MCP Dev Summit NYC 2026

Clients? Servers? Agents?
The beautiful asymmetry of the MCP Spec

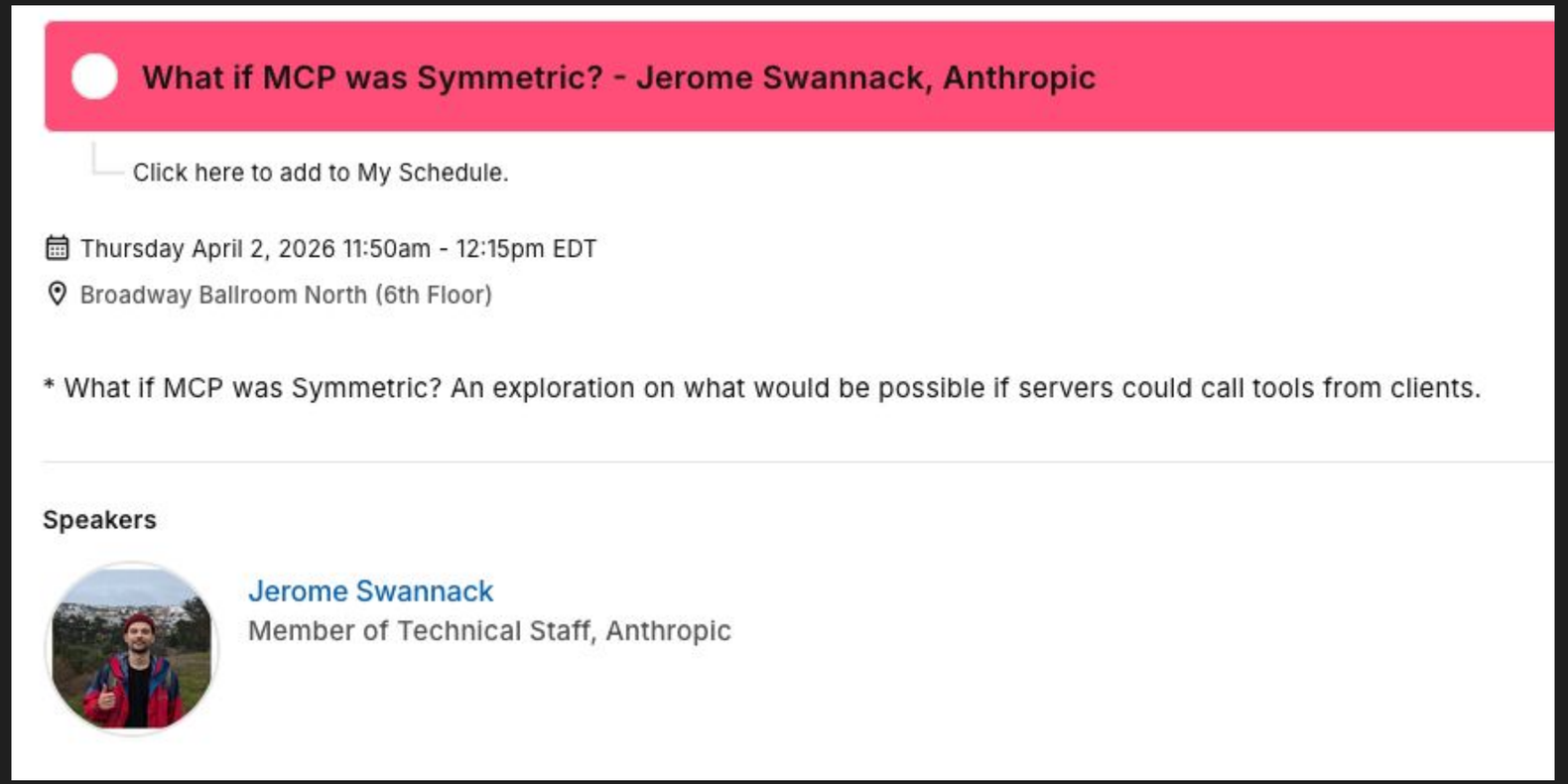
Rohit Ganguly

About Me

- I work at **Descope**, an auth/identity company **(Booth G8!)**
- NYC based!
- Before this, I worked at Microsoft on a few teams:
  - Azure SDK
  - Azure MCP / GitHub Copilot for Azure
  - VS Code

https://rohit.info

Shoutout

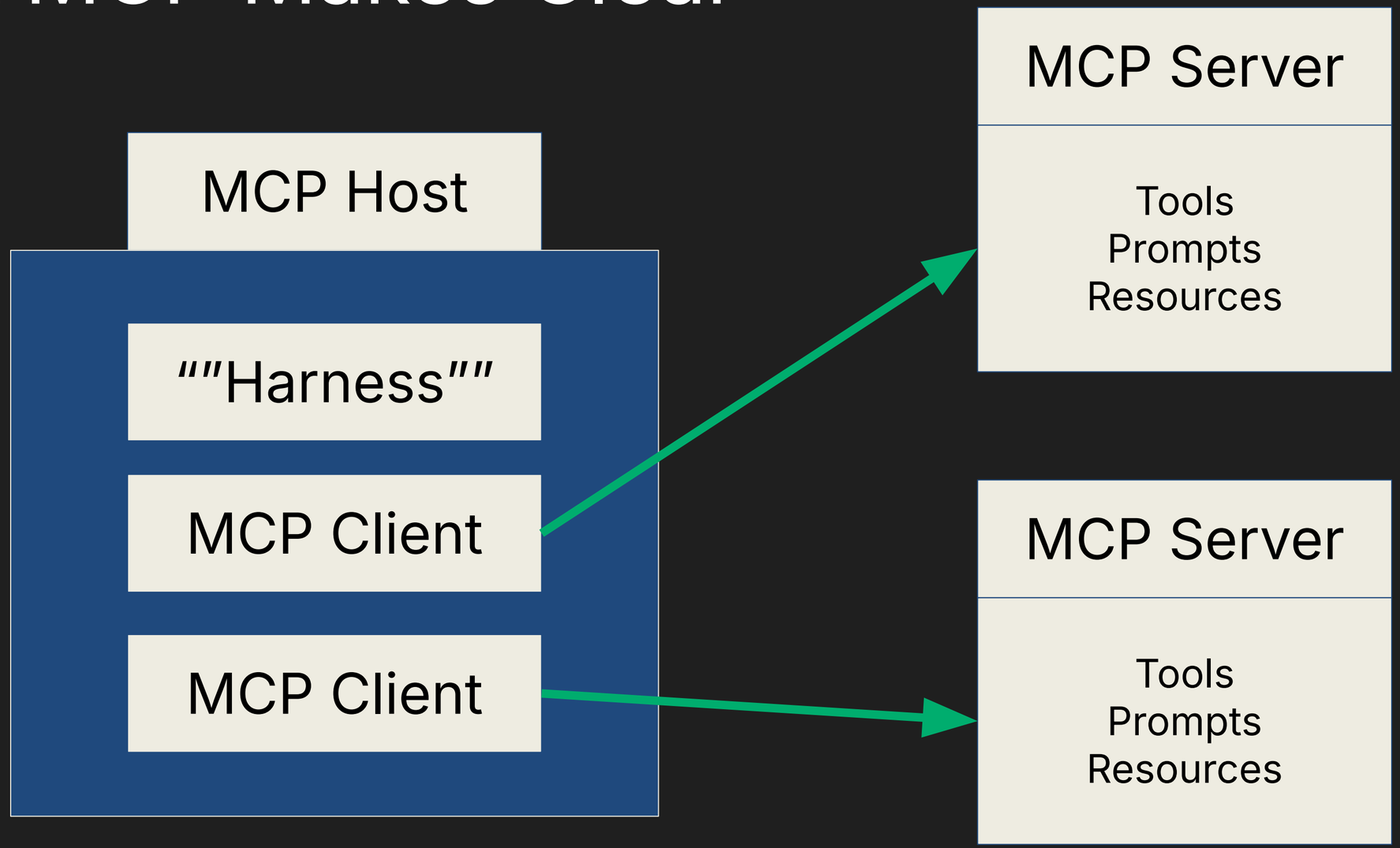


If you missed this talk, tune into the recording soon!

Talk Goals

- Inspire curiosity
- Have fun
- Oversimplify
- Poke holes
- Think a little unrealistically

What MCP Makes Clear



What Servers (can) give Clients

MCP’s bread and butter!

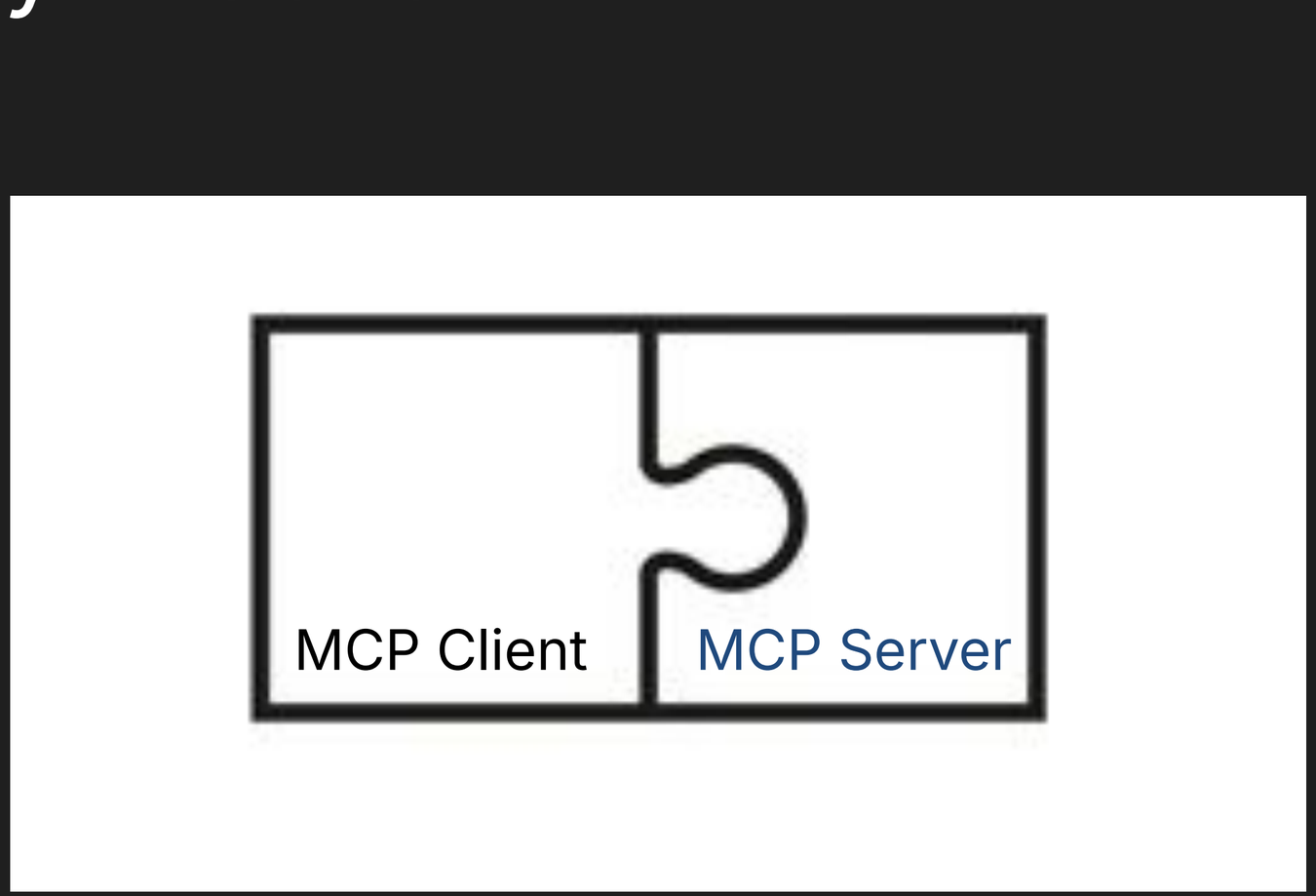
- ❏ **Tools:** functions that the model can invoke to take actions
- ❏ **Resources:** read-only data and context for the model or user
- ❏ **Prompts:** templated messages and workflows

What Clients (can) give Servers

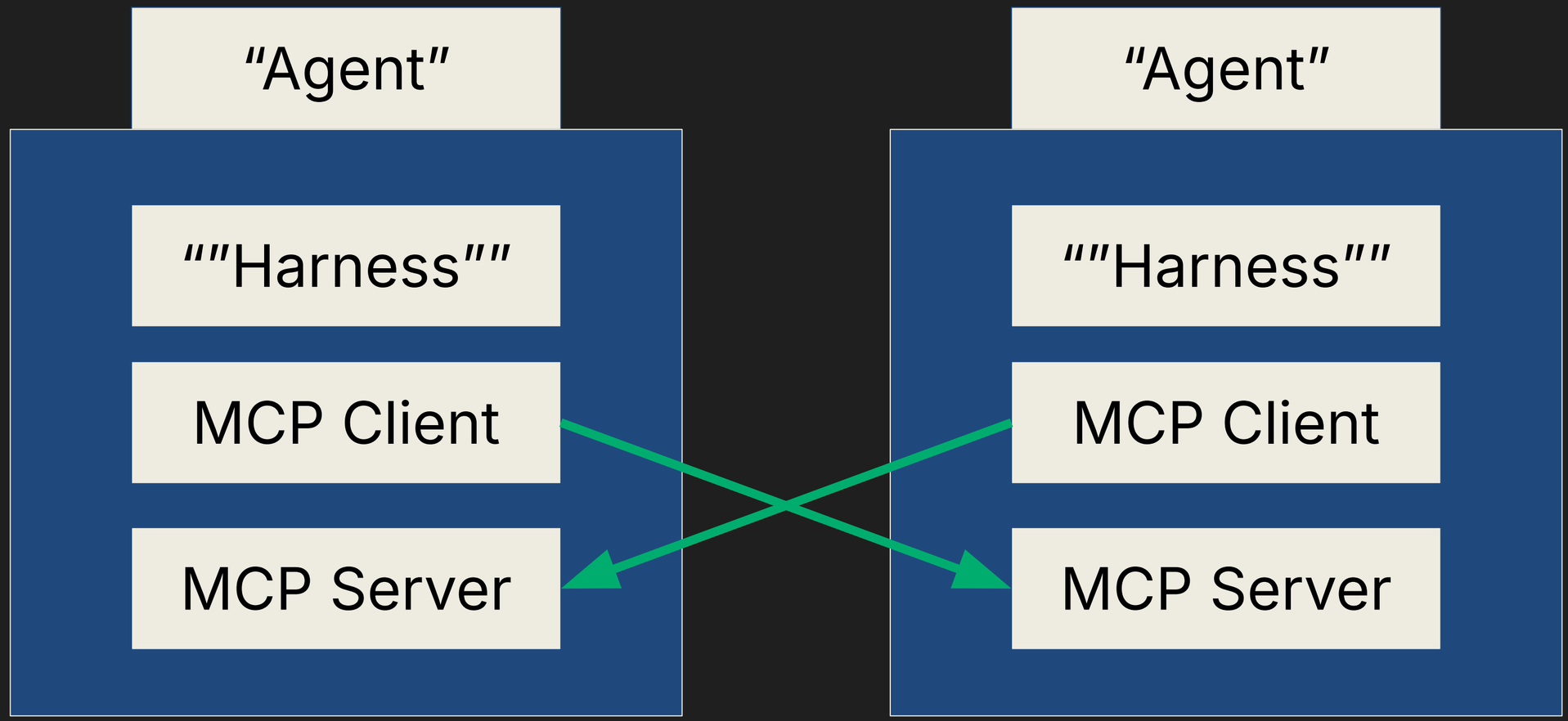
This one is a lot more unique :)

- **Sampling:** server requesting LLM completions from the client
- **Roots:** filesystem or URI boundary **hints** that tell the server where it *should* operate
- **Elicitation:** server asking the end user for additional information
- **URL Elicitation**: send user to the browser, away from the client

Asymmetry in action

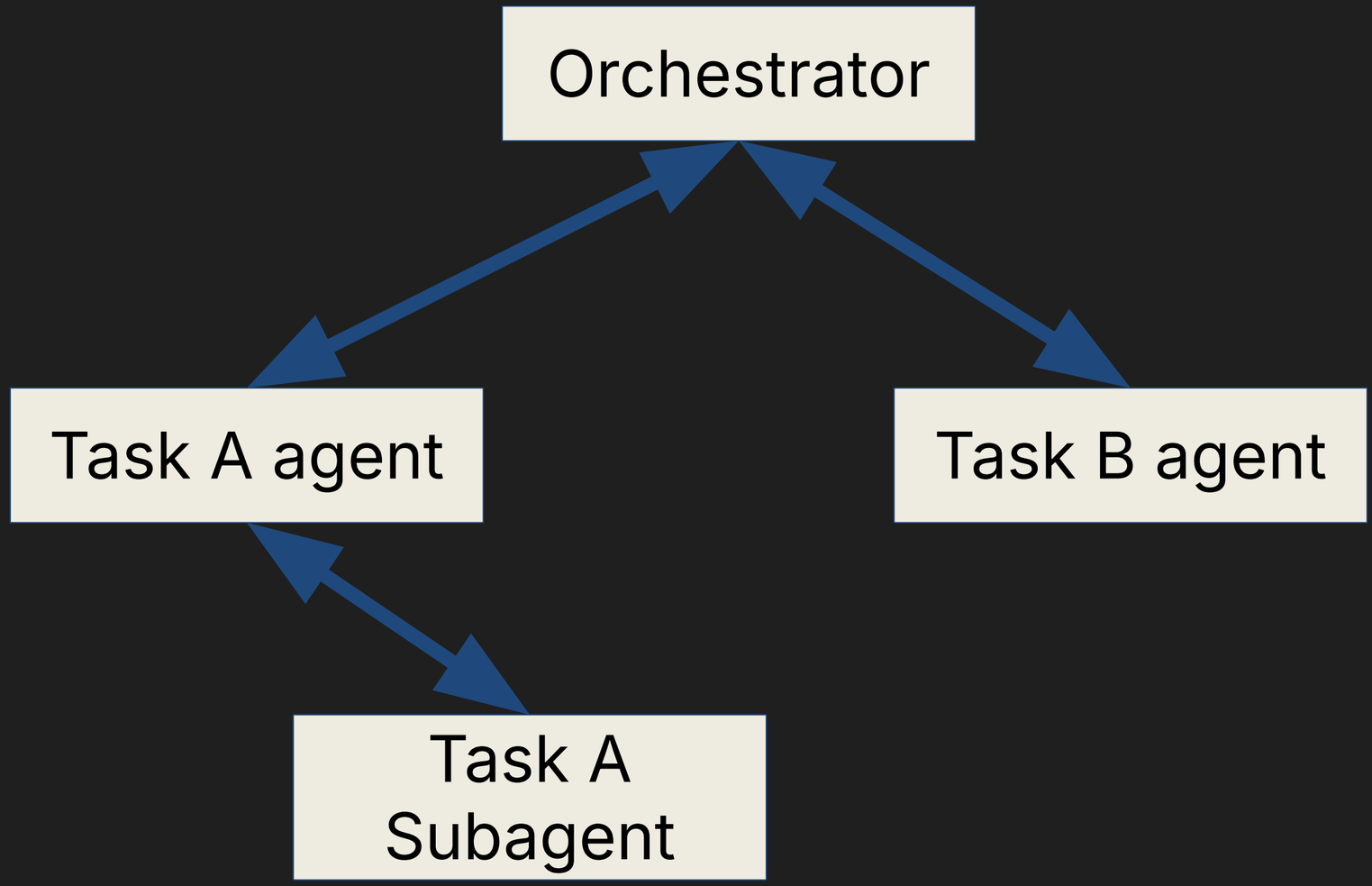


What happens if we zoom out?



Simplify...

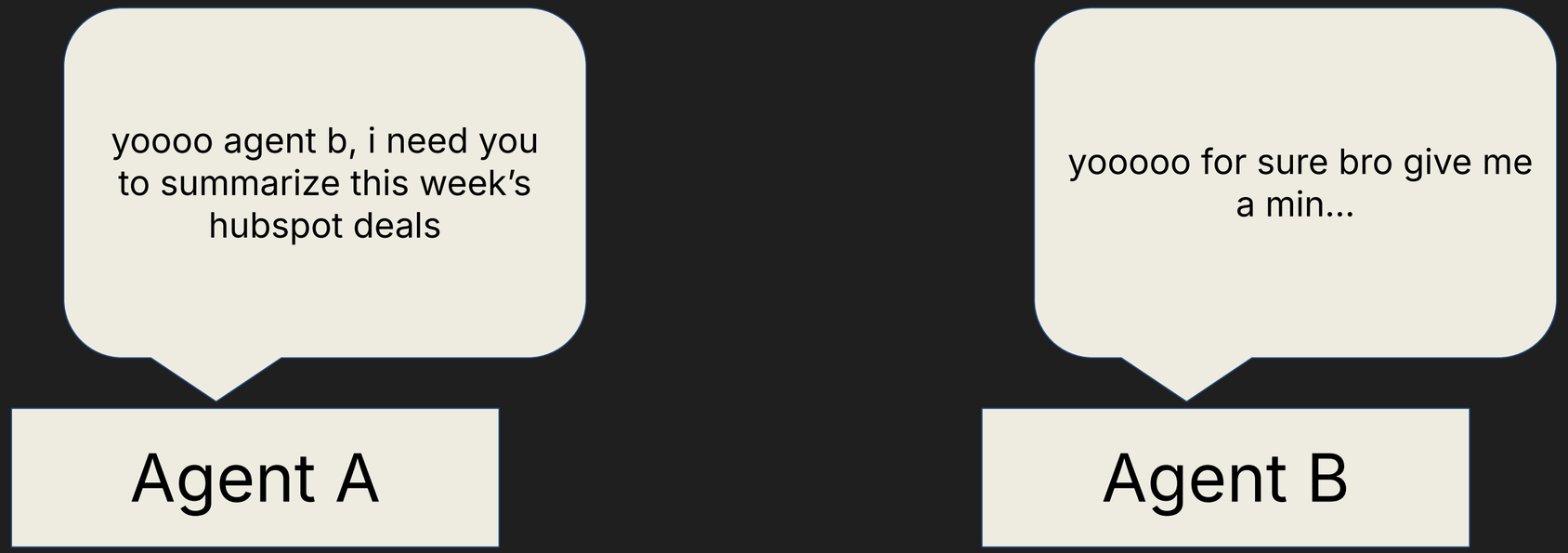
Think ~~sillier~~ bigger



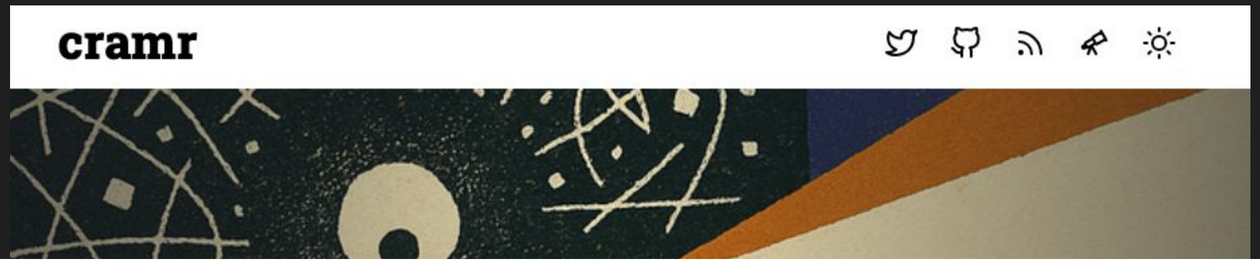
Why the heck would we do that?

Why agent-to-agent via MCP?

- ❏ Today’s “Agent-to-agent” is usually kinda client-server anyways
- ❏ Context management/isolation is like, most of the agent game
- ❏ Agents might not be in the same domain/org boundary



Agents as MCP Servers - a fun experiment



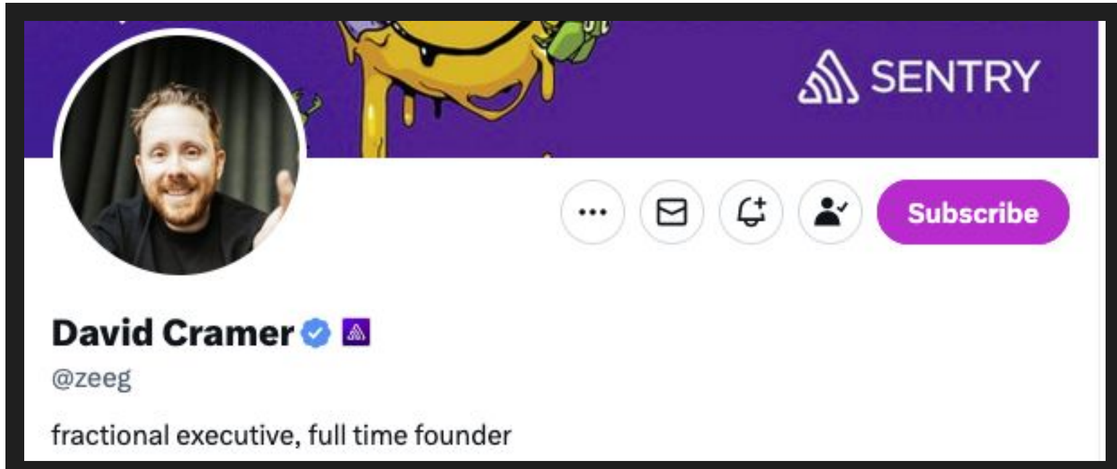

```
Use Sentry's MCP Agent to answer questions related to Sentry (sentry.io).

You should pass the entirety of the user's prompt to the agent. Do not interpret the prompt
in any way. Just pass it directly to the agent.",
```

Subagents with MCP

Oct 29, 2025 7 min read MCP

A big gripe people have with MCP is the always-on context hit. For Sentry in a basic form, this is ~14,000 tokens. That's not an enormous amount, but its also not nothing. People have been trying to come up with inventive solutions to work around it, and many have sworn off MCP for CLIs for this reason. We've also seen proliferation of poorly designed MCPs that make Sentry's not insignificant 14,000 tokens look like nothing.

What if we could eliminate the token cost of MCP and at the same time enable smarter agents?

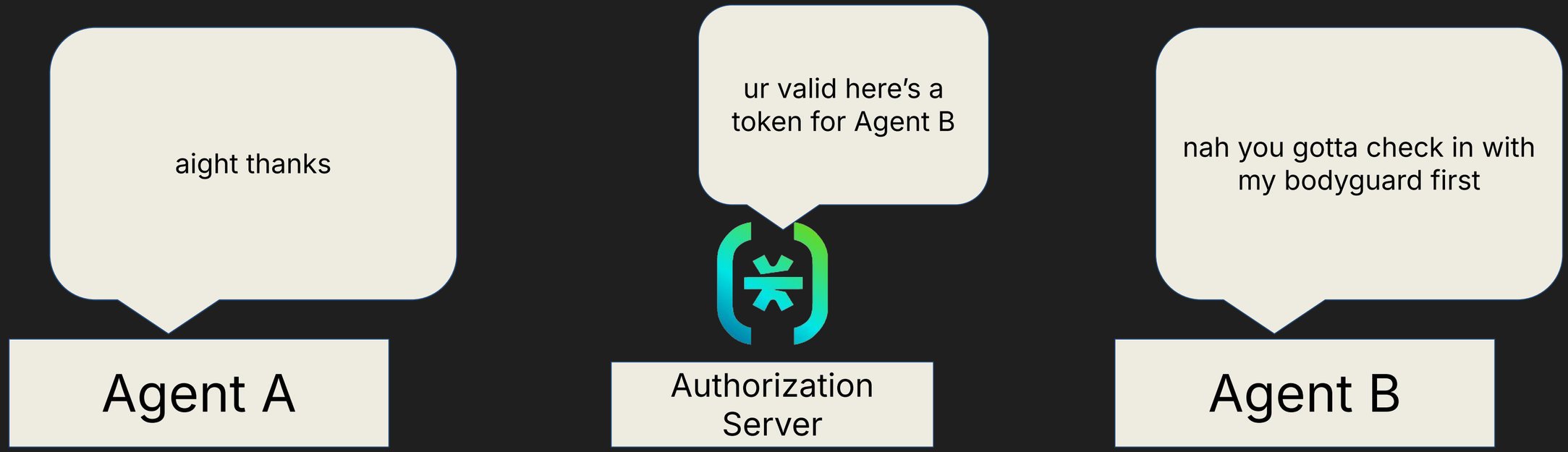


but... what about auth?

(mcp has auth?)

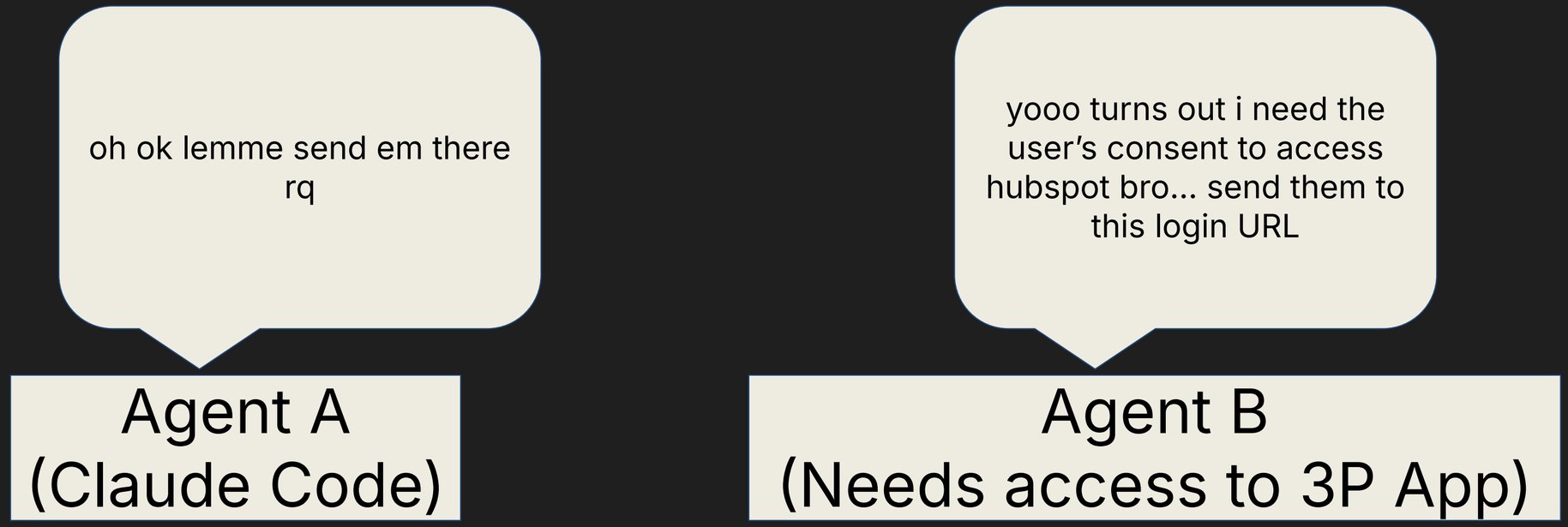
Initial Auth - Client to Server

- Clients get tokens that follow Resource Indicators (RFC 8707)
  - “This token is for this resource and only this resource (Agent B)”

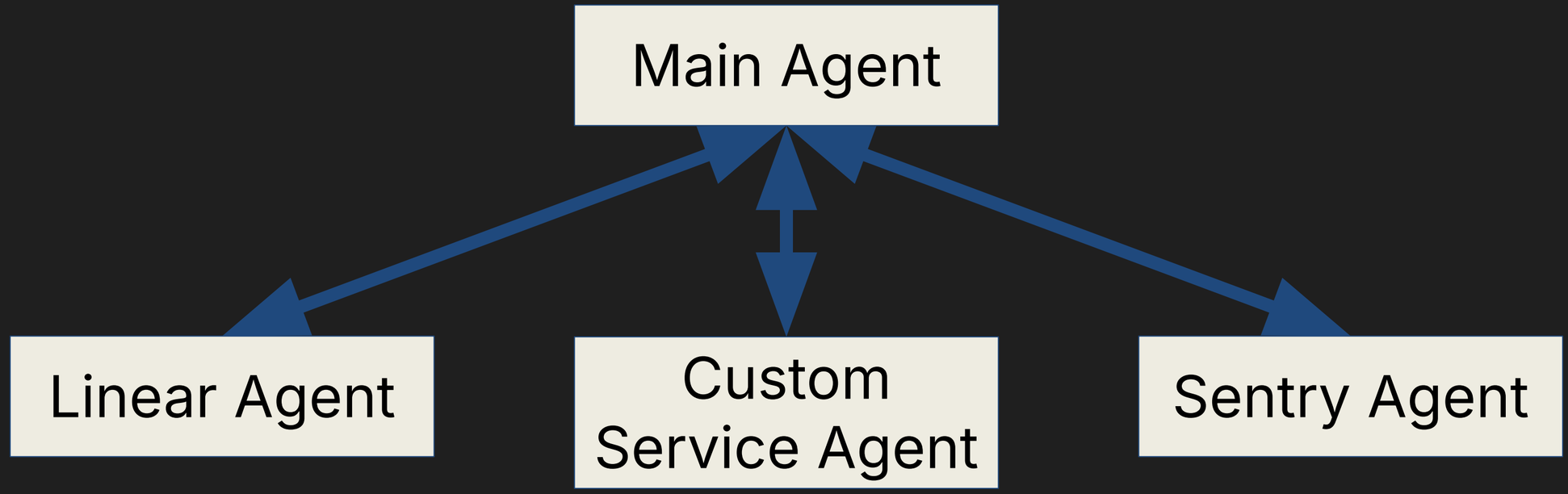


Downstream auth - URL Elicitation!

- If an MCP Server needs a third-party OAuth token/credentials:
  - It can ask the client ‘send the user here to auth!’



Let’s complicate this



This is where things get hard

- Main Agent may need to auth/consent for every other agent AND...
  - Every other agent may need auth/consent for their 3P services
  - This also assumes there is **always** a human with a client

- Agents need a good way to discover what other agents can do
- Client/server design = most stuff works 1 hop at a time
- Peer-to-peer collaboration is not a priority
- State is mostly on the client (servers just handle requests)

Conclusion

- MCP wasn’t designed for this perfectly, but it’s solving some of these problems!
  - Enterprise-Managed Authorization (MCP Auth Extension)
  - SEP-1649 Server Cards
  - SEP-1686 Tasks, Nested Task Execution (not an SEP yet, but mentioned)
  - Triggers and Event-Driven Updates

- Agents exposing some of their functionality as a server is **really cool** and unlocks a lot

- I hope you had fun!
  - Maybe follow me on X, the Everything App Formerly Known As Twitter: @rohitiwnl
  - Visit Descope at Booth G8 for Agent/MCP Auth conversations!
  - Check the MCP 2026 roadmap! It’s neat!