

The AI Commerce Architecture Guide

What to Build. What to Trust. How They Work Together.

A practical framework for deciding what to build, what to trust, and how modern commerce AI fits together.

To Build or Buy, That is The Question

You've probably had the same thought many commerce teams are having today.

You've watched someone build a working application with Claude Code in an afternoon. You've connected an MCP to Amazon Seller Central and watched an AI analyze campaign performance in minutes. Maybe you've even started building internal tools yourself.

It naturally raises the question: Why do we still need a commerce platform?

It's the right question. But it's also the wrong way to think about the problem. The future isn't a choice between building with AI and buying software. It's understanding which parts of your commerce operation AI should own, and which require systems purpose-built for production.

AI excels at analysis, content creation, workflow automation, and internal tools. Those are areas where teams should absolutely build.

Production commerce execution is different. Managing bids, pricing, promotions, inventory-aware advertising, and cross-signal decisions isn't primarily an AI problem. It's a reliability problem. These systems operate continuously, against imperfect data, changing marketplace conditions, and decisions where mistakes immediately affect revenue, margin, and market share.

The gap between a successful prototype and a production platform isn't measured in lines of code. It's measured in years of operational knowledge. That distinction is becoming increasingly important as more brands experiment with AI-generated commerce applications.

This guide introduces a framework for deciding where agentic AI belongs, where specialized commerce systems still create the most value, and how the two work together. It explains why the future isn't one model running your business, but a coordinated architecture where agentic AI, shared context, market intelligence, and execution systems each play a distinct role.

The goal isn't to discourage building. It's to help you build the right things. AI has dramatically lowered the cost of creating software. It hasn't changed where competitive advantage comes from. That still depends on knowing which capabilities are uniquely yours and which are better delivered by production-ready platforms.



Dani Nadel
President & COO, Feedvisor

Executive Summary

AI has fundamentally changed how commerce teams build software. In a matter of hours, brands can now create dashboards, automate workflows, analyze campaign performance, and build internal applications that would have required months of engineering only a year ago.

That changes an important question. Not whether something can be built, but whether it's something worth building.

It's a reasonable question. The challenge is that the answer isn't a single yes or no. It depends on which part of your operation you're talking about.

The central idea is simple: not every AI application carries the same operational risk. **Some applications create insight, improve workflows, or build internal tools. Others make autonomous business decisions with real money at stake.** The first group rewards experimentation. The second demands reliability.

The pages that follow introduce a practical framework for deciding where AI belongs, where specialized commerce systems belong, and how the two work together.

You'll find:

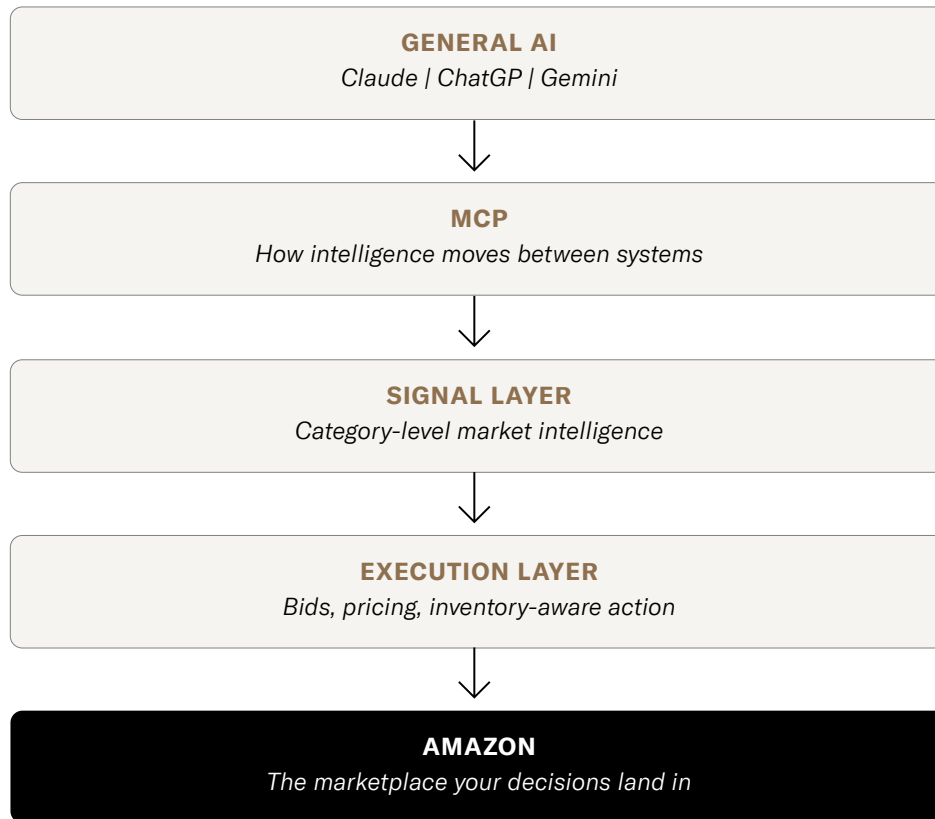
- The Layers of The AI Stack: what to build and what to trust
- Why Execution is Different: the three conditions that break homegrown automation on Amazon
- The Hidden Signal Layer: where proprietary market intelligence, not the AI itself, creates advantage
- The Future Architecture: how AI, MCP, Signal Layers, and Execution Layers work together

We've spent more than fifteen years building production commerce systems like those described in this guide, and we have watched many smart teams rediscover the same lesson the hard way: building software is easier than operating software.

AI has fundamentally changed what's possible. But it hasn't changed what reliable execution requires. That's the framework you would want in hand before writing a single line of code.

The AI Commerce Stack: How the Systems Connect

Every layer has a different job. The future belongs to organizations that know which system owns which decision.



Read the rest of this guide with that stack in mind. General AI reasons. MCP connects. The signal layer sees what your account alone cannot. The execution layer acts safely, reliably, in real time. Confuse the jobs, and you build the wrong thing well.

The Complexity Iceberg

You've seen what Claude, ChatGPT, and other AI tools can do. You've watched someone build a working app in twenty minutes. Maybe you've already connected an MCP to your Amazon Seller Central or Vendor Central account and asked Claude to pull your campaign data. It worked. It felt like magic.

And the natural question follows: why am I paying for a platform when I could build something myself? It's the right question. Whether the goal is reducing costs, gaining control, or building something tailored to your workflows, the impulse makes sense. But there's a meaningful difference between what AI tools can do well today and operating a production commerce system.

The difference isn't whether AI is capable. It's the operational risk attached to the decision it's making. Generating a report and changing bids across thousands of products are both AI applications. They are not the same operational problem.

The challenge is that the first few days of experimentation only reveal the simplest part of the problem, the happy path.

Building a prototype is relatively easy. Connecting an MCP to your Amazon seller account, pulling campaign data, generating recommendations, and creating dashboards can all be accomplished surprisingly quickly.

What takes years isn't building the first version. It's everything that comes next: edge cases, Amazon-specific nuances, API instability, retroactive attribution changes, conflicting signals, data-quality issues, policy shifts, and countless operational exceptions.

Those problems rarely show up in a proof of concept. They appear months later, in production, when reliability matters and every decision carries real business consequences.

That's the complexity iceberg. The prototype is what you see. Production is everything beneath the waterline.

10% above the waterline

The initial build:
connect, pull, recommend,
visualize.

90% below it

The accumulated
knowledge, infrastructure,
and operational resilience
required to keep the system
working day after day.
That's where most of the
complexity lives.

"Every production platform is really a collection of lessons learned from failures its customers never have to experience themselves. That's what lives below the waterline."

- Dani Nadel, President & COO, Feedvisor

The deeper you move below the waterline, the less the challenge becomes writing code and the more it becomes engineering resilience.

The Framework: The Four Layers of AI

Every AI initiative lives in one of four layers. Often, the mistake isn't building, but building the wrong layer.

DON'T BUILD	BUILD		
Execution Bids, Pricing, Promotions, Guardrails <i>If it's wrong: real money, real time, often irreversible</i>	Internal Tools Dashboards, Calculators, Automations, Slack alerts <i>If it's wrong: a noisy channel, not lost margin</i>	Content Listing copy, A+ content, Ad creative, Review synthesis <i>If it's wrong: caught in review before it ships</i>	Analysis Data patterns, Insights, Reports, Opportunity ID <i>If it's wrong: a second look, not a lost sale</i>

The three lower layers are where AI creates leverage: fast to build, inexpensive to iterate, forgiving when they're wrong, and always reviewed by a human. The execution layer is different in kind, not degree. It operates in real time, with real money at stake. Everything that follows is the proof.

Start Where AI Genuinely Wins

This isn't an anti-AI argument, quite the opposite. The tools are remarkable, and for the right use cases they're transformative. Every brand should be using AI for the work below. The question is what comes after.

Analysis and interpretation

Give a model a data export and ask what patterns it sees across the last 90 days of campaign performance. It will surface insights, anomalies, and opportunities in minutes that might otherwise take an analyst hours.

Content and creative work

Writing listing copy, generating A+ content variations, drafting ad copy, turning raw data into reporting narratives.

Strategic reasoning and decision support

"Given these margin constraints and this competitive landscape, what should my advertising strategy prioritize?" is a question AI handles well.

One-time data tasks and lightweight tools

Reformatting bulk files, merging sources, cleaning spreadsheets, and building simple dashboards, calculators, and workflow automations.

Why Autonomous Execution on Amazon Is a Different Problem

Three things make autonomous commerce execution fundamentally harder than the demos suggest.

01 The operating data isn't deterministic

Amazon's data is messier than most realize. APIs don't always return accurate information, reports lag by hours, metrics contradict one another, inventory counts can be wrong, and attribution shifts retroactively. A mature platform distinguishes a glitch from a real signal because it's comparing against millions of similar data points across thousands of accounts. A system built on one account's data has none of that context, and a bad API response becomes a bad decision executed at speed.

02 LLMs reason probabilistically, not deterministically

They generate plausible outputs, not verified ones. For content and analysis, that's manageable. For a real-time system putting real money at stake, one wrong read cascades: bids adjusted on faulty logic, budgets shifted on a bad read, pricing triggered by an imagined pattern. Unlike a human analyst, the system won't flag when its own logic breaks down.

03 The operating environment never stands still

Amazon changes fee structures, ad placements, Buy Box logic, category requirements, and API behavior, often without notice. A solution provider learns of these through direct communication with Amazon, often before the change hits your account. A homegrown system has no rep to call, gets no advance notice, and can't file a ticket. It responds after the fact, if it detects the change at all, and by then tens of thousands in spend or margin may already be lost.

That's the iceberg in practice. The initial build handles the visible 10%. The daily reality of Amazon operations is the other 90%, and it surfaces one failure mode at a time, on its own schedule, usually at the worst possible moment. You can't prompt your way to that knowledge.

A Short Story About a Pricing Assistant

No customer names. Just a scenario that plays out more often than you'd think.

STEP 1: SETUP

A brand connects Claude to Seller Central and builds a pricing assistant.

STEP 2: TRIGGER

It works beautifully. Then Amazon has a lag in inventory reporting.

STEP 3: THE MISTAKE

Claude reads the low inventory numbers as excess inventory, and drops price to move units that were never actually sitting still.

STEP 4: ACCELERATION

Advertising, tuned to chase conversions, accelerates behind the lower price.

STEP 5: CONSEQUENCE

The real inventory, already tight, disappears in days.

STEP 6: THE COST

Margins collapse.

The bestseller goes out of stock at its lowest price of the year.

THE OUTCOME

Nothing in that chain was a bug. Every step was a reasonable inference from the data on hand. The model behaved exactly as designed.

The failure was asking a probabilistic reasoner to own a real-time, real-money business decision based on data that lied. This scenario plays out every week across Amazon. The only variable is the scale of the loss.

The Decision Matrix

One misconception from the current AI wave is that the future belongs to a single model running your business: connect an MCP, expose your data, let the AI handle the rest. That’s not where this is heading. The future of commerce operations is a coordinated stack of specialized systems, each optimized for a different job and a different tolerance for error. Generating an insight and acting on it are not the same problem.

LAYER	RISK	BUILD WITH AI?	EXAMPLES
Analysis	Low	Yes*	Data patterns, insights, reports, opportunity ID
Content	Low	Yes	Listing copy, A+ content, ad creative, review synthesis
Internal Tools	Low	Yes	Dashboards, calculators, automations, Slack alerts
Execution	High	No, needs a domain system	Bid management, pricing, inventory-aware advertising, guardrails

**With a caveat, a human still catches the mistakes.*

The dividing line is simple. If a mistake costs a few hours of rework, AI is often the right answer. If a mistake can cost money, margin, or market share in real time, you need a system engineered specifically to prevent it.

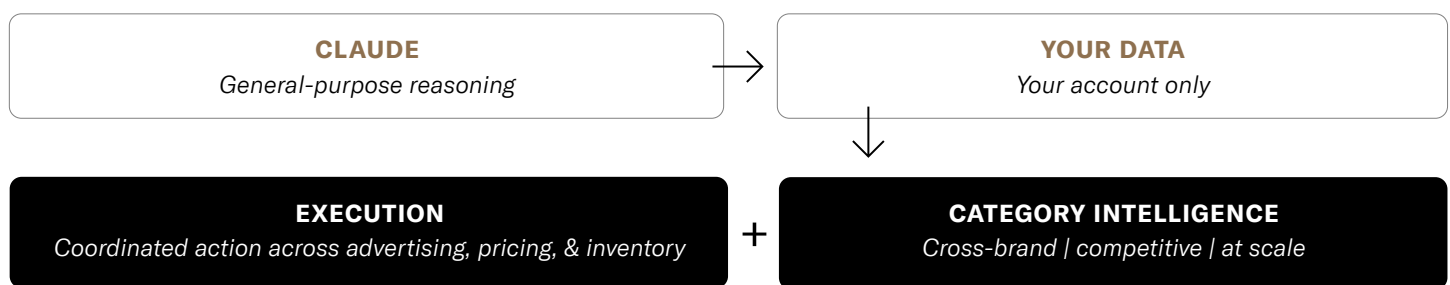
*“AI isn’t replacing commerce platforms.
It’s redefining what commerce platforms should do.”*

- Dani Nadel, President & COO, Feedvisor

The Hidden Signal Layer

**“Every Brand Has AI.
Almost None Have the Signal Layer.”**

AI is excellent at analyzing the data you have. But the most valuable analysis requires signals most brands don't have: competitive pricing movements across your category, cross-brand benchmarks, demand patterns visible only at scale, and pattern recognition from observing thousands of brands over years.



Your Seller or Vendor Central data tells you what happened in your account. *Category-level intelligence tells you why, and what to do next.*

That signal layer sits between the AI tools brands build for themselves and the execution systems that act on decisions. It's the difference between a model that reasons well about your account and a system that knows what your whole category is doing right now. Almost anyone can now stand up an AI model. Almost no one can manufacture the signal layer, because it isn't built from your data. It's built from the market's.

YOUR DATA TELLS YOU

“ROAS dropped 15% this week.”

“Conversion rate is declining.”

“Ad spend is up but revenue is flat.”

CATEGORY INTELLIGENCE TELLS YOU

“Your top competitor cut price 12%, shifting demand across the category.”

“3 new sellers entered your keywords with lower prices and higher review counts.”

“Category search volume dropped 18%; you're spending more to capture a shrinking pool.”

Real Examples

The framework gets concrete fast. Here is how four common builds sort themselves the moment you ask what a mistake costs.

“I built a Claude dashboard.”

A wrong number costs a second look, not a sale. Human stays in the loop.

Great. Build it.

“I built autonomous pricing.”

One bad read at 2 AM can trigger a race to the bottom across your catalog.

Don't.

“I built Slack alerts.”

An internal tool. Worst case is a noisy channel, not lost margin.

Great.

“I built inventory-aware bidding.”

Real money, real time, on Amazon data that lags and contradicts itself.

Don't.

The framework isn't about what AI can build. It's about the cost of being wrong.

- If a mistake costs an afternoon of rework, build it.
- If a mistake can cost margin, market share, or customer trust in real time, trust it to a system engineered for exactly that job.
- **Green** means a mistake costs you an afternoon. **Red** means it costs you the quarter.

Before You Build Anything

The most useful thing this guide can leave you with isn't an answer. It's the order of the questions. Before you build anything, ask these four, in this order:

1 Which layer am I building?

3 Who owns it in production?

2 What happens if it fails?

4 Is this where my engineering team creates the most value?

Only then ask the question everyone starts with: **“Should we build it?”**

The Future Is Coordination

Once you separate reasoning from execution, a different architecture emerges. The future won't be one AI replacing every system. It will be specialized intelligence connected through open standards, shared context, and coordinated execution.

The framework throughout this guide has been vendor-neutral on purpose because it applies regardless of which AI models or commerce platforms you use. But it naturally leads to one question:

“So how do we think about solving this?”

This doesn't mean brands should have less control over their data and intelligence. Quite the opposite. The future we see is not AI replacing platforms, or brands surrendering control to a single model. It's intelligence, systems, and autonomous execution working together, each where it creates the most value.

The companies that gain the greatest advantage from AI won't build everything themselves, nor will they rely on a single model to run their business. They'll build an architecture where every layer does what it does best: General AI for reasoning and productivity, MCP and open standards for connectivity, a proprietary signal layer that captures business context, and specialized execution systems that can act safely, reliably, and autonomously.

Where MCP Fits

Everyone is asking the same question right now: “Do you have an MCP?” It's the right question, but it's often asked for the wrong reason.

MCP changes how intelligence moves between systems. It does not eliminate the need for execution systems. Connecting a model to your data is plumbing, not a decision engine.

The signal layer and the execution layer still have to exist, and still have to be reliable, whether or not an MCP is carrying messages between them.

Putting the Architecture to Work

Agentis

Agentis wasn't built to replace general-purpose AI. It was built to provide what general AI lacks: a commerce-specific signal layer and a deterministic execution layer that understands how Amazon actually operates.



AGENTIS™

Agentis is our implementation of this architecture: the Signal Intelligence Layer and Execution Layer, unified in a single platform built for Amazon. It continuously observes the market, reasons across category intelligence, advertising, pricing, inventory, promotions, demand, and competition, turning isolated recommendations into coordinated autonomous decisions that improve business performance, not just campaign metrics.

The question isn't whether AI can pull your campaign data and suggest a bid change. It can. The question is whether you'd trust it to make that change autonomously, at scale, across your entire catalog, at 2 AM, when Amazon's API just returned bad data, your competitor just dropped price, and your inventory file hasn't synced in six hours.

The future isn't choosing between AI and commerce platforms. It's building an architecture where every layer does what it does best.

See the architecture mapped to **your business.**

Connect your account and Agentis runs a full analysis, identifying where AI belongs, where it doesn't, and where you're leaving margin on the table. **Let us run your free diagnostic: feedvisor.com/agentis**

How Coordinated is Your Business?

Most brands can tell you their ROAS. Far fewer can tell you how pricing, inventory, advertising, and profitability are influencing one another.

See where coordination gaps are limiting growth and discover how Agentis helps brands connect every signal that drives profitable growth.



"Market share up, TACoS down, and our team finally focused on strategy."

STORELLI

THE AGENTIS ADVERTISING ENGINE

- | | | |
|--|---|--|
| ✓ FULL-FUNNEL
Every channel connected.
More conversions from the same budget. | ✓ DISCOVERY-LED
The terms that move the needle. Product by product. | ✓ PLACEMENT-DRIVEN
Precision bidding that maximizes every placement. |
| ✓ SIGNAL-AWARE
Bids that respond to pricing, inventory, and competition instantly. | ✓ SPEND-OPTIMIZED
Money dynamically moves to where it earns most. | ✓ PROFIT-FOCUSED
TACoS is the metric. Profit is the goal. |

Your margins tell the real story.

Let Agentis show you.

See a live demo built around your products, your competitive landscape, and your margin structure.

No slides. No generic walkthrough. A real conversation, not a sales pitch.

TRY AGENTIS TODAY