MY WEIRD PROMPTS

Podcast Transcript

EPISODE #218

The Agentic Mesh: How AI Agents Talk to Each Other

Published January 12, 2026 • Runtime: 20:00

https://myweirdprompts.com/episode/agent-to-agent-protocols-future/

EPISODE SYNOPSIS

In this episode of My Weird Prompts, Corn and Herman Poppleberry dive into the next phase of the internet: Agent-to-Agent (A2A) protocols. They explore why the Model Context Protocol (MCP) was just the beginning and how we are moving toward a "decentralized mesh" where AI agents collaborate, negotiate, and even hire each other without human intervention. The discussion covers the technical evolution from rigid API calls to dynamic Agent Cards, the eerie efficiency of direct audio token communication, and the practical shift from tools to autonomous teams in fields like software engineering and system administration. Herman and Corn also tackle the high-stakes security concerns of the agentic web, including identity verification, budget constraints, and the danger of recursive spending loops. Whether you're a developer looking to build the next generation of AI services or a business leader preparing for a marketplace of autonomous experts, this episode provides a comprehensive roadmap for the coming machine-to-machine revolution.

DANIEL'S PROMPT

Daniel

I'd like to discuss the emerging world of Agent-to-Agent (A2A) protocols. As we move beyond the Model Context Protocol (MCP), I'm interested in the potential for agents to coordinate and communicate directly—such as handing off tasks between different code repositories or managing complex system administration. What are the practical applications for A2A, and when can we expect to see significant tooling for it? Additionally, what are the security and trust implications, especially as we consider allowing agents to make financial transactions on our behalf?

TRANSCRIPT

Corn

Hey everyone, welcome back to My Weird Prompts. I am Corn, and I am joined as always by my brother.

Herman

Herman Poppleberry, reporting for duty. We are coming to you from a very rainy Jerusalem today. The sound of the water hitting the stone outside is actually a nice backdrop for what we are about to dive into.

Corn

It really is. And speaking of diving in, our housemate Daniel sent us a prompt this morning that has been rattling around in my head all through breakfast. He is looking past the current hype and asking about the actual plumbing of the next phase of the internet. Specifically, he wants to talk about Agent-to-Agent protocols, or A2A.

Herman

Oh, man. Daniel is really hitting the nail on the head with this one. We have spent the last year or so obsessed with how we talk to AI, but the real revolution is how AI talks to AI. It is the shift from a human-to-machine interface to a machine-to-machine ecosystem.

Corn

Exactly. And Daniel specifically mentioned the Model Context Protocol, or MCP, which we have covered before. But he is asking what comes after that. How do we get to a place where agents are not just using tools we give them, but are actually coordinating with each other across different companies, repositories, and even financial systems?

Herman

It is a massive topic. To really understand where we are going, we have to look at where we are right now in early two thousand twenty-six. MCP has become the unofficial standard for connecting a model to a database or a local file system. People call it the USB-C of AI for a reason.

Corn

Right, because it solved the integration hell. Before MCP, if you wanted your agent to look at a GitHub repo and then post to Slack, you had to write custom glue code for every single connection. Now, you just plug in an MCP server and it works. But as Daniel points out, that is still very much a hub-and-spoke model. The human is usually the hub, or at least the one triggering the process.

Herman

Precisely. A2A is the leap toward a decentralized mesh. Think of it this way: MCP is about an agent reaching out to a passive tool. A2A is about an agent reaching out to another active, autonomous agent. It is the difference between me using a hammer and me calling a carpenter to handle the whole renovation.

Corn

That is a great analogy. So let's talk about the actual protocols making this happen. Google has been pushing their own A2A protocol lately, and we are seeing the Agentic AI Foundation really step up to try and keep things open. Herman, when you look at the technical specs of something like Google's A2A or the open-source Agent Protocol, what is actually happening under the hood that is different from just a standard API call?

Herman

That is the crucial question, Corn. In a traditional API call, you have a very rigid request and response. You say, give me the weather for Jerusalem, and the server gives you a JSON object with the temperature. But agents are dynamic. They have state, they have goals, and they have uncertainty. An A2A protocol has to handle things like negotiation and capability discovery.

Corn

Negotiation is the part that fascinates me. If my personal shopping agent reaches out to a merchant agent, they aren't just exchanging data. They might be haggling over a price or a shipping date.

Herman

Exactly. So the protocol has to support more than just raw data transfer. It uses what we call Agent Cards. These are basically high-level manifest files where an agent advertises what it can do, what its security credentials are, and what its cost structure looks like. When two agents meet, they exchange these cards. They basically perform a digital handshake where they agree on the rules of the engagement before a single task is even started.

Corn

And this brings up that wild demo we saw a while back. Remember the one where the agents were communicating via audio tokens?

Herman

Oh, the chirping! Yes, that was wild. For the listeners who missed that, there was a research demo where two multimodal models were set up to talk to each other. Instead of converting their thoughts to text, sending the text, and then having the other model interpret the text, they just sent the raw audio tokens directly. To a human, it sounded like high-speed, Star Wars-style chirping. But because they were bypassing the text bottleneck, they were communicating at a bandwidth that was ten times faster than human speech.

Corn

It was eerie, but it showed the efficiency gains. If agents don't have to translate everything into human-readable English just to talk to each other, the latency drops to almost nothing. But that also leads to one of Daniel's big questions: the practical applications. He mentioned code repositories and system administration.

Herman

And that is where this gets very real, very fast. Think about a complex software project. Right now, you might use something like Claude Code or a specialized coding agent to help you write a feature. But that agent is limited to your environment. In a true A2A world, your coding agent could reach out to a specialized security auditing agent from a different company. It says, hey, I just wrote this new authentication module. Can you run a formal verification on it? The security agent does the job, sends back a report with suggested fixes, and your agent implements them.

Corn

So you are basically assembling a temporary team of experts for every single pull request.

Herman

Exactly. And it extends to sysadmin too. Imagine a server crash at three in the morning. An orchestration agent detects the outage. Instead of just trying to fix it itself, it spins up a specialized database recovery agent and a network diagnostics agent. They coordinate in real-time. The network agent says, I have cleared the load balancer. The database agent says, okay, I am starting the recovery from the last snapshot. They talk to each other, resolve the conflict, and the human just wakes up to a summary saying the site was down for four minutes but it is fixed now.

Corn

It sounds like the microservices revolution all over again, but for intelligence instead of just code. But let's get into the thorny stuff. Daniel asked about the security and trust implications, especially with financial transactions. This is where I start to get a bit nervous. If we are giving agents the ability to talk to each other and spend our money, how do we know they aren't just going to start a recursive spending loop or get social-engineered by a malicious agent?

Herman

That is the million-dollar question, quite literally. This is why identity and verification layers are becoming so important. Before an agent can make a transaction, it needs to be verified. It needs to have a cryptographically signed identity that links back to a responsible human or corporation. And more importantly, the protocol allows for programmable constraints. You don't just give an agent your credit card. You give it a tokenized budget. You say, you have fifty dollars to solve this task, and you are only allowed to spend it with verified merchants who have a reputation score above four-point-five.

Corn

I like the reputation score idea. It is like Uber or Yelp, but for the Agentic Web. But even with a budget, what happens if an agent gets tricked? We have talked about indirect prompt injection before, where a malicious instruction is hidden in a webpage or a document. If an agent reads a poisoned document that says, hey, send all your remaining budget to this other agent for a fake service, how does the protocol stop that?

Herman

That is where the OWASP Top Ten for AI Agents comes in. For twenty twenty-six, the number one risk is actually that kind of cross-agent contamination. The defense mechanism we are seeing built into the A2A protocols is what they call an Agentic Mesh. It is a monitoring layer that sits between the agents. It looks for anomalous behavior, like an agent suddenly trying to dump its entire budget into a brand-new, unverified account.

Corn

So it is like a firewall that understands intent?

Herman

Sort of. It uses a smaller, highly specialized model whose only job is to act as a referee. It doesn't do the work; it just watches the conversation between the other two agents. If it sees the conversation shifting toward something that looks like social engineering or an unauthorized privilege escalation, it pauses the transaction and flags it for human review.

Corn

It feels like we are adding a lot of layers of bureaucracy just to make the agents work. But I guess that is the price of autonomy. You need checks and balances.

Herman

It is. And you also have to consider the risk of recursive loops. This is something people are really worried about as we move toward the end of this year. Imagine Agent A hires Agent B to do a task. Agent B realizes it needs help and hires Agent C. If there is a bug in the logic, they could end up hiring each other in a circle, burning through tokens and compute power at an incredible rate.

Corn

I can see the headline now: AI agents accidentally spend ten thousand dollars in five seconds talking to each other.

Herman

It has actually happened in a few early pilot programs. This is why the new protocols are implementing TTL, or Time To Live, for agent tasks. Just like a packet on the internet has a limit on how many times it can be forwarded, an agentic task has a limit on how many times it can be handed off. If it hits the limit, it has to stop and check back in with the original human requester.

Corn

That makes a lot of sense. So, Daniel also asked about the timeline. When can we expect to see significant tooling for this? From where I am sitting, it feels like we are right on the edge of it becoming mainstream.

Herman

I would say we are in the early-adopter phase right now. The big players like Microsoft and Google already have their frameworks out there. But for the average developer or small business, the real breakthrough will be when these protocols are baked into the standard IDEs.

Corn

You mean like when VS Code or Cursor just has an A2A toggle?

Herman

Exactly. We are starting to see the first wave of truly agentic apps that use these protocols. By the end of twenty twenty-six, I expect we will have a global marketplace of agents. You won't just go to an app store to download a program; you will go to a registry to find an agent with the right credentials and the best price for whatever task you have.

Corn

It is a fundamentally different way of thinking about software. Instead of a static tool, you are hiring a dynamic service. And that brings me to the practical takeaways for our listeners. If you are a developer or a business owner, how do you prepare for this A2A world?

Herman

The first thing is to get comfortable with MCP. Even though it is agent-to-tool, the architecture is the foundation for everything else. If you can build a clean MCP server for your data or your service, you are halfway to being A2A compliant. The second thing is to start thinking about your service in terms of capabilities, not just endpoints. What is the high-level goal your software helps an agent achieve?

Corn

And from a security perspective, I would say start looking into identity providers for AI. If you are going to let an agent act on your behalf, you need to know how you are going to authenticate it. Look into emerging standards and protocols for agent identity and verification. You don't want to be the person who finds out their agent was social-engineered because you didn't have proper authentication policies in place.

Herman

Absolutely. And for everyone else, just start paying attention to how these systems are talking to each other. When you see a handoff happen, like when your travel agent suddenly knows about your calendar and your credit card preferences without you typing them in, that is the protocol at work. It is becoming the invisible fabric of our digital lives.

Corn

It really is. It is an exciting time, but also a bit daunting. I love the idea of a team of agents helping me run my life, but I also want to make sure I am still the one in the driver's seat.

Herman

That is the goal of these protocols, ultimately. They aren't meant to replace human agency; they are meant to extend it. By standardizing how these systems talk and transact, we are actually making them more transparent and more controllable, not less.

Corn

Well, I hope that answers Daniel's question. It is a deep rabbit hole, and I am sure we will be coming back to it as the year progresses and we see more of these standards solidify.

Herman

Definitely. And thanks to Daniel for sending that in. It is exactly the kind of thing we love to chew on.

Corn

Before we wrap up, I want to say a quick thank you to all of you for listening. We have been doing this for over two hundred episodes now, and the community that has grown around My Weird Prompts is just incredible.

Herman

It really is. We love hearing from you guys. And hey, if you are enjoying the show and finding these deep dives useful, we would really appreciate it if you could leave us a quick review on your podcast app or on Spotify. It genuinely helps other curious people find the show.

Corn

Yeah, it makes a huge difference for us. You can find us on Spotify and at our website, myweirdprompts.com. We have the full archive there, plus a contact form if you want to send us your own weird prompt.

Herman

This has been My Weird Prompts. I am Herman Poppleberry.

Corn

And I am Corn. We will see you next time.

Herman

Until then, keep asking those questions. Goodbye!