# Why Anthropic Just Gave Away Their Secret Weapon (And What It Means for Enterprise AI)

The War of AI Titans has just started and how can we can take the advantage of it.

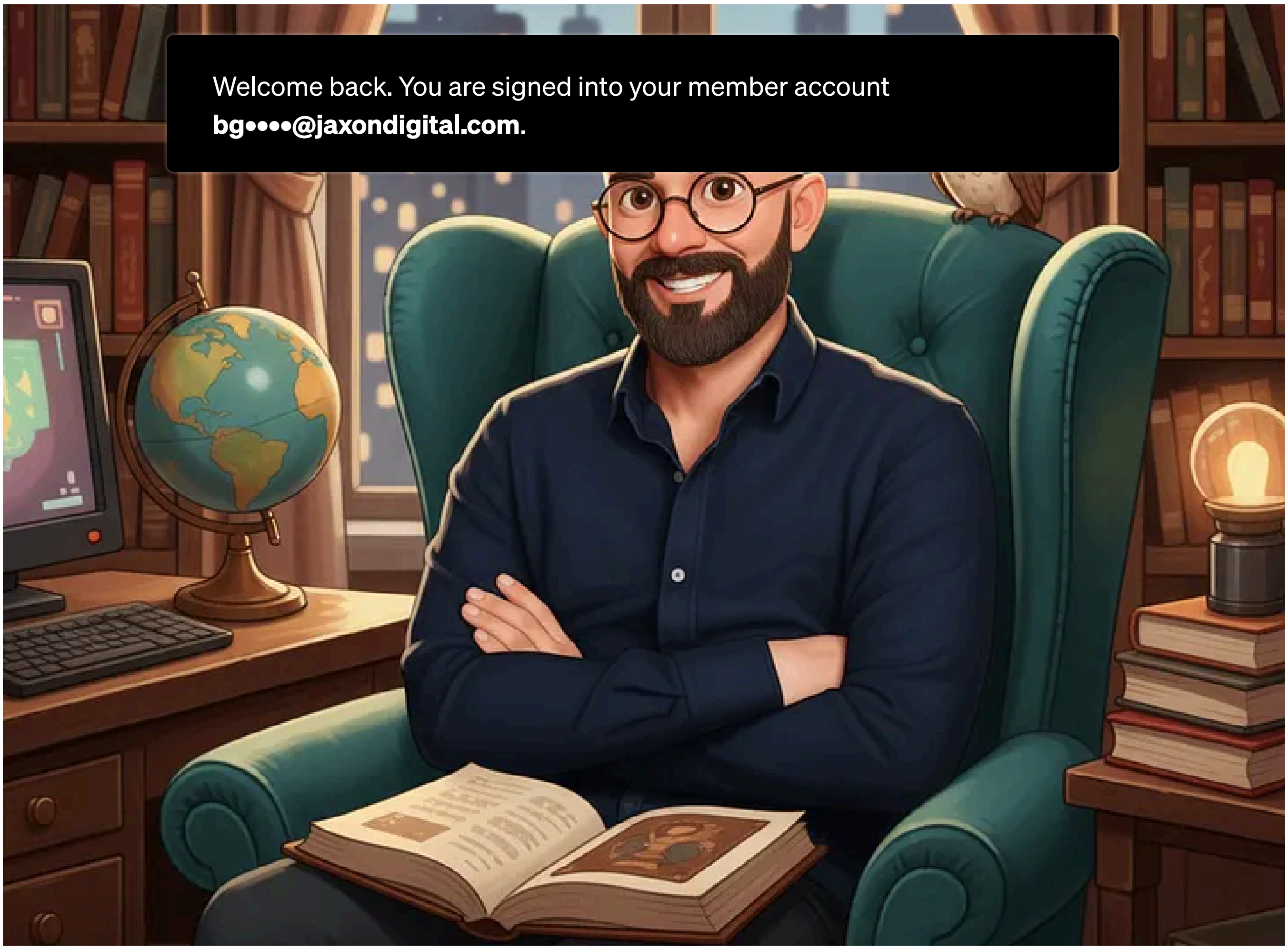
14 min read · 1 day ago

Reza Rezvani Following

Listen Share More

This is my weekly Friday Give-A-Way …

Last Wednesday, I sat across from a CTO at a Series A startup. He'd just killed their AI agent project after burning through $170,000 and six months of engineering time. The agent worked perfectly in demos — impressive enough to show investors.



A New Chapter of LLM and AI Competition (Anthropic vs. OpenAI)

But in production? It crashed after ninety minutes when context windows overflowed. The checkpoint system they built from scratch couldn't handle edge cases when the agent encountered unexpected file structures. And their permission controls created a Catch-22: either too restrictive to be useful, or too permissive to be safe.

*"We're back to square one,"* he told me. *"Except now we're six months behind our roadmap and half a million dollars poorer."*

Two days later, Anthropic announced something that made me immediately think of that conversation: the Claude Agent SDK. They're releasing the exact infrastructure stack they use to build Claude Code — context management, checkpointing, sub-agents, permission systems — all of it. Free. Public. Available to everyone.

**My first reaction:** *"This makes no business sense."*

**My second reaction, ten minutes later:** *"Wait. This is brilliant."*

**Because this isn't about AI models anymore. This is about who controls the layer where eve** **es while Anthropic**

. . .

### What Happened While Everyone Watched ChatGPT

Here's the market shift the headlines missed.

Between 2023 and mid-2025, Anthropic captured 32% of the enterprise LLM market while OpenAI's share fell from 50% to 25%. That's not a minor fluctuation — that's a complete market reversal in roughly eighteen months.

In coding specifically, where enterprise spending concentrates, Claude now holds 42% market share compared to OpenAI's 21%. More than double. In the segment that generates real revenue.

I've watched enterprise software markets for fifteen years. Market share doesn't flip this fast unless something fundamental changed in buyer psychology. And what changed is simple: **enterprises stopped buying based on brand and started buying based on what actually works when you scale beyond proof-of-concept.**

But here's where the story gets interesting. Roughly $1.2 billion of Anthropic's $4 billion annual revenue comes from just two customers: Cursor and GitHub Copilot. That's 30% of their business sitting on two contracts.

Then, three weeks ago, OpenAI launched GPT-5 with pricing that reportedly undercuts Claude by a significant margin for comparable tasks. Not a minor discount — aggressive, market-share-recapture pricing.

**So Anthropic faced a classic strategic trap:** Cut prices to defend those two massive customers and destroy your margins across the entire business, or hold pricing and risk losing 30% of revenue overnight.

*Instead of choosing between bad options, they changed what game they're playing.*

. . .

## The Move

The Claude disguised as developer goodwill.

Anthropic released the infrastructure foundation they use internally to build Claude Code — virtual machines, memory management, context engineering, agent coordination, checkpoint systems. Everything. Open. Free to use.

**Why would they do this? Because they spotted something everyone else missed:** *every company building production AI agents solves identical infrastructure problems, and nobody's turned that into a platform play yet.*

Here's what I mean. Over the past six months, I've consulted with eleven companies building AI agents — from seed-stage startups to public companies.

**Every single one struggles with the same five infrastructure challenges:**

> Context windows that overflow.

Your agent works great for an hour, then loses critical information and starts making nonsensical decisions. The Agent SDK's compact feature automatically summarizes previous messages when approaching context limits, solving what everyone else rebuilds from scratch.

**Master Claude Agent SDK: A 5-Step Integration Guide to Cut Development Time 70% with...**

Stop building custom agent loops from scratch. Follow this battle-tested integration to deploy secure, scalable AI...

alirezarezvani.medium.com

IASTER CLAUDE AGENT SDK:
5-STEP INTEGRATION GUIDE to Cut
evelopment Time 60% with Production Ready Agents
1
2
3
4
4
5
CUT DEVELOPMENT
TIME 70%
PRODUCTION READY
AGENTS

**No reliable resumption after failures.** Your agent encounters an error three hours into a complex task. You fix the error. Now what? Start from zero? Try to resume from some arbitrary point and hope it works? The SDK's checkpoint system automatically saves state before each change, letting you rewind to any previous point.

**Permission systems that don't scale.** Either your agent can't do anything useful (too restrictive) [illegible] middle ground wh[illegible]ission controls that work in production environments.

**No audit trail for compliance.** If your agent makes a change, can you prove what it did and why? Regulated industries can't deploy AI systems without this. The SDK handles comprehensive logging systematically.

**Context management across specialized tasks.** The SDK's sub-agents delegate specialized tasks that run in parallel, with proper context isolation and coordination. Building this yourself takes months of careful engineering.

These aren't model problems. These are infrastructure problems. And every development team wastes three to six months rebuilding the same solutions because until now, nobody provided them as a platform.

**Anthropic looked at this and asked the right question:** "What if we compete on who ships to production fastest, not who has better benchmarks?"

*That question changes everything about how this market evolves over the next two years.*

. . .

## The Metric That Actually Matters (And Why Nobody Talks About It)

**Every article about Claude Sonnet 4.5 leads with the same headline:** *"maintains focus for over thirty hours on complex, multi-step tasks"*.

**Impressive? Technically, yes. Relevant?** *Almost never.*

I've had this conversation with CTOs at six different companies in the past month. None of them want thirty-hour unsupervised agent runs.

### What they actually need:

**Two-hour workflows that complete reliably.** Not thirty hours of autonomous operation where you pray nothing breaks. Two hours of reliable work where you can validate progress and continue.

**Checkpoint systems they can actually use.** *"The agent worked for six hours then failed"* is useless if you can't resume from hour five. You need granular checkpoints that let you validate and proceed.

**Clear visib** ce. For trust. Enterprise teams need to understand exactly what the agent did, not just see the final output.

**Controlled intervention points.** The ability to pause, review, adjust, and continue — without restarting from zero or breaking the entire workflow.

The thirty-hour capability enables specific use cases: comprehensive security audits of massive codebases, large-scale refactoring across hundreds of files, deep research projects spanning multiple domains. These matter. But they're not the primary blocker for most production deployments.

What blocks production is mundane infrastructure: context management that doesn't break, checkpointing that works reliably, permissions that make sense, audit trails that satisfy compliance.

**The Agent SDK provides exactly this.** Not the sexy benchmark everyone tweets about. Just infrastructure that works when you need it to work.

*And that's why this matters more than any benchmark improvement could.*

**Agentic Engineering: Multi-Agent Orchestration for Modern Software Devs**

Learn the 3-pillar orchestration system that transforms AI-assisted development. Specialized agents, unified control...

alirezarezvani.medium.com

• • •

## The AWS Parallel Nobody's Discussing (But Should)

In 2006, Amazon did something that seemed strategically insane: they started selling the internal infrastructure they'd built for Amazon.com to anyone who wanted it, including direct competitors.

Industry analysts thought Amazon was making a mistake. Why help competitors by selling them the same infrastructure Amazon uses? Why not keep that competitive advantage proprietary?

We know h ear — more than Amazon's entire North American retail business.

**The exact same strategic pattern is playing out with Anthropic and the Agent SDK.**

Anthropic built sophisticated agent infrastructure to power Claude Code internally. They looked at the market and realized: every developer team is attempting to build equivalent systems. Every team is solving the same problems. Every team is reinventing the same wheels.

**And they saw the opportunity everyone else missed:** *the real moat isn't "better models" — it's "the platform everyone builds on."*

Inside Anthropic, Claude Code now powers almost all their major agent loops — not just coding, but research, video creation, note-taking, and numerous other applications. They built it as general-purpose infrastructure, not just a coding tool.

Now they're making it available to everyone. Because once developers standardize on your infrastructure:

- Switching models becomes trivial *(just change an API endpoint)*
- Infrastructure lock-in is exponentially stickier than model preference
- You become the default platform regardless of which model powers individual agents

This is infrastructure-as-competitive-moat. It's the same playbook that made AWS more valuable than retail. And it fundamentally changes how the AI market evolves.

*Because once you control the infrastructure layer, model wars become far less relevant.*

. . .

## Why This Repositions Every Player in the Market

OpenAI's strategy has been crystal clear since ChatGPT launched: dominate consumer awareness, then leverage that brand recognition into enterprise contracts.

It worked spectacularly in consumer markets. ChatGPT achieved cultural penetratio I for hundreds c

But in enterprise markets, OpenAI's position eroded from 50% market share to 25% over roughly two years, while Anthropic grew from 12% to 32%.

Why the divergence? **Because enterprises don't buy based on consumer brand. They buy based on what works reliably in production when millions of dollars depend on it.**

And now Anthropic is offering something fundamentally different: "Don't just use our models. Use our infrastructure. Build your entire agent stack on our platform. Lock in at the infrastructure layer, not the model layer."

This directly undermines OpenAI's enterprise strategy in a way that pricing or model improvements can't. Because once development teams standardize on the Agent SDK:

**Switching becomes exponentially harder.** It's not about trying a different model API. It's about rebuilding your entire operational infrastructure — context management, checkpointing, sub-agent coordination, permission systems, audit logging. That's three to six months of engineering work to switch. Maybe more.

**Infrastructure decisions compound.** As you build more agents on the platform, switching costs multiply. Your first agent might take three months to migrate. Your twentieth agent makes migration practically impossible.

**The platform improves continuously.** JetBrains integrated Claude Agent into their IDEs within days of the SDK's release. As more teams adopt the SDK, more integrations appear, making the platform more valuable for everyone. Network effects kick in.

OpenAI can compete on model quality. They can compete on pricing. They can compete on brand. But competing against infrastructure means building an entire platform ecosystem — something that requires years of focused development effort and developer mindshare they don't currently have.

**And the most expensive commodity in technology isn't money. It's time.**

This [illegible] Claude Sonn[illegible]

. . .

### The Benchmarks That Actually Determine Production Success

Claude Sonnet 4.5 scores 77.2% on SWE-bench Verified and 61.4% on OSWorld. Every article leads with these numbers. They're easy to report, easy to compare, easy to tweet.

But they're not what determines whether your AI agent succeeds or fails in production.

**The questions that actually matter:**

**Can I deploy without rebuilding foundational infrastructure?** Every month spent building infrastructure is a month not shipping features. The Agent SDK eliminates this entirely. You get production-grade context management, checkpointing, and coordination systems on day one.

**Do I have the safety guarantees compliance actually requires?** Claude Sonnet 4.5 achieved a 98.7% safety score compared to 89.3% for the previous version. But more importantly, the SDK provides permission systems, approval workflows, and audit infrastructure built for regulated environments. SOC 2? HIPAA? GDPR? The logging and access controls are built in.

**Can I resume work after failures without starting over?** Production systems fail. Always. The question is whether your infrastructure handles failure gracefully. The SDK's automatic checkpoint system makes resumption straightforward instead of requiring custom engineering.

**Do I have audit trails that satisfy auditors?** Three months after deployment, when something unexpected happens, can you trace exactly what your agent did and why? The SDK's comprehensive logging answers this question by default.

Devin, an AI coding platform, reported that Claude Sonnet 4.5 increased their planning performance by 18% and end-to-end evaluation scores by 12%. Those improvements matter. But the bigger strategic story is this: teams using the Agent

SDK avoid building checkpoint systems, context management, and sub-agent coordinati

**Time-to-production drops from six months to three weeks.** That's not an exaggeration based on the projects I've advised. That's the competitive advantage that actually determines who wins.

*And that advantage compounds the longer you wait to switch infrastructure.*

. . .

## The Strategic Fork Every Team Faces Right Now

If you're building AI agents, you're standing at a strategic decision point right now. Most teams don't realize it yet. But the choice you make in the next sixty days determines whether you ship in weeks or months.

### Path 1: Build Custom Infrastructure

You spend four to six months building the foundational layer:

- Context management that handles overflow gracefully
- Checkpoint systems that actually work when you need them
- Permission frameworks that scale beyond toy examples
- Audit logging that satisfies compliance requirements
- Sub-agent coordination for parallel workflows

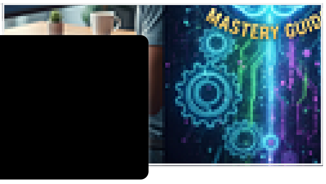
You hope you don't encounter edge cases that break production. You maintain all this infrastructure while competitors ship features. You compete on implementation details instead of unique business value.

**Real example:** A healthcare AI startup I advised spent seven months building custom agent infrastructure. When they finally shipped, two competitors had already captured their target market using the Agent SDK. Their infrastructure was technically superior. Their market position was fatally compromised.

**Claude Code Complete Mastery Guide: For Solo Developers Using Claude Code**

Why 85% of developers use AI tools, but only 15% build the context systems th

alirezarezvar



### Path 2: Leverage Platform Infrastructure

You integrate the Agent SDK in three days:

- Production-grade infrastructure immediately available
- Engineering time focused on business logic that differentiates
- Continuous platform improvements you get automatically
- Integration ecosystem that grows as more teams adopt

You deploy in weeks. You iterate rapidly based on user feedback. You compete on unique value instead of infrastructure quality.

**Real example:** JetBrains integrated the Claude Agent SDK into their IDE products in under a week. Their engineering team focused on integration UX and developer workflows instead of building context management systems from scratch.

The SDK is available now in Python and TypeScript. Installation takes minutes. Integration takes days, not months.

**This isn't a convenience tool. It's a strategic decision about where you compete and how fast you ship.**

*And that decision determines whether you lead your market or follow it.*

. . .

## What the Market Data Reveals About Model Competition

Here's what enterprise LLM spending data from mid-2025 reveals: **model quality differentiates less than operational excellence once you reach "good enough" capability.**

Claude excels at coding tasks. GPT-5 offers aggressive pricing. Gemini demonstrates specific advantages. But Anthropic captured enterprise market leadership not by having the best model, but by having the best production experience.

When I interview enterprise buyers about their LLM selection process, here's what they actual

1. **Time-to-production:** How fast can we deploy in a way that actually works?
2. **Operational reliability:** Will this work consistently at scale?
3. **Compliance infrastructure:** Can we prove to auditors that this is safe and controlled?
4. **Integration ecosystem:** Does this play well with our existing tools?
5. **Model quality:** Is the underlying model good enough for our use case?

Notice what's fifth on that list.

Anthropic bet that infrastructure matters more than benchmark leadership. Their enterprise market share growth from 12% to 32% in two years while OpenAI dropped from 50% to 25% suggests the market validates this bet.

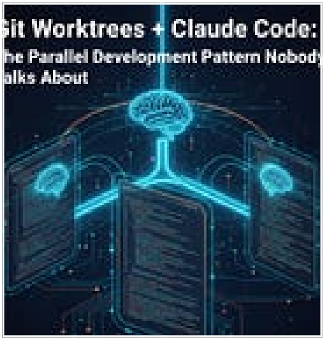
**The model wars are evolving into platform wars.** And platform wars are won with infrastructure that makes deployment easy, not models that edge benchmarks by a few percentage points.

*If you're still optimizing model selection while ignoring infrastructure strategy, you're solving last year's problem.*

**Git Worktrees + Claude Code: Parallel AI Development Guide**

Stop context switching. Learn how engineering teams use git worktrees with Claude Code for parallel development…

alirezarezvani.medium.com


it Worktrees + Claude Code:
he Parallel Development Pattern Nobody
alks About

. . .

## The Question That Clarifies Everything

Here's the question that makes Anthropic's strategy crystal clear:

**If Claude Sonnet 4.5 represents such a significant competitive advantage — why give away t**

Think about it. If your infrastructure is a key competitive moat, why make it available to everyone, including direct competitors?

The answer reveals something fundamental about how Anthropic thinks about competition: **they're not competing on models anymore. They're competing on platforms.**

OpenAI dominates consumer mindshare. ChatGPT is the default AI for hundreds of millions of people. That's valuable — but it's not enough to win enterprise markets.

Because enterprise competition centers on operational excellence: reliability, compliance, auditability, deployment speed, integration quality, production readiness. Those are infrastructure problems, not model problems.

The Agent SDK is Anthropic declaring: "We'll win by being the platform everyone builds on, regardless of which model they ultimately choose."

It's the same strategic logic that made AWS valuable: control the infrastructure layer, and model superiority becomes less relevant.

**It's strategically sound.** Especially when facing customer concentration risk with 30% of revenue depending on two contracts while competitors slash prices aggressively.

*This is what strategic repositioning looks like in practice — and why it's so effective.*

• • •

## What This Means When You Make Your Next Architecture Decision

The CTO I mentioned at the beginning? The one who burned six months and half a million dollars?

He called me back last Friday. His team spent the previous weekend evaluating the Agent SDK. They rebuilt their core agent workflow as a proof-of-concept.

**His exact words:** *"We just eliminated four months of infrastructure work we thought was unavoidable. We're deploying to initial customers next Tuesday instead of next quarter.*

*This changes everything about our roadmap.”*

**That’s not a altering competitive timelines.**

**Here’s what this means for your next architecture decision:**

> Stop asking “which model has the best benchmarks.” Start asking “which infrastructure gets us to production fastest with the least operational risk.”

The competitive advantage in AI development is shifting from model selection to infrastructure leverage. Teams that recognize this shift early will ship faster and operate more reliably than those still optimizing model choice in isolation.

The Claude Agent SDK provides a computer environment where agents can write files, run commands, and iterate on work autonomously — but more importantly, it provides the operational infrastructure production systems actually require: context management, checkpointing, permission control, audit logging, sub-agent coordination.

**My recommendation based on fifteen years advising enterprise technology teams:** Evaluate the Agent SDK for your next agent project. Not because Claude has superior benchmarks — though it does in several categories. But because the infrastructure might save you four to six months of development time and countless operational headaches.

While most teams debate benchmark percentages and model pricing, Anthropic is building the infrastructure layer for production AI agents.

And if they execute this strategy successfully — and early adoption patterns suggest they are — the model wars become significantly less relevant than the platform wars.

**That’s the strategic shift most people are missing.** And missing it means falling behind teams that see it clearly.

. . .

### Three Act

**First:** If you ek exploring the Agent SDK documentation. Specifically look at the checkpoint system, context management, and sub-agent coordination. Ask yourself: “How long would building this ourselves take?”

**Second:** Talk to your team about the build-versus-leverage tradeoff. Are you building infrastructure someone else already solved? Could you ship features faster by leveraging platform infrastructure? What’s the strategic opportunity cost of building from scratch?

**Third:** Watch how the market evolves over the next ninety days. Which development tools integrate the Agent SDK? Which companies ship production agents quickly? Which teams spend months building infrastructure? The pattern will become clear.

. . .

*What’s your experience building production AI agents? What infrastructure problems consumed the most engineering time? How are you thinking about the build-versus-leverage strategic tradeoff?*

*Share your perspective in the comments. I read every response, and the most valuable discussions happen there — people sharing real production experiences that don’t make it into press releases.*

*If this analysis changed how you think about AI infrastructure strategy, hit the clap button and share it with your team. This conversation matters for everyone making technology decisions right now.*

. . .

*Follow for analysis that cuts through AI hype to focus on what actually matters for production systems.*

*Next article: “The Agent Architecture Pattern That’s Replacing Microservices (And Why Your Stack Needs It).”*

Welcome back. You are signed into your member account **bg••••@jaxondigital.com**.

✨ *Thanks for reading! If you'd like more practical insights on AI and tech, hit* ***subscribe*** *to stay updated.*

*I'd also love to hear your thoughts — drop a comment with your ideas, questions, or even the kind of topics you'd enjoy seeing here next. Your input really helps shape the direction of this channel.*

### About the Author

*Me, Alireza Rezvani work as a CTO @ an HealthTech startup in Berlin and architect AI development systems for my engineering and product teams. I write about turning individual expertise into collective infrastructure through practical automation.*

**Connect:** Website | LinkedIn
**Read more:** Medium Reza Rezvani

**Explore my other open source projects:** GitHub

Artificial Intelligence | Ai Agent | Agentic Ai | Enterprise Architecture | Future

Following ⌄

## Written by Reza Rezvani

1.1K followers · 77 following

As CTO of a Berlin AI MedTech startup, I tackle daily challenges in healthcare tech. With 2 decades in tech, I drive innovations in human motion analysis.

## No responses yet

Bgerby

What are yo

Welcome back. You are signed into your member account **bg••••@jaxondigital.com**.

## More from Reza Rezvani

In nginity by Reza Rezvani

### How Cursor and Claude Code Plugins Turned Me Into a 20x Developer – The Agentic Coding Setup That...

My Background Agent just submitted a PR that made our senior architect ask, “Who wrote this?”

✦ Oct 14 77

Welcome back. You are signed into your member account **bg••••@jaxondigital.com**.

Reza Rezvani

## Gemini CLI: What Happened When I Replaced My IDE With a Free AI Terminal Agent for 30 Days

I gave Google's new Gemini CLI full access to my development workflow and tested it on real production code. Here's what actually worked...

✦ Oct 10 20

Reza Rezvani

## "7 Steps" How to Stop Claude Code from Building the Wrong Thing (Part 1): The Foundation of...

Learn how to s. This
guide introd

Welcome back. You are signed into your member account **bg••••@jaxondigital.com**.

Sep 17

Reza Rezvani

## The AI Agent That Became Our Team’s Silent Partner: A Journey from Chaos to Flow

When everything changed, it wasn’t the code – it was how we worked

Oct 11 4

See all from Reza Rezvani

## Recommended from Medium

Welcome back. You are signed into your member account **bg••••@jaxondigital.com**.

In AI Software Engineer by Joe Njenga

## Anthropic Just Solved AI Agent Bloat—150K Tokens Down to 2K (Code Execution With MCP)

Anthropic just released smartest way to build scalable AI agents, cutting token use by 98%, shift from tool calling to MCP code execution

✦ 2d ago 286 18

In Level Up Coding by Fareed Khan

## Building a Training Architecture for Self-Improving AI Agents

RL Algorithms, Policy Modeling, Distributed Training and more.

4d ago

Welcome back. You are signed into your member account **bg••••@jaxondigital.com**.

Daniel Avila

## Running Claude Code Agents in Docker Containers for Complete Isolation

Running AI-generated code directly on your machine can be risky.

5d ago 👏 40

In Coding Nexus by Algo Insights

## DeepAgent: How the New AI Agent Learns, Thinks, and Builds Its Own Tools

Welcome back. You are signed into your member account **bg•••••@jaxondigital.com**.

When I first r … es it truly mean for an AI to discover its own tools?

✦ Nov 1 👏 168 💬 2

Agen.cy

## MCP Ecosystem Is Exploding: Here Are 20+ Launches You Shouldn’t Miss

MCP is the USB-C for AI agents. One port, every tool.

Oct 31

In Silicon Valley Gradient by dravian

Welcome back. You are signed into your member account **bg••••@jaxondigital.com**.

**“gpt-6 is**

says sam altman as the model created its own language to process a new way of thinking,

6d ago 244 10

See more recommendations