



GUIDE 02 OF 03 · APPLY

Putting MCP to Work

Practical Use Cases and a 30-Day Pilot Plan

Real workflows across fundraising, grants, programs, communications, and operations — plus a step-by-step 30-day pilot your team can run without a developer.

A deep research guide from MCP.NGO · Reading time: about 12 minutes · Last reviewed: September 2026

IN THIS SERIES

Guide 01 — Understanding MCP

Guide 02 — Putting MCP to Work (this guide)

Guide 03 — Connecting with Care

Who this guide is for

You've read the plain-language explanation (Guide 01: *Understanding MCP*) or you already know the gist: the Model Context Protocol lets your AI assistant securely connect to the software your organization runs on. This guide answers the next two questions — *what would we actually use it for?* and *how do we try it without betting the farm?*

You need three things to act on this guide: an AI assistant subscription that supports connectors (all the major platforms do as of 2026), an inventory of the tools your team uses, and one hour a week for a month. You do not need a developer.

One caution before the fun part: everything below assumes you follow the safety basics in Guide 03 — vetted connectors, minimum permissions, read-only by default, and a human reviewing anything that goes out the door. Sector data shows why this matters: nearly half of nonprofits report having no AI policy at all, and organizations that skip governance tend to stall at shallow gains. Twenty minutes of setup discipline protects months of progress.

Where connected AI earns its keep

The pattern across every use case below is the same: work that used to require a staff member to *gather* information from several systems before they could *think* about it. Connection removes the gathering. Here's what that looks like across a typical nonprofit.

Fundraising and donor relations

This is where most organizations see value first, because donor work is drowning in exactly the kind of lookup-and-draft tasks connected AI handles well.

With a read-only connection to your CRM or donor database, a development officer can ask for donors who gave last year but not yet this year, get briefed before a donor meeting on someone's full giving and contact history, or request a list of lapsed monthly givers above a certain amount — all in plain English, all from live data. Pair the CRM with your email or document tools and the assistant can draft personalized thank-yous and check-ins that reference each donor's actual gifts, ready for human review and sending. Year-end appeal season, which usually means a week of spreadsheet wrangling, becomes an afternoon of reviewing drafts.

A note on judgment: personalization built on real data is powerful, which is exactly why a person should read every donor-facing message before it goes out. Speed is the assistant's job; the relationship is still yours.

Grants and institutional giving

Grant work is document-heavy, deadline-driven, and scattered across drives, spreadsheets, and inboxes — ideal terrain. Connect your cloud drive and a grants tracker, and the assistant can pull last year's program outcomes and budget figures into a first-draft narrative shaped to a funder's questions, check a draft against the RFP's requirements, summarize a funder's reporting obligations from the executed agreement, or give the grants manager a Monday-morning digest of every deadline in the next 60 days. Reporting gets the same lift in reverse: interim and final reports assembled from the actual program data your team already tracks.

Programs, data, and impact reporting

Most program teams keep their truth in spreadsheets and lightweight databases. Once connected, "How many people completed the workshop series this quarter, and how does that compare to last year?" becomes a question anyone can ask — not just the one staffer who knows where the pivot table lives. Assistants can draft impact summaries for newsletters and board reports from live figures, flag data-entry gaps ("14 intake records are missing zip codes"), and turn a messy end-of-program debrief document into a structured lessons-learned summary. For organizations pursuing outcomes-based funding, having reportable numbers a question away is quietly transformative.

Communications and marketing

Connect the shared drive, and newsletter drafts can be grounded in what actually happened this month — the real event recap, the real client story (with consent and privacy rules applied), the real numbers. Connect design and scheduling tools, and a comms manager can go from "draft a four-week social calendar for the fall campaign" to reviewable posts in one sitting. Website and blog content can be checked against your existing materials for consistent language about programs. The gain isn't that the AI writes; it's that the AI writes *from your source material* without anyone assembling a briefing packet first.

Operations, leadership, and internal knowledge

The unglamorous wins are often the biggest. Meeting transcripts become assigned tasks on the project board. The employee handbook, once connected, answers its own questions ("What's our mileage reimbursement rate?") instead of interrupting the operations manager. Board-packet assembly — finance summary, program dashboard, prior minutes — becomes a first draft the ED edits rather than a day the ED loses. Finance staff can ask natural-language questions of the budget spreadsheet without building a new report for every board member's curiosity.

How connecting actually works

Inside your AI assistant's settings you'll find a connectors or integrations directory. For mainstream tools — the major office suites, cloud drives, project trackers, communication platforms, and a growing list of CRMs — connecting is a click-and-sign-in flow: you choose the connector, authenticate through the familiar "grant access" screen (OAuth, the same mechanism you've used to link apps for years), and decide what it's allowed to do. These are **remote connectors**, hosted by the software vendor, and they're the right starting point for almost every nonprofit.

If a tool you rely on isn't in your assistant's built-in directory, check whether the vendor offers an official MCP server — many now do, and the MCP project maintains a public registry where servers can be found. Prefer official, first-party connectors over community-built ones for anything touching sensitive data; Guide 03 gives you a ten-question vetting list.

Local servers — connectors that run on a staff computer — exist too, and technical teams use them for custom or on-premise systems. Most nonprofits can ignore them entirely for now.

The 30-day pilot

Resist the urge to connect everything on day one. A narrow pilot produces a real decision; a broad one produces a mess. Here's a month-long plan sized for a small team.

Week 1 — Choose and prepare. Pick *one* workflow that is frequent, annoying, and low-risk — donor-meeting prep briefings and grant-deadline digests are classic first choices. Name a pilot owner. Write down the current process and how long it takes (you'll want the before-and-after). Adopt the short list of ground rules from Guide 03: what data may be connected, read-only to start, human review on all outputs, and what must never go into an AI tool. If leadership hasn't blessed a one-page AI policy yet, this is the week to do it.

Week 2 — Connect and test. Set up the one connector your workflow needs, with the minimum permissions it needs — read-only unless the workflow truly requires writing. Have the pilot owner run the real task five or ten times. Keep notes: What did it get right? What did it get wrong? How long did review take?

Week 3 — Measure and stress-test. Put the workflow in front of a second staff member and see whether the gains survive contact with someone who didn't design it. Track hard numbers (minutes per task, tasks per week) and soft ones (does the output need light edits or heavy rescue?). Deliberately test failure: ask questions the data can't answer and confirm the assistant admits it rather than inventing figures.

Week 4 — Decide and document. Three honest outcomes are possible. *Scale it*: the time savings are real — document the workflow in a shared how-to, and pick the next one. *Fix and retry*: the idea works but the data is too messy or the permissions too tight — clean up and rerun. *Stop*: the juice isn't worth the squeeze for this workflow — disconnect the connector (don't leave unused access lying around) and spend your energy elsewhere. All three are wins; the pilot's job was to buy you a cheap, fast answer.

A benchmark of the sector's AI use found that the small minority of organizations seeing genuine impact are distinguished not by using AI more, but by exactly this kind of deliberate practice: chosen workflows, measurement, shared documentation, and governance. The pilot is how you join them.

Build or buy?

For at least the first year, nearly every nonprofit should use pre-built connectors and nothing else. Custom MCP servers make sense only when a mission-critical system has no official connector *and* the workflow

value is high — a homegrown case-management database, for instance. If you reach that point, treat it as a small software project: budget for a contractor or a skilled technical volunteer, insist on the authentication and security practices in the official specification, and plan for maintenance, because a connector is a living piece of infrastructure, not a one-time purchase.

What it costs

The protocol is free, and most official connectors are included with the products they connect. Real costs come in three familiar shapes: **subscriptions** (the paid tier of an AI assistant for the staff who'll use it — connector features generally live on paid plans), **time** (a few hours for the pilot owner in month one, plus an hour of team training), and **attention** (someone must own the quarterly review of what's connected and who has access). For most small organizations, the all-in cost of finding out whether this matters is under a few hundred dollars and one person's spare hours for a month.

Five pitfalls to step around

1. **Connecting everything immediately.** Every connection is access to manage. Add them one proven workflow at a time.
2. **Skipping the permissions screen.** The moment you click "Allow" is the moment that matters most. Read what you're granting; choose read-only when offered.
3. **Trusting outputs unreviewed.** Connected AI is dramatically more accurate about your organization than unconnected AI — and still not infallible. Review anything that reaches a donor, funder, or community member.
4. **Piloting on your most sensitive data.** Start with grant deadlines, not case files. Sensitive-data workflows come later, if ever, and only after Guide 03.
5. **Letting it be one person's magic trick.** If the workflow isn't documented and shared, it leaves when its inventor does. Write the how-to.

Where to go next

Before you run Week 2 of the pilot, read Guide 03, *Connecting with Care* — it's the seatbelt for everything in this guide. If you skipped the fundamentals, Guide 01, *Understanding MCP*, will fill in the vocabulary in ten minutes.

Sources & further reading

- Model Context Protocol — official documentation, specification, and ecosystem resources: <https://modelcontextprotocol.io>
- Model Context Protocol blog — "The 2026 MCP Roadmap" (where the standard is headed): <https://blog.modelcontextprotocol.io/posts/2026-mcp-roadmap/>

- Virtuous & Fundraising.AI — "The 2026 Nonprofit AI Adoption Report" (adoption, governance gap, and the "efficiency plateau"): <https://virtuous.org/blog/the-2026-nonprofit-ai-adoption-report-blog/>
- NonProfit PRO — "Nonprofit AI Adoption Hits 92%, But Only 7% See Major Impact": <https://www.nonprofitpro.com/article/nonprofit-ai-adoption-hits-92-but-only-7-see-major-impact/>
- WorkOS — "Everything your team needs to know about MCP in 2026": <https://workos.com/blog/everything-your-team-needs-to-know-about-mcp-in-2026>

This guide is for general information only and is not technical, legal, or security advice. Product features and plans change quickly; verify details with vendors before purchasing or connecting.