The best agentic systems

don't have the most agents

They have the right ones.

Here's how to know which is which 👉

swipe →

Not everything needs to be agentic

And that's okay.

A perfectly good API wrapped in an "agent" just adds latency and debugging nightmares.

swipe →

Before you decompose, ask:

- 🧠 Does it need autonomous decision-making?
- 💬 Will natural language reasoning help?
- 🔗 Is coordination genuinely complex?

"No" to all three?

Use a function. Skip the agent.

swipe →

The Decomposition Mindset

5 steps from problem to purposeful agents

1. Define the Goal State (be specific!)
2. Map every Decision Point
3. Find Natural Boundaries
4. One sentence per agent
5. Map the Handoffs

Decomposition is thinking. Diagrams come after.

swipe →

5 Patterns That Work

⛓️

Specialist Chain

Sequential experts: A ➔ B ➔ C

🔌

Router + Specialists

Classify intent, delegate to expert

✏️

Generator + Critic

Create, evaluate, iterate

📊

Parallel Gatherers

Fan out, synthesize back

👥

Human-in-the-Loop Checkpoints

Autonomous work with human approval gates

Pick the pattern that fits. Don't force it.

swipe →

The #1 Mistake

COMMON ANTI-PATTERN

One Agent Per Tool

An agent that only calls one API isn't autonomous.
It's just a wrapper with extra latency.

Tool access is a capability, not a reason to exist.

If it doesn't make decisions, it's not an agent.

swipe →

Start with the problem.

Decompose with purpose.

Then diagram with clarity.

📝 **Read the full framework**
myyearindata.com

👥 **Follow me on LinkedIn**
Scott Bell

⚡ **DailyDatabricks.Tips**

📰 **Databricks.News**