

FROM SYSTEMS TO LEVERAGE

An advanced guide to **AI agents**.

You run systems that work without you. Now compose them: chain specialist agents, build your own workflows, feed them your data, and run the whole thing like a small ops team.

The advanced leap: from using agents to composing them

Beginner was one task. Intermediate was systems. Advanced is composition.

The beginner move was handing over one task with a finish line. The intermediate move was building systems that run without you asking each time. The advanced move is composing: chaining specialist agents together, feeding them your own data, and building workflows bigger than any single prompt.

Think of it as going from playing an instrument to conducting. One agent is a talented employee. A composed team of agents, wired to your data and your tools, is a department.

One giant prompt

"Do my market research, write the report, make the slides, and email it." One agent, one breath, fingers crossed.

A composed workflow

A researcher gathers. A planner organizes. A maker drafts. A checker reviews. Each one has a tight brief and hands off a finished artifact.

The big idea

Beginners delegate tasks. Intermediates build systems. Advanced users compose teams. The leverage compounds at every level.

Chain agents together: specialists beat generalists

One agent doing everything is a bottleneck. Split the work.

A single agent juggling research, analysis, writing, and review does each one worse than a specialist would. The advanced pattern is the assembly line: small agents, each with one job and a tight brief.

The handoff rule makes it work: each agent passes the next one a finished artifact, not a chat log. "Here are the 10 sources, ranked" beats "here is our whole conversation, good luck."

● The Researcher

Gathers raw material: sources, prices, options, facts. Never judges, just collects.

● The Planner

Turns raw material into structure: outlines, comparisons, shortlists. Never writes the final draft.

● The Maker

Produces the artifact: the draft, the sheet, the summary. Works only from the plan, never from raw chaos.

● The Checker

Reviews the maker's output against your rules before anything ships. Cheap insurance. Expensive to skip.

Start with two

You do not need four agents on day one. Split your hardest workflow into researcher and maker first. Add the planner and checker once the handoffs feel natural.

Build your own workflows: the recipe pattern

Name the steps, define the handoffs, set the checkpoints.

A workflow is a recipe your agents follow every time. Write it once, run it forever. Every good recipe has four parts.

1

Name it. "The Friday client report." A name turns a prompt into a thing the team can run.

2

List the steps in order, with what each step produces. Gather, organize, draft, review.

3

Define the handoffs: what exactly passes from one step to the next. Artifacts, not conversations.

4

Set the checkpoints: where it pauses and waits for you. Money moves, public output, and judgment calls always checkpoint.

Copy-paste: turn anything into a workflow

"Turn this into a reusable workflow: [describe the outcome]. Break it into steps. For each step, tell me what it needs as input and what it produces as output. Mark the checkpoints where you must pause and wait for my approval."

The big idea

A workflow you write once and run fifty times is fifty tasks you never delegate again.

Feed it your data: APIs, docs, and knowledge bases

An agent is only as good as what it can reach.

The biggest jump in output quality comes from inputs, not prompts. Give your agents your world. Your docs: paste the manual, upload the price sheet, share the policy. Your data: the spreadsheet, the notes, the customer list. Your live tools: connected through APIs so the agent can check the real numbers instead of guessing.

This is where plugmyagent earns its keep: the catalog lists 450+ free APIs your agents can call, and the MCP server directory wires tools straight into your agent with copy-paste config.

Generic answers

An agent answering from training data alone. Confident, outdated, and wrong about your prices.

Your answers

The same agent with your price sheet, your return policy, and your shipping API. Now it answers like your best rep, at 2am, instantly.

The data rule

Never let an agent guess what it could look up. If the answer exists in your docs or your tools, connect it before you prompt.

Guardrails and evals: trust, but verify

Advanced users do not hope. They check.

The more your agents do on their own, the more you need ways to catch mistakes early. Three mechanisms, all plain English.

1

Show your work. "Cite your sources for every claim." If it cannot point at the source, it does not ship.

2

Spot checks. Audit a random sample every week. Five minutes, once a week, catches drift before it becomes damage.

3

Golden examples. Keep 3 to 5 cases where you know the right answer cold. When you change a workflow, run it against those cases first. If it fails a golden example, it does not go live.

The eval habit

Before you trust a new workflow, run it on three cases where you already know the answer. New workflows earn trust. They do not start with it.

Run it like an ops team

Give the team structure and it stops feeling like a toy.

At this level you are managing, not prompting. Borrow the structure real teams use.

● Roles

Who does what: your researcher, your planner, your maker, your checker. Write it down once.

● The standup

Your morning briefing becomes the team's roll call: what ran overnight, what needs you, what is next.

● Escalation rules

The advanced approval habit: "Handle it unless money moves, a customer is involved, or you are less than 80% sure. Then bring it to me with your recommendation."

● The weekly review

Fifteen minutes every Friday: what worked, what broke, one improvement for next week. Systems that do not get reviewed quietly rot.

The big idea

You are not using an AI anymore. You are running a small ops team that happens to be software. Manage it like one.

Know the costs: tokens, limits, and picking your model

Leverage is cheap. Waste is not.

Every request costs compute, and the bill is real whether you see it or not. The advanced skill is spending it where it matters.

1

Big models for judgment calls. Negotiations, strategy, anything where nuance matters.

2

Small, fast models for grunt work. Sorting, summarizing, reformatting, first drafts.

3

Batch the routine. One scheduled sweep beats twenty ad-hoc questions.

4

Respect rate limits. Space out heavy jobs. Run the big ones overnight.

The 80/20 rule

80% of the value comes from cheap, consistent automation. Save the heavyweight model for the 20% where judgment actually matters. Most people do the reverse.

Four advanced plays

Copy, paste, customize. Each one composes multiple agents.

1

The research swarm. "Break [question] into 4 angles. Research each angle in parallel. Then merge everything into one brief: the consensus, the disagreements, and the 3 sources I should read myself."

2

The content pipeline. "Take my [rough notes] and run the pipeline: planner makes the outline, maker drafts all three formats [post, email, one-pager], checker enforces my voice rules [short, no fluff, no hype]. Show me only the final drafts."

3

The data watchdog. "Watch [spreadsheet / API / price feed]. Alert me the moment anything moves more than [X] or breaks the pattern. Include what changed and a draft of the response."

4

The personal CRM. "After every meeting I log, record: what was decided, what I promised, and the follow-up date. Every Friday, send me the open loops: what is overdue, what is due next week, and who I have not followed up with."

What's next

Pass it on

Know someone still doing everything themselves? Send them guide 001. New to Muse? Enter the code **DGC4EX** in Settings within 48 hours of joining: you get 1 billion tokens, and so does the person who invited you.

Keep going

This is guide 003.

You have gone from chatbot to coworker to systems builder to leverage. You delegate tasks, build systems, and compose teams of agents wired to your data.

The free API catalog, the MCP server directory, and the component library are live now. Next up: setup walkthroughs with keys, auth, and copy-paste config blocks that work the first time.

PLUGMYAGENT.COM // © 2026 HIDDENINWEST