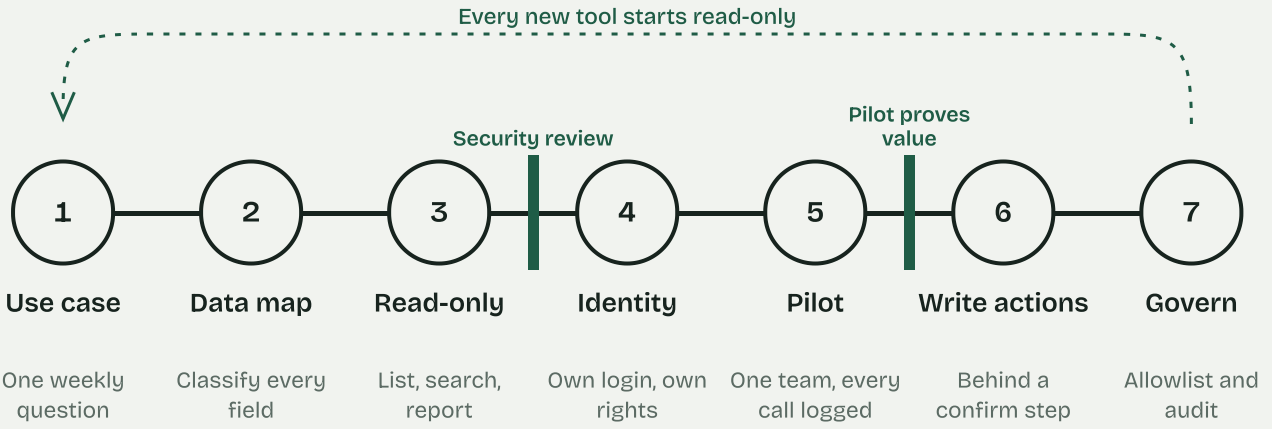


Dominik Heinrich, multi-cloud DevOps & AI solution architect

Connect your company to AI with MCP

How I connect tools and data to Claude, ChatGPT and coding agents through the Model Context Protocol, safely. Seven steps from the first read-only server to company-wide use.



Seven steps, two gates. The AI only ever acts with the rights of the person using it.

From two production platforms I built, anonymized

100% of the app's UI also usable from AI agents via MCP	**0** personal data in a client's read-only API and MCP server	**Weekly** planning a client automates through MCP	**2** production platforms with their own MCP server

The seven steps

From one read-only server to company-wide use. Tick them off as they go live.

01 Start with one question people ask every week

- [ ] Pick a use case with a clear owner: weekly planning, a sales report, ticket triage, cost questions.
- [ ] Write down which data the answer needs and who is allowed to see it.
- [ ] Measure how long it takes today, so you can show the difference later.

02 Know your data before you expose it

- [ ] **Classify every field:** public, internal, personal, secret.
- [ ] Personal and secret data stay out of the first version. Aggregates are usually enough.
- [ ] Data that must never leave your network: serve it to a local model behind the same MCP server.

03 Read-only first

- [ ] **The first server only reads:** list, search, report. No create, update or delete.
- [ ] One tool, one job, with a clear name and description the model can understand.
- [ ] Small, structured answers instead of whole tables: faster, cheaper and less to leak.

04 Identity and keys

- [ ] Each user connects with their own identity (OAuth or OIDC), so the agent acts with that user's rights, never more.
- [ ] API keys stored hashed, scoped per tool, with an expiry and a way to revoke them.
- [ ] No shared admin token, ever.

05 Pilot with one team

- [ ] Five to ten people, two weeks, one channel for feedback.
- [ ] **Log every tool call:** who, which tool, which parameters, how long, what came back.
- [ ] Rate limits and a cost cap per user from day one.

06 Add write actions behind a confirm step

- [ ] Writes only where the pilot proved value, each one confirmed by the user before it runs.
- [ ] Destructive actions (deleting, payments, permissions) come last, with an explicit confirmation and the same permissions as in the user interface.
- [ ] Every write is reversible, or at least recorded with the state before and after.

07 Scale and govern

- [ ] An approved list of MCP servers and AI clients for the company; everything else blocked by policy.
- [ ] **New tools reviewed like code:** pull request, security check, a named owner.
- [ ] A monthly report on usage, errors and hours saved.

Rolling it out, by company size

Startup: one server for your own product, so customers and your team can work in it from their AI assistant.

Mid-sized company: one server per core system (ERP, CRM, ticketing) behind single sign-on.

Enterprise: a central gateway with an allowlist, audit logs and data classification, plus local models for confidential data.

Mistakes I see most often

- Starting with write access because the demo looks better.
- One shared admin key for everyone, so the audit trail shows nobody.
- Returning whole database tables, then wondering about cost and leaks.
- Connecting personal data before anyone asked the data protection officer.
- Unreviewed community MCP servers on laptops with company data.

Ways to work together

The same quality for a ten-person startup and a global group. The format adapts to you.

Startups

Fractional DevOps & AI lead

A few days a month: architecture, the release pipeline and AI features, so the founders can focus on customers. From prototype to production without a rewrite.

Mid-sized companies

A project with a clear outcome

A defined goal like a lower cloud bill, a migration or the first AI agents in production, with a fixed scope, weekly progress and a handover your team can run.

Enterprises

An architect inside your teams

Embedded with your platform, security and product teams: architecture decisions, governance for AI and regulated environments. For larger scopes I bring a trusted senior team.

How I work

Decisions written down: architecture decision records with the trade-offs, so your team knows why, not just what.

Secure by default: OIDC and managed identities, least privilege, secrets in a vault, an audit trail.

Operated, not just delivered: SLOs, alerts on business outcomes and an incident routine.

Measured: cost, lead time, uptime and hours saved, reported every week.

Knowledge stays with you: runbooks, docs and workshops for your team.

One point of contact who talks to engineering and to management, in German, English and Spanish.



Want your systems to talk to AI, safely?

Book a 30-minute call

E-mail mail@dominikheinrich.ch
Web dominikheinrich.ch

Available worldwide, remote. On-site on request, depending on the country.