WORKSHOP / 1 HOUR

From Explorer to Architect

Model Context Protocol: the standard that lets AI assistants take action in your apps

Olga Safonova

Women in Product Nordics
TechWomen Copenhagen

THE PROBLEM

AI that knows everything and does nothing is an expensive search bar.

Current AI can answer questions and generate text.
But it can't create a Miro board or update your roadmap.

THE DISTINCTION

RAG vs MCP

RAG

Information flows IN

Retrieval-Augmented Generation.
AI pulls data from documents,
databases, knowledge bases.

Result: Better answers.

MCP

Commands flow OUT

Model Context Protocol.
AI sends actions to apps,
APIs, local software.

Result: Things get done.

WHAT MCP ACTUALLY IS

A protocol.

Like USB-C for AI.

One standard that lets any AI assistant connect to any app.

ARCHITECTURE

How it works

01
Client

Claude Desktop
ChatGPT
Cursor

→

02
Server

Miro MCP Server
Playwright MCP Server
GLEIF MCP Server

→

03
App

Miro
Your browser
GLEIF API

ARCHITECTURE

What servers provide

01
Tools

Actions

Things the AI can do

→ Search for a company
→ Create a board
→ Take a screenshot

02
Resources

Data

Things the AI can read

→ Files on your computer
→ Database records
→ Live API data

03
Prompts

Templates

Reusable instructions

→ Code review checklist
→ Summary format
→ Report template

ARCHITECTURE

Apps can be anywhere.

Cloud Apps

- Miro
- ProductPlan
- GLEIF API
- Jira, Slack, Notion...

Local Apps

- Your browser
- Files on your machine
- Local git repos
- Installed apps

MCP works with both. Data stays where you want it.

TIER 2 / USER

How to Use MCPs

Can follow instructions, no coding required.
Three live demos.

Get Claude Desktop

claude.ai/download

MAC

→ Download the .dmg file

→ Drag Claude to Applications

→ Open and sign in

WINDOWS

→ Download and run installer

→ Follow the wizard

→ Open and sign in

Try it: Add GLEIF

github.com/olgasafonova/gleif-mcp-server/releases

STEPS

→ Download the binary for your system

→ Open config folder

→ Add the config (see GitHub README)

→ Restart Claude Desktop

CONFIG FOLDER

→ Mac: Cmd+Shift+G, paste path

→ Windows: Win+R, paste path

DEMO 1

GLEIF

Cloud App Pattern

LIVE PROMPTS

```
"Use the GLEIF server to look up Apple Inc's LEI"
```

```
"Who is Apple's ultimate parent company according to GLEIF?"
```

UNDER THE HOOD

→ Data lives in the cloud (GLEIF API)

→ Lookup, validate, trace ownership

→ Structured output, not web scraping

→ Real financial identifiers (LEI)

DEMO 2

Playwright

Local App Pattern

LIVE PROMPTS

```
"Open a browser and go to producthunt.com"
```

```
"Take a screenshot of the page"
```

THE DIFFERENCE

→ Your browser, your machine

→ Navigate, fill forms, capture pages

→ Privacy: data stays on your machine

→ No API keys needed for local apps

DEMO 3

Configuration

What you actually do

```
// claude_desktop_config.json

{
  "mcpServers": {
    "gleif": {
      "command": "gleif-mcp-server"
    },
    "playwright": {
      "command": "npx",
      "args": ["@anthropic/mcp-playwright"]
    }
  }
}
```

WHAT IT TAKES

→ Edit one file

→ Give it a name, tell it how to start

→ Restart to activate

→ No coding to use an MCP

TIER 3 / ARCHITECT

How to Build MCPs

For developers who want to create their own.
Real patterns from production servers.

STRUCTURE

Architecture Overview

```
server/
├── main.go # Entry point
├── tools/
│   ├── definitions.go # What can it do?
│   └── handlers.go # How does it do it?
└── client/
    └── client.go # API client
```

main.go

Entry point. Registers tools, starts the server.

definitions.go

Tool names, descriptions, parameter schemas. What Claude sees.

handlers.go

The actual implementation. Calls APIs, processes data, returns results.

client.go

HTTP client with auth, rate limiting, caching. Talks to the real API.

LESSONS LEARNED

Production Patterns

Error Handling
Wrap API errors with context. Catch failures gracefully. Return clear error messages.

Caching
Remember recent results. Fewer API calls, faster responses.

Bulk Operations
Group similar requests. One batch beats many singles.

Authentication
Store API keys safely, not in code. Check they exist at start.

Rate Limiting
Don't hit APIs too fast. Slow down when needed.

Retries
Retry temporary failures (timeouts, server errors). Skip bad requests. Wait longer between attempts.

YOUR FIRST MCP SERVER

Getting Started

RECOMMENDED

Fork GLEIF MCP

Simplest, 12 tools

Clean, minimal codebase. Read-only API. Perfect for learning the MCP SDK patterns.

Fork Miro MCP

Most advanced, 77 tools

Full CRUD operations, OAuth flow, bulk actions. Reference for complex servers.

Start fresh

Go, Python, or TypeScript SDK

Official SDKs handle protocol. You focus on tools. Docs at modelcontextprotocol.io

Python and **TypeScript** are the most popular—most tutorials and the AI/ML community use them.

Why Go?

Single binary, no runtime dependencies, starts instantly, cross-compiles for any OS. Python needs a runtime, TypeScript needs Node. **Go** just works.

You decide what works for you

WRAP-UP

Distilled

EXPLORERS **MCP lets AI take action, not just answer questions**

USERS **Setup is simple: download, configure, restart**

ARCHITECTS **Start small. Build for real value**

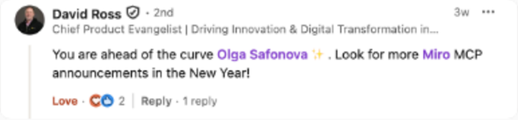
REAL VALUE

"Impressive work on Miro MCP, thank you!"

Andrey Khusid CEO & Co-founder, Miro **Miro MCP**

"You are ahead of the curve. Look for more Miro MCP announcements!"

David Ross Chief Product Evangelist, Miro **Miro MCP**



tieto

"SO useful in making everyday work more efficient"

Jenny Felldin Head of Product **ProductPlan MCP**

"A golden nugget to help sales with bid requirements"

Jenny Felldin Head of Product **MediaWiki MCP**

"A golden opportunity to strengthen our product positioning"

Sandra Wallin Head of Enablement **MediaWiki MCP**

"This type of reports is really useful"

Sandra Wallin Head of Enablement **MediaWiki MCP**

"Nyttig verktøy" — Glossary is very useful

Per Borgersen Delivery Manager **MediaWiki MCP**

"This might be one of the best gifts ever"

Ari Räisänen Lead Software Architect **MediaWiki MCP**

"It works beautifully 💪"

Ari Räisänen Lead Software Architect **MediaWiki MCP**

"Design Team is super excited"

Aleksandra Bratek UI Designer **MediaWiki + Miro MCP**

"Love it!"

Morten Lerfald Senior Business Developer **MediaWiki MCP**

RESOURCES

Learn More

MCP Specification modelcontextprotocol.io

Go SDK github.com/modelcontextprotocol/go-sdk

TypeScript SDK github.com/modelcontextprotocol/typescript-sdk

Python SDK github.com/modelcontextprotocol/python-sdk

MCP servers I built github.com/olgasafonova

Thank you!

Olga Safonova

AI Product Leader ⁞ Building AI tools in Go
Ex-Workday, Siteimprove, Saxo Bank ⁞ Keynote Speaker

Connect

LINKEDIN

GITHUB

SUBSTACK