Model Context Protocol

André Schaaff, Thomas Boch, Sébastien Derriere, Matthieu Diebolt (UTBM intern)

Centre de Données astronomiques de Strasbourg

Astro-CC Trieste, 7-9 October 2025

Context

- Continous R&D at CDS around AI to access the CDS data and services (and more…)

- Mainly a background of   experiments to build chatbots presented during astroinformatics, ESA NLP workshop, IVOA sessions, etc.

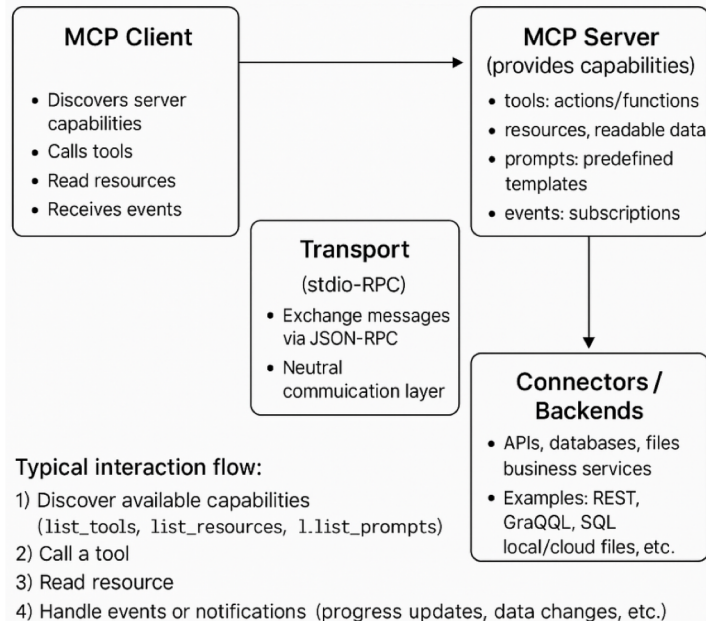- Switch to experiments with OpenAI API since ChatGPT advent with now a recent focus on MCP

MCP ?

- Model Context Protocol, initiated by Anthropic, first announcement in November 2024

- An open Standard for Connecting Models, Tools, and Data
- Why ?
- AI models require external context for accurate reasoning like an access to business tools in large companies
- Major AI actors are adopting it like OpenAI since March 2025 and we started to experiment it last in September !

☐ MCP server

- Enabling a generic exchange between LLMs and APIs
- FastMCP is the official MCP SDK for Python
- Easy MCP server implementation and JSON-RPC conversion for the exchange
- prompts are meta-capabilities: they tell the LLM *how to ask*, not *what to do*.
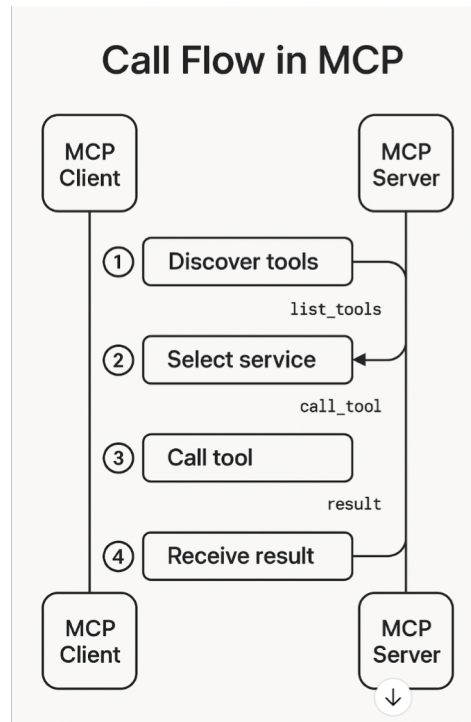
Principle of an MCP Server

MCP Client
- Discovers server capabilities
- Calls tools
- Read resources
- Receives events

MCP Server
(provides capabilities)
- tools: actions/functions
- resources, readable data
- prompts: predefined templates
- events: subscriptions

Transport
(stdio-RPC)
- Exchange messages via JSON-RPC
- Neutral commuication layer

Connectors / Backends
- APIs, databases, files business services
- Examples: REST, GraQQL, SQL local/cloud files, etc.

Typical interaction flow:
1) Discover available capabilities (`list_tools`, `list_resources`, `1.list_prompts`)
2) Call a tool
3) Read resource
4) Handle events or notifications (progress updates, data changes, etc.)

MCP server call flow

- Discovering, selection, call and result
- Descriptions to provide like list of tools, etc.

- Configuration file examples easy to find online

Call Flow in MCP

MCP @ CDS

- In previous experiments we trained models with Simbad structure, to generate ADQL queries, etc.
- With MCP the link to the services becomes more natural

- Example : « give me 5 images of M1 with a size of 3 degrees in infrared » will chaine tools : Sesame to obtain the coordinates , MocServer to obtain the infrared HiPS at these coordinates), and then HiPS2FITS to provide the infrared images. => it works

MCP @ CDS (2)

- A description of the possible services to query was given to the AI

- Difference with previous experiments :
- give to an AI the means to do it rather than ingesting plenty of examples and hoping that the AI will be able to reproduce (without hallucinations !)

- Improving is ongoing => more examples at Görlitz IVOA.

MCP server in a few lines

- Service or tool description to help the AI to « know » when to call it.
- following a query, an IA will then select the tool which seems the most adapted to answer to it.

Idea concerning MCP and OpenAPI

- OpenAPI and MCP could be linked as an interoperability bridge between existing web APIs and AI-driven systems.
- they are conceptually compatible
- both OpenAPI and MCP describe structured ways for a client to understand and interact with services.

Idea concerning MCP and OpenAPI (2)

- OpenAPI describes HTTP APIs (endpoints, parameters, schemas, responses) and MCP describes capabilities (tools, resources, prompts, events) in a transport-agnostic way using JSON-RPC.

- Points to evaluate
- Existing REST APIs (already using OpenAPI) immediately usable by MCP clients ?
- => LLMs could automatically discover and call APIs without manual integration.

Conclusion

- MCP is very promising, probably the missing link.
- More MCP / CDS integrations in IVOA Görlitz with Thomas Boch during DCP / KD joint session

- Ongoing study around bridging OpenAPI and MCP

- Working with AI tools is becoming more and more a real time work to integrate new tools and features and it is motivating.

- Rendez-vous with Thomas in Görlitz for the last updates !