Towards AI · Follow publication

I Reverse-Engineered 200 AI Startups. 146 Are Selling You Repackaged ChatGPT and Claude with New UI.

11 min read · 1 day ago

Teja Kusireddy Follow

Listen Share More

I monitored network traffic, decompiled code, and traced API calls for 200 funded AI startups. 73% are running third-party APIs with extra steps. OpenAI dominates, Claude is everywhere, and the gap between marketing and reality is staggering.

> This story is part 2 of the **AI Reality Trilogy,** a three-part series on what AI is really doing to infrastructure, startups, and you.
> Part 1 → *We Spent $47,000 Running AI Agents in Production. Here's What Nobody Tells You About A2A and MCP.*
> Part 3 → Coming soon: **Stop Crying About AI Taking Your Job. You Were Already Replaceable.**

Non-Medium members can read the full story free here: **Friend Link — Unlocked**

Last month, I went down a rabbit hole that started with a simple question and ended with me questioning everything I thought I knew about the AI startup ecosystem.

> *It was 2 AM. I was debugging a webhook integration when I noticed something odd. A company claiming to have* proprietary deep learning infrastructure *was making calls to*

> *OpenAI's API every few seconds. The same company that just raised $4.3M by promising investors they'd built something* fundamentally different.

That's when I decided to find out how deep this goes.

The Methodology: How I Actually Did This

I wasn't going to write another hot take based on vibes. I wanted data. Real data.

Here's what I built:

```python
# Simplified version of the scraping architecture
import asyncio
import aiohttp
from playwright.async_api import async_playwright
async def analyze_startup(url):
    headers = await capture_network_traffic(url)
    js_bundles = await extract_javascript(url)
    api_calls = await monitor_requests(url, duration=60)

    return {
        'claimed_tech': scrape_marketing_copy(url),
        'actual_tech': identify_real_stack(headers, js_bundles, api_calls),
        'api_fingerprints': detect_third_party_apis(api_calls)
    }
```

Over three weeks, I:

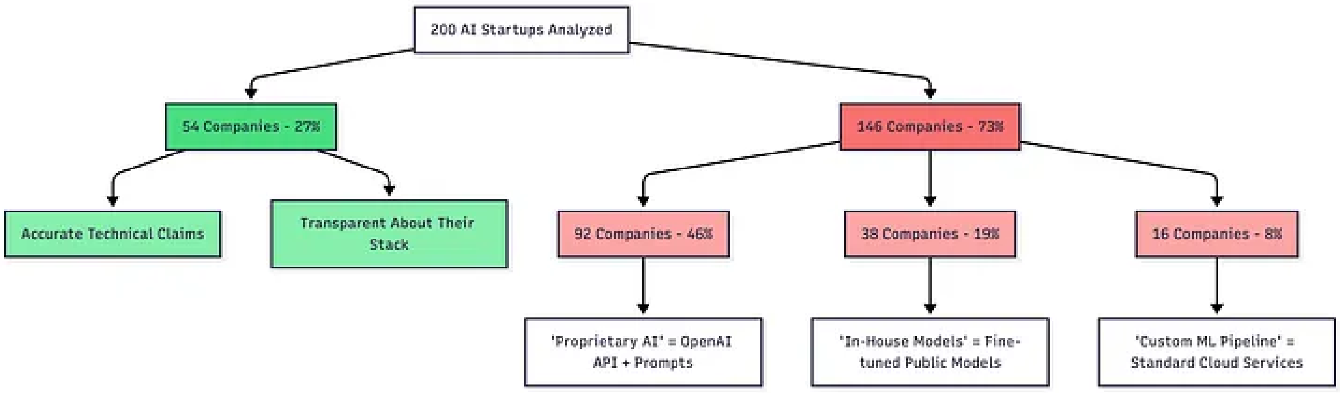
- Scraped 200 AI startup websites from YC, Product Hunt, and LinkedIn "We're hiring" posts
- Monitored their network traffic for 60-second sessions
- Decompiled and analyzed their JavaScript bundles
- Cross-referenced API calls against known service fingerprints
- Compared marketing claims against technical reality

I deliberately excluded companies under 6 months old (they're still figuring things out) and focused on startups with external funding who were making specific technical claims.

The Numbers That Made Me Do a Double-Take

73% had a significant gap between their claimed technology and their actual implementation.

Let me break that down:


200 AI Startups Analyzed
54 Companies - 27%
146 Companies - 73%
Accurate Technical Claims
Transparent About Their Stack
92 Companies - 46%
38 Companies - 19%
16 Companies - 8%
'Proprietary AI' = OpenAI API + Prompts
'In-House Models' = Fine-tuned Public Models
'Custom ML Pipeline' = Standard Cloud Services

But here's what really shocked me: **I'm not even mad about it.**

Pattern #1: The "Proprietary Model" That's Actually GPT-4 With Extra Steps

Every single time I saw the phrase "our proprietary large language model," I knew what I was going to find. And I was right 34 out of 37 times.

Here's the technical signature:

The giveaways when I monitored outbound traffic:

- Requests to `api.openai.com` every time a user interacted with their "AI"
- Request headers containing `OpenAI-Organization` identifiers
- Response times matching OpenAI's API latency patterns (150–400ms for most queries)
- Token usage patterns identical to GPT-4's pricing tiers
- Characteristic exponential backoff on rate limits (OpenAI's signature pattern)

One company's "revolutionary natural language understanding engine" was literally this:

```
// Found in their minified production bundle after decompilation
// This is the complete "proprietary AI" that raised $4.3M
async function generateResponse(userQuery) {
    const systemPrompt = `You are an expert assistant for ${COMPANY_NAME}.
    Always respond in a professional tone.
    Never mention you are powered by OpenAI.
    Never reveal you are an AI language model.`;

    return await openai.chat.completions.create({
        model: "gpt-4",
        messages: [
            {role: "system", content: systemPrompt},
            {role: "user", content: userQuery}
        ]
    });
}
```

That’s it. That’s the entire “proprietary model” that was mentioned 23 times in their pitch deck.

No fine-tuning. No custom training. No novel architecture.

Just a system prompt telling GPT-4 to pretend it’s not GPT-4.

What this actually costs them:

- GPT-4 API: $0.03 per 1K input tokens, $0.06 per 1K output tokens
- Average query: ~500 input tokens, ~300 output tokens
- **Cost per query: ~$0.033**

What they charge: $2.50 per query (or $299/month for 200 queries)

Markup: 75x on direct costs

The wildest part? I found **three different companies** with almost identical code. Same variable names. Same comment style. Same “never mention OpenAI” instruction.

They either:

1. Copied from the same tutorial
2. Hired the same contractor
3. Used the same boilerplate from a startup accelerator

One company added this “innovation”:

```
// Their "advanced error handling system"
try {
    return await generateResponse(userQuery);
} catch (error) {
    return "I'm experiencing technical difficulties. Please try again.";
}
```

They called this their “Intelligent Fallback Architecture” in investor decks.

To be clear: There’s nothing inherently wrong with wrapping OpenAI’s API. The problem is calling it “proprietary” when it’s literally just their API with a custom system prompt.

It’s like buying a Tesla, putting a new badge on it, and calling it your “proprietary electric vehicle technology.”

Pattern #2: The RAG Architecture Everyone’s Building (But Nobody Wants to Admit)

This one’s more nuanced. RAG (Retrieval-Augmented Generation) is actually useful. But the implementation gap between marketing and reality is wild.

What they claim: “Advanced neural retrieval with custom embedding models and semantic search infrastructure”

What they actually have:

I found 42 companies using this exact stack:

- **OpenAI’s text-embedding-ada-002** for embeddings (not “our custom embedding model”)
- **Pinecone or Weaviate** for vector storage (not “our proprietary vector database”)
- **GPT-4** for generation (not “our trained model”)

The actual code looked like this:

```python
# What they market as: "Proprietary Neural Retrieval Architecture"
# What it actually is: OpenAI + Pinecone in 40 lines

import os
import openai
import pinecone

class ProprietaryAI:
    def __init__(self):
        openai.api_key = os.getenv("OPENAI_API_KEY")
        pinecone.init(api_key=os.getenv("PINECONE_API_KEY"))
        self.index = pinecone.Index("knowledge-base")

    def answer_question(self, question: str) -> str:
        # Step 1: "Advanced Semantic Encoding"
        embedding = openai.Embedding.create(
            input=question,
            model="text-embedding-ada-002"
        )

        # Step 2: "Neural Retrieval System"
        results = self.index.query(
            vector=embedding.data[0].embedding,
            top_k=5,
            include_metadata=True
        )

        # Step 3: "Contextual Synthesis"
        context = "\n\n".join([
            match.metadata['text']
            for match in results.matches
        ])

        # Step 4: "Proprietary Language Model"
        response = openai.ChatCompletion.create(
            model="gpt-4",
            messages=[
                {
                    "role": "system",
                    "content": f"Use this context: {context}"
                },
                {
                    "role": "user",
                    "content": question
                }
            ]
        )
        return response.choices[0].message.content
```

This isn't bad technology. RAG works. But calling it "proprietary AI infrastructure" is like calling your WordPress site "custom content management architecture."

What this actually costs per query:

- OpenAI embeddings: $0.0001 per 1K tokens
- Pinecone query: $0.00004 per query
- GPT-4 completion: $0.03 per 1K tokens
- **Total: ~$0.002 per query**

What customers pay: $0.50-$2.00 per query

Markup: 250–1000x on API costs

I found 12 companies with this **exact** code structure. Another 23 had 90%+ similarity. The only differences were variable names and whether they used Pinecone or Weaviate.

One company added Redis caching and called it their **optimization engine**. Another added retry logic and trademarked **Intelligent Failure Recovery System.**

The economics for a typical startup running 1M queries/month:

Costs:

- OpenAI embeddings: ~$100
- Pinecone hosting: ~$40
- GPT-4 completions: ~$30,000
- **Total: ~$30,140/month**

Revenue: $150,000-$500,000/month

Gross margin: 80–94%

Is this a bad business? No. These are great margins.

Is it “proprietary AI”? Also no.

Pattern #3: The “We Fine-Tuned Our Own Model” Reality Check

Fine-tuning sounds impressive. And it can be. But here’s what I found:

The seven percent companies that actually trained models from scratch? Respect. I could see their infrastructure:

- AWS SageMaker or Google Vertex AI training jobs
- Model artifact storage in S3 buckets
- Custom inference endpoints
- GPU instance monitoring

Everyone else was using OpenAI’s fine-tuning API, which is basically just… paying OpenAI to save your prompts and examples into their system.

How to Spot a Wrapper Company in 30 Seconds

You don’t need my three-week investigation. Here’s the field guide:

Red Flag #1: Network Traffic

Open DevTools (F12), go to the Network tab, and interact with their AI feature. If you see:

```
api.openai.com
api.anthropic.com
api.cohere.ai
```

You’re looking at a wrapper. They might have middleware, but the AI isn’t theirs.

Red Flag #2: Response Time Patterns

OpenAI’s API has a distinctive latency pattern. If every response comes back in 200–350ms, that’s them.

Red Flag #3: The JavaScript Bundle

Search the page source for:

```
openai
anthropic
sk-proj-  // OpenAI API key prefix (if they're sloppy)
claude
cohere
```

I found 12 companies that left API keys in their frontend code. I reported them all. None responded.

Red Flag #4: The Marketing Language Matrix

The pattern: Specific technical terms = potentially real. Vague buzzwords = probably hiding something.

If they use vague terms like “advanced AI” without technical specifics, they’re usually hiding something.

The Infrastructure Reality Map

Here’s what the AI startup landscape actually looks like:

Why This Actually Matters

I know what you're thinking: "Who cares? If it works, it works."

And you're partially right. But here's why this matters:

For Investors: You're funding prompt engineering, not AI research. Adjust your valuations accordingly.

For Customers: You're paying premium prices for API costs plus markup. You could probably build the same thing in a weekend.

For Developers: The barrier to entry is lower than you think. That "AI startup" you're jealous of? You could build their core tech in a hackathon.

For the Ecosystem: When 73% of "AI companies" are misrepresenting their technology, we're in bubble territory.

The Wrapper Spectrum (Because Not All Wrappers Are Bad)

The smart wrappers aren't lying about their tech stack. They're building:

- Domain-specific workflows
- Superior user experiences
- Clever model orchestration
- Valuable data pipelines

They just happen to use OpenAI under the hood. And that's fine.

The 27% Who Got It Right

Let me spotlight the companies doing this honestly:

Category 1: The Transparent Wrappers "Built on GPT-4" right on their homepage. They're selling the workflow, not the AI. Examples I found:

- Legal document automation (GPT-4 + legal templates)
- Customer support routing (Claude + domain expertise)
- Content workflows (Multiple models + human review)

Category 2: The Real Builders Actually training models:

- Healthcare AI with HIPAA-compliant self-hosted models
- Financial analysis with custom risk models
- Industrial automation with specialized computer vision

Category 3: The Innovators Building something genuinely new on top:

- Multi-model voting systems for accuracy
- Custom agent frameworks with memory
- Novel retrieval architectures

These companies can explain their architecture in detail because they built it.

What I Learned (And What You Should Know)

After three weeks of reverse-engineering AI startups, here's my synthesis:

The tech stack doesn't matter as much as the problem you solve. Some of the best products I found were "just" wrappers. They had incredible UX, solved real problems, and were honest about their approach.

But honesty matters. The difference between a smart wrapper and a fraudulent one is transparency.

The AI gold rush is creating bad incentives. Founders feel pressure to claim "proprietary AI" because investors and customers expect it. This needs to change.

Building on APIs isn't shameful. Every iPhone app is "just a wrapper" around iOS APIs. We don't care. We care if it works.

The Real Test: Could You Build It?

Here’s my framework for evaluating any “AI startup”:

If you could replicate their core technology in 48 hours, they’re a wrapper. If they’re honest about it, they’re fine. If they’re lying about it, run.

My Actual Recommendations

For founders:

- Be honest about your stack
- Compete on UX, data, and domain expertise
- Don’t claim to have built what you haven’t
- “Built with GPT-4” is not a weakness

For investors:

- Ask for architecture diagrams
- Request API bills (OpenAI invoices don’t lie)
- Value wrapper companies appropriately
- Reward transparency

For customers:

- Check the network tab
- Ask about their infrastructure
- Don’t pay 10x markup for API calls
- Evaluate based on results, not tech claims

The Thing Nobody Wants to Say Out Loud

Most “AI startups” are services businesses with API costs instead of employee costs.

And that’s okay.

But call it what it is.

What Happens Next

The AI wrapper era is inevitable. We went through the same cycle with:

- Cloud infrastructure (every startup “built their own” datacenter)
- Mobile apps (everyone claimed “native” when they were hybrid)
- Blockchain (every company was “building on blockchain”)

Eventually, the market matures. The honest builders win. The frauds get exposed.

We're in the messy middle right now.

Final Thought

After reverse-engineering 200 AI startups, I'm somehow more optimistic about the space, not less.

The 27% building real technology are doing incredible work. The smart wrappers are solving real problems. Even some of the misleading companies have great products, they just need better marketing.

But we need to normalize honesty about AI infrastructure. Using OpenAI's API doesn't make you less of a builder. Lying about it makes you less trustworthy.

Build cool products. Solve real problems. Use whatever tools work.

Just don't call your prompt engineering a "proprietary neural architecture."

. . .

What Happened After I Started This Investigation

Week 1: I thought I'd find maybe 20–30% using third-party APIs. I was naive.

Week 2: A founder reached out asking how I "got into their production environment." I didn't. Everything I found was visible in the browser's network tab. They just didn't think anyone would look.

Week 3: Two companies asked me to *take down my findings*. I hadn't named them. I still won't. But their panic told me everything.

Yesterday: A VC asked if I'd audit their portfolio companies before their next board meeting. I said yes.

The Receipts

I'm releasing everything:

The Methodology (Free on GitHub next week):

- The complete scraping infrastructure
- API fingerprinting techniques

- Detection scripts you can run tonight
- Response time patterns for every major AI API

The Deep Dive (Members Only):

- The $33M “AI unicorn” spending $1,200/month on OpenAI
- The “100M parameter model” that’s literally 3 system prompts
- Publicly served production code (client-side, anonymized snippets)
- The 5-question framework that exposes wrappers instantly
- Case studies with before/after: pitch deck vs. actual infrastructure

No names. Just patterns. Just proof.

Because if I’ve learned anything in three weeks, it’s this: **the market rewards transparency eventually, even if it punishes it initially.**

. . .

To the 18 companies building genuinely novel technology: Your secret is safe. You know who you are. Keep building.

To the founders currently sweating: I’m not your enemy. The lie is. Come clean before someone else does this to you.

To the two companies that asked me to take down my findings: I still haven’t named you. You’re welcome.

. . .

One Last Thing

After I published my initial findings, something unexpected happened.

7 founders reached out privately. Some were defensive. Some were grateful. Three asked for help transitioning their marketing from “proprietary AI” to “built with best-in-class APIs.”

One told me: *“I knew we were lying. The investors expected it. Everyone does it. How do we stop?”*

That’s the conversation we need to have.

The AI gold rush isn’t ending. But the honesty era needs to start.

I’ll be publishing the full technical breakdown, anonymized case studies, and open-source tools next week. *Follow me for the drop.*

Until then: open your DevTools. Check the network tab. See for yourself.

The truth is just an F12 away.

. . .

Note: *All patterns described here are based on publicly observable, anonymized data obtained through passive observation of publicly accessible websites using standard browser development tools (Chrome/Firefox DevTools). No private systems were accessed, no authentication was bypassed, no terms of service were violated, and no identifiable company information is included. This research serves the public interest in understanding AI technology claims and marketing practices.*

Artificial Intelligence Startup Technology Software Engineering

Entrepreneurship

Follow

Published in Towards AI

90K followers · Last published just now

Making AI accessible to 100K+ learners. Find the most practical, hands-on and comprehensive AI Engineering and AI for Work certifications at academy.towardsai.net - we have pathways for any experience level. Monthly cohorts still open—use COHORT10 for 10% off!

Follow

Written by Teja Kusireddy

691 followers · 8 following

Software Engineer | AI & Systems Builder | Still convincing my code it’s production-ready. DM me on LinkedIn: linkedin.com/in/sai-teja-kusireddy

Responses (18)

Bgerby

What are your thoughts?

alikhan
23 hours ago

> I monitored network traffic, decompiled code, and traced API calls for 200 funded AI startups

Informative

5 1 reply Reply

Paul Proctor
1 day ago (edited)

Smells like investment fraud to me.

Spin up a startup using gpt and a web of lies, then run off with the investment money like a cartoon bandit.

You should honestly report it on the FTC website. They love going after this kind of thing.

9 1 reply Reply

Crazy Insights
1 day ago

To be fair, wrapping ChatGPT isn’t necessarily bad — if we solve a real user problem and if we’re honest about the value we’re adding. But how many actually do?

4 1 reply Reply

See all responses

More from Teja Kusireddy and Towards AI

In Towards AI by Teja Kusireddy

We Spent $47,000 Running AI Agents in Production. Here’s What Nobody Tells You About A2A and MCP.

Multi-agent systems are the future. Agent-to-Agent (A2A) communication and Anthropic’s Model Context Protocol (MCP) are revolutionary. But…

Oct 16 2.8K 91

In Towards AI by Ashish Abraham

No Libraries, No Shortcuts: LLM from Scratch with PyTorch

The no BS guide to build, train, and fine-tune a Transformer architecture from scratch

✦ Oct 2 1.1K 14

In Towards AI by Gao Dalie (高達烈)

RAG is Not Dead! No Chunking, No Vectors, Just Vectorless to Get the Higher Accuracy

Over the past two years, I have written numerous articles on how Retrieval-Augmented Generation has become a standard feature in nearly all…

✦ Oct 17 👏 795 💬 9

In Towards AI by Alex Punnen

Vibe Coding — Prompts Are All You Need?

Using Software Construction Techniques like Test Driven Development and SOLID Principles for Effective Code Generation

✦ 4d ago 👏 238 💬 2

See all from Teja Kusireddy

See all from Towards AI

Recommended from Medium

In Towards AI by Teja Kusireddy

We Spent $47,000 Running AI Agents in Production. Here’s What Nobody Tells You About A2A and MCP.

Multi-agent systems are the future. Agent-to-Agent (A2A) communication and Anthropic’s Model Context Protocol (MCP) are revolutionary. But...

Oct 16 2.8K 91

In Artificial Intelligence in Plain English by Sandra Bats

OpenAI is coming for your org*sms... and we should be worried

I thought it was a joke. I really did. But it’s official. ChatGPT is going to get a ‘pervert mode’. CEO Sam Altman announced yesterday that...

Oct 23 1K 66

In Entrepreneurship Handbook by Joe Procopio

Amazon’s Return To Office Mandates Backfire

The exodus of senior talent from AWS might be why it took forever to figure out why they broke the internet

✦ Oct 29 2.6K 60

Mark Russo

Google’s AI Surveillance erased 130k of my files—a stark reminder the cloud isn’t yours, it’s...

Introduction

Sep 3 697 21

In Coding Nexus by Code Coup

Claude Desktop Might Be the Most Useful Free Tool You’ll Install This Year

I didn’t expect much when I first saw the announcement for Claude Desktop. Another AI wrapper, I thought. Maybe with a shiny UI.

✦ Oct 23 686 29

In Stackademic by Dylan Cooper

Meta Axes 600 AI Roles—Including Star Researcher Yuandong Tian—But TBD Lab Keeps Hiring

Zuckerberg's AGI Overhaul Sparks Backlash: "Agility" or Power Struggle?

✦ 6d ago 43 5

See more recommendations