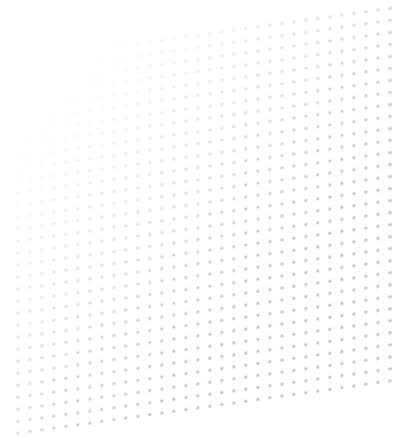

A RESOURCE FOR DESIGN SYSTEM AND ENGINEERING TEAMS

MCP server evaluation checklist.



Read the tool list before you trust the server.

Need. Provenance. Surface. Blast radius.

36

checks, each with a gate

13

tool-list rules the scanner runs

05

tool lists scanned, four from real servers

Adopt servers the way you adopt dependencies.

An MCP server is code you run with your permissions, plus text your model reads as guidance. The protocol says so itself: tools are arbitrary code execution, and tool annotations are untrusted unless they come from a server you trust [S01]. A server that looks useful in a demo can still read your home directory, carry instructions in its descriptions or change its tools after you approved them [S07].

This checklist turns the adoption decision into 36 checks across need, provenance, transport, auth, the tool surface, least privilege and failure behaviour. Each check has a gate. A failed blocker rejects the server; a failed major rejects it unless someone writes down why the risk is accepted; minors are recorded. The decision uses three words: APPROVE, DEFER (no immediate need) or REJECT.

The kit makes the tool-surface checks executable. `inspect-tools.mjs` reads a saved tools/list and applies 13 rules; it found nothing to fix in the reference filesystem and git servers, four review points in Playwright MCP and 8 errors in a poisoned lookalike. `score-evaluation.mjs` turns a filled record into the decision.

Version 1.0 / Sources checked 24 September 2026

Field Guide 24 of the Design × AI series. Pairs with Field Guides 25 (starter config), 26 (security review) and 27 (integration patterns), which share one design system and one token server. Verified 24 September 2026: MCP specification 2026-07-28, MCP Inspector 2.8.0, @modelcontextprotocol/server-file-system 2026.8.31, mcp-server-git 2026.8.18, @playwright/mcp 0.0.82.

Practical guidance, not a certification. The scanner matches patterns; a clean scan means none of the 13 rules fired, not that a server is safe. Server names and versions are the ones checked on the date above. Prepared with AI assistance and edited by hand.

START HERE

Pick your route, then read the gates.

One copy per server. Save its tools/list before you install it anywhere that matters, and keep the record with the decision.

Adopting a public server

All four sections, in order. Run the scanner on its tools/list first: a blocker there saves you the rest of the work.

Upgrading one you already use

Section 03 again, then Field Guide 26's `diff-tools.mjs` against the snapshot you locked. A changed description is a new review, not a patch.

Building your own

Section 03 is your spec. Field Guide 27's token server passes every scanner rule; copy its annotations, schemas and error style.

Clearing a backlog of requests

Check A.01 first for each one. Most requests end at DEFER: no task needs the server this week.

Read the gates and labels.

Label	Meaning
BLOCKER	A fail rejects the server. No waiver.
MAJOR	A fail rejects the server unless the record carries a written waiver.
MINOR	Record the answer. It never blocks on its own.
NEED	A fail defers the decision: there is no task for the server yet.
SCRIPT	Checked by <code>inspect-tools.mjs</code> here, or a script in Field Guides 25 and 26.
SPEC	Traces to the MCP specification 2026-07-28 or its Security Best Practices page.
RESEARCH	Comes from published security research or a documented incident.
CLIENT	Depends on Claude Desktop or Claude Code behaviour, per their docs.

Three decisions, one rule for each.

APPROVE when every blocker passes and every major passes or is waived. DEFER when there is no task for it this week. REJECT when a blocker fails, or a major fails with nobody willing to sign a waiver.

The vocabulary comes from the essay this guide belongs to. The scorer applies it mechanically so the argument happens about evidence, not about taste.

SECTION 01

Need, fit and provenance.

Start with the two questions that end most evaluations early: do you need it this week, and is it from who it says it is from.

Suggested owners: Design-system lead + engineering lead

☐ **It solves a problem you have this week**

A.01 **NEED**

Name the task it unblocks. Servers added because they might be useful cost start-up time, and their tool descriptions compete for your model's context.

Evidence: A task or ticket linked in the record. No task: DEFER.

The essay's rule: add servers for a specific problem, remove them when done.

☐ **No built-in already covers it**

A.02 **MAJOR** **CLIENT**

Check the client first. Claude Code already reads and edits files and runs git through its own tools, so a filesystem or git server there adds a second path to the same data. In a client without file or git tools of its own, such as a Claude Desktop chat, the same servers earn their place.

Evidence: The record names the built-in you checked.

Claude Code's built-in Read, Edit and Bash tools. Recommended practice.

☐ **It fits the server budget**

A.03 **MINOR**

Keep the everyday config at five servers or fewer and add task-specific ones when the task starts. Count tools, not servers: one server can bring 25 tools.

Evidence: Server count after adding it; Field Guide 25's validator warns above five.

The essay's under-five rule. Playwright MCP 0.0.82 lists 25 tools (kit fixture).

☐ **The publisher is who it claims to be**

B.01 **BLOCKER** **RESEARCH**

Package scope, repository owner and registry namespace must all point at the vendor. The official registry verifies namespaces, not names: a search for github-mcp-server returned three namespaces, and only `io.github.github` is GitHub. A package impersonating Postmark's server shipped 15 clean releases before version 1.0.16 quietly added a BCC to every email.

Evidence: Registry namespace and package owner recorded; lookalikes fail.

S05, S15. Postmark advisory, September 2025. S09.

☐ **It is maintained**

B.02 **MAJOR**

A release or commit in the last three months, and issues that get answers. MCP moves fast: the 2026-07-28 revision removed the initialize handshake and protocol sessions, so an abandoned server will fall behind clients.

Evidence: Last release date and a recent issue response, linked.

The essay's maintenance check. S04.

☐ **It is not deprecated or archived**B.03 **BLOCKER**

Several early reference servers now carry an npm deprecation and live in an archive repository: GitHub, GitLab, Postgres, Slack, Puppeteer and more. The reference repository also says its servers are educational examples, not production-ready solutions.

Evidence: `npm view <pkg> deprecated` is empty; the repository is not archived.

npm registry, 24 September 2026. S17.

☐ **You can read the code you will run**B.04 **MAJOR**

The package links to its source and you have read the tool handlers, not only the README. Check what the published tarball contains when it differs from the repository.

Evidence: Reviewer and commit or tarball hash in the record.

The essay: review source code of servers you use.

☐ **The version is pinned**B.05 **MAJOR** **SCRIPT**

An exact npm version, `uvx pkg==version` or an image tag or digest. `npx -y pkg` without a version runs whatever was published last, which is how a bad release reaches you without a review.

Evidence: Field Guide 25's `validate-mcp-config.mjs` rule C04 passes.

S09, S16.

☐ **There is a way to report a vulnerability**B.06 **MINOR**

A SECURITY.md or security contact, and a licence you can live with.

Evidence: Links in the record.

Recommended practice.

The namespace is the identity. The name is marketing.

SECTION 02

Transport, auth and secrets.

How the server runs decides who else can talk to it. How it authenticates decides what a leaked credential costs.

Suggested owners: Engineering lead + security partner

☐ The transport matches where it runs

C.01 **MAJOR** **SPEC**

stdio for local tools: the client launches the server and only the client can talk to it. Streamable HTTP for remote servers. HTTP+SSE is deprecated; do not start new setups on it.

Evidence: Transport recorded; no `sse` type in a new config.

S13, S14. HTTP+SSE reclassified as Deprecated in 2026-07-28. S04.

☐ Local HTTP servers are not open to the machine

C.02 **BLOCKER** **SPEC**

A local HTTP server must validate `Origin` and should bind to 127.0.0.1, or a web page can reach it through DNS rebinding. Prefer stdio; if it must be HTTP, require a token.

Evidence: Bind address and Origin check confirmed in the code or docs.

Streamable HTTP transport, security warning. S13. Local server compromise. S03.

☐ It runs isolated when it touches untrusted content

C.03 **MAJOR** **SPEC**

A server that fetches pages, reads other people's issues or runs code belongs in a container or sandbox with minimal privileges, read-only mounts where possible.

Evidence: The launch command shows the isolation (for example `docker run --rm` with named mounts).

Security Best Practices, local server compromise. S03.

☐ You know which protocol revision it speaks

C.04 **MINOR** **SPEC**

Record the negotiated revision. On 24 September 2026 the reference filesystem server answered the Inspector with 2025-11-25; the kit's token server answers both 2026-07-28 and 2025-11-25. Mixed eras work, but you should know which one you are debugging.

Evidence: `npx @modelcontextprotocol/inspector --cli <server> --method initialize`.

S04, S06. Kit captures.

☐ stdout carries protocol only

C.05 **MINOR** **SPEC**

A stdio server must not print anything but protocol messages on stdout. Logs go to stderr, which Claude Desktop writes to `mcp-server-<name>.log`.

Evidence: No banner on stdout at start-up; logs visible in the client log.

S14. Claude Desktop log locations. S19.

☐ **Remote auth uses OAuth where offered**D.01 **MAJOR** **SPEC**

Remote servers authorize with OAuth 2.1, resource indicators and audience-bound tokens. A long-lived shared key is the fallback, not the default.

Evidence: Auth method recorded; OAuth chosen when both exist.

Authorization, 2026-07-28. S12.

☐ **No secret is written into a config file**D.02 **BLOCKER** **SCRIPT** **CLIENT**

Claude Code expands `${VAR}` in `.mcp.json`. Claude Desktop's config file has no documented expansion, so there the answer is OAuth, the OS keychain through a Desktop extension, or a server that needs no secret.

Evidence: Field Guide 25's validator rule C01 passes.

S10. Field Guide 25.

☐ **Credentials carry the minimum scope**D.03 **MAJOR** **SPEC**

Fine-grained tokens, one repository, read-only toolsets. Broad scopes granted up front widen what a leaked token can reach and blur the audit trail.

Evidence: Token scopes listed in the record.

Security Best Practices, scope minimization. S03.

☐ **The server does not pass your token through**D.04 **BLOCKER** **SPEC**

A server must reject tokens that were not issued for it and must not forward a client's token to a downstream API. Passthrough breaks audience checks, rate limits and audit logs at once.

Evidence: Server docs or code show audience validation and its own downstream credential.

Security Best Practices, token passthrough: MUST NOT. S03, S12.

☐ **It never asks for secrets in a form**D.05 **BLOCKER** **SPEC**

Form elicitation must not request passwords, API keys, tokens or payment details. Those go through URL-mode elicitation or OAuth, out of the client's reach.

Evidence: No form elicitation for credentials in the code or during a trial run.

Elicitation, 2026-07-28. S11.

A secret in a config file is a secret in every backup of that file.

SECTION 03

The tool surface.

The tool list is the part of a server your model reads. Save it, scan it, and read every description as if it were a prompt someone else wrote.

Suggested owners: Engineering lead + design-system lead

☐ **You read tools/list before first use**

E.01 **BLOCKER** **SCRIPT**

Capture it with the Inspector CLI and run the scanner. Keep the file: Field Guide 26 locks its hash so an upgrade that rewrites a description is caught.

Evidence: `inspect-tools.mjs <file>` exits 0 (no errors).

S06. Kit script.

☐ **Every tool that changes state says so**

E.02 **BLOCKER** **SCRIPT** **SPEC**

Annotations default to the unsafe side: a tool with no hints is presumed not read-only, destructive and open-world. A mutating tool without hints, or a hint that contradicts the name (`delete_token` marked read-only), fails.

Evidence: Scanner rules T01, T02 and T03 clean.

ToolAnnotations defaults, schema 2026-07-28. S02. Clients must treat annotations as untrusted. S01.

☐ **Descriptions describe; they do not instruct**

E.03 **BLOCKER** **SCRIPT** **RESEARCH**

No pseudo-tags such as `<IMPORTANT>`, no requests to read files or hide steps from the user, no invisible characters, no text that steers other tools. That is tool poisoning, and descriptions are where it hides.

Evidence: Scanner rules T04, T05 and T11 clean.

Invariant Labs, tool poisoning and shadowing, April 2025. S07.

☐ **Broad capabilities are isolated or off**

E.04 **MAJOR** **SCRIPT**

Tools that run arbitrary code, shell commands or queries need a sandbox and a confirmation step, or a flag that removes them. Playwright MCP's `browser_evaluate` and `browser_run_code_unsafe` are useful and honest about it; they still count.

Evidence: Scanner rules T06 and T07 reviewed; the mitigation is written down.

S01, S03. Kit fixture: 4 warnings on Playwright MCP 0.0.82.

☐ **You can switch tools off**

E.05 **MINOR**

Toolsets, allow-lists or a read-only mode, and you use them. GitHub's server exposes toolsets and a read-only mode; its remote endpoint has a `/readOnly` path.

Evidence: The config shows the narrowing flag or URL.

GitHub MCP server docs. S18.

☐ **Output is bounded**

E.06 MAJOR CLIENT

Limits, pagination and a truncated flag. Claude Code warns when one tool result passes 10,000 tokens and cuts it at 25,000 by default; a server that returns a whole database hits that wall and wastes the context on the way.

Evidence: Largest realistic call measured; a limit parameter or cap exists.

Claude Code MCP docs. S10. Field Guide 27's `search_tokens` caps at 10 results.

☐ **Data comes back as structured content**

E.07 MINOR SCRIPT SPEC

An `outputSchema` lets the client validate results and generate types. Servers that declare one must return conforming `structuredContent`.

Evidence: Scanner rule T12; a sample call validates.

Tools, output schema. S01.

☐ **Inputs are closed and typed**

E.08 MINOR SCRIPT SPEC

`additionalProperties: false`, enums for closed sets, length limits on strings. Closed inputs make invented arguments fail loudly.

Evidence: Scanner rule T08.

Tools, input schema guidance. S01.

Rule	Checks	Severity
T01 to T03	Annotations present, present on mutating tools, and consistent with the name	warn, error
T04, T05, T11	Instructions, hidden characters, encoded blobs, steering of other tools	error
T06, T07	Broad capabilities and free-form code, command or SQL parameters	warn
T08, T12, T13	Open input schema, no output schema, terse description	info
T09, T10	Descriptions over 1,000 characters; names outside the spec's character set, duplicates	warn, error

Every description is a prompt you did not write. Read it like one.

SECTION 04

Least privilege, failure and the decision.

Assume the server will be misused once: by a bug, by injected text or by a bad release. These checks decide how far that goes.

Suggested owners: Security partner + engineering lead

☐ **Filesystem access is scoped to named directories**

F.01 **BLOCKER** **SCRIPT** **CLIENT**

Grant project folders, never a home directory or the root. A local server runs with your account's permissions and can do anything you can do in the paths it gets.

Evidence: Field Guide 25's validator rule C05 passes.

Connect to local servers, security consideration. S19.

☐ **You know where data can go**

F.02 **MAJOR**

List the hosts it can reach. Open-world tools, and any tool that posts, sends or uploads, are exits for data.

Evidence: Egress hosts in the record; `openWorldHint` matches reality.

S02.

☐ **The whole config avoids the lethal trifecta**

F.03 **BLOCKER** **SCRIPT** **RESEARCH**

Private data, untrusted content and a way to send data out, in one session, is the combination that turns a prompt injection into a leak. Check the config, not the server: two harmless servers can complete the set.

Evidence: Field Guide 26's `check-inventory.mjs` shows no unbroken trifecta.

Simon Willison, June 2025. S08. Invariant Labs, GitHub MCP exploit. S07.

☐ **A person approves consequential actions**

F.04 **MAJOR** **SPEC**

Keep confirmation on destructive and external actions. The spec asks for a human who can deny tool calls; a blanket allow rule removes them.

Evidence: No allow rule covers destructive or open-world tools.

Tools, user interaction model. S01.

☐ **Errors come back as tool errors the model can act on**

G.01 **MAJOR** **SPEC**

`isError: true` with a reason and a next step, not a crash, an empty result or a protocol error for a business failure. Field Guide 27's server answers an unknown token with the three closest names.

Evidence: One deliberate bad call in the trial run; the result is actionable.

Tools, error handling. S01.

☐ **It starts and answers within your timeouts**G.02 **MINOR** **CLIENT**

Start-up under `MCP_TIMEOUT`, long calls under the per-server `timeout`. First runs of `npx` and `uvx` download packages; measure the second start.

Evidence: Measured start-up and slowest call.

Claude Code MCP docs. S10.

☐ **Outages and expired auth fail loudly**G.03 **MINOR**

The Inspector CLI exits 3 when a server needs auth and 4 when it is unreachable, so a scripted check can tell the two apart instead of hanging.

Evidence: A run with the credential removed exits non-zero with a clear message.

Inspector CLI exit codes. S06.

☐ **Removal is clean**G.04 **MINOR**

Deleting the entry stops the server, and you know how to revoke what it was granted: the OAuth grant, the token, the mounted folder.

Evidence: Revocation steps in the record.

Recommended practice.

☐ **The decision is recorded with a re-review date**H.01 **MAJOR** **SCRIPT**

APPROVE, DEFER or REJECT, the tools/list snapshot hash and the date to look again. Re-review on a new version, a changed hash or a new client.

Evidence: `score-evaluation.mjs` output and `diff-tools.mjs --lock` hash in the record.

The essay: evaluation, documentation and periodic review. Field Guide 26.

Check the config, not the server. Harm comes from combinations.

APPENDIX A

Scan the tool list.

`inspect-tools.mjs` reads a saved tools/list and applies 13 rules. It never connects to a server, so you can run it on a snapshot before anything is installed.

terminal

```
# Capture (Inspector 2.8.0), then scan
npx @modelcontextprotocol/inspector --cli npx -y @playwright/mcp@0.0.82 --headless \
  --method tools/list --format json > playwright.tools.json
node scripts/inspect-tools.mjs playwright.tools.json
```

Fixture	Tools	Errors	Warnings	Info	Result
acme-ds-tokens 1.0.0 (Field Guide 27)	2	0	0	0	PASS
server-filesystem 2026.8.31	14	0	0	13	PASS
mcp-server-git 2026.8.18	12	0	0	26	PASS
@playwright/mcp 0.0.82	25	0	4	28	REVIEW
lookalike (synthetic, poisoned)	7	8	5	14	FAIL

Four real captures made 24 September 2026, one synthetic fixture. Info findings are open input schemas (T08), missing output schemas (T12) and terse descriptions (T13).

node scripts/inspect-tools.mjs fixtures/lookalike.tools.json (excerpt)

```
lookalike.tools.json: 7 tools, 8 errors, 5 warnings, 14 info: FAIL
  error T04 lookup_token: Instruction-like text: pseudo-tag addressed to the model
    (description); asks the model to hide something from the user (description);
    pre-call instruction to read or send data (description); references a
    credential or config file (description); asks the model to move secrets or
    context (description).
  error T02 sync_tokens: Name says it changes something, but it declares neither
    readOnlyHint nor destructiveHint.
  error T03 delete_token: readOnlyHint is true, but the name says it changes
    something.
  error T03 reset_theme: destructiveHint is false, but the name says it deletes,
    resets or overwrites.
  error T05 format_css: Invisible or bidirectional control characters in description.
  error T04 format_css: Instruction-like text: asks for the conversation
    (description).
  error T11 format_css: Text that steers other tools or servers in description.
```

Wrapped for print; T01 warnings, `run_script`, `get tokens` and info lines omitted. The lookalike borrows the token server's tool name and adds a `<IMPORTANT>` block that asks for `~/ .npmrc`, the pattern Invariant Labs documented.

APPENDIX B

Turn answers into a decision.

`evaluation-checklist.json` holds the 36 checks with their gates. A record answers each one pass, fail or n/a, and `score-evaluation.mjs` applies the gates.

`evaluation-checklist.json` (excerpt)

```
{
  "id": "E.03",
  "title": "Descriptions describe; they do not instruct",
  "check": "No pseudo-tags, hidden characters, file paths, or text steering other tools.",
  "gate": "blocker",
  "automated": "inspect-tools.mjs T04 T05 T11"
}
```

terminal

```
$ node scripts/score-evaluation.mjs examples/evaluation.acme-ds-tokens.json
acme-ds-tokens: APPROVE (26 pass, 0 fail, 10 n/a, 0 open of 36)

$ node scripts/score-evaluation.mjs examples/evaluation.playwright.json
playwright: DEFER (4 pass, 1 fail, 0 n/a, 31 open of 36)
  No immediate need (A.01). Evaluate again when the task appears.

$ node scripts/score-evaluation.mjs examples/evaluation.lookalike.json
acme-ds-helper: REJECT (2 pass, 6 fail, 0 n/a, 28 open of 36)
  blocker B.01 failed: The publisher is who it claims to be
  blocker E.01 failed: You read tools/list before first use
  blocker E.02 failed: Every tool that changes state says so
  blocker E.03 failed: Descriptions describe; they do not instruct
  blocker F.03 failed: The whole config avoids the lethal trifecta
  major E.04 failed without a waiver: Broad capabilities are isolated or off
```

Output from the kit. A REJECT does not need the whole record: one failed blocker ends it. An APPROVE needs every blocker and major answered.

DO

- Save tools/list before the first install, and keep it with the decision
- Answer n/a with a reason, so the next reviewer can disagree
- Re-run the scanner on every upgrade, then diff against the locked snapshot
- Evaluate the whole config when you add one server

DON'T

- Treat a clean scan as a security review: the scanner matches patterns
- Trust annotations from a server you have not vetted
- Waive a blocker; change the decision instead
- Install from a registry search result without checking the namespace

KEEP WITH THE DECISION

Leave an evaluation record.

One record per server and version. It is the trail that lets the next person see what was checked, what was waived and when to look again.

Server / package / exact version	Registry namespace / publisher
Task it unblocks (A.01)	Clients it will run in
tools/list snapshot file / lock hash	Scanner result (errors / warnings)
Blockers and majors: pass, fail, n/a	Waivers, with who signed them
Decision: APPROVE / DEFER / REJECT	Re-review date and triggers

A DEFER is a decision too.

Write it down with the task that would change it. The next request for the same server starts from your record, not from zero.

SOURCES / MAINTENANCE

Keep the guide current.

Sources checked 24 September 2026. Scanner and scorer output was produced by running the kit on that date; the four real tool lists were captured with MCP Inspector 2.8.0 from the package versions on the cover.

S01 / MCP specification 2026-07-28, Tools

Annotations are untrusted unless from trusted servers; human in the loop; output schema; error handling; security considerations.

<https://modelcontextprotocol.io/specification/2026-07-28/server/tools>

S02 / MCP schema 2026-07-28, ToolAnnotations

Defaults: readOnlyHint false, destructiveHint true, idempotentHint false, openWorldHint true.

<https://github.com/modelcontextprotocol/modelcontextprotocol/blob/main/schema/2026-07-28/schema.ts>

S03 / MCP, Security Best Practices

Confused deputy, token passthrough, SSRF, local server compromise, scope minimization.

https://modelcontextprotocol.io/docs/2026-07-28/tutorials/security/security_best_practices

S04 / MCP specification 2026-07-28, Key changes

Stateless protocol, server/discover, removal of sessions and the initialize handshake.

<https://modelcontextprotocol.io/specification/2026-07-28/changelog>

S05 / The MCP Registry

Preview; namespace authentication; security scanning delegated to package registries and aggregators.

<https://modelcontextprotocol.io/registry/about>

S06 / MCP Inspector, CLI client

tools/list capture, --format json, exit codes 3 (auth) and 4 (unreachable).

<https://modelcontextprotocol.io/docs/2026-07-28/tools/inspector/cli>

S07 / Invariant Labs, MCP tool poisoning attacks

Instructions hidden in descriptions, rug pulls and cross-server shadowing (1 April 2025).

<https://invariantlabs.ai/blog/mcp-security-notification-tool-poisoning-attacks>

S08 / Simon Willison, The lethal trifecta

Private data, untrusted content and external communication in one agent (16 June 2025).

<https://simonwillison.net/2025/Jun/16/the-lethal-trifecta/>

S09 / Postmark, malicious postmark-mcp package

An impersonating package added a BCC in version 1.0.16 (September 2025).

<https://postmarkapp.com/blog/information-regarding-malicious-postmark-mcp-package>

S10 / Claude Code, Connect Claude Code to tools via MCP

Scopes, \${VAR} expansion in .mcp.json, output warning at 10,000 tokens and 25,000 cap, timeouts.

<https://code.claude.com/docs/en/mcp>

S11 / MCP specification 2026-07-28, Elicitation

Form mode must not request passwords, keys or tokens; URL mode for sensitive interactions.

<https://modelcontextprotocol.io/specification/2026-07-28/client/elicitacion>

S12 / MCP specification 2026-07-28, Authorization

OAuth 2.1, resource indicators, audience validation; stdio servers take credentials from the environment.

<https://modelcontextprotocol.io/specification/2026-07-28/basic/authorization>

S13 / MCP specification 2026-07-28, Streamable HTTP
Validate Origin against DNS rebinding; bind local servers to 127.0.0.1.
<https://modelcontextprotocol.io/specification/2026-07-28/basic/transport/streamable-http>

S14 / MCP specification 2026-07-28, stdio transport
stdout carries only MCP messages; logs go to stderr.
<https://modelcontextprotocol.io/specification/2026-07-28/basic/transport/stdio>

S15 / Official MCP Registry API, search for github-mcp-server
Three namespaces returned on 24 September 2026; io.github.github lists ghcr.io/github/github-mcp-server:1.12.2.
<https://registry.modelcontextprotocol.io/v0/servers?search=github-mcp-server&version=latest>

S16 / JFrog, CVE-2025-6514 in mcp-remote
Command injection through a malicious authorization endpoint; fixed in 0.1.16.
<https://jfrog.com/blog/2025-6514-critical-mcp-remote-rce-vulnerability/>

S17 / modelcontextprotocol/servers
Seven active reference servers, an archived list, and a warning that they are not production-ready solutions.
<https://github.com/modelcontextprotocol/servers>

S18 / GitHub MCP server
Toolsets, read-only mode, remote /readonly endpoints, lockdown mode.
<https://github.com/github/github-mcp-server>

S19 / MCP docs, Connect to local MCP servers
Claude Desktop config location and the warning that servers run with your account's permissions.
<https://modelcontextprotocol.io/docs/2026-07-28/develop/connect-local-servers>

Maintenance: re-run the four captures and the scanner after each MCP specification revision and each Inspector major release, and re-check the deprecated package list with `npm view` quarterly. Update the PDF, HTML, Markdown, JSON and the fixtures together.