



Building a Workflow Engine on MCP: Orchestrating Processes with Tasks

Thursday, April 2nd, 2026 | MCP Dev Summit NA 2026

Who am I?



Open-Source
Maintainer

Donnie Adams
Software Architect

Basketball Coach

thedadams
 robot

Obot Engineering Team

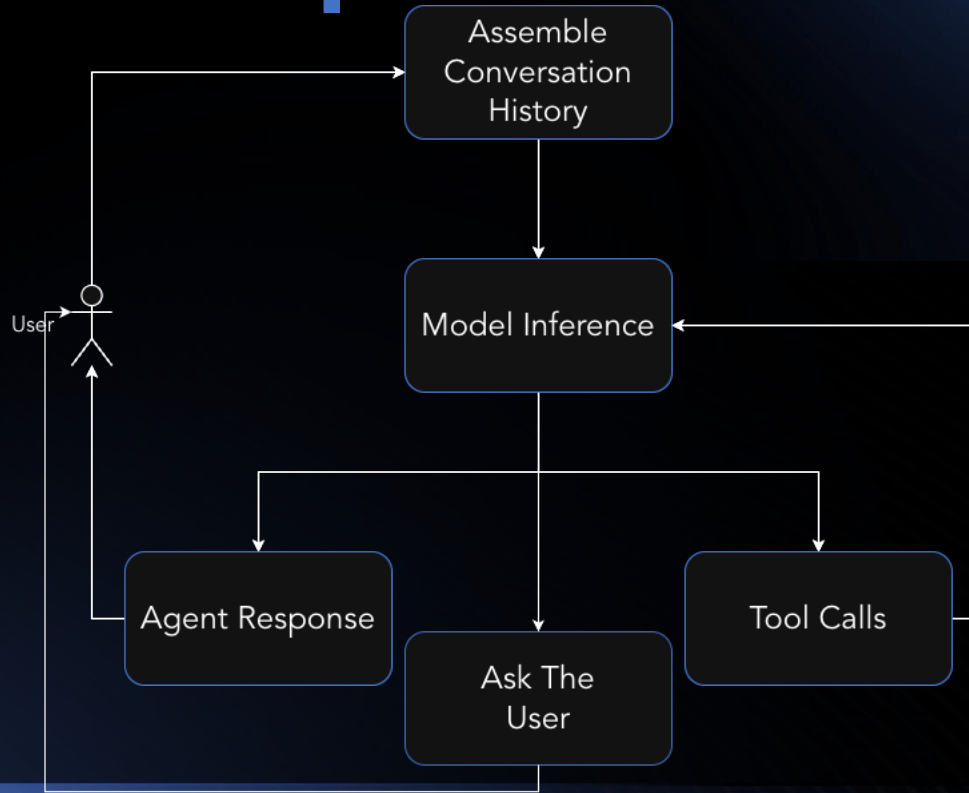
Bill Maxwell
Calvin McLean
Craig Jellick
Darren Shephard
Grant Linville
Ivy Jeong
Nick Hale
Sangeetha
Hariharan
Taylor Price

Agenda

- MCP is Everything
- Chat as an MCP Tool
- Ad-hoc Tasks
- The Real Tasks

This will be a technical medium-dive

Agent Loop

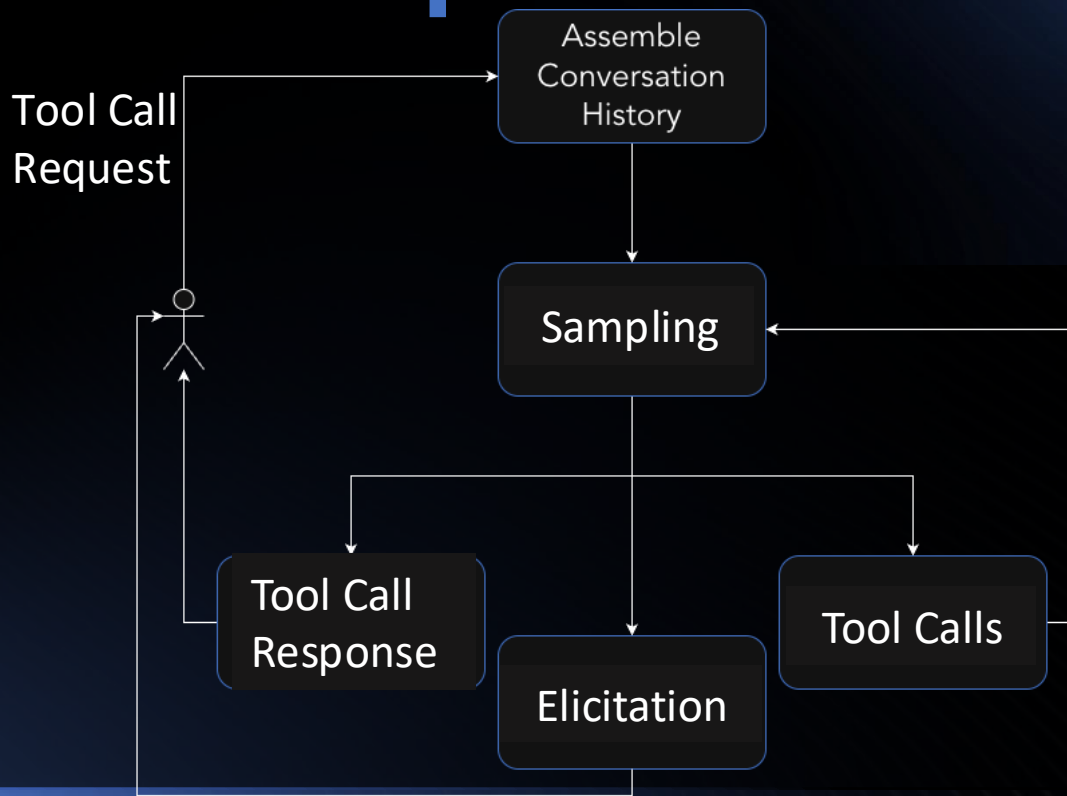


Agent Loop



MCP

Agent Loop



Demo Time

Sync Issue

Where does this breakdown?

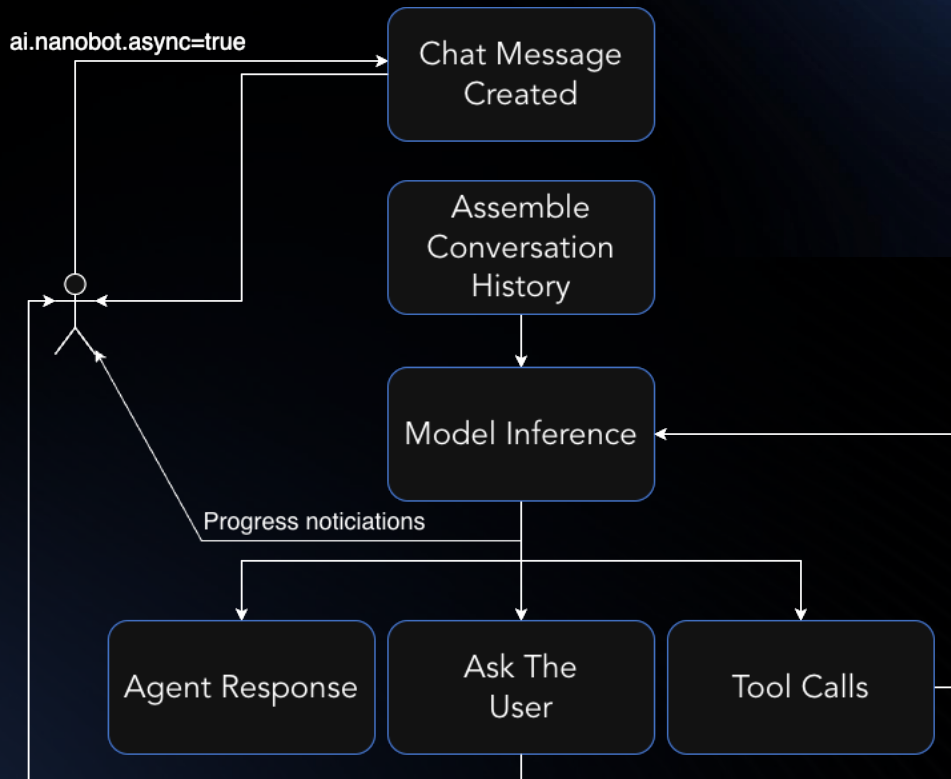
- Reliability
 - Servers restart, connections disconnect
- Timeouts
 - Long messages or agent loops could take a while

A solution to many of these issues is asynchronous tool calls.
Exactly what tasks were meant to do.

But we ran into this issue before tasks were added to the specification.

Async without Tasks

The caller can make a request to get the chat://progress resource



More Demo

Review of Tasks

Servers can declare that some of their tools support or are require to run as tasks.

Tasks are then run asynchronously.

Task start in the `working` status, and can transition to `completed`, `failed`, `cancelled` or `input_required`.

Servers can support sending task status notifications

- One net-positive here is that the task status notification contains the whole task object

Tasks natively support user input with the `input_required` status.

The Elephant

Wasn't this presentation supposed to be about running workflows as tasks?

Yes! But things change quickly at a startup.

We decided to go with the “build a workflow through chat experience” for two two main reasons:

1. We wanted the workflow experience to be interactive.
2. The chat experience we had gave us 80% of the functionality we needed.

Give the people what
they came for...

thedadams

<https://thedadams.com>

<https://github.com/thedadams/gleam-mcp>

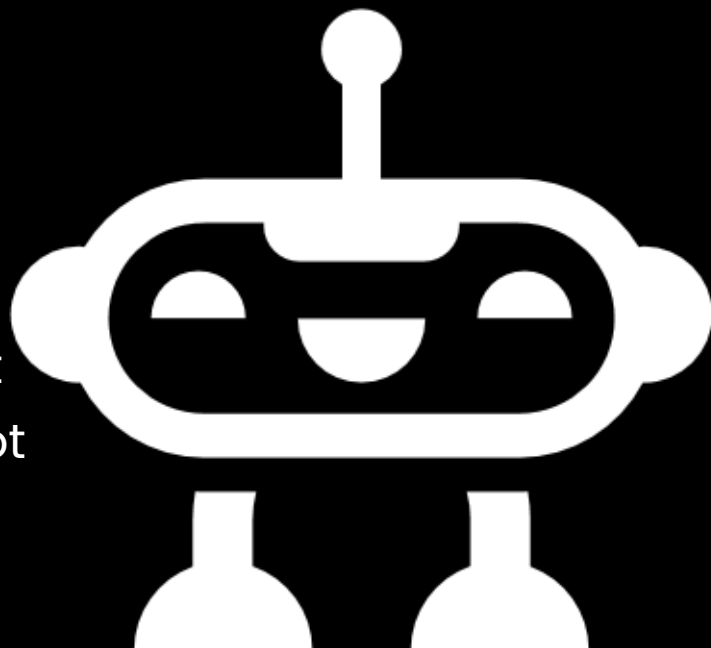
(gleam-mcp is still a work in progress)

Obot AI

<https://obot.ai>

<https://github.com/obot-platform/obot>

<https://github.com/nanobot-ai/nanobot>



Thank you

