El Protocolo de Contexto de Modelo (MCP): Un nuevo estándar para la integración de herramientas en asistentes de IA

Imagina un asistente de IA que pueda acceder a tu calendario, programar reuniones, redactar correos electrónicos e incluso generar informes financieros con solo pedírselo. Esta es la promesa de los agentes de IA, y el Protocolo de Contexto de Modelo (MCP) es la clave para hacer realidad esta visión. El MCP es un protocolo abierto que estandariza la forma en que las aplicaciones, como los asistentes de IA, proporcionan contexto a los Modelos de Lenguaje Grande (LLM). En términos simples, el MCP es como un puerto USB-C para aplicaciones de IA: proporciona una forma universal de conectar los modelos de IA a diferentes fuentes de datos y herramientas.

¿Por qué es importante el MCP?

A medida que los asistentes de IA se vuelven más sofisticados, necesitan acceder a una gama cada vez mayor de datos y herramientas para realizar tareas complejas. Sin embargo, cada nueva integración requiere una solución personalizada, lo que dificulta la escalabilidad y el mantenimiento. El MCP aborda este desafío al proporcionar un estándar abierto y universal para conectar sistemas de IA con fuentes de datos, reemplazando las integraciones fragmentadas con un solo protocolo. El resultado es una forma más simple y confiable de brindar a los sistemas de IA acceso a los datos que necesitan.

¿Qué son los agentes de IA?

Antes de profundizar en el MCP, es importante entender el concepto de "agentes de IA". Un agente de IA es un sistema que puede realizar acciones de forma autónoma para lograr un objetivo específico. A diferencia de los chatbots tradicionales que simplemente responden a preguntas, los agentes de IA pueden tomar decisiones, interactuar con sistemas externos y ejecutar tareas sin intervención humana.

Arquitectura del MCP

El MCP sigue una arquitectura cliente-servidor, donde una aplicación host se conecta a varios servidores para acceder a diferentes recursos.

Componente	Descripción	Ejemplo
Host MCP	Aplicaciones que utilizan el MCP para acceder a datos.	Claude Desktop, Zed Editor, IDEs
Clientes MCP	Conectores que establecen conexiones con los servidores.	Conector de API específico

Componente	Descripción	Ejemplo
Servidores MCP	Programas que exponen capacidades específicas a través del MCP.	Servidor de archivos, servidor de base de datos
Fuentes de datos locales	Datos almacenados en la computadora del usuario.	Archivos, bases de datos
Servicios remotos	Sistemas externos accesibles a través de Internet.	APIs, servicios web

Funcionalidades del MCP

Los servidores MCP ofrecen las siguientes funciones a los clientes:

- **Recursos:** Contexto y datos para el usuario o el modelo de IA.
- **Indicaciones (Prompts):** Plantillas de mensajes y flujos de trabajo.
- **Herramientas:** Funciones que el modelo de IA puede ejecutar.

Los clientes pueden ofrecer la siguiente función a los servidores:

- **Muestreo:** Permite a los servidores solicitar respuestas a los LLM.

Aplicación práctica del MCP en la construcción de asistentes de IA

El MCP está transformando la forma en que los sistemas de IA interactúan con las fuentes de datos externas, lo que lleva a aplicaciones más eficientes e inteligentes. A continuación, se presentan algunos ejemplos notables de cómo se está utilizando el MCP en la construcción de asistentes de IA:

Asistentes de codificación

- **Sourcegraph Cody:** Al integrar MCP, Cody puede acceder a bases de código y documentación, proporcionando a los desarrolladores sugerencias e información precisa sobre el código.
- **Zed Editor:** Zed ha incorporado MCP para permitir que sus funciones de IA interactúen con varias herramientas y recursos de desarrollo.

Integraciones empresariales

- **Block:** Block ha adoptado MCP para conectar sus sistemas de IA con repositorios de datos internos de forma segura.

Consultas de datos impulsadas por IA

- **AI2SQL:** AI2SQL permite a los usuarios generar consultas SQL a través de indicaciones en lenguaje natural.

Aplicaciones de IA de escritorio

- **Claude Desktop:** Claude Desktop integra MCP para permitir que los asistentes de IA

accedan a archivos, aplicaciones y servicios locales de forma segura.

Sistemas de gestión de contenido

El MCP también tiene un gran potencial en los sistemas de gestión de contenido. Permite un acceso unificado a diferentes proveedores de almacenamiento, una API consistente para las operaciones de contenido y un manejo estandarizado de metadatos. Además, facilita operaciones inteligentes como el análisis de contenido impulsado por IA, la categorización automatizada y las capacidades de búsqueda inteligente.

Entornos de desarrollo

En los entornos de desarrollo, el MCP puede mejorar los flujos de trabajo de codificación al proporcionar herramientas integradas para el análisis de código, la generación de documentación y las pruebas automatizadas. También ofrece asistencia contextualizada al comprender la estructura del proyecto, acceder a la documentación relevante e integrarse con el control de versiones.

Llamadas a herramientas: IA accionable

Una característica clave de los agentes de IA es la capacidad de realizar "llamadas a herramientas". Esto les permite interactuar directamente con otros sistemas y ejecutar acciones específicas. Por ejemplo, si un agente de IA recibe la instrucción "Corrige el error en este código", puede acceder a herramientas para leer el archivo, analizar el código e incluso ejecutar pruebas para identificar y solucionar el problema.

Beneficios del uso de MCP para la integración de herramientas

El MCP ofrece una serie de beneficios para la integración de herramientas en asistentes de IA:

- **Acceso estandarizado a los datos:** Simplifica las conexiones a diversas fuentes de datos.
- **Modularidad:** Facilita la extensión para admitir múltiples servidores MCP.
- **Eficiencia:** Reduce la sobrecarga al optimizar la interacción entre los modelos de IA y las fuentes de datos.
- **Versatilidad:** Funciona en varios dominios.
- **Seguridad mejorada:** Incorpora funciones de seguridad para proteger los datos confidenciales.
- **Desarrollo simplificado:** Simplifica el desarrollo de aplicaciones impulsadas por IA.

Estandarización como impulsor clave: La estandarización que ofrece el MCP es fundamental para su adopción. Simplifica el desarrollo, reduce los errores y promueve la interoperabilidad entre diferentes sistemas de IA y herramientas.
Seguridad y consentimiento del usuario: El MCP se centra en la seguridad y el consentimiento del usuario en la integración de herramientas. Garantiza que los usuarios tengan control sobre el acceso a los datos y la ejecución de las herramientas.

Limitaciones del MCP

A pesar de sus ventajas, el MCP también presenta algunas limitaciones:

- **Complejidad de la integración:** Integrarlo en sistemas existentes puede requerir un esfuerzo inicial significativo.
- **Gestión de la seguridad:** Es crucial garantizar la transferencia y el almacenamiento seguros de datos.
- **Experiencia técnica:** La implementación de MCP puede requerir un conjunto de habilidades especializadas.
- **Protocolos estructurados vs. lenguaje natural:** El MCP se basa en protocolos estructurados, lo que puede limitar la flexibilidad en la comunicación en lenguaje natural.

MCP vs. otros protocolos para la integración de herramientas

El MCP se compara favorablemente con otros protocolos y marcos para la integración de herramientas:

Llamadas a funciones

Si bien las llamadas a funciones son útiles para tareas con límites claros, el MCP es más adecuado para conversaciones de varios turnos y requisitos complejos.

RAG y concatenación de prompts

El MCP ofrece una forma más estructurada y modular de proporcionar contexto a los modelos de IA en comparación con RAG y la concatenación de prompts.

MCP vs. Composio

Composio es otro marco de integración de herramientas que ofrece una amplia biblioteca de integraciones preconstruidas y una especificación de herramientas propia. Sin embargo, a diferencia del MCP, que es de código abierto, Composio tiene componentes de código cerrado, lo que puede generar preocupaciones sobre el bloqueo del ecosistema.

Implementaciones de código abierto del MCP

El MCP es un proyecto de código abierto con un ecosistema en crecimiento. Anthropic proporciona SDK para varios lenguajes de programación. Además, existe un repositorio de código abierto de servidores MCP. Algunos ejemplos de implementaciones de código abierto incluyen:

- **mcp-agent:** Un marco para construir agentes de IA que utilizan MCP.
- **Quarkus MCP Extension:** Una extensión para el framework Quarkus que facilita la creación de servidores MCP en Java.
- **Awesome-mcp-servers:** Una lista de servidores MCP de código abierto.
- **Servidores para datos y sistemas de archivos:** Filesystem, PostgreSQL, SQLite,

Google Drive.
- **Servidores para herramientas de desarrollo:** Git, GitHub, GitLab, Sentry.
- **Servidores para automatización web y de navegadores:** Brave Search, Fetch, Puppeteer.
- **Servidores para productividad y comunicación:** Slack, Google Maps, Memory.
- **Servidores para IA y herramientas especializadas:** EverArt, Sequential Thinking, AWS KB Retrieval.

Conclusión

El Protocolo de Contexto de Modelo (MCP) representa un paso significativo hacia sistemas de IA más inteligentes, conscientes del contexto e interactivos. Al proporcionar un marco estandarizado para que los modelos de IA se conecten con recursos externos, el MCP abre nuevas posibilidades para las aplicaciones impulsadas por IA en diversas industrias. A medida que el ecosistema de MCP continúa creciendo y madurando, está listo para desempeñar un papel fundamental en la configuración del futuro del desarrollo y la integración de la IA.
El "efecto dominó del MCP" se refiere a la creciente adopción del MCP por parte de las empresas y los usuarios, lo que está impulsando a otras empresas a adoptar el protocolo para no quedarse atrás. Este efecto está creando un ecosistema más amplio y robusto de herramientas y servicios compatibles con MCP, lo que a su vez está acelerando la innovación en el campo de la IA.
En el futuro, se espera que el MCP evolucione para admitir conexiones remotas a servidores, lo que ampliará aún más su alcance y flexibilidad. Además, se está trabajando en el desarrollo de un registro centralizado para el descubrimiento de herramientas, lo que facilitará a los desarrolladores encontrar e integrar las herramientas adecuadas para sus necesidades.

Síntesis

El MCP ofrece una solución estandarizada y segura para la integración de herramientas en asistentes de IA. Sus principales beneficios incluyen:
- **Simplificación del desarrollo:** Reduce la necesidad de integraciones personalizadas.
- **Mejora de la seguridad:** Asegura el control del usuario sobre el acceso a los datos y la ejecución de las herramientas.
- **Mayor eficiencia:** Optimiza la interacción entre los modelos de IA y las fuentes de datos.
- **Escalabilidad:** Permite la conexión a múltiples servidores y herramientas.

A pesar de algunas limitaciones, como la complejidad de la integración en sistemas existentes, el MCP está impulsando la innovación en el campo de la IA y se espera que tenga un impacto significativo en el futuro del desarrollo y la integración de la IA.

Obras citadas

1. docs.anthropic.com, https://docs.anthropic.com/en/docs/agents-and-tools/mcp#:~:text=MCP%20is%20an%20open%20protocol,C%20port%20for%20AI%20applications. 2. Model Context Protocol: Introduction, https://modelcontextprotocol.io/introduction 3. Introducing the Model Context Protocol - Anthropic, https://www.anthropic.com/news/model-context-protocol 4. MCP solves the automation issue with AI agents; here's how to use it - TechGig,

https://content.techgig.com/career-advice/mcp-solves-the-automation-issue-with-ai-agents-heres-how-to-use-it/articleshow/118575694.cms 5. Specification (Latest) – Model Context Protocol Specification, https://spec.modelcontextprotocol.io/specification/2024-11-05/ 6. Model Context Protocol (MCP): A Beginner's Guide - AI IXX, https://aiixx.ai/blog/model-context-protocol-mcp-a-beginners-guide 7. Model Context Protocol (MCP): The USB-C of AI Data Connectivity - HackerNoon, https://hackernoon.com/model-context-protocol-mcp-the-usb-c-of-ai-data-connectivity 8. MCP Documentation - Introduction, https://www.claudemcp.com/docs/introduction 9. Model Context Protocol (MCP): A New Standard for AI Agents | by Gokcer Belgusen | Feb, 2025 | Medium, https://medium.com/@gokcerbelgusen/model-context-protocol-mcp-a-new-standard-for-ai-agents-878a1378f41d 10. What You Need to Know About Model Context Protocol (MCP) - TypingMind Blog, https://blog.typingmind.com/what-you-need-to-know-about-model-context-protocol-mcp/ 11. Why the Model Context Protocol (MCP) Misses the Mark | by Cosmic Iron | Medium, https://medium.com/@cosmiciron/why-the-model-context-protocol-mcp-misses-the-mark-0ba107d9d248 12. Model Context Protocol (MCP) - Understanding the Game-Changer - Runloop AI, https://www.runloop.ai/blog/model-context-protocol-mcp-understanding-the-game-changer 13. Still Confused About How MCP Works? Here's the Explanation That Finally Made it Click For Me : r/ClaudeAI - Reddit, https://www.reddit.com/r/ClaudeAI/comments/1ioxu5r/still_confused_about_how_mcp_works_heres_the/ 14. Why We're All-In on MCP - Mastra, https://mastra.ai/blog/mastra-mcp 15. Model Context Protocol - GitHub, https://github.com/modelcontextprotocol 16. lastmile-ai/mcp-agent: Build effective agents using Model Context Protocol and simple workflow patterns - GitHub, https://github.com/lastmile-ai/mcp-agent 17. Implementing a MCP server in Quarkus, https://quarkus.io/blog/mcp-server/ 18. Awesome MCP Servers - A curated list of Model Context Protocol servers - GitHub, https://github.com/appcypher/awesome-mcp-servers 19. Example Servers - Model Context Protocol, https://modelcontextprotocol.io/examples 20. Engineering AI systems with Model Context Protocol · Raygun Blog, https://raygun.com/blog/announcing-mcp/