MCP
Dev Summit
Mumbai

The 5 Wrong Reasons to Build an MCP Server (And What to Do Instead)

Daniel Oh - Developer Advocate at IBM (@danieloh30)
Java Champion | CNCF Ambassador | CNCF TAG DevEx Co-chair | Microsoft MVP

AI Landscape

THE AGENTIC BOOM

CENTRAL REGISTRIES

The explosion of the “Agentic IDE” ecosystem.

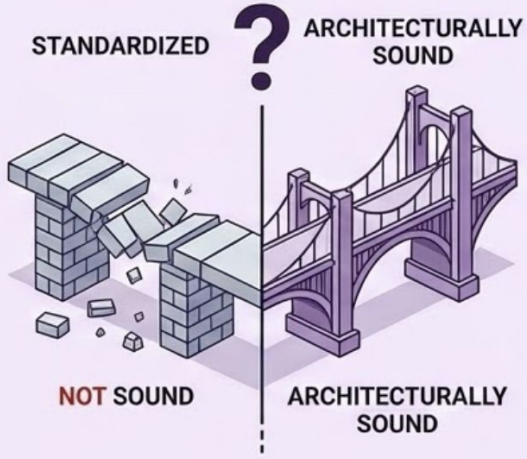
5,000+ community MCP servers listed on central registries.

CORPORATE REALITY & RISKS

The corporate pressure to “AI-enable” every internal API.

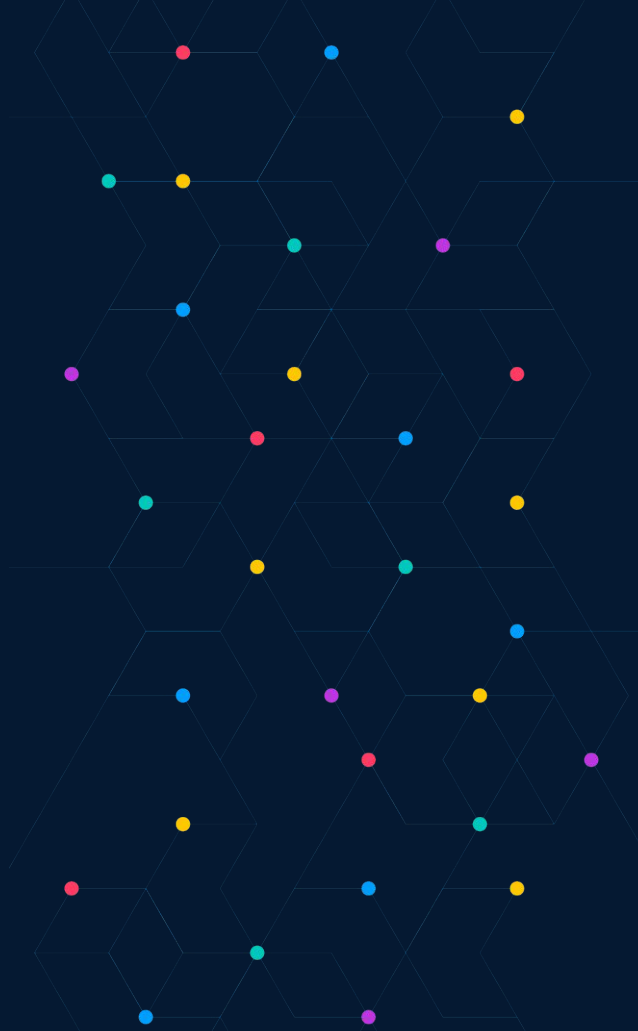
The risk of the “Everything looks like a nail” syndrome.

STANDARDIZATION VS. SOUNDNESS



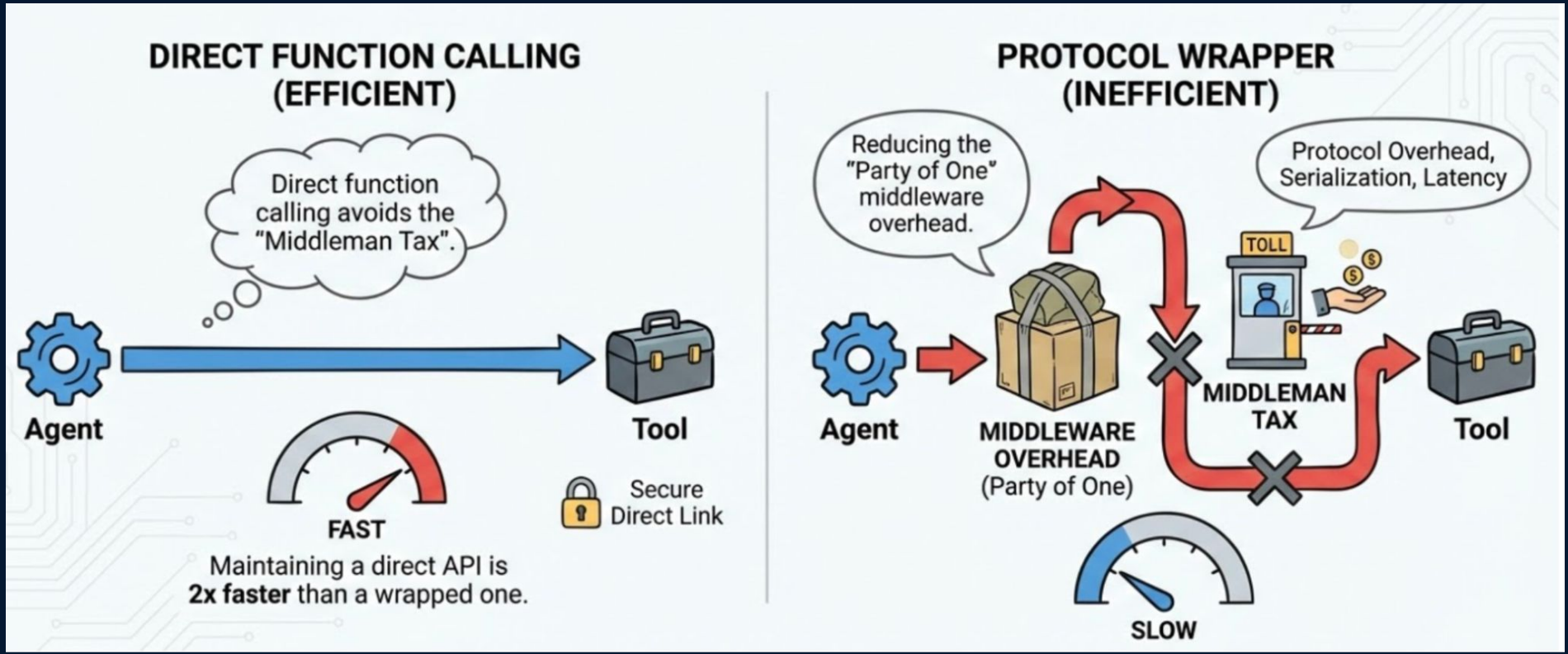
Why “Standardized” does not always mean “Architecturally Sound.”

No is the Right Answer



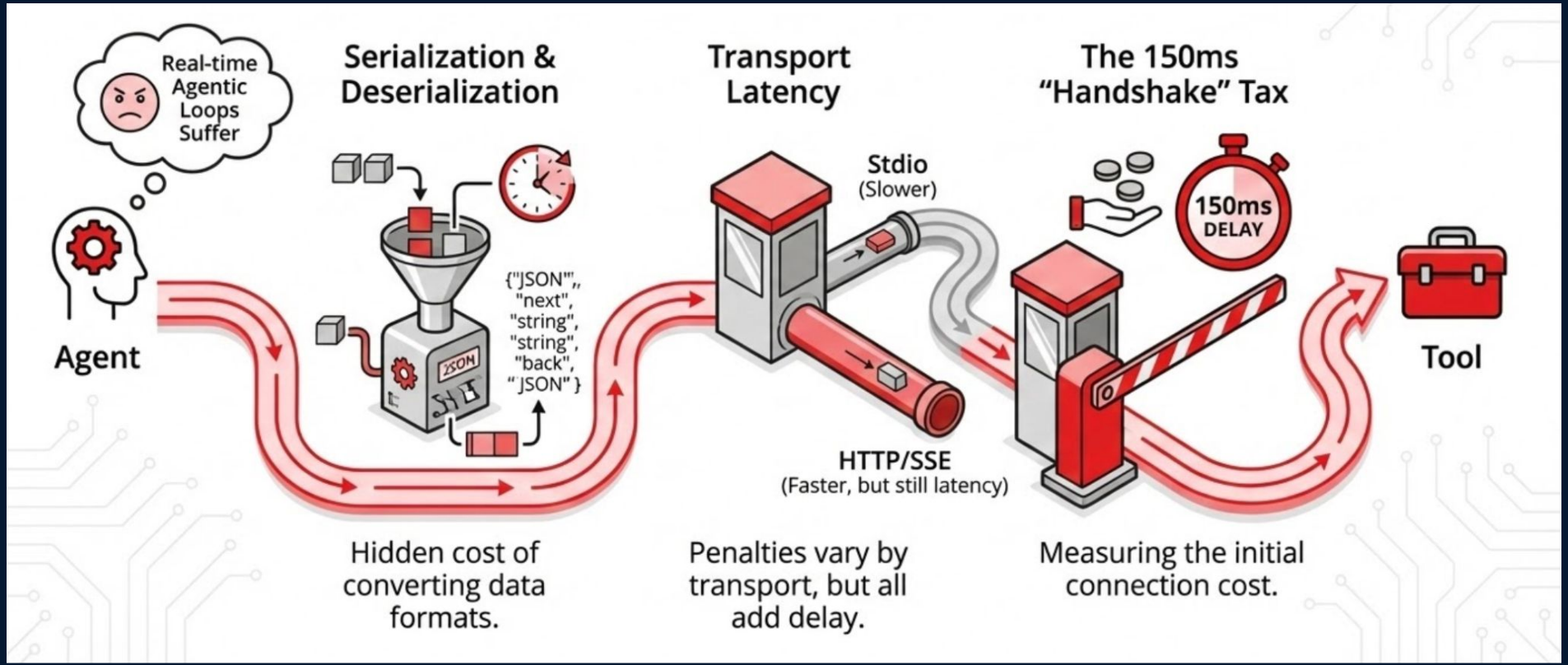
Wrong Reason #1: Single Consumer Trap

One Agent + One Tool = Zero reason for a protocol!



Wrong Reason #2: Performance Myth

JSON-RPC is a convenience, not a performance optimization!!



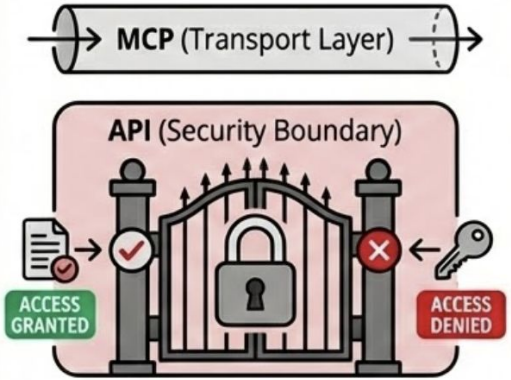
Wrong Reason #3: Security Fallacy

MCP is a transport layer, not a security boundary

The Danger of Blind Trust

Trusting the LLM client blindly.
Increasing the 'Prompt Injection' surface area via tools.

Authorization Belongs at the API Level



MCP is just a transport mechanism.
API level is the correct place for authorization.

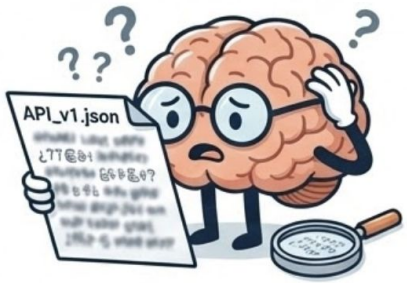
The "Exposed Internal Tool" Nightmare

Exposed Internal Tool nightmare scenario.

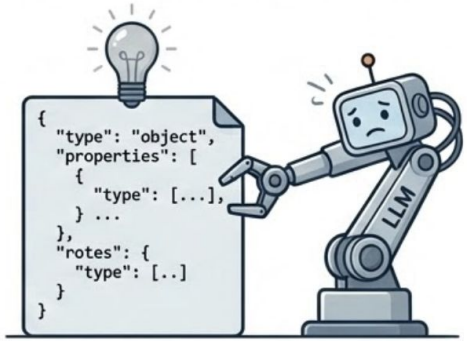
Wrong Reason #4: Documentation Replacement

MCP won't fix a poorly designed, undocumented API

- **LLMs still require** high-quality semantic context to work.

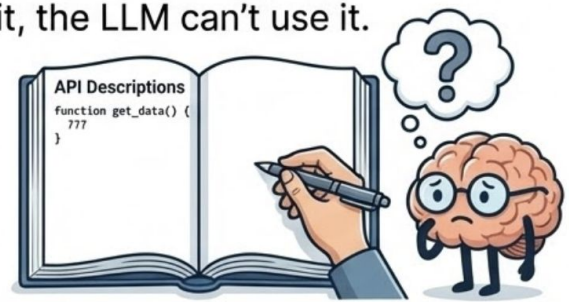


- **The JSON Schema** is not "Auto-Intelligence."



- **"Garbage In, Garbage Out"**: Standardizing a mess just makes it a standard mess.

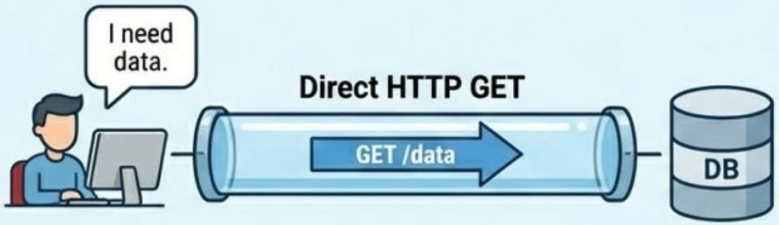
- **Descriptions are code**: If you can't describe it, the LLM can't use it.



Wrong Reason #5: Simple CRUD Operations

HTTP verbs are already a "Standard Protocol”!

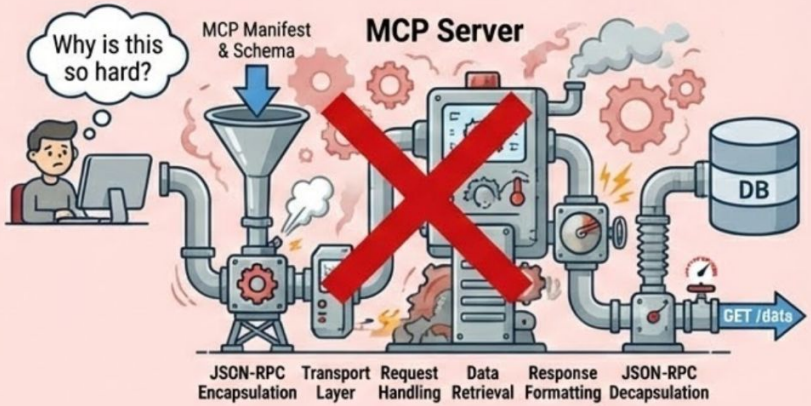
The Simple Way: Standard Protocol



HTTP verbs (GET, POST, etc.) are the standard.
cURL or shell scripts are sufficient “tools”.
Direct, efficient, and easy to maintain.

```
$ curl https://api.example.com/data
```

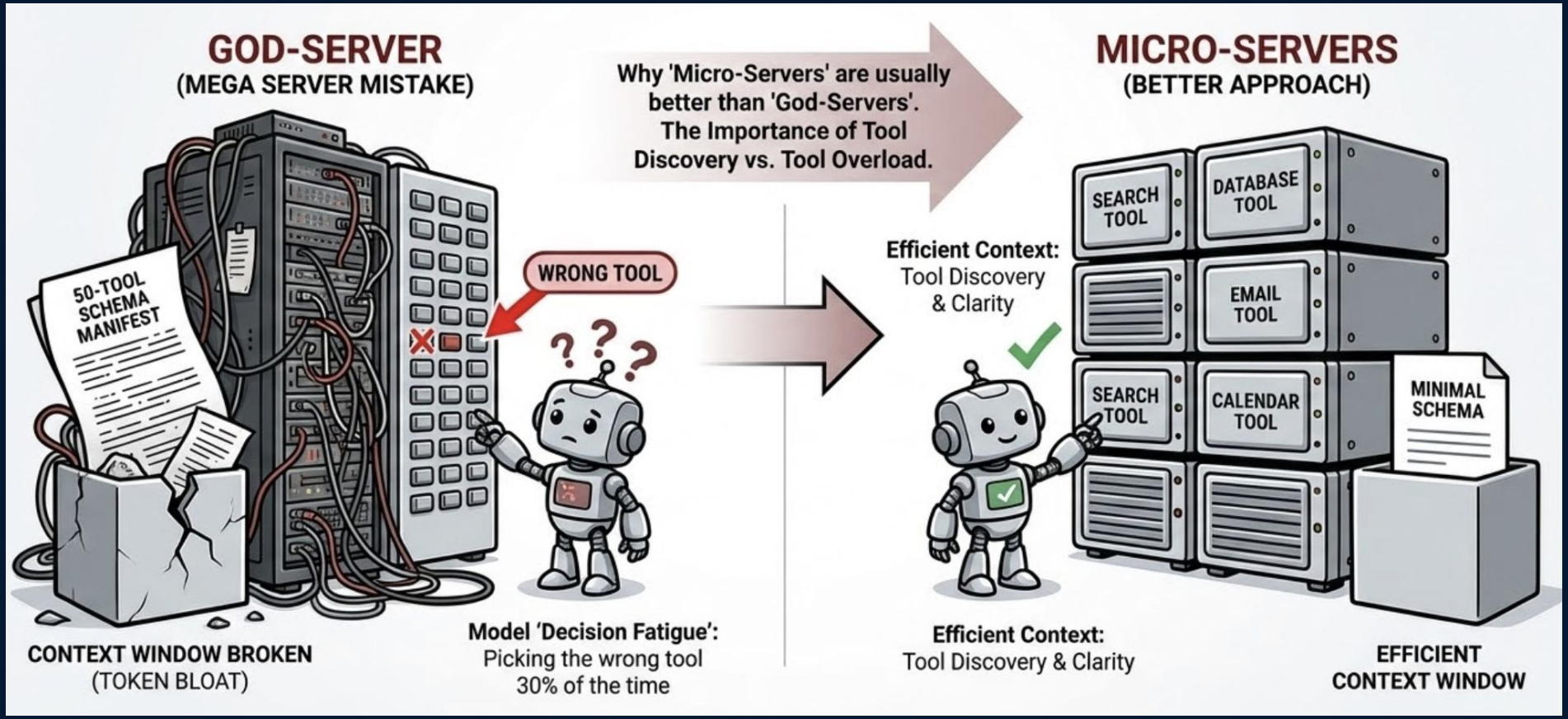
The Wrong Way: MCP Overkill



Wrapping a single endpoint in a server manifest is unnecessary.
High maintenance burden for a single endpoint.
Creates “Interface Bloat” for basic tasks.

Architecture & Battle Scars

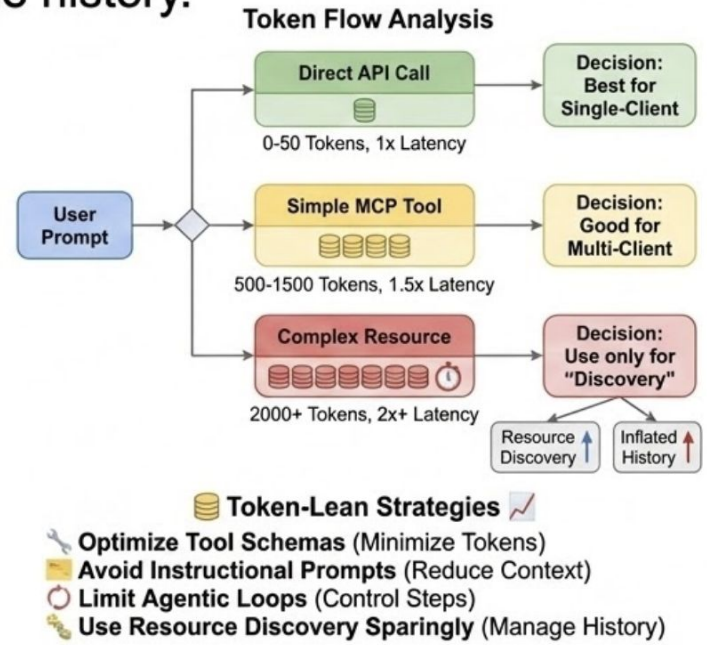
Case Study: Mega Server Mistake



Token Efficiency Table

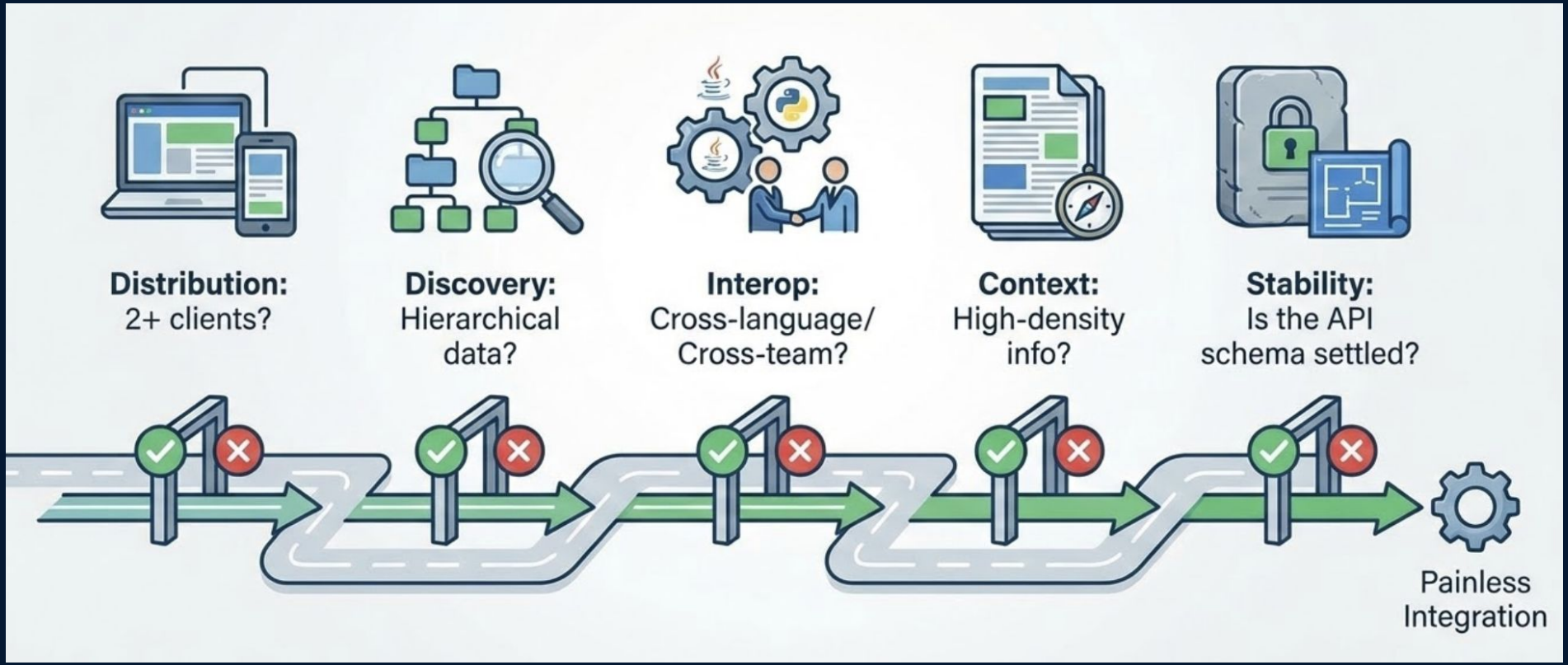
- Quantitative comparison of Tool schemas vs. Function signatures.
- Impact of “Instructional” prompts within the MCP server.
- The price of multi-step “Agentic Loops.”
- How “Resource Discovery” inflates the message history.
- Strategies for “Token-Lean” MCP design.

Token Overhead & Latency Comparison			
Interaction Type	**Est. Token Overhead**	**Latency Factor**	**Decision**
Direct API Call	0 - 50	1x	Best for Single-Client
Simple MCP Tool	500 - 1,500	1.5x	Good for Multi-Client
Complex Resource	2,000+	2x+	Use only for “Discovery”



Worth It Framework

5-Point "Painless" Checklist

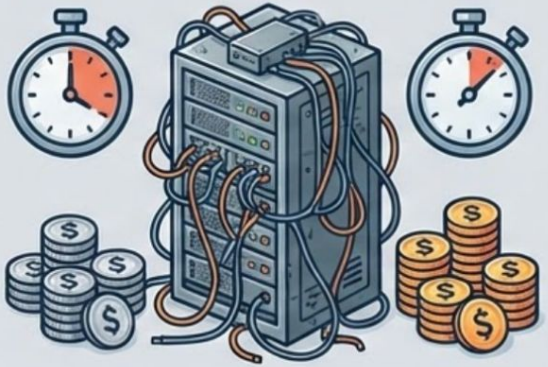


Live Demo:
Over-Engineered Helpdesk

MCP
Dev Summit
Mumbai

Scenario: The "Over-Engineered" Helpdesk Server.

Over-Engineered Server



Latency & Token Use:
High / Inefficient.

The "Aha!" Moment:
When MCP finally pays for itself.

Clean SDK

Latency & Token Use:
Low / Optimized.

Tech Stack

MCP SDK

Quarkus

LangChain4j

OpenAI

Expectations
Setting expectations for "Live Code" failures.

https://github.com/danieloh30/over-engineered-tool.git

Conclusion & The Pragmatic Path Forward

ARCHITECT FOR AGILITY

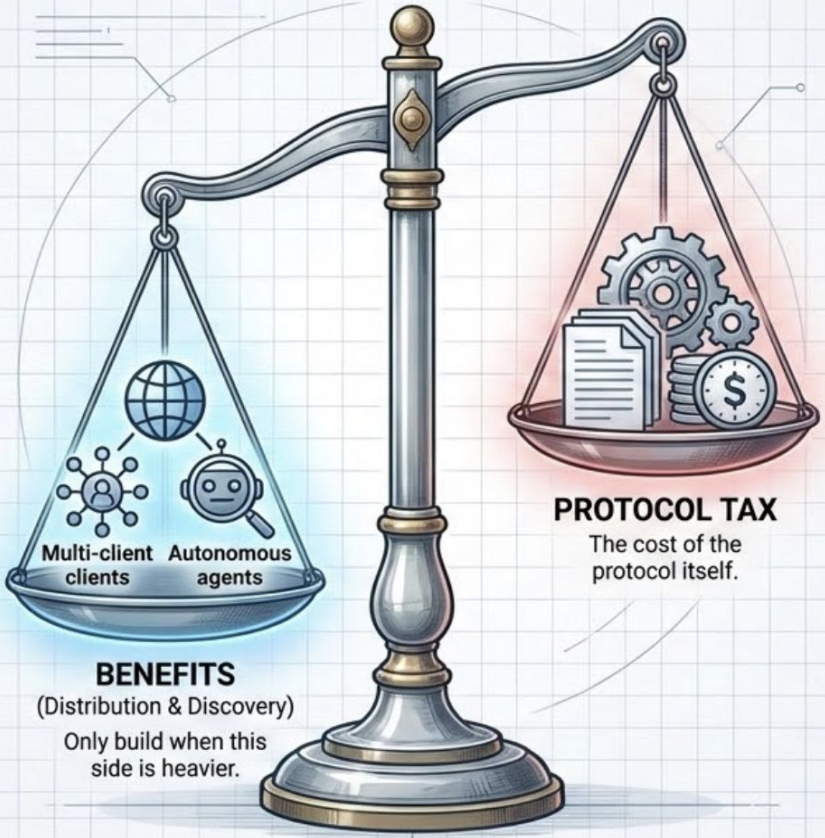
Clean Java Service
(Business Logic)

Disposable MCP Interface
(Disposable Layer)

Keep business logic in clean, protocol-agnostic Java services; treat MCP as a "disposable" interface layer.

FOCUS ON THE "WHY"

Does the LLM actually need a dynamic tool registry, or just a well-documented API endpoint?



STANDARDIZE WITH INTENT

10-line SDK
(Superior)

500-line JSON-RPC wrapper
(Over-Engineered)

Don't let the "Standard" label fool you. A 10-line SDK is often superior to a 500-line JSON-RPC wrapper.

FINAL CALL TO ACTION

Audit your current "Agentic" projects today—identify one MCP server that should have been a simple function call and refactor it.

THANK YOU!
QUESTION?

MCP Dev Summit Mumbai