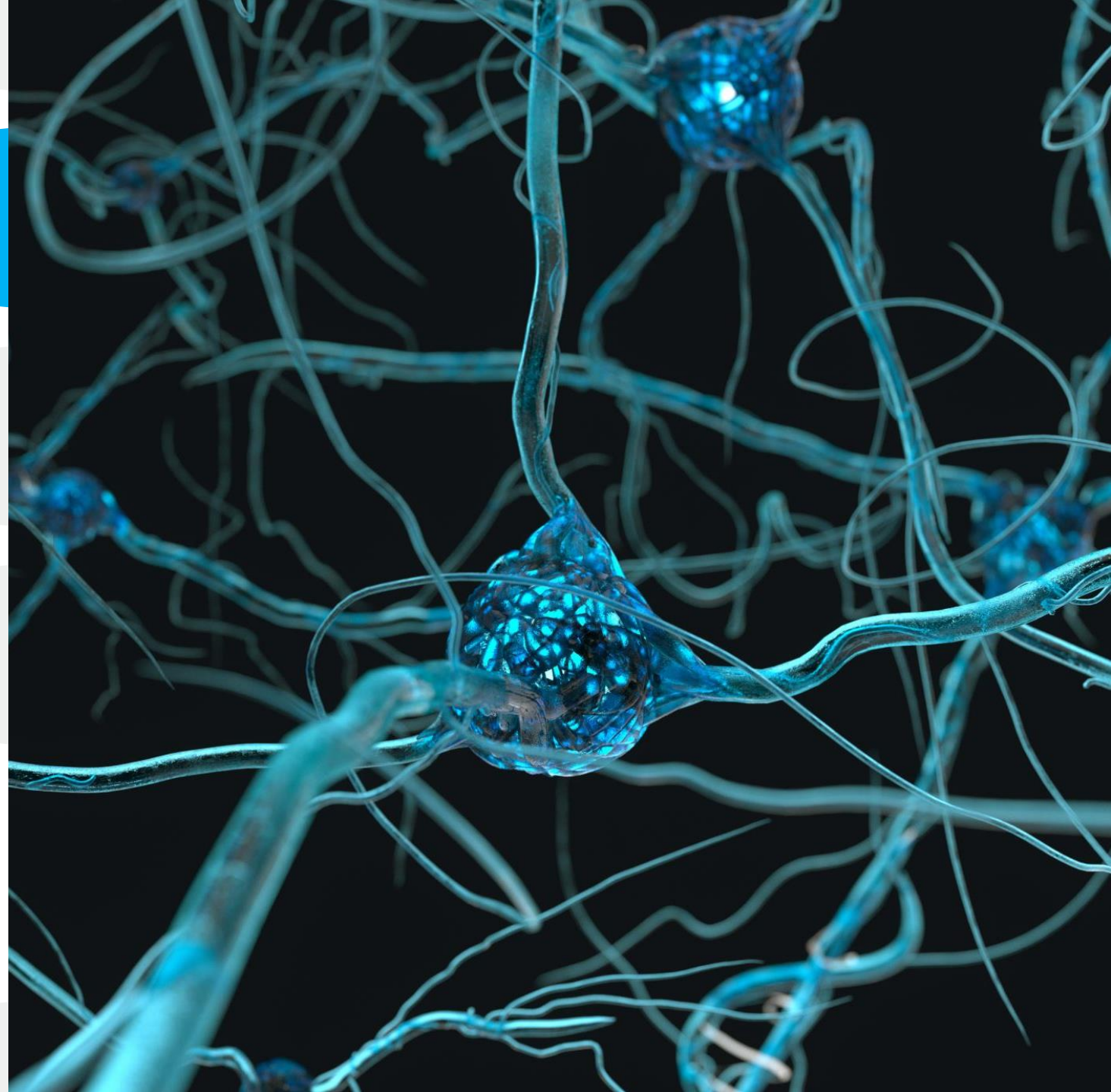
AI Task Agent Using More Flexible Generic Tools via Model Context Protocol (MCP)

Jadon Duff, Jacob Ewasko, Ioannis Limniatis, Jacob Lebkuecher



Goals and Motivations

- Motivations: Tasks can be inconvenient and repetitive. Using our AI Task Agent, users can save time on these tasks and prioritize other things.
- Goal: Enable AI to perform tasks flexibly, securely, and to be deployable across many environments.

- Existing AI Agents lack flexibility and respond statically.
- Repetitive computer tasks.
- Limited Large Language Model (LLM) knowledge and hallucinations when handling data and obscure tasks.
- Current systems are too open, and deployment can be lengthy.

Task Execution using Generic MCP Tools (Feature 1)

- The user can ask agent to perform tasks
- Agent can operate computer via tools
- MCP enables agent to autonomously complete tasks

Improved AI Knowledge via User-Supplied Documentation (Feature 2)

- The user can provide documentation
- Documentation improves agent memory and knowledge
- Reduce hallucinations & domain-specific

Versatile Deployment with Widespread Hardware Compatibility (Feature 3)

- The user can securely deploy
- Configurable deployment (air-gapped/connected)
- Updates

Novel Features

- Full environment, securely deployable (e.g. differs from OpenClaw and other attempted solutions)
- Robust knowledge improvement & memory management
- Generic MCP tools, unlike domain-specific MCP

Algorithms and Tools

Feature 1 - Task Execution using Generic MCP Tools

- MCP: Python, FastMCP, Uvicorn, Linux, Docker
- LLM: Gemma4 31B, Qwen3.8 27B, Transformers, PyTorch, OpenAI-Agents, Unsloth
- Tools: Python, FastMCP, Linux, PyAutoGUI, Computer-control libraries/tools

Feature 2 - Improved AI Knowledge via User-Supplied Documentation

- RAG: Python, OpenAI-Agents, FastMCP, text-embedding-3-large
- Embeddings: text-embedding-3-large, OpenAI, Python, Unsloth

Feature 3 – Versatile Deployment with Widespread Hardware Compatibility

- Deployment/Versioning: Docker, FastMCP, Python, Nginx, GitHub
- Subsystem consolidation: Docker, Linux, FastMCP, Gemma4 31B, Qwen3.8 27B, Unsloth, HuggingFace, PyTorch
- Air-gapped: Linux, Docker

Technical Challenges

Feature 1 - Task Execution using Generic MCP Tools

- Is the MCP properly sandboxed?
- Can we balance a small LLM with complex tool selection and reasoning?
- What about edge-cases or obscure goals?
- Can the small model sufficiently handle the tools in its context window?
- Will the user understand how and which tools to disable?

Feature 2 – Improved AI Knowledge via User-Supplied Documentation

- How do we communicate to the user that their documents must be reliable?
- What if the LLM is not good at searching context with the tool?
- How do we choose good parameters for how much, and which, context to pull from the vector store?
- How do we provide this to the model efficiently?

Feature 3 – Versatile Deployment with Widespread Hardware Compatibility

- Can we trust the user to set deployment up correctly?
- How can we implement a versioning system?
- How can we integrate LLM, MCP, and client on resource constrained systems?
- How do we ensure our system continues to function without Internet?

Milestones

Milestone 1 Itemized Tasks (Sept. 28):

- Compare and select technical tools for all subsystems
- Provide small (“hello world”) demos
- Resolve technical challenges
- Compare and select collaboration tools for software development, documents/presentations, communication, task calendar
- Create Requirement Document
- Create Design Document
- Create Test Plan

Milestone 2 Itemized Tasks (Oct. 26)

- Prototype for Task Execution using Generic MCP Tools,
- Prototype AI Agent
- POC RAG memory
- Prototype for Versatile Deployment with Widespread Hardware Compatibility

Milestone 3 Itemized Tasks (Nov. 23):

- MVP for Task Execution using Generic MCP Tools
- MVP AI Agent
- Prototype RAG memory
- MVP for Versatile Deployment with Widespread Hardware Compatibility

Collaborations

https://en.wikipedia.org/wiki/File:U.S._Coast_Guard_Auxiliary_Mark.svg

https://commons.wikimedia.org/wiki/File:U.S._Coast_Guard_Auxiliary_Seal.svg

Questions

AI Task Agent Using More Flexible Generic Tools via Model Context Protocol (MCP)