AI POWER HOUR · COMMUNITECH × MOMENTUM · JUNE 22, 2026

It's Who's Running the Loop.

Momentum Product Co. · momentumproductco.com

Every booth at every conference says "agentic" now. Most of it is just packaging. A true agent has four ingredients — a defined job, written instructions, real context, and a trigger that makes it run without you. Get those right and you have an agent. Miss the trigger and you have a saved assistant. Miss the context and you have a fancy chat. Underneath is every question the room asked — answered live.

THE FOUR INGREDIENTS — EVERY AGENT NEEDS ALL FOUR

01
Job.

One specific outcome the agent owns. Not "help me with email" — "draft a reply to every new client inquiry by 8am."

02
Instructions.

How it does the job, plus what the output looks like. "Color-code the canvas red/yellow/green" beats "give me feedback."

03
Context.

The reference material it needs every time — resume, meeting brief, VIP list, scoring rubric. No context, no agent.

04
Trigger.

What makes it run. Schedule, event, or button. This is the bit "AI assistants" skip — and the bit that makes it real.

QUESTIONS ANSWERED LIVE IN THE ROOM

Q. What's the actual difference between a chat, a saved assistant, and a true agent?

It's who's running the loop. Chat = you re-explain every time, nothing persists. Saved assistant = chat plus saved instructions, but it waits for you. True agent = reaches across context AND acts on a trigger. Pulls live info, uses your files, saves output, runs the work instead of waiting.

Q. What's RAG, and when do I want it instead of an agent?

Retrieval-augmented generation. Dump documents in one place, query the corpus. Best for large content lookup — RFP responses, knowledge bases, internal docs. If you just need answers based on data you already have, you don't need an agent. RAG is cheaper and simpler.

Q. In the free plan, do I just copy-paste the prompt into a new chat each time?

That works. Cleaner: save it as a Claude skill and call the skill from any chat with one line. Or paste the instructions into a project so they persist. Auto-triggering (on schedule, file drop, email) needs Cowork or Claude Code on a paid surface.

Q. Can I put my agent's instructions inside a Claude project?

Yes — and you should. Open a project, drop your context file in, paste the four ingredients into the custom-instructions box. Every chat in that project runs against them. Persistent context without re-pasting.

Q. How do I actually set up the automatic trigger?

In Cowork or Claude Code, build the workflow once then say "turn this into a scheduled task." It creates a task that wakes up on a trigger — Monday 8am, every 3 hours, on file drop in a folder. Each run is a fresh session with no memory of the chat, so the prompt has to be self-contained.

Q. Do scheduled tasks have to live inside a project?

No. Projects help when you want the agent to reference the same files and instructions every run. For something like meeting prep, a project keeps context tight. For a one-shot scheduled job, no project required.

Q. Is my data safe when I paste business ideas into Claude?

Depends on the plan. Free → your inputs may train models. Pro ($20/mo) → you can opt out of training. Teams & Enterprise → closed off. Don't put proprietary or commercializable work into a free plan. Testing a real business idea? Get at least Pro.

Q. Can I package my agent prompt as a reusable Claude skill?

Yes. Tell Claude "package this into a skill called Lean Canvas Reviewer." It builds the skill from the conversation. After that, you call the skill from any chat with one line — no re-pasting the prompt every time. Borrow open-source expert skills too; they're just files.

Q. How do I get a visual output instead of a wall of text?

Add "create me a visual representation" or "create me an artifact" to the prompt. Better still: build one artifact format you love (e.g. a color-coded scorecard) and reuse it across every run, so you have a comparative framework across ideas.

Q. When I run a business idea through it, what's the highest-value output to ask for?

Don't just ask for feedback — ask for the top 3 experiments that would validate the idea. A paid 2-hour taster. 20 problem interviews. One channel test. Those become your action list, not a comment thread.

Q. How much does it cost when an agent is hitting APIs all day?

Claude Code charges per token on API calls. Solo person, tight scope: not bad. Scroll a full site every run and hit every API: gets out of hand fast. Scope tight, turn it off when you're not using it, and write results to a file so you're not hoarding context.

Q. I'm burning time on Gemini's free tier — is that just because I picked the wrong tool?

Sometimes. The right answer isn't always a better prompt — sometimes it's $20/mo for Pro to get materially more context and controls. But also: if the workflow isn't actually agent-shaped, no tool-switching fixes that. Make sure you really need an agent first.

WHEN YOU DON'T NEED AN AGENT AT ALL

You need RAG.

Large content repository, answers from existing data. Cheaper and simpler than an agent — query the corpus, get the answer.

You need workflow automation.

Sequential steps where the path is known: A → B → C. No autonomous decisions. Just wire the steps; you don't need an agent's judgment.

You need a one-off chat.

Quick question, no repetition, no triggers. Open Claude, ask, close. Agents are for the work you'd hand off — not the question you'd Google.

GUARDRAILS FOR AGENTS THAT TOUCH THE WEB

Least privilege.

Only connect to sites and tools it actually needs. Resist the urge to wire up everything "in case."

Read-only by default.

Wherever possible, no write access. Read, summarize, draft — let you do the send.

Human in the loop.

Anything that sends, spends, or deletes goes through you. Gmail's connector only drafts by design.

Treat the web as untrusted.

Pages can carry hidden prompt injections. Don't let an agent crawl the open web on its own.

Scope tight.

Don't scroll someone's whole site when you need one page. Turn the agent off when it's idle.

Save, don't hoard.

Write results to a file instead of accumulating everything in running context. Cheaper, faster, sharable.

An agent isn't AI getting smarter — it's you not running the loop.