)	Latest release (version)
AmoyLab/Unla	MIT	false	2,235	2026-08-27	v0.10.0 (2026-08-27)
awslabs/mcp	Apache-2.0	false	9,729	2026-09-23	2026.09.20260923 (2026-09-22)
badlogic/pi-mono → earendil-works/pi	MIT	false	109,292	2026-09-25	v0.87.1 (2026-09-25)
basicmachines-co/basic-memory	AGPL-3.0	false	4,037	2026-09-24	v0.23.2 (2026-09-24)
block/goose → aaif-goose/goose	Apache-2.0	false	54,640	2026-09-25	gdk-v0.1.0-alpha (2026-09-25) chore(GDK): release v0.1.0-alpha (2026-09-24)
brave/brave-search-mcp-server	MIT	false	1,469	2026-09-17	v2.1.4 (2026-09-17)
browser-use/browser-use	MIT	false	116,250	2026-09-15	0.13.10 (2026-09-15)
browserbase/mcp-server-browserbase	Apache-2.0	true	3,413	2026-07-20	v3.0.0 (2026-03-20)
CherryHQ/cherry-studio	AGPL-3.0	false	52,142	2026-09-25	v2.1.3 (2026-09-25)
ChromeDevTools/chrome-devtools-mcp	Apache-2.0	false	52,592	2026-09-25	chrome-devtools-mcp v1.10.1 (2026-09-25)
cisco-ai-defense/mcp-scanner	Apache-2.0	false	1,079	2026-09-19	4.8.4 (2026-08-20)
cline/cline	Apache-2.0	false	69,283	2026-09-25	Desktop v0.0.36 (2026-09-25)
cloudflare/mcp-server-cloudflare	Apache-2.0	false	4,283	2026-09-24	workers-observability (2026-08-11)
continuedev/continue	Apache-2.0	false	36,022	2026-07-21	v2.1.0-vscode (2026-07-21)

Repo (as fetched → final)	License (GitHub SPDX)	Archived	Stars	Last commit (UTC)	Latest release (date)
crystaldba/postgres-mcp	MIT	false	3,340	2026-08-16	PostgreSQL MCP (2025-05-16)
cyanheads/git-mcp-server	Apache-2.0	false	241	2026-08-24	v2.15.3: Flag allocation regression: working (2026-08-24)
danny-avila/LibreChat → LibreChat-AI/LibreChat	MIT	false	44,933	2026-09-25	v0.8.8-rc4 (2026-09-25)
docker/mcp-gateway	MIT	false	1,584	2026-09-16	v0.44.1 (2026-09-16)
docker/mcp-registry	MIT	false	557	2026-09-16	—
doobidoo/mcp-memory-service	Apache-2.0	false	1,965	2026-09-25	v11.14.0 (2026-09-25)
envoyproxy/ai-gateway → theagentrouter/agent-router	Apache-2.0	false	2,142	2026-09-25	v1.1.0 (2026-08-26)
exa-labs/exa-mcp-server	MIT	false	5,049	2026-09-23	—
executeautomation/mcp-playwright	MIT	false	5,655	2025-12-13	—
firecrawl/firecrawl-mcp-server	MIT	false	7,512	2026-09-25	v3.24.1: release: v3.24.1 (2026-08-26)
getzep/graphiti	Apache-2.0	false	31,152	2026-09-24	v0.30.2 - FalkorT (2026-09-08)
ggml-org/llama.cpp	MIT	false	129,487	2026-09-25	b11178 (2026-09-25)
github/github-mcp-server	MIT	false	33,188	2026-09-16	GitHub MCP Server (2026-09-16)
GLips/Figma-Context-MCP	MIT	false	15,910	2026-06-24	v0.13.2 (2026-06-24)

Repo (as fetched → final)	License (GitHub SPDX)	Archived	Stars	Last commit (UTC)	Latest release (if any)
google-gemini/gemini-cli	Apache-2.0	false	107,162	2026-09-24	Release v0.62.0-nightly.20260921 (2026-09-25)
googleapis/genai-toolbox → googleapis/mcp-toolbox	Apache-2.0	false	16,490	2026-09-25	v1.13.0 (2026-09-25)
hangwin/mcp-chrome	MIT	false	12,453	2026-01-06	v1.0.0 (2025-12-20)
highflame-ai/ramparts	Apache-2.0	false	96	2026-08-21	v0.8.9 (2026-09-25)
IBM/mcp-context-forge	Apache-2.0	false	4,528	2026-09-25	v1.0.7-20260921 Upstream MCP (2026-09-21)
idosal/git-mcp	Apache-2.0	false	8,423	2026-05-08	—
ihor-sokoliuk/mcp-searxng	MIT	false	1,260	2026-09-22	v2.4.0 (2026-09-22)
invariantlabs-ai/mcp-scan → snyk/agent-scan	Apache-2.0	false	3,090	2026-09-25	Agent Scan v0.6a6af81c-1733 (2026-09-25)
jina-ai/MCP	Apache-2.0	false	865	2026-09-18	—
joenorton/comfyui-mcp-server	Apache-2.0	false	410	2026-02-17	v1.1.1 - QoL, Bugfixes (2026-02-17)
jonigl/mcp-client-for-ollama	MIT	false	826	2026-09-11	Release v0.35.0 (2026-09-11)
Kilo-Org/kilocode	MIT	false	27,415	2026-09-25	v7.8.0 (pre-release 25)
Kong/kong	Apache-2.0	false	44,195	2026-09-23	3.9.3 (2026-06-10)
lasso-security/mcp-gateway	MIT	false	390	2026-01-22	Lasso Guardrail: Support (2026-09-25)

Repo (as fetched → final)	License (GitHub SPDX)	Archived	Stars	Last commit (UTC)	Latest release (version)
lastmile-ai/mcp-agent	Apache-2.0	false	8,561	2026-01-25	v0.2.6 (2025-12-25)
mark3labs/mcphost	MIT	true	1,596	2026-04-13	v0.34.0 (2026-03-25)
mcp-router/mcp-router	NOASSERTION	true	2,147	2026-09-18	v0.6.4 — Final release v0.6.4 終了 (2026-09-18)
mcp-use/mcp-use	MIT	false	10,676	2026-09-24	mcp-use 2.7.1- canary (2026-09-24)
mcpjungle/MCPJungle	MPL-2.0	false	1,280	2026-08-02	0.4.6 (2026-08-02)
mem0ai/mem0	Apache-2.0	false	65,987	2026-09-25	Mem0 Python Sdk v0.1.1 (2026-09-23)
metatool-ai/metamcp	MIT	false	2,686	2026-06-22	2.4.22 Security update custom headers (2025-12-19)
microsoft/markitdown	MIT	false	186,980	2026-09-21	Version 0.1.8 (2026-09-21)
microsoft/mcp	MIT	false	3,710	2026-09-24	Azure.Mcp.Service beta.47 (2026-09-24)
microsoft/mcp-gateway	MIT	false	851	2026-08-25	—
microsoft/playwright-mcp	Apache-2.0	false	37,559	2026-09-18	v0.0.82 (2026-09-18)
Mintplex-Labs/anything-llm	MIT	false	66,457	2026-09-24	AnythingLLM v1.1.0 (2026-09-22)
modelcontextprotocol/conformance	NOASSERTION	false	127	2026-09-11	v0.1.16 (2026-03-25)
modelcontextprotocol/csharp-sdk	NOASSERTION	false	4,541	2026-09-19	v2.2.0 (2026-08-25)

Repo (as fetched → final)	License (GitHub SPDX)	Archived	Stars	Last commit (UTC)	Latest release (date)
modelcontextprotocol/ext-apps	NOASSERTION	false	2,872	2026-09-25	v2.0.1 (2026-09-25)
modelcontextprotocol/go-sdk	NOASSERTION	false	5,147	2026-09-24	v1.8.0 (2026-09-24)
modelcontextprotocol/inspector	NOASSERTION	false	10,954	2026-09-23	2.8.0 (2026-09-23)
modelcontextprotocol/java-sdk	MIT	false	3,711	2026-09-18	v2.0.1 (2026-08-25)
modelcontextprotocol/kotlin-sdk	NOASSERTION	false	1,462	2026-09-25	0.15.0 (2026-07-28)
modelcontextprotocol/modelcontextprotocol	NOASSERTION	false	9,303	2026-09-24	2026-07-28 (2026-07-28)
modelcontextprotocol/php-sdk	NOASSERTION	false	1,613	2026-09-15	v0.8.1 (2026-08-25)
modelcontextprotocol/python-sdk	MIT	false	24,394	2026-09-23	v2.2.0 (2026-09-23)
modelcontextprotocol/registry	NOASSERTION	false	7,285	2026-09-22	v1.8.1 (2026-08-25)
modelcontextprotocol/ruby-sdk	NOASSERTION	false	918	2026-09-25	v1.6.0 (2026-09-25)
modelcontextprotocol/rust-sdk	NOASSERTION	false	3,950	2026-09-24	rmcp-v3.4.1 (2026-09-24)
modelcontextprotocol/servers	NOASSERTION	false	90,586	2026-09-22	Release 2026.8.1 (2026-09-21 13:31)
modelcontextprotocol/servers-archived	MIT	true	304	2025-05-28	—
modelcontextprotocol/swift-sdk	NOASSERTION	false	1,498	2026-04-29	0.12.1 (2026-05-29)
modelcontextprotocol/typescript-sdk	NOASSERTION	false	13,454	2026-09-23	1.30.1 (2026-09-23)

Repo (as fetched → final)	License (GitHub SPDX)	Archived	Stars	Last commit (UTC)	Latest release (version)
motherduckdb/mcp-server-motherduck	MIT	false	524	2026-08-19	v1.0.8 (2026-08-19)
NVIDIA/garak	Apache-2.0	false	9,353	2026-09-16	v0.17.0 (2026-09-16)
obot-platform/obot	MIT	false	1,050	2026-09-24	v0.25.6 (2026-09-24)
ollama/ollama	MIT	false	181,682	2026-09-24	v0.40.0 (2026-09-24)
open-webui/mcpo	MIT	false	4,386	2026-02-27	v0.0.20 (2026-02-27)
open-webui/open-webui	NOASSERTION	false	153,141	2026-09-21	v0.11.4 (2026-09-21)
openai/codex	Apache-2.0	false	126,416	2026-09-25	0.158.0-alpha.14 (2026-09-25)
oraios/serena	NOASSERTION	false	29,793	2026-09-24	mit-final (2026-09-24)
perplexityai/modelcontextprotocol	MIT	false	2,542	2026-08-27	—
pomerium/pomerium	Apache-2.0	false	5,016	2026-09-25	v0.33.3 (2026-09-25)
promptfoo/promptfoo	MIT	false	25,447	2026-09-25	0.123.1 (2026-09-25)
punkpeye/awesome-mcp-servers	MIT	false	95,514	2026-09-23	—
redis/mcp-redis	MIT	false	626	2026-09-21	0.5.1 (2026-08-01)
riseandignite/mcp-shield	MIT	false	555	2025-04-26	—
RooCodeInc/Roo-Code	Apache-2.0	true	24,297	2026-05-15	Release v3.54.0 (2026-05-15)

Repo (as fetched → final)	License (GitHub SPDX)	Archived	Stars	Last commit (UTC)	Latest release (version)
smithery-ai/cli → arcadeai-labs/smithery-cli	AGPL-3.0	false	836	2026-05-31	v1.2.0 (2026-05-31)
snyk/agent-scan	Apache-2.0	false	3,090	2026-09-25	Agent Scan v0.6.0 (2026-09-25) a6af81c-1733 (2026-09-25)
sparfenyuk/mcp-proxy	MIT	false	2,767	2026-05-14	v0.12.0 (2026-05-14)
sst/opencode → anomalyco/opencode	MIT	false	209,973	2026-09-25	v2.0.16 (2026-09-25)
stacklok/toolhive	Apache-2.0	false	2,206	2026-09-25	v0.51.2 (2026-09-25)
stripe/agent-toolkit → stripe/ai	MIT	false	1,832	2026-09-25	—
supabase-community/supabase-mcp → supabase/mcp	Apache-2.0	false	2,921	2026-09-23	mcp-server-supabase v0.13.0 (2026-09-23)
tavily-ai/tavily-mcp	MIT	false	2,407	2026-09-16	—
TBXark/mcp-proxy	MIT	false	734	2026-09-15	v1.1.0 (2026-09-15)
trailofbits/mcp-context-protector	Apache-2.0	false	226	2026-02-13	—
upstash/context7	MIT	false	62,417	2026-09-24	@upstash/context7-sdk@1.0.2 (2026-09-24)
vllm-project/vllm	Apache-2.0	false	92,665	2026-09-25	v0.30.1rc0: [ROCm] MI355 dense NV (2026-09-23)
wonderwhy-er/DesktopCommanderMCP	MIT	false	9,742	2026-09-25	Release Notes – Desktop Commander MCP (2026-09-17)
zed-industries/zed	NOASSERTION	false	90,869	2026-09-25	nightly: cloud_agent Update yawc to 0.1.0 (2026-09-24)

Sources

1. MCP Versioning: <https://modelcontextprotocol.io/specification/versioning>
2. MCP 2026-07-28 Changelog: <https://modelcontextprotocol.io/specification/2026-07-28/changelog>
3. MCP Blog, "The 2026-07-28 Specification" (2026-07-28): <https://blog.modelcontextprotocol.io/posts/2026-07-28/>
4. Spec releases on GitHub: <https://github.com/modelcontextprotocol/modelcontextprotocol/releases>
5. Transports (2026-07-28): <https://modelcontextprotocol.io/specification/2026-07-28/basic/transports>
6. Authorization (2026-07-28): <https://modelcontextprotocol.io/specification/2026-07-28/basic/authorization>
7. Elicitation (2026-07-28): <https://modelcontextprotocol.io/specification/2026-07-28/client/elicitation>
8. Tools (2026-07-28): <https://modelcontextprotocol.io/specification/2026-07-28/server/tools>
9. MCP Roadmap (updated 2026-08-22): <https://modelcontextprotocol.io/development/roadmap>
10. Governance and Stewardship: <https://modelcontextprotocol.io/community/governance>
11. MCP Blog, "MCP joins the Agentic AI Foundation" (2025-12-09):
<https://blog.modelcontextprotocol.io/posts/2025-12-09-mcp-joins-agentic-ai-foundation/>
12. Linux Foundation AAIF press release: <https://www.linuxfoundation.org/press/linux-foundation-announces-the-formation-of-the-agentic-ai-foundation>
13. The MCP Registry (about): <https://modelcontextprotocol.io/registry/about>
14. Registry repo README: <https://github.com/modelcontextprotocol/registry>
15. Registry API (own count, 2026-09-25 06:51 PT): <https://registry.modelcontextprotocol.io/v0.1/servers?version=latest>
16. SDK Tiering System: <https://modelcontextprotocol.io/community/sdk-tiers>
17. SDK list with tiers: <https://modelcontextprotocol.io/docs/sdk>
18. TypeScript SDK README: <https://github.com/modelcontextprotocol/typescript-sdk>
19. MCP Blog index (Ruby SDK 1.0, 2026-07-27): <https://blog.modelcontextprotocol.io/>
20. Reference servers README + LICENSE: <https://github.com/modelcontextprotocol/servers>
21. FastMCP releases: <https://github.com/PrefectHQ/fastmcp/releases>
22. Open WebUI MCP docs: <https://docs.openwebui.com/features/extensibility/mcp/>
23. opencode MCP servers docs: <https://opencode.ai/docs/mcp-servers/>
24. Zed MCP docs: <https://zed.dev/docs/ai/mcp>
25. Codex MCP docs: <https://developers.openai.com/codex/mcp>
26. LM Studio MCP docs: <https://lmstudio.ai/docs/app/mcp>
27. Cline, running models locally: <https://docs.cline.bot/running-models-locally/overview>
28. Cline blog, local models (compact prompt drops MCP): <https://cline.bot/blog/local-models>
29. Cline MCP overview: <https://docs.cline.bot/mcp/mcp-overview>
30. goose README: <https://github.com/aaif-goose/goose>
31. Continue README (read-only notice): <https://github.com/continuedev/continue>
32. Roo Code repo (archived): <https://github.com/RooCodeInc/Roo-Code>
33. Ollama PR #13700 (closed, unmerged): <https://github.com/ollama/ollama/pull/13700>

34. mcp-client-for-ollama (ollmcp): <https://github.com/jonigl/mcp-client-for-ollama>
35. mcphost (archived): <https://github.com/mark3labs/mcphost>
36. mcpo README: <https://github.com/open-webui/mcpo>
37. GitHub MCP Server README: <https://github.com/github/github-mcp-server>
38. Playwright MCP README: <https://github.com/microsoft/playwright-mcp>
39. Chrome DevTools MCP README: <https://github.com/ChromeDevTools/chrome-devtools-mcp>
40. Browserbase MCP (archived): <https://github.com/browserbase/mcp-server-browserbase>
41. Context7 README: <https://github.com/upstash/context7>
42. Serena LICENSE and `mit-final` tag: <https://github.com/oraios/serena/blob/main/LICENSE> ;
<https://github.com/oraios/serena/releases/tag/mit-final>
43. mcp-searxng README: <https://github.com/ihor-sokoliuk/mcp-searxng>
44. Firecrawl MCP README: <https://github.com/firecrawl/firecrawl-mcp-server>
45. Graphiti MCP server README: https://github.com/getzep/graphiti/tree/main/mcp_server
46. basic-memory README: <https://github.com/basicmachines-co/basic-memory>
47. mcp-memory-service README: <https://github.com/doobidoo/mcp-memory-service>
48. MCP Toolbox for Databases README: <https://github.com/googleapis/mcp-toolbox>
49. postgres-mcp README: <https://github.com/crystaldba/postgres-mcp>
50. Supabase MCP README: <https://github.com/supabase/mcp>
51. DuckDB/MotherDuck local MCP README: <https://github.com/motherduckdb/mcp-server-motherduck>
52. MCP for Blender README: <https://github.com/ahujasid/mcp-for-blender>
53. Figma, Guide to the Figma MCP server: <https://help.figma.com/hc/en-us/articles/32132100833559-Guide-to-the-Figma-MCP-server>
54. Docker MCP Gateway README: <https://github.com/docker/mcp-gateway>
55. ToolHive README: <https://github.com/stacklok/toolhive>
56. IBM ContextForge README: <https://github.com/IBM/mcp-context-forge>
57. agentgateway README: <https://github.com/agentgateway/agentgateway>
58. Snyk Agent Scan README: <https://github.com/snyk/agent-scan>
59. Cisco AI Defense MCP Scanner README: <https://github.com/cisco-ai-defense/mcp-scanner>
60. Trail of Bits mcp-context-protector README: <https://github.com/trailofbits/mcp-context-protector>
61. MCP Security Best Practices (2026-07-28): https://modelcontextprotocol.io/specification/2026-07-28/basic/security_best_practices
62. Invariant Labs, Tool Poisoning Attacks (2025-04-01): <https://invariantlabs.ai/blog/mcp-security-notification-tool-poisoning-attacks>
63. Invariant Labs, GitHub MCP Exploited (2025-05-26): <https://invariantlabs.ai/blog/mcp-github-vulnerability>
64. OWASP MCP Top 10: <https://owasp.org/www-project-mcp-top-10/>
65. NVD CVE-2025-6514 (mcp-remote): <https://nvd.nist.gov/vuln/detail/CVE-2025-6514>
66. NVD CVE-2025-49596 (MCP Inspector): <https://nvd.nist.gov/vuln/detail/CVE-2025-49596>
67. NVD CVE-2025-53109 (filesystem symlink): <https://nvd.nist.gov/vuln/detail/CVE-2025-53109>
68. NVD CVE-2025-53110 (filesystem prefix): <https://nvd.nist.gov/vuln/detail/CVE-2025-53110>

69. NVD CVE-2025-68143 / 68144 / 68145 (mcp-server-git): <https://nvd.nist.gov/vuln/detail/CVE-2025-68143> ;
<https://nvd.nist.gov/vuln/detail/CVE-2025-68144> ; <https://nvd.nist.gov/vuln/detail/CVE-2025-68145>
70. NVD CVE-2025-66414 (TS SDK DNS rebinding): <https://nvd.nist.gov/vuln/detail/CVE-2025-66414>
71. NVD CVE-2026-52869 (Python SDK session authz): <https://nvd.nist.gov/vuln/detail/CVE-2026-52869>
72. NVD CVE-2026-46519 (mcp-server-kubernetes): <https://nvd.nist.gov/vuln/detail/CVE-2026-46519>
73. NVD CVE-2026-75130 (Context7 prompt injection): <https://nvd.nist.gov/vuln/detail/CVE-2026-75130>
74. NVD CVE-2026-73496 (MCP Atlassian path traversal): <https://nvd.nist.gov/vuln/detail/CVE-2026-73496>
75. OSV MAL-2025-47604 (postmark-mcp): <https://osv.dev/vulnerability/MAL-2025-47604>
76. The Hacker News on postmark-mcp (2025-09-29): <https://thehackernews.com/2025/09/first-malicious-mcp-server-found.html>
77. Pi (pi.dev): <https://pi.dev>
78. GitHub repo pages and Atom feeds (/commits.atom , /releases.atom) for every repo in Appendix A, fetched 2026-09-25. Example: <https://github.com/microsoft/playwright-mcp/releases.atom>
79. mcp-router (archived, final release): <https://github.com/mcp-router/mcp-router>
80. Anthropic, Donating MCP and establishing the AAIF: <https://www.anthropic.com/news/donating-the-model-context-protocol-and-establishing-of-the-agentic-ai-foundation>