

From Agents to Systems: Building MCP Servers for Practical AI Agents



Dr. Matthias Liebeck



LinkedIn



GitHub Copilot Newsletter



Made Possible by



Table of Contents

1. Introduction and Motivation
2. Core Concepts MCP
3. Integration and Hosting
4. Board Game MCP Server for Gesellschaftsspieler-gesucht
5. Integration in ChatGPT
6. Further Concepts
7. Conclusion




Introduction and Motivation

The Original LLM Model

Originally, interaction with LLMs was purely text-based

Prompt  LLM  Response

- A prompt is sent to the model.
- The model generates a response.
- **The model only knew what was in the prompt or in its training data.**
 In the past, the training cutoff date of the model was very important.

In the Past: Limits of original LLMs

No access to:

- databases
- internal APIs
- filesystems

No ability to:

- execute actions
- verify results

Answers were:

- based on a fixed knowledge state
- not verifiable
- potentially hallucinated

LLMs could "think", but not "act"

Today: Agentic AI Systems

- LLMs are now part of larger systems and can
 - plan
 - make decisions
 - select tools
 - combine results
- Typical examples:
 - agentic coding (e.g., GitHub Copilot)
 - research agents
 - automated workflows (e.g., n8n)

The New Problem: Tool Integration

- LLMs need access to external resources, but
 - every tool has a different API
 - every platform integrates differently
 - every solution is proprietary
- Result:
 - high integration effort
 - vendor lock-in
 - poor reusability
- **The model is not the problem, but the lack of a standard for tools.**

The Solution: MCP (Model Context Protocol)

Key questions: How do we give LLMs standardized access to tools?

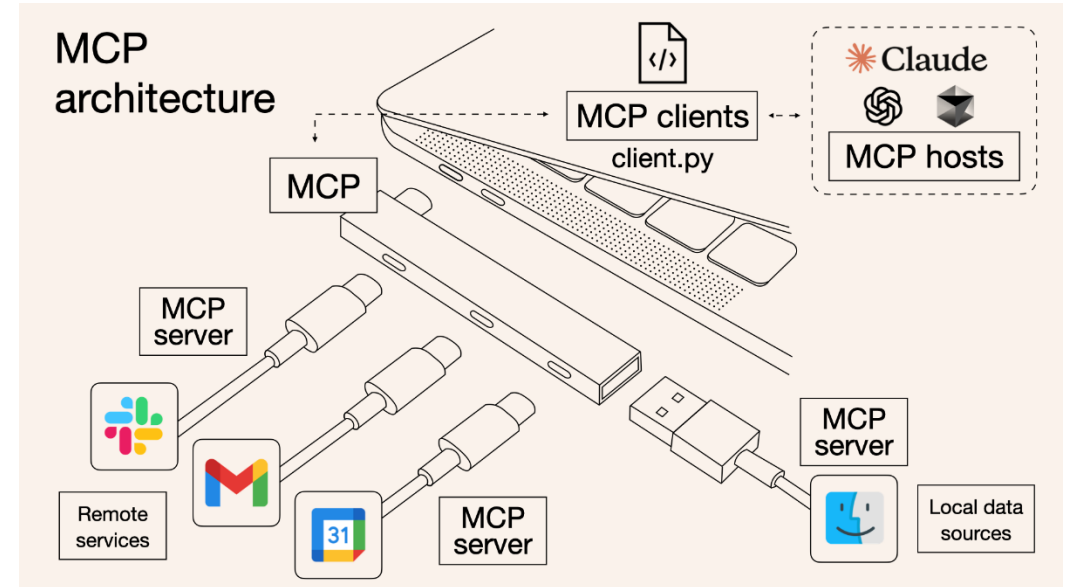
- How do we describe what tools exist?
- What they do?
- How they are called?
- How do we uncouple models, tools, and platforms?

 **MCP (Model Context Protocol) is the solution**

Model Context Protocol (MCP)

Model Context Protocol (MCP)

- MCP is an **open protocol**
 - It standardizes
 - how tools are described
 - how they are called
 - how results are returned
 - MCP is
 - model-independent
 - platform-independent
 - vendor neutral
- 💡 „MCP is like USB for LLMs“



Check out mcp-servers.org

Awesome MCP Servers

A collection of servers for the Model Context Protocol.

Search MCP servers...



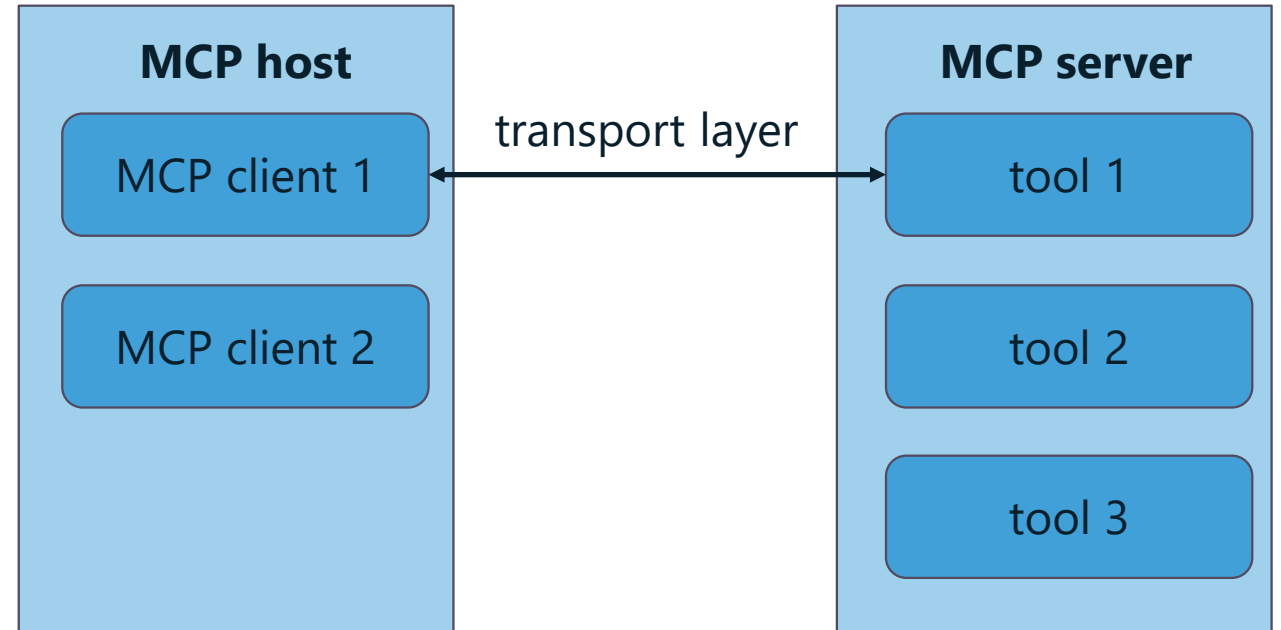


Core Concepts MCP

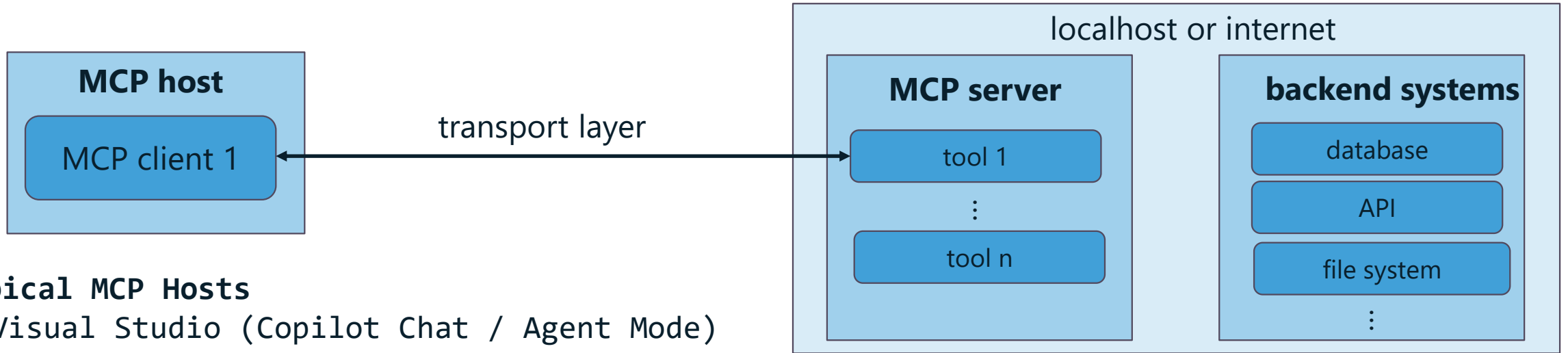
MCP Architecture

MCP Architecture

- MCP clearly separates:
 - host (LLM + Agent + MCP Client)
 - server (tools & resources)
- The MCP server is:
 - no plugin
 - not an SDK
 - a standalone service / process



MCP Architecture: MCP Host



Typical MCP Hosts

- Visual Studio (Copilot Chat / Agent Mode)
- ChatGPT
- VS Code
- custom chat or agent apps

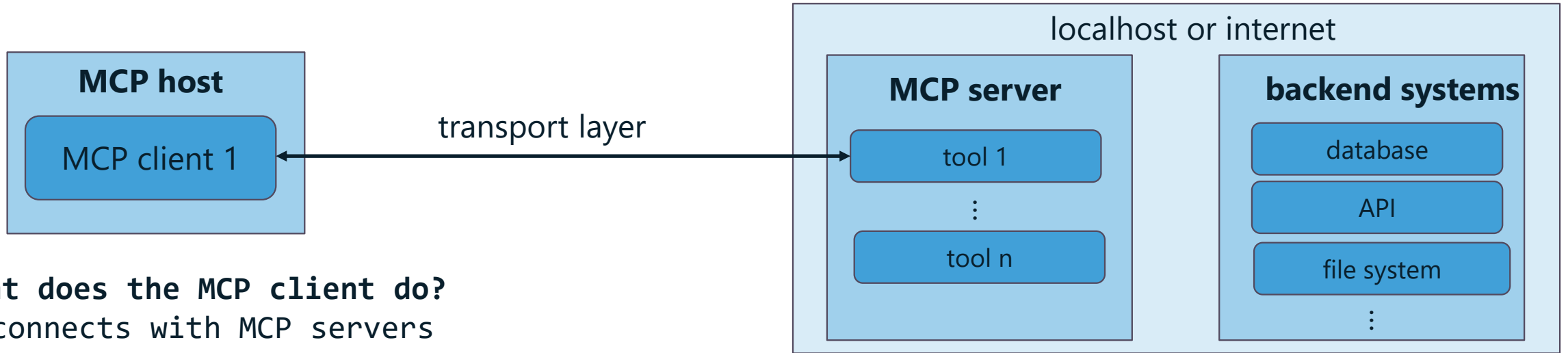
Responsibilities:

- contains the LLM
- processes user prompts
- plans steps / „What do I need?“
- decides:
 - **if** a tool is needed
 - **which** tool should be used
- integrates results back into reasoning

What is the MCP host?

The **MCP host** is an application that runs the LLM and **acts agentic**.

MCP Architecture: MCP Client



What does the MCP client do?

- connects with MCP servers
- calls `/tools/list`, `/tools/call`, etc.
- serializes/deserializes MCP requests
- takes care of:
 - transport (stdio / HTTP)
 - auth headers
 - reconnects

Where does the MCP client live?

- in the MCP host
- invisible to most users

Example:

In Visual Studio: GitHub Copilot contains a **MCP Client**

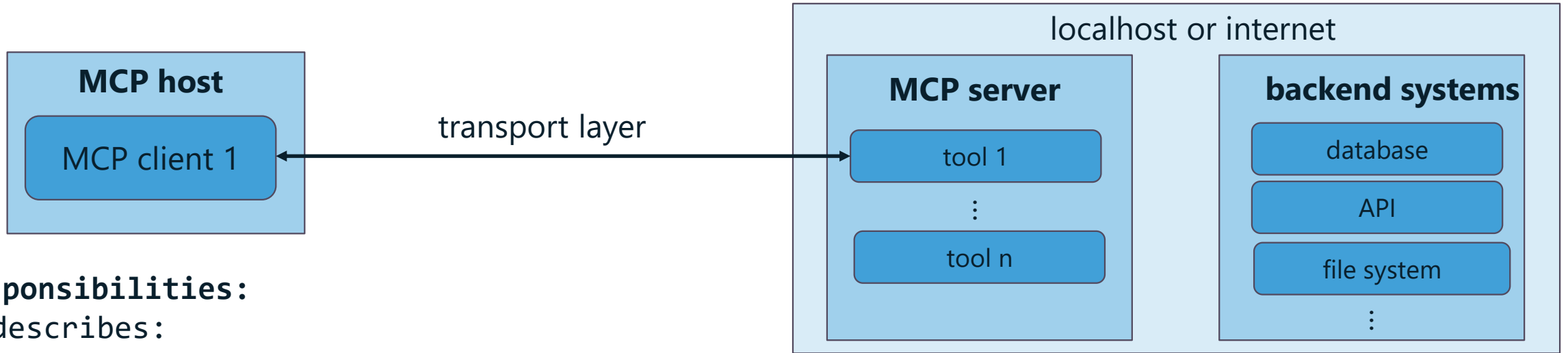
What is the MCP client?

The MCP client is the **technical implementation of the MCP protocol** in the host.

🚨 Important:

The MCP client **does not make decisions**. It only executes what the host decides.

MCP Architecture: MCP Server



Responsibilities:

- describes:
 - which tools exist
 - which parameters they have
- executes actions like:
 - API calls
 - database queries
 - access to the file system
- returns structured results

What is the MCP server?

It's an **own process or service**, which offers **tools**.

🚨 Important:

The MCP server **does not know why** a tool was called, just **how** it was called.

The MCP server is a classic backend architecture, but for agents.

Tool Descriptions – The Most Important Part

Tool descriptions are the API design for LLMs.

They determine:

- whether a tool is used
- when a tool is used

bad description 👉 tool is ignored

Best practice:

- action-oriented
- explicit (“Use this tool when...”)

The description is not a comment. It is control logic.

Transport Options

Transport: Standard Input/Output vs. HTTP/HTTPS

stdio

- subprocess of host
- communication via stdin/stdout
- ideal for:
 - local development
 - single user

HTTP / HTTPS

- standalone web service
- ideal for
 - cloud hosting
 - team usage
 - ChatGPT
- supports:
 - TLS
 - auth

💡 stdio is for development. HTTP is for production.

Authentication in MCP

- MCP does not define its own auth, but can use it
- Auth happens on transport level (HTTP)
- Common approaches:
 - Bearer tokens / API keys
 - **OAuth 2.1** (user based)
 - network-based security
- 💡 **Important: The MCP host must support the chosen auth method.**

Summary: MCP Architecture

Summary

- MCP separates:
 - decision logic (LLM)
 - execution (Server)
- Tools are:
 - clearly defined
 - reusable
 - securely operable

 MCP is not a new kind of AI. It is a new way to provide backend functionality to agents.



Integration and Hosting

Hosting Options for MCP Servers

How can I host an MCP server?

An MCP server is:

- a normal process
- or a normal web service

Hosting is independent of the MCP protocol.

Typical options (examples):

- Cloud App Service / Web App hosting
- API Gateway / API Management
- Serverless Functions
- Docker / Containers

The hosting model is **intentionally not prescribed by MCP.**

Hosting with App Service / Web App

Hosting with App Service (Web App Hosting)

Very well suited for MCP servers

Advantages:

- permanent HTTPS URL
- easy deployments
- built-in logging & monitoring

Ideal for:

- streamable HTTP
- production usage

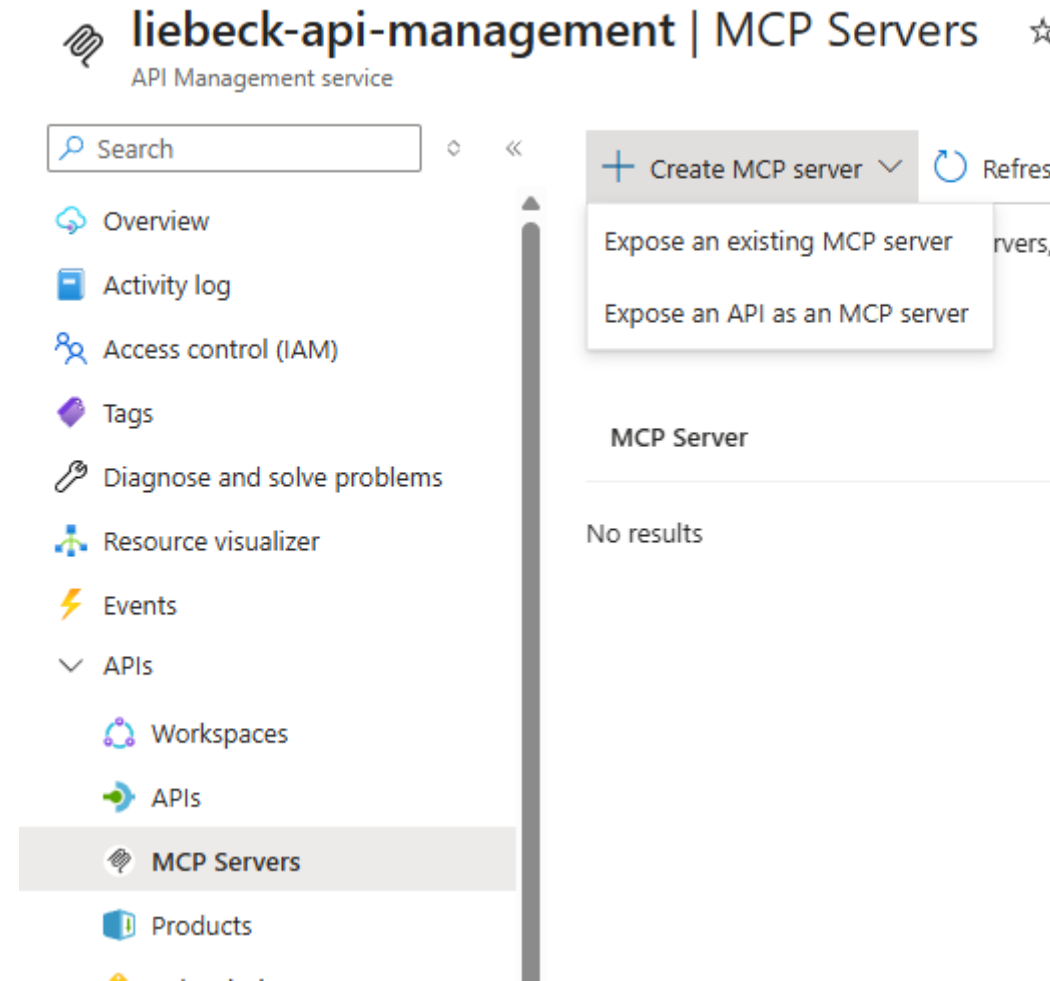
Hosting with API Management / API Gateway

Hosting with API Management / API Gateway

- **REST API as MCP server:** Expose a REST API hosted in API Management as an MCP server
- **Existing MCP server:** Put an existing MCP server behind API Management

Disadvantage:

- Additional infrastructure cost (developer 40€, basic v2 127€ / month)



Hosting with Serverless Functions

Hosting with Serverless Functions (e.g., Azure Functions)

Advantages:

- serverless
- low cost for low traffic

Challenges:

- cold starts
- always-on is not ideal for agents  costs will be higher

Hosting with Docker / Containers

Hosting with Docker / Containers

MCP servers run perfectly fine in containers

Suitable for:

- Container Apps / managed container platforms
- Kubernetes
- local development

Advantages:

- maximum control
- reproducible environments



Board Game MCP Server for Gesellschaftsspieler-gesucht

Board Game MCP Server Gesellschaftsspieler-gesucht.de

Gesellschaftsspieler-gesucht: German board game community platform

- 19.000 users
- 3.000 games
- hobby project



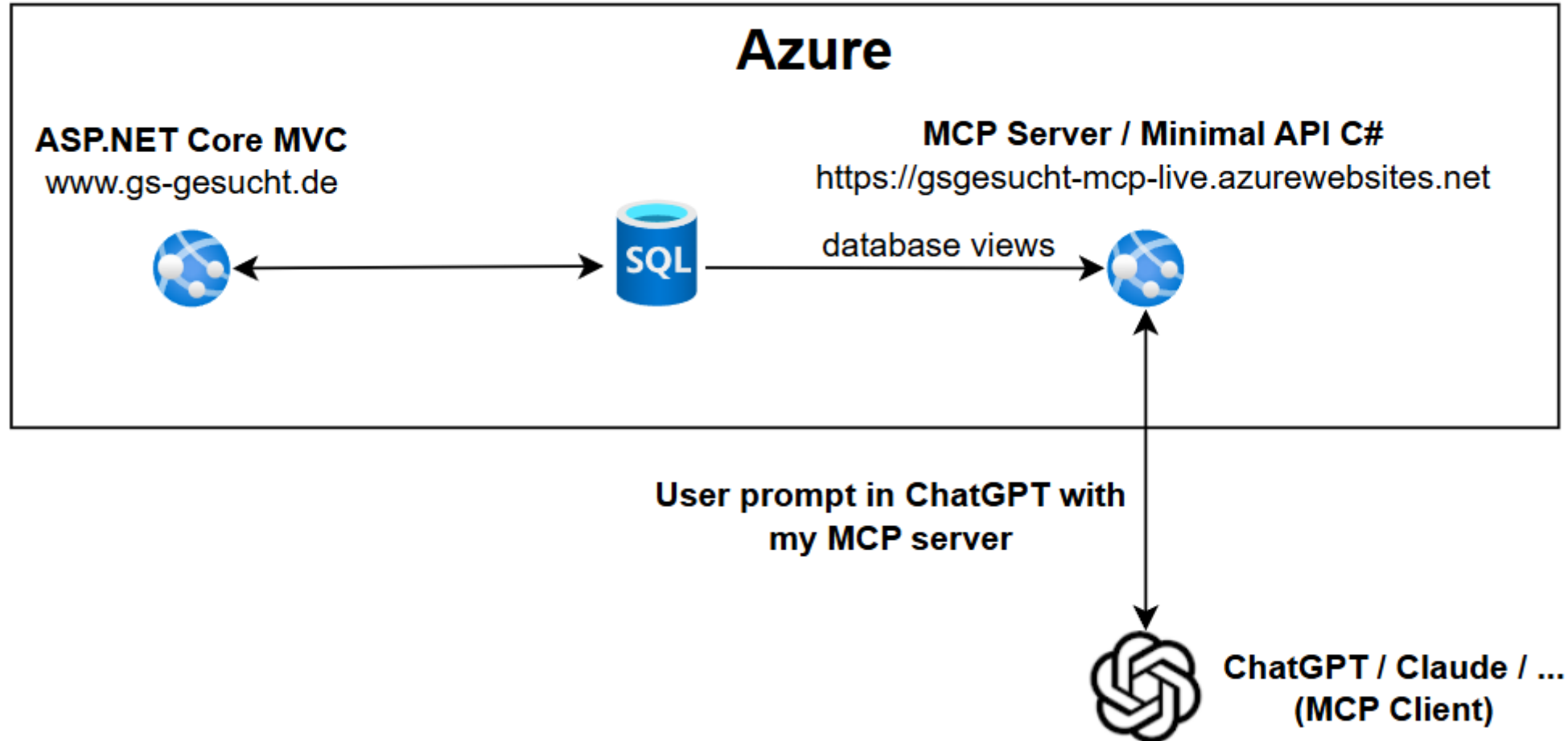
New:

MCP server enables AI assistants to search & recommend games 🤖

The screenshot shows the website's main page. At the top, there is a navigation bar with a home icon, 'Startseite', a search bar, 'Spiele', 'Community', 'Weitere Inhalte', and a user profile 'DerSpieler' with 138 notifications and 900 followers. Below the navigation bar, the page is divided into several sections:

- Mitgliederbereich** (Members Area): A sidebar menu with options like 'Übersicht', 'Nachrichten (138)', 'Einstellungen', 'Meine Mitspielergesuche', 'Meine Treffen', 'Merkzettel', 'Meine Spielesammlung', 'Meine Meldungen', 'Spieleerklärer suchen', 'Prototypen-Tester suchen', and 'Logout'.
- Gesellschaftsspieler-gesucht: Die kostenlose Internetcommunity für Gesellschaftsspieler**: A main banner with text: 'Du spielst gerne Gesellschaftsspiele, dir fehlen aber die Mitspieler? Suche bei uns!', 'Aktuell sind wir **19044** Mitglieder auf der Suche nach Mitspielern!', and 'Suche und finde Mitspieler in deiner Gegend für Gesellschaftsspiele, Brettspiele, Kartenspiele, Rollenspiele, Würfelspiele oder Geschicklichkeitsspiele.' An image of colorful board game pieces is shown on the right.
- Neuste Mitglieder: (» mehr)**: A row of six user avatars with names: Ushandul, Alex2503, Rohremer, Schnute89, Sunny86, and seestern2026.
- Neuigkeiten: (» mehr)**: A section for news, featuring an 'Update Januar 2026' by Matthias from 20.01.2026. The update text says: 'Wir hoffen, dass ihr gut ins neue Jahr gestartet seid! 😊'. Below this, there are three bullet points under 'Updates':
 - Meine Meldungen:** Es gibt nun einen neuen Menüpunkt „Neue Meldungen“, wo ihr den Status eurer Meldungen prüfen könnt.
 - Community Hilfe Seite:** Es wurde eine Pagination eingebaut, um noch besser die zu verbessernden Einträge sehen zu können.
 - Adminbereich:** Benutzern können nun die neuen Rollen Verlag, Autor und PowerUser zugeordnet werden.
- Kennst du schon?**: A section for game recommendations, featuring 'Munchkin Warhammer' with 2 hearts and 4 thumbs up. The text says: 'Munchkin Warhammer 40.000 (2019) verlegt von Pegasus Spiele'.

Architecture Overview



Data Strategy

Read-Only Views

- 6 dedicated views for MCP (vs. 50+ internal tables)
- Decouples MCP from internal schema
- Exposes only what AI needs
- Easy to evolve independently

In-Memory Cache

- Load all game data into memory at startup
- ~3.000 games → millisecond queries
- No database roundtrip per tool call

Trade-offs:

- ✓ Blazing fast responses
- ✓ Reduces database load
- ⚠ Needs periodic refresh for updates


Ideal for read-heavy, relatively static datasets

MCP Tools, Security & Future

3 key tools

- `search`: full-text search (name, authors, publishers)
- `get_top_games`: returns the top community-rated games
- `get_random_games`: returns random games

Security Current State

-  Secrets in Azure Key Vault → safe to open-source
-  Public data only (no user authentication)

Future Considerations


- OAuth integration for user-specific content
- games in personal collection
- personalized recommendations


Takeaways & Resources

What I Learned

- MCP servers are straightforward
 - In my case: C# minimal API + a few attributes
- Design for AI clients:
 - structured data
 - full responses
 - no localization / return in English
- Think of production early on:
 - caching
 - read-only views

Resources

 github.com/Liebeck/gesellschaftsspieler-gesucht-mcp

 www.gs-gesucht.de



Integration in ChatGPT

MCP & ChatGPT

Using MCP servers in ChatGPT (as an MCP host)

ChatGPT can use external MCP servers and call tools dynamically

Requirements:

- The MCP server must be reachable via HTTPS
- The endpoint must be MCP-compatible

Integration is done via:

- Developer Mode
- MCP App / Connector configuration

Developer Mode in ChatGPT

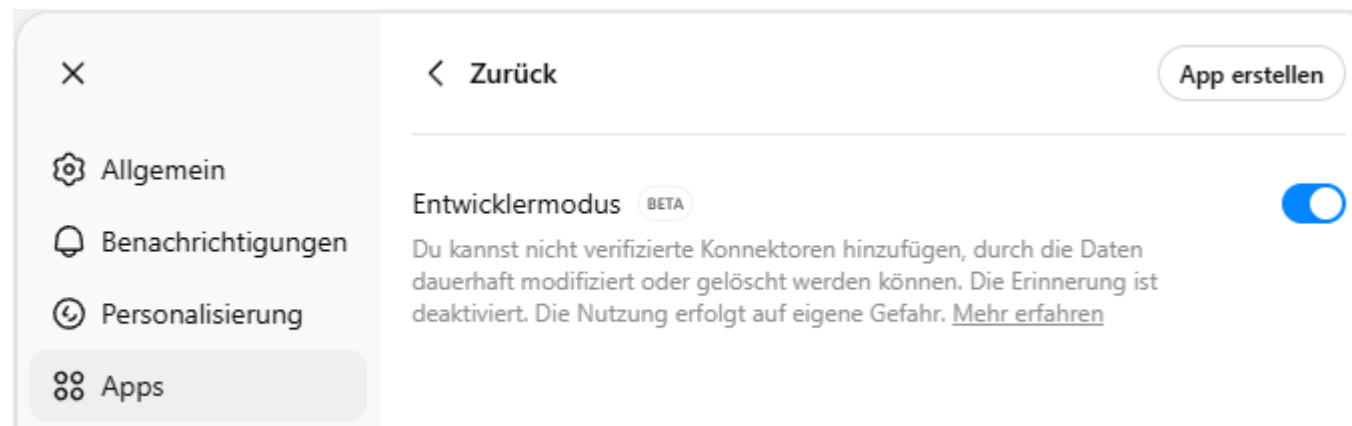
ChatGPT Developer Mode enables

- tool usage
- MCP server integration

Required for:

- connection to custom MCP servers
- local development

Configuration is done directly in the ChatGPT UI



Local Development with ngrok

Local MCP Server + ngrok

- **Problem**: localhost is not reachable from ChatGPT
- **Solution**: HTTPS tunnel using **ngrok** (recommended example by OpenAI)

Typical workflow:

- start MCP server locally
- open an ngrok tunnel
- get a public HTTPS URL

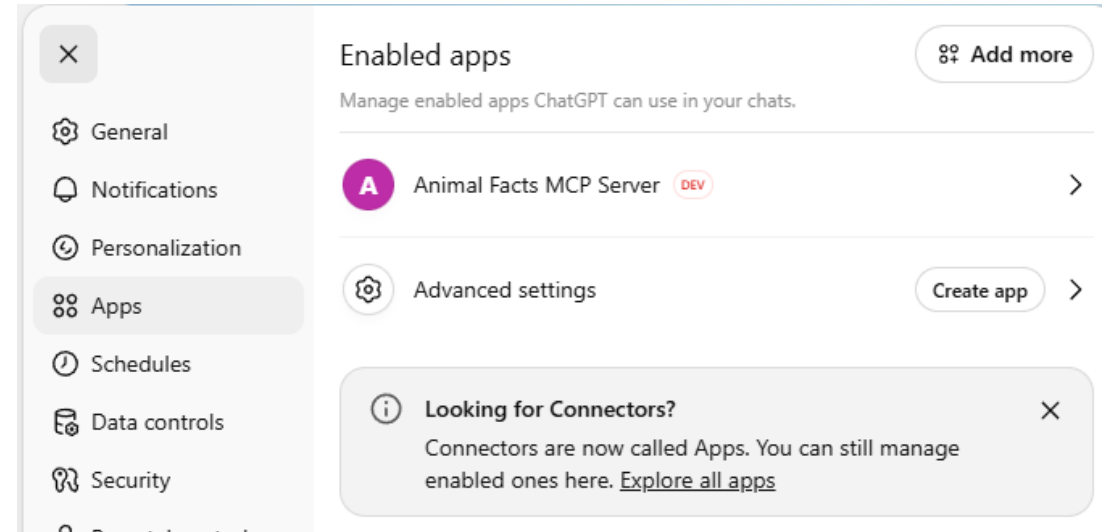
Advantages:

- ideal for local development
- no deployment required

Add MCP Server to ChatGPT

Add MCP Server to ChatGPT

- you need to provide
 - name
 - description
 - MCP endpoint URL




The description determines **when** ChatGPT will use the server

Recommendation: **use explicit, action-oriented descriptions**

Add MCP Server to ChatGPT

New App BETA ×

 **Icon (optional)**
Minimum size: 128 x 128 px

Name

Board Game MCP from Gesellschaftsspieler-gesucht

Description (optional)


This MCP server returns information about board games, such as

MCP Server URL

https://gsgesucht-mcp-live.azurewebsites.net/mcp


Authentication

No Auth ▾

 **Custom MCP servers introduce risk.** [Learn more](#)

I understand and want to continue

OpenAI hasn't reviewed this MCP server. Attackers may attempt to steal your data or trick the model into taking unintended actions, including destroying data.

 [Read the guide](#) Create

Tool Invocation from the Chat

Tool Invocation from the Chat

- ChatGPT decides whether a tool should be used
- For demos, explicit prompts are recommended

Tool calls are:

- visible in the UI
- traceable
- easy to debug

Example prompt:

Please show the top 10 best ranked board games from Gesellschaftsspieler-gesucht. Please show the average ranking, the number of players, the play time and the number of Likes.

Gesellschaftsspieler-gesucht MCP server example

Please show the top 10 best ranked board games from Gesellschaftsspieler-gesucht. Please show the average ranking, the number of players, the play time and the number of likes.

{≡} Called tool ▾

Below are the **Top 10 board games on Gesellschaftsspieler-gesucht (by GSG community rank)**, including **average rating, player count, play time, and likes**.

GSG Rank	Game	Avg. rating	Players	Play time (min)	Likes
1	Twilight Imperium: 4. Edition	4.7069	3–6	480	207
2	Die Burgen von Burgund	4.5625	2–4	60–120	302
3	Brass: Birmingham	4.5588	2–4	60–120	332
4	Arche Nova	4.5577	1–4	120–240	644
5	Dune: Imperium	4.5543	1–4	60–120	377
6	Terraforming Mars	4.5448	1–5	120–240	593
7	Spirit Island	4.5341	1–4	120–240	429
8	Scythe	4.5323	1–5	60–120	293
9	Agricola	4.5000	1–5	120–240	168
10	Root	4.5000	2–4	60–120	156

If you want, I can also add **ratings count, favorites, release year**, or filter the top list for a specific **player count** (e.g., "best at 2 players").

Remote MCP Server Best Practices

Remote MCP Server Best Practices

- **Security considerations**
 - Always verify the authenticity of remote MCP servers before connecting
 - Only connect to trusted servers
 - Review permissions requested during authentication
 - Be cautious when granting access to sensitive data or systems
- **Managing multiple connectors**
 - Multiple remote MCP servers can be connected simultaneously
 - Organize connectors by purpose or project
 - Regularly review and remove unused connectors to keep the workspace secure

Source: <https://modelcontextprotocol.io/docs/develop/connect-remote-servers#best-practices-for-using-remote-mcp-servers>

Integration Summary

MCP servers can be used by:

- IDEs
- ChatGPT
- Custom chat applications

Integration is:

- standardized
- reusable

 **The MCP server remains the same regardless of the host.**



Further Concepts

Overview: Further Concepts

- Billing
- Rate Limiting
- Read vs. Write Tools
- Environment Separation
- Caching
- Tool Discovery & Capability Filtering
- Prompt Injection

Most of these topics are

- not part of the MCP protocol itself
- but they determine whether an MCP server is production-ready.

MCP solves the tool integration problem. Architecture, security, and operations solve the rest.

Billing & Cost Models

Billing & Cost Models

- billing is not part of the MCP protocol
- charging is handled:
 - in the MCP host, or
 - in front of the MCP server (e.g., via a gateway)

Typical billing models:

- per tool call
- weighted calls (expensive tools cost more)
- quotas / credits per time period

Important distinction:

MCP standardizes how tools are invoked, not how they are billed.

Rate Limiting

Rate Limiting & Quotas

- also not an MCP feature
- typically implemented via:
 - API gateways
 - Middleware (rate limiting by key / token)

Possible limits:

- per user
- per tool
- per time window

Best practice:

- coarse limits at the gateway
- fine-grained limits inside the MCP server

Read vs. Write Tools

Read vs. Write Tools

- not all tools are equally critical

Typical distinction:

- Read tools: Search, lookup, analysis
- Write tools: Create, update, delete

Consequences:

- different approval requirements
- different rate limits
- possibly explicit user confirmation

Rule of thumb: Reading is cheap. Writing is dangerous.

Environment Separation

Environment Separation (Dev / Test / Prod)

MCP servers are new APIs and they require environments.

Typical separation:

- development
- test / staging
- production

Differences per environment:

- available tool sets
- data sources
- authentication keys

Critical rule: An agent should never accidentally write to production

Caching

Caching & Efficiency

- many tool results change infrequently
- Caching reduces:
 - costs
 - latency
 - unnecessary tool calls
- Especially suitable for:
 - reference data
 - lookups
 - search results

Notes:

- Caching belongs in the MCP server or upstream infrastructure, not in the prompt.
- **Not every tool call needs to be recomputed every time.**

Tool Discovery & Capability Filtering

Tool Discovery & Capability Filtering

- not every user or agent should see every tool
- **Tools can be filtered by:**
 - role
 - scope
 - OAuth claims
- OAuth enables:
 - fine-grained permissions
 - tool access “on behalf of the user”

Key insight: Tool access is an **authorization problem**, not a prompt problem.

Prompt Injection

Prompt Injection & Input Validation

- tool inputs originate indirectly from the user
- attacks aim to:
 - trigger unauthorized actions
 - bypass security rules
- **Essential countermeasures:**
 - strict input validation
 - whitelists instead of blacklists
 - least-privilege tools
- **Warnings:**
 - LLMs are **not** a security component.
 - trust no prompt, not even your own.



Conclusion

What MCP Really Is (and Why It Matters)

MCP is not:

- a new model
- a prompting framework

MCP is:

- a standard / protocol
- infrastructure for agentic systems

💡 MCP does not solve an AI problem. It solves the tool and integration problem.

Why MCP Is Powerful (Especially for Backend Teams)

MCP Makes Tools AI-Usable

MCP cleanly decouples:

- LLMs ↔ business logic
- Tools ↔ platforms
- Agents ↔ implementations

Results:

- reusable tools
- interchangeable hosts
- lower integration effort
 - auth
 - logging
 - deployment

For backend developers:

- MCP servers are **normal services**
- Built with familiar concepts:
 - APIs
 - auth
 - logging
 - deployment

 **We will talk less about models themselves but and more about their capabilities.**



Thank you!

LinkedIn

<https://globalai.community/chapters/vienna>

GitHub Copilot Newsletter

Dr. Matthias Liebeck



Made Possible by

