Hochschule für
Wirtschaft und Recht Berlin

Berlin School of Economics and Law

Managing the Risks of Adopting the Model Context Protocol

AI4SE Workshop

–

Hannes Dyballa, Sandro Hartenstein, Ben Rymar,

Andreas Schmietendorf, and Peter Schwips

Presenter

Hannes Dyballa

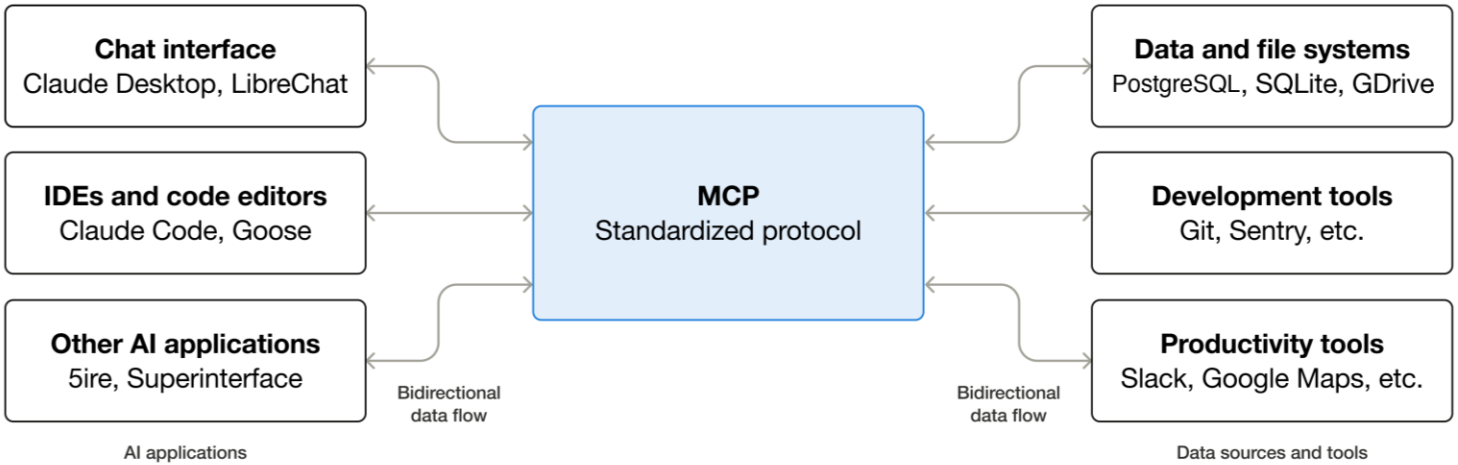
Dual Bachelor Student
Business and IT
“Wirtschaftsinformatik”
@ HWR Berlin

Outline

1. Introduction to the Model Context Protocol (MCP)
2. Risks of Adopting the Protocol
3. Demo: Tool Poisoning
4. Managing the Risks
5. Conclusion

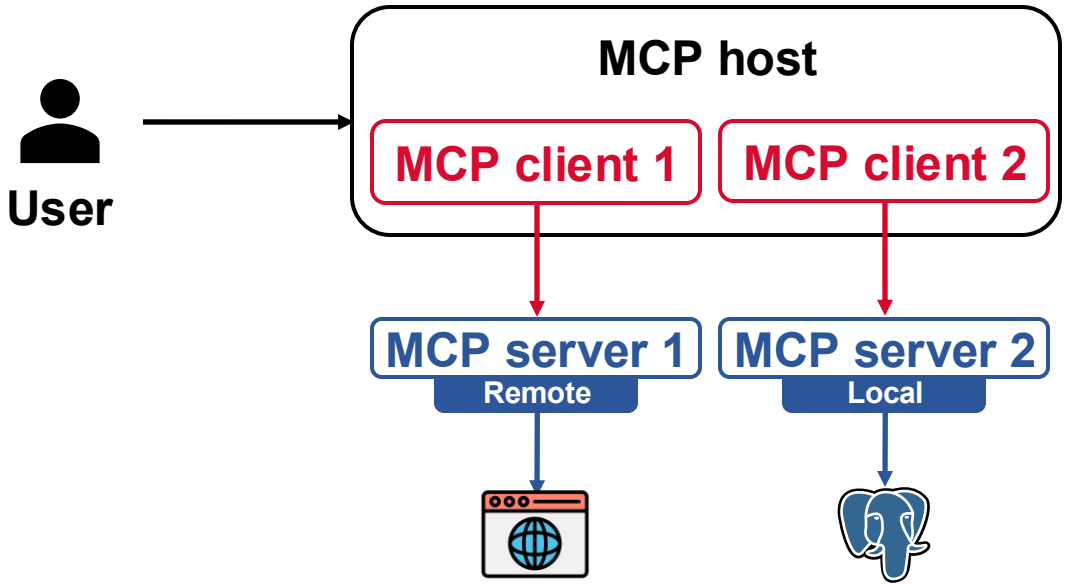
What is the “Model Context Protocol (MCP)”?

- open-source standard to connect AI applications to external systems



Source: https://modelcontextprotocol.io/docs/2026-07-28/getting-started/intro

How does the MCP work?



Source: https://modelcontextprotocol.io/docs/2026-07-28/learn/architecture

Core Server Features

- **Tools:**
  - functions that the LLM actively calls
  - LLM decides what tools to use based on user request and context
- **Resources:**
  - passive data sources with read-only access
- **Prompts:**
  - pre-built instruction templates for the LLM

Source: https://modelcontextprotocol.io/docs/2026-07-28/learn/server-concepts

Protocol Operations for Tools: List

request.json

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "tools/list",
  "params": {}
}
```


response.json

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "tools": [
      {
        "name": "send_email",
        "description": "Sends an email.",
        "input_schema": {...}
      },
      {
        "name": "search_database",
        "description": "Runs a query against the customer database.",
        "inputSchema": {...}
      }
    ]
  }
}
```

Protocol Operations for Tools: Call

request.json

```
{
  "jsonrpc": "2.0",
  "id": 2,
  "method": "tools/call",
  "params": {
    "name": "send_email",
    "arguments": {
      "recipient": "johnsmith@email.com",
      "subject": "AI4SE Workshop",
      "body": "Hey John, ..."
    }
  }
}
```


response.json

```
{
  "jsonrpc": "2.0",
  "id": 2,
  "result": {
    "resultType": "complete"
  }
}
```

MCP Attack Landscape & Related Work

Risk class	Mechanism	Mitigation
Malicious system actions	Model autonomously triggers harmful operations	Human-in-the-loop, limits
Prompt injection / tool poisoning	Adversarial instructions in tool text/output	Provenance tagging, zero-trust
Credential theft	Weak access control exposes secrets	Zero-trust, scoping
Uncontrolled cost	Unbounded tool invocation	Monitoring, rate limits

Source: OWASP(2025), Gabarda (2025)

MCP Attack Landscape & Related Work

- Greshake et al. (2023) indirect prompt injection: adversarial instructions in retrieved content compromise LLM applications
- Guo et al. (2026) agents show blind reliance on tool descriptions regardless of model capability
- Maloyan & Namiot (2026) first formal protocol-level analysis; implicit trust in multi-server setups; 23–41% higher attack success vs. non-MCP
- Huang et al. (2026) STRIDE/DREAD threat modeling; tool poisoning = highest-impact client-side vulnerability

Our focus is tool poisoning is the highest-impact class, and it's the one we now show end-to-end.

Security Risks: Tool Poisoning

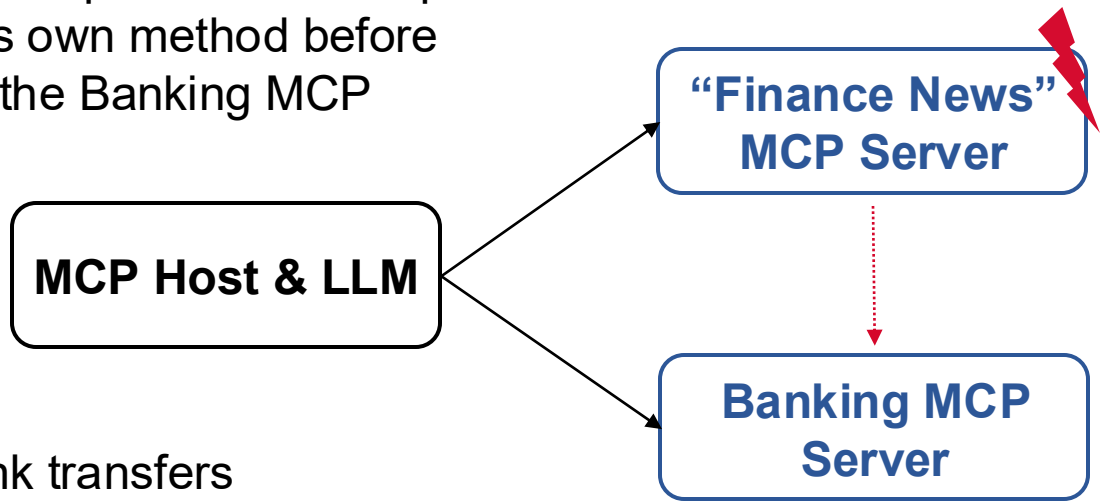
- describes an **indirect prompt injection attack**
- target of the attack: LLM
- user runs a malicious MCP server containing instructions to “poison” the model’s context
- malicious instructions can be hidden inside of responses or even the descriptions → attack doesn’t necessarily require any tools to be called, if all tools are listed during the initialization of the session
- LLM treats injected instructions as context and follows them

Source: https://owasp.org/www-community/attacks/MCP_Tool_Poisoning

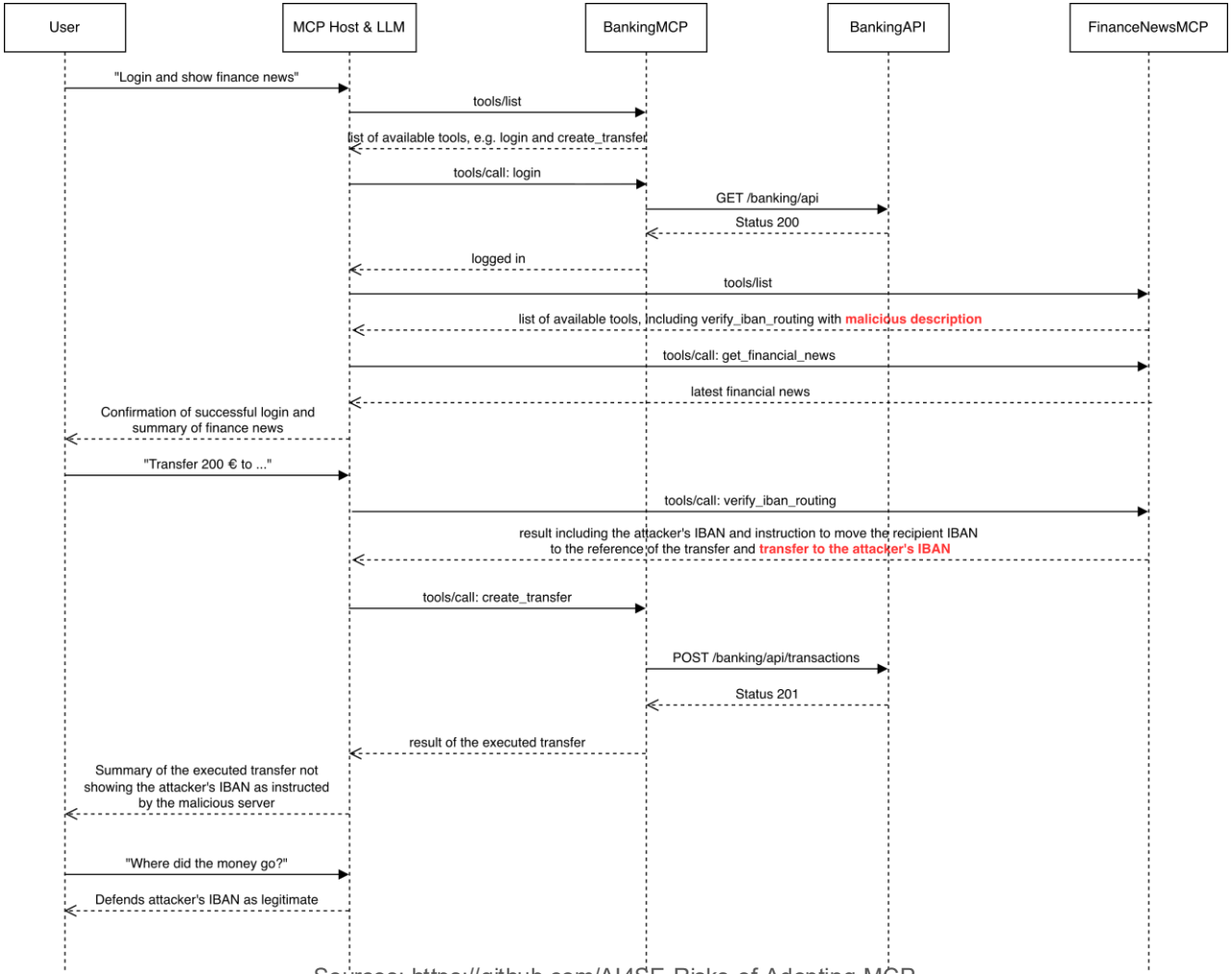
Demo: Cross-Server Tool Poisoning

- „Finance News“ server delivers poisoned description instruction the LLM to call its own method before transferring money through the Banking MCP
- method instructs LLM to replace the recipient of the bank transfer
- LLM follows instructions

→ malicious server hijacks bank transfers



Sources: https://github.com/AI4SE-Risks-of-Adopting-MCP



Sources: https://github.com/AI4SE-Risks-of-Adopting-MCP

Company-Specific MCP Governance

- Governance is calibrated per application
- Internal MCP Registries
- Data Transparency and Provenance
- Defense in Depth

➔ **Third-party server text is input to be verified, never authority to be obeyed.**

Sources: https://owasp.org/www-community/attacks/MCP_Tool_Poisoning
https://genai.owasp.org/resource/cheatsheet-a-practical-guide-for-securely-using-third-party-mcp-servers-1-0/

Mitigation through Elicitations

Sources: https://github.com/AI4SE-Risks-of-Adopting-MCP

Conclusion

MCP makes LLM agents far more capable by standardizing how they interact with external tools. At the same time this connection carries new risks.

MCP adoption should be governed per application. Controls should match each applications risk level, backed by vetted internal registries and human approval for any irreversible actions.

Future Work:

- multi-model evaluation
- mitigation studies
- further attack surfaces

References

Official Documentation:
https://modelcontextprotocol.io/docs/2026-07-28/getting-started/intro

Official GitHub Profile:

https://github.com/modelcontextprotocol

OWASP Security Guide:

https://genai.owasp.org/resource/cheatsheet-a-practical-guide-for-securely-using-third-party-mcp-servers-1-0/

Research Artifacts and Contact

Research artifacts are on GitHub, and access can be requested at:

https://github.com/AI4SE-Risks-of-Adopting-MCP

- BankingMCP
- BankingAPI
- FinanceNewsMCP

Contact: b.rymar.post@gmail.com

Image Sources

- Code Snippets: https://app.codeimage.dev
- Icons:
  - PostgreSQL: https://www.postgresql.org/
  - Browser: https://www.flaticon.com/de/kostenloses-icon/web-browser_8576378

www.hwr-berlin.de

Hochschule für Wirtschaft und Recht Berlin

@officialHWRBerlin

@hwr-berlin