MCP: the MVP for Agentic AI Interoperability?

The world of generative AI (GenAI) continues to evolve rapidly, and new AI engines, features and capabilities are emerging more quickly than almost anyone can keep up. As AI agents emerge (agents being where AI can be given a goal, work out a plan to achieve that goal and do its best to identify tools it can use in meeting that goal), the need to streamline communication between different GenAIs and tools is increasingly important. One answer to that is an open standard called the Model Context Protocol, or MCP.[1]

As the spelled-out name would suggest, it is a *protocol* (a set of guidelines for how things should be done) to provide *context* (the circumstances around communicating) for *models* (different AI agents) acting as a go between for AI agents and resources. It was developed by Anthropic (the folks behind the Claude chatbot) but is receiving broad acceptance in the industry. Eventually, like most enabling standards, users won't have to know the term or think about how to use it, any more than we have to think about enabling TCP/IP in our operating systems to access the Internet. But, in these early days, we'll see the term MCP, need to identify and select MCP servers (the go-betweens that enable specific MCP uses) and be more in the weeds than we will in the future. (Sadly, those of us of a certain age may remember when (1960s and 70s) MCP stood for "male chauvinist pig." Techies amongst us may recall being encouraged to get an MCP credential in the '90s – "Microsoft Certified Professional").

Speaking of adoption, the Google Gemini folks just announced MCP support in their development tools, and a demo app in AI Studio is the first place I saw MCP "in the wild"; I had previously seen it referenced as part of the future Agentic AI stack, along with A2A (Google's Agent to Agent protocol, for agents negotiating costs of use) and two Apache open source specifications, Kafka and Flink.

I hope to tell you more about MCP as it becomes more easily available; it is important for financial professionals to understand the potential benefits of opening access to your files, your data, your programs and other resources to your agents, while understanding the risks of imperfect tools, especially those acting like a "black box," having that access, as well as the challenges of access rights and authorization.

My own career in standards began when I saw an advertisement in *The Wall Street Journal* talking about a new interoperability solution from Digital Equipment (DEC, later acquired by Compaq) promising the moon – they would make applications from different vendors work together as if they were developed by a single vendor. The ad trumpeted, "NAS – the perfect solution for an imperfect world." I will, however, point out the Network Application Support (NAS) did *not* take off or prove itself to be that perfect solution. That was 1990.

Not soon after, the Object Management Group (OMG) published the first CORBA specification. The Common Object Request Broker Architecture (CORBA) is a standard that facilitates communication between different software components, even if they are written in different languages and run on different platforms. It essentially acts as a middleware layer, enabling applications to work together over a network.

---

[1] https://docs.anthropic.com/en/docs/agents-and-tools/mcp.

Interoperability moved forward more with the World Wide Web Consortium's Web Services and Service Oriented Architecture, based on XML (and later JSON) with the catchy components like SOAP, UDDI and WSDL. I jumped on the interoperability train with XBRL hoping that Web Services would take off, and still look to UDDI (Universal Description, Discovery and Integration) for a model of future self-explanatory service communication.

So, now we are at MCP (in partnership with A2A and other emerging parts of the AI interoperability stack). It is early days, and I've been wrong before, but MCP looks to be something we should be monitoring for sure.