

MCP checklist & vendor questionnaire

Before you connect an MCP server to your research repository, transcripts, or participant tracker, you’re building a live interface between a language model and your most sensitive qualitative data. Seven checks to run first—and ten questions to ask the vendor before you trust them. Grounded in the research, not the hype.

The seven MCP checks *before* you wire it up...

- 1

Scope — did you narrow it, or just open it?

☐ Connected one curated project, read-only, with the exposed tool count kept under ~10–15.

Why: Tool-selection accuracy slips past 10–15 tools—a fast model was right about 91% at 10 tools, down to 87% at 15 (Rodrigues & Vas, 2026). Every extra tool is also extra attack surface (see check 2).
- 2

Read/write gating — are destructive actions locked behind a human?

☐ Sorted every tool into read-only vs write/modify/send, and put a human confirmation in front of anything irreversible.

Why: MCP measurably widens the attack surface—compromise rose from 26.4% to 52.8% versus non-MCP setups, and worse across chained servers (Maloyan & Namiot, 2026). Write access you didn’t scope is write access an attacker inherits.
- 3

Traceability — can every insight point back to the transcript?

☐ Confirmed the connection returns immutable source identifiers, so each synthesized insight cites the exact, timestamped source—and you can spot-check it fast.

Why: Stakeholders lose trust in AI-assisted findings the moment provenance is obscured. A cited answer is checkable; a confident summary isn’t.
- 4

Tool clarity — does the description match the behavior?

☐ Read the server’s actual tool list and confirmed it does only what it says—no undocumented write, delete, send, or network tools.

Why: Studies of real-world MCP servers find a meaningful share whose behavior doesn’t match their descriptions (Shi et al., 2026). The description is a claim, not a guarantee.
- 5

Consent & PII — is the data allowed to flow at all?

☐ Confirmed PII handling and a valid lawful basis for sending this data to a model provider, with redaction and permissioning enforced in the MCP layer—paying special attention to older records.

Why: Connecting sends your content to the model. MCP carries no participant-consent concept; under GDPR you need a lawful basis, and consent for one purpose doesn’t stretch to a new one. (Not legal advice—loop in privacy/legal.)
- 6

Access logging — can you scope a breach?

☐ Confirmed every MCP tool call is logged with user, timestamp, and parameters—and that you can reach those logs yourself during an incident, not file a support ticket.

Why: Traceability tells you whether to trust a finding; access logging tells you what happened when something goes wrong—a different axis. MCP widens the attack surface it rides on (Maloyan & Namiot, 2026); the wider that surface, the more you’ll need to reconstruct who reached what, when.
- 7

Injection defense — can hostile content hijack the connection?

☐ Confirmed the connection can’t be steered by the content it reads—a hidden instruction buried in a transcript, file, or field can’t make the AI act on its own.

Why: A connector runs on whatever flows through it. An instruction hidden inside your own data—the phishing attachment now living in your repository—can act through the connection without you asking, and MCP measurably widens that attack surface (Maloyan & Namiot, 2026).

The audit matrix

Check	Verify	Grounded in	Pass / Fail
Scope	One project, read-only, ≤10–15 tools?	Tool-selection accuracy (Rodrigues & Vas)	☐
Read/write gating	Destructive tools behind a human gate?	MCP attack amplification (Maloyan & Namiot)	☐
Traceability	Immutable source IDs on every insight?	Provenance and stakeholder trust	☐
Tool clarity	Description matches actual behavior?	Description-code inconsistency research	☐
Consent & PII	Lawful basis confirmed, PII handled?	GDPR purpose limitation + vendor data-flow disclosure	☐
Access logging	Tool calls logged with user, timestamp, params —reachable mid-incident?	Incident-response practice (Maloyan & Namiot)	☐
Injection defense	Hidden instructions in your data can’t trigger actions?	MCP attack surface (Maloyan & Namiot)	☐

Ten questions to ask your vendor *before* you trust them...

- 1

What exactly leaves our environment when we query, and where does it physically go?

Data residency and boundary — get it in writing (in the DPA).
- 2

How does authentication work — per-user, or a shared key?

Per-user auth, not a shared key.
- 3

Is your MCP server read-only, or can the model write, delete, or message on our behalf — and can we scope that ourselves?

Read/write governance.
- 4

Do you pass immutable, timestamped source identifiers so every synthesized insight traces back to the exact transcript?

Traceability.
- 5

Is there a security audit trail — who accessed what, and when?

Reachable by you, not a support ticket.
- 6

Will you disclose the full tool list, and does each tool’s description match what its code actually does?

Description-behavior integrity.
- 7

What are the retention and training policies for anything sent through the connector — both yours and the model provider’s?

“Not used for training” and “not transmitted” are different promises.
- 8

What certifications and compliance commitments can you show us?

SOC 2, GDPR/HIPAA — verifiable in writing.
- 9

Have you red-teamed your MCP layer — and can you share the report?

A pentest of the MCP surface, not just SOC 2.
- 10

Bonus, if participant data is involved: What mechanism, if any, do you provide to enforce participant-consent scope—or is that entirely on us?

Usually the answer is “on you.” Better to hear it now.

Don’t just check it—verify it, first-hand

A vendor’s docs are a claim, not proof. Real MCP servers ship tools that don’t do what their descriptions say (Shi et al., 2026). So the checklist has a partner: a verifier that connects to the server and tests it against the seven checks—reading the real tool list, sorting read from write, flagging hidden or undocumented tools.

What it caught in testing: Against a sample server, it caught a “save note” tool that quietly overwrites—its description never says so—and an “export” tool the manifest never listed: a hidden data-egress path. Synthetic data only; no real participant records touch a server under test.

A probe proves a failure, never safety. The contractual questions—retention, DPAs, jurisdiction—still get answered on paper.

Stay Curious. Stay Critical. Stay Cool.

Run it before you connect, and forward it to whoever’s about to “just wire it up.”
Reply to the newsletter with the question your vendor couldn’t answer.

Book a governance review heykaleb.com/aixuxr-transformation-chat

