APR 7, 2026

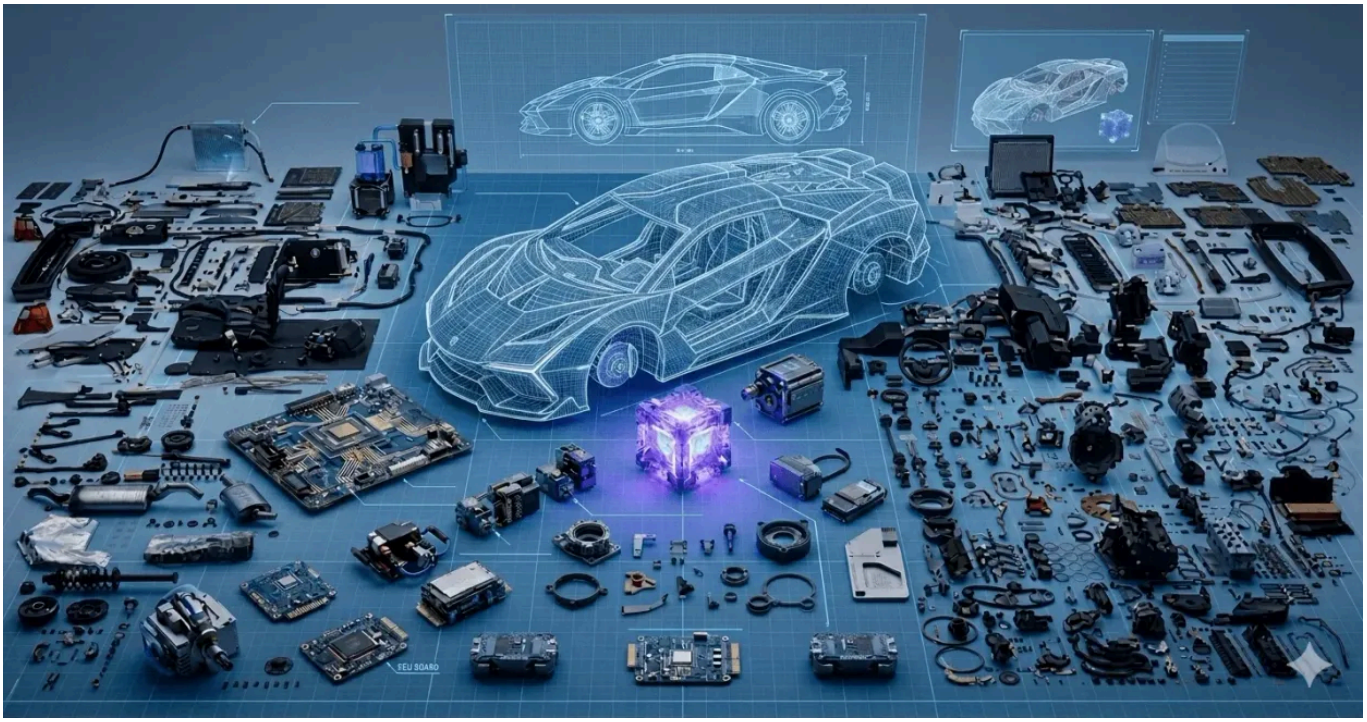
# Inside the Claude Code Leak

The Model Is the Engine. We Still Need to Build the Car



By Jenny Xiao, Liang Wu, and Jay Zhao

Leonis Capital · leoniscap.com/research/inside-the-claude-code-leak/



On March 31, Anthropic accidentally uploaded 512,000 lines of source code to npm. Within 48 hours, every serious AI founder on earth had read it. And what they found wasn't a model breakthrough. It was a harness. The most commercially successful AI agent in the world is, at its core, **a very well-built car around an engine**, where the engine is the model, and the car is everything built around it.

Steve Jobs called the computer a "bicycle for the mind." AI agents are something far more powerful than bicycles. They're cars. And a car is mostly not the engine. It's the transmission, the brakes, and the navigation. Anthropic just accidentally published the factory manual for theirs, and the most important thing it revealed is how little of the car is the engine.

**The consensus read on the leak is bearish for startups.** Before it, founders could hand-wave about "proprietary architecture" as a moat. Now 512,000 lines of production code are public, the architecture turns out to be a well-executed design pattern, and the ambiguity is gone. If people all had the same architecture handbook, what exactly are VCs paying billion-dollar valuations for?

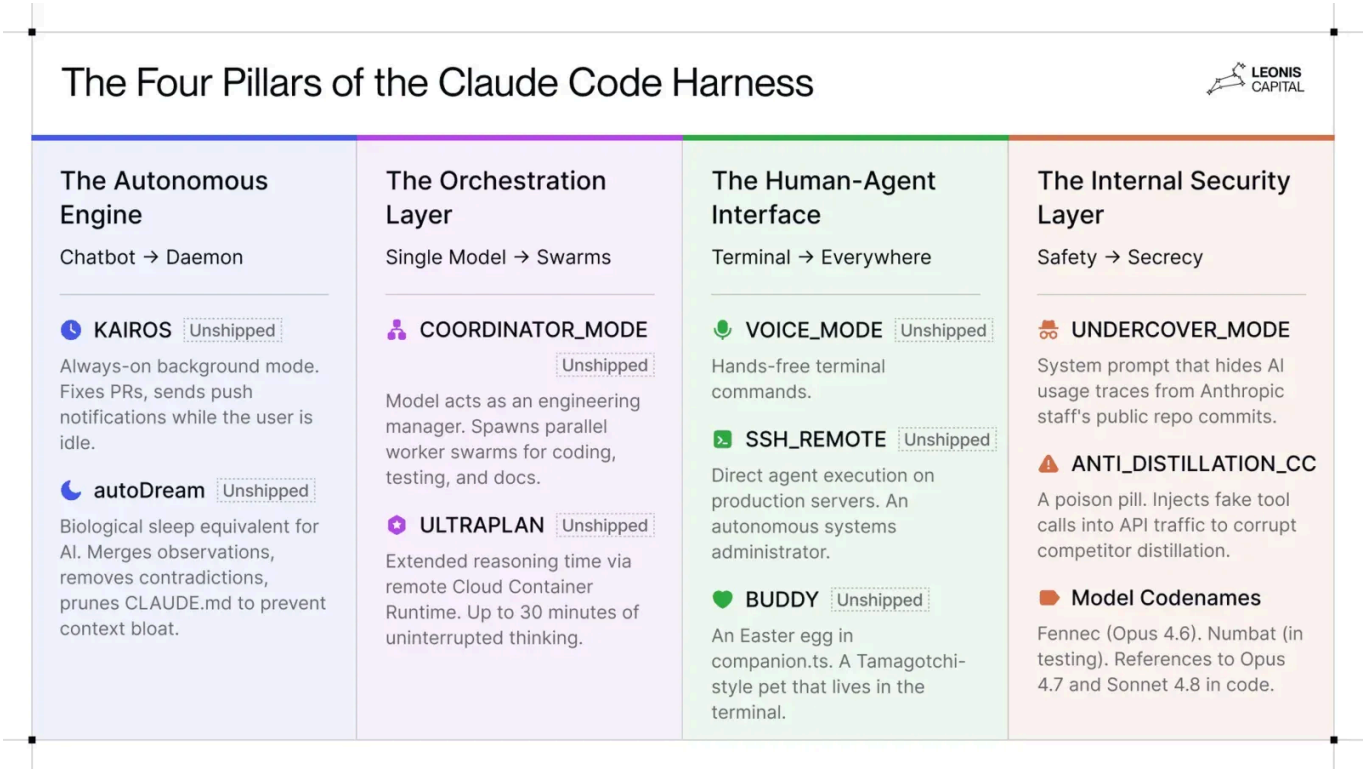
We think that read is wrong, or at least it's asking the wrong question. The most important thing the leak revealed isn't how much has been built. **It's how much is left to build.**

## The Leak Showed We're Barely Scratching the Surface

Here's what's in those 512,000 lines of code. The codebase spans 1,900 TypeScript files and includes over 40 built-in tools, each a permission-gated plugin, a 46,000-line query engine that manages all LLM API calls, streaming, and caching, a sub-agent orchestration system, and a three-layer self-healing memory architecture.

It takes 512,000 lines of code in the harness to make the underlying LLM work. This validates one of our predictions for 2026: the harness is becoming the main event. The leak is the first public proof.

But the real takeaway isn't what Anthropic has built. The leak reveals four pillars of the Claude Code harness, many of which are still hidden behind feature flags. This highlights how much capability is still sitting inside the models, unexplored and unshipped.




These four pillars, and the unshipped features within them, sketch where AI agents are heading.

- **The Autonomous Engine:** Perhaps the most exciting unshipped feature is KAIROS, an "always-on" background mode that allows Claude Code to work proactively (fixing PRs, sending push notifications) while the user is idle. It even runs a process called **autoDream**, a biological sleep equivalent for AI: merging observations, removing contradictions, and pruning CLAUDE.md files to prevent context bloat. This is the shift from chatbot to daemon, which suggests a future where AI Agents don't wait for instructions but continuously improve the codebase on their own.
- **The Orchestration Layer:** This is the part where the harness is moving away from single model interactions toward a managerial structure. **COORDINATOR_MODE** enables the model to act as an Engineering Manager, spawning parallel worker swarms to handle implementation, testing, and documentation simultaneously. **ULTRAPLAN**, another key unreleased feature, gives the model extended reseasoning time by offloading complex architectural tasks to a remote Cloud Container Runtime, allowing it up to 30 minutes of uninterrupted "thinking time" to solve difficult problems. This moves the agent from executing tasks one at a time to managing entire projects in parallel.
- **The Human-Agent Interface:** The feature flags reveal plans for expanded interaction modes beyond typing in a terminal. **VOICE_MODE** enables hands-free terminal commands. **SSH_REMOTE** allows direct agent execution in production servers, effectively turning Claude into an autonomous systems administrator. There is even **BUDDY**, an Easter egg in companion.ts that launches a Tamagotchi-style pet in your terminal, called **BUDDY**.

- **The Internal Security Layer**: This layer is one of the most controversial parts of the leak. **Undercover Mode** instructs the model to cover up traces of AI usage from Anthropic staff's public repo commits. **ANTI_DISTILLATION_CC** is a poison pill that injects fake tool calls into API traffic to corrupt the training data of competitors trying to distill Anthropic's models. Claude Code even comes up with **codenames** for its models: Fennec is Opus 4.6, Numbat is in testing, and references to Opus 4.7 and Sonnet 4.8 are already in the code.

To a systems engineer, the individual components of the Claude Code leak look like standard practice. The tool system's split between read-only concurrent and mutating serialized operations is a textbook safety-concurrency tradeoff. The agent loop is well-understood. The memory layering is smart but not novel. But taken together, these incremental choices activate model capability in a way that produces results most people would call magical.

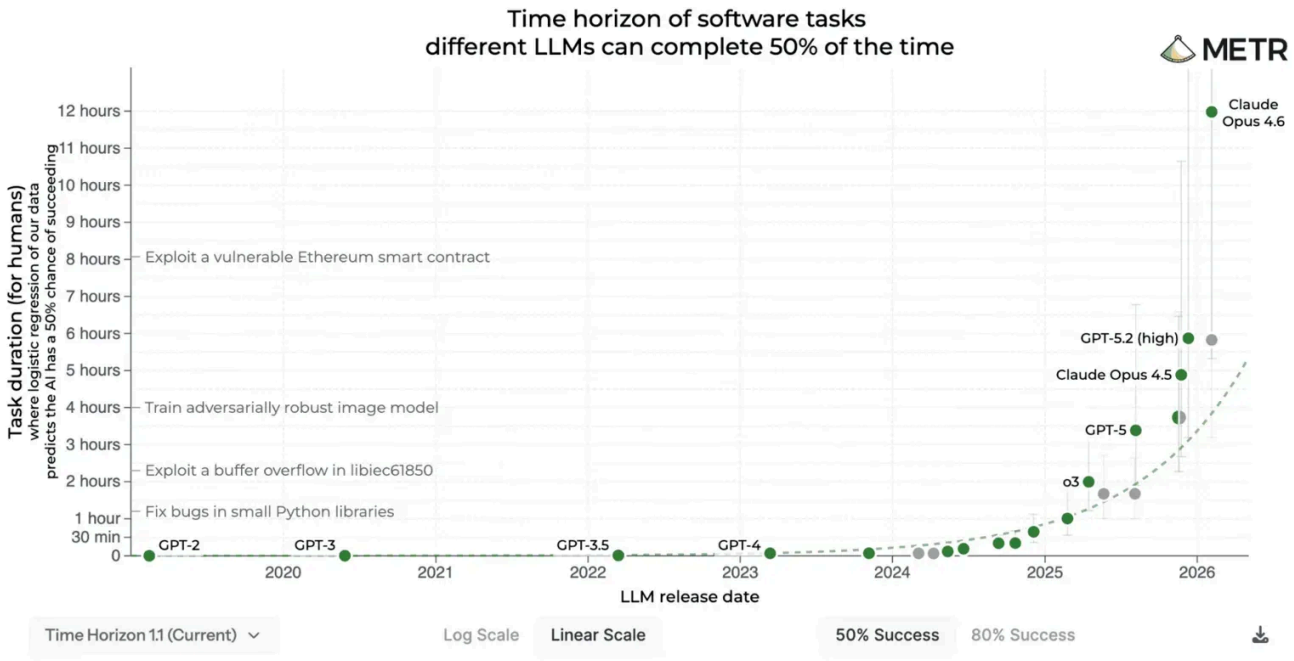
Anthropic's own engineering research supports this. In one experiment, their team asked Claude Opus 4.5 to create a retro video game maker. They ran the model two ways. Once as a single agent with no harness and once wrapped in a three-agent harness with a planner, a generator, and an evaluator. The first attempt finished in 20 minutes, cost $9, and produced broken code. The second run with the full harness took six hours, cost $200, and shipped working software end-to-end. [1] In both experiments, it was the same model, same weights, the only variable was the harness.

The 44 unshipped feature flags represent Anthropic's best guesses at what's possible. If the company that built the model is still figuring out how to fully activate it , we're all in the early innings.

## The "Engine Market" is Slowing and Consolidating

Model architecture innovation is hitting the top of an S-curve. As the pace of model improvement slows down, the engine market is consolidating into 3-4 players: Anthropic, OpenAI, Google, and maybe an open-source player. The pace of fundamental model intelligence breakthroughs is slowing relative to the pace of harness innovation.

A lot of what we're seeing in model performance, especially on long-duration and sophisticated tasks, comes from improvements in model harnesses. For example, the METR benchmark shows a near-vertical jump from Opus 4.5 to Opus 4.6 in long-horizon task performance. Much, if not most, of this capability jump is driven by system-level scaffolding.




The clearest example is Opus 4.6's Adaptive Thinking Mechanism, which lets the model dial reasoning efforts across different levels. Other examples include Opus 4.6's Agent Teams framework, which allows the model to spin up sub-agents to handle different parts of the task, and the context compaction feature that solves the "context rot" problem, the performance degradation that happens when the model's working memory fills up. This feature alone produced a significant jump on MRCR v2, a long-context benchmark. In other words, the model isn't necessarily "smarter" at its core, but it's now better at allocating compute resources, managing agent swarms, and compressing its own context.

This isn't to say that there weren't genuine model-level breakthroughs in Opus 4.6, what we are seeing is a Model-System fusion. [2] But the jumps in capabilities are increasingly coming from everything around the model, not the model itself.

> What About the Bitter Lesson?
>
> Some careful readers might object and "so what about the "bitter lesson"? They are right to ask. Historically, the bitter lesson in AI shows that scaling compute almost always wins against clever engineering. The model vs. harness debate sounds like the "bitter lesson" playing out again, where one can argue that harnesses are just duct tape for model limitations and get deleted as models improve. In fact, additional scaffolding might be harmful for model performance as the model reaches the next level. [3]
>
> We think the bitter lesson still applies on a macro-level, but the timeline is longer than people assume. We are still pretty far away from models that autonomously manage ten-hour sessions, handle enterprise security, and operate with audit trails without the support of harnesses.
>
> More importantly, **while the bitter lesson applies to** *raw* **model performance, it's less relevant for real-world use cases**. In enterprise settings, harnesses aren't workarounds, they're the product. Enterprises don't want a raw model that can do anything, they want a system that does specific things reliably, with governance and compliance. As models get more powerful, the harness shifts from compensating for model weakness to activating model capability in domain-specific ways.

## The "Car Market" is Wide Open

One of our biggest predictions going into 2026 is that foundation model companies will prioritize their own first-party tools. This is essentially **engine-makers building their own cars**. We saw this with Anthropic's fast iteration on Claude Code, Cowork, and Dispatch, and their ban on third-party harnesses. Meanwhile, Google and OpenAI are also building out their product ecosystems. Model companies have a structural advantage in building scaffolding for their own models, and they're moving fast.

Many think that the Claude Code leak is an existential threat to AI startups whose value proposition relied on "secret sauce" orchestration logic. We think it's the opposite.

Anthropic spent years and hundreds of millions figuring out production AI agent architecture. That R&D is now public. The repos analyzing the code became some of the fastest-growing on GitHub in history. Founders are already studying the playbook. A startup that would have spent 12-18 months on architectural exploration can skip straight to differentiation. The Claude Code leak neutralizes some of the advantages the model companies have, at least temporarily.

More importantly, **the leak decoupled two problems that used to be bundled**: **how to build a production agent** (now solved publicly) and **how to apply it to a specific domain** (still unsolved, still defensible). The leaked harness design patterns are domain-agnostic and can transfer anywhere from finance to healthcare. **What the leak didn't touch is the vertical moat**: domain data, regulatory expertise, workflow depth, and customer trust.

## The Vertical AI Thesis Accelerates

That decoupling is the key and it maps directly to what we've been arguing: vertical often beats horizontal. Broad horizontal solutions are the first to be subsumed by foundation models whereas vertical applications, though narrower in reach, retain users and justify premium pricing through domain depth and stickiness.

The leak didn't change the thesis. It made executing on it cheaper, faster, and more urgent. The infrastructure question, how to build a production agent that manages context, routes between tools, consolidates memory, and orchestrates sub-agents, is answered. The remaining question is who has the domain depth to build where foundation models can't follow.

What that looks like in practice plays out across three dimensions: how startups get funded, what separates winners from losers, and where the product gaps actually are.

## The Capital Paradox

The leak is **simultaneously deflationary and inflationary for startup capital**.

Deflationary, because R&D costs drop. The architecture exploration phase that used to require a principal engineer and a year of burn is now a weekend of reading.

Inflationary, because competition costs rise. When everyone gets the same playbook at the same time, the field gets crowded fast. You no longer differentiate on architecture, instead you have to differentiate on distribution, domain depth, customer relationships, and speed.

The net effect is that total capital required for a winning AI startup probably hasn't changed much. It's just been reallocated, from less R&D to go-to-market and forward-deployed engineering.

### The Sorting Function

Before the leak, startups could hand-wave about "proprietary architecture" as a moat. That ambiguity is gone. Everyone now knows what good engineering looks like, which forces competition onto what actually matters, things like UX, data flywheels, and enterprise trust.

This is a sorting function. If you're duct tape around someone else's engine, you're a feature, not a company, and the model makers just showed you exactly how they plan to ship that feature themselves. If you're a vertical agent in a domain the model company doesn't touch, you just got the hardest part of your stack for free with minimal competitive threat from below.

The wave of "Claude Code for X" companies that win won't be founders who read the leaked code most carefully. It'll be founders who were already deep in a domain and suddenly got an engineering roadmap to build at Claude Code's level of performance.

## Where Safety Ends and Governance Begins

The leaked code shows Anthropic has invested heavily in safety engineering. There are tiered permission systems, append-only audit logs, risk explanations, and sequences of checks around sensitive operations like shell commands.

But safety is not governance.

What's missing from Claude Code is a framework for the end customer and enterprise to define their own policies, control their own auditability, or verify agent behavior to a third party. The system is safe by Anthropic's standards, and Claude Enterprise adds another layer on top, but it still isn't configurable to the standards of a bank, a hospital, or a regulator.

The difference is subtle but critical. Safety is about preventing bad behavior inside the system. Governance is about making behavior legible, controllable, and accountable outside of it. A bank doesn't just need an agent that won't misbehave, it needs an agent whose behavior its own compliance team can configure, audit, and defend to a regulator.

This is where Claude Code stops and real enterprise agents begin. Model companies will keep building safety, b**ut governance that is customizable, industry-specific, and tied to real-world accountability is where startups still have a wedge.** For startups selling enterprise buyers, "we built what Anthropic skipped" becomes a genuine pitch.

## Where the “Cars” Are Going

$10B+ has been deployed into AI harnesses in 24 months. The leak suggests much of that was priced on architectural IP that turned out to be a design pattern. The next wave of investment should price on something different.

This maps to our investment thesis. We've argued for two years that durable AI value will concentrate not in the models themselves, but in the systems that execute real work. We call these **Systems of Action**. The Claude Code leak accelerates the shift because it commoditizes the infrastructure question of how to build a production agent. The application question is wide open, everything from what domain to own, what workflows to execute, what institutional memory to accumulate.

The leak shifted where moats live. Harness architecture used to be defensible, but now it's a design pattern. What remains defensible are things that don’t transfer like years of proprietary data, embedded workflows, institutional memory, and regulator relationships. That's what category-defining companies will build on.

The engine market is consolidating. The car market is wide open. The next $10B startup won't be built by someone who copied Claude Code's agent loop. It'll be built by someone who never needed to, because they were already deep in a domain the model companies will never touch.

That's where we're investing.

---

FOOTNOTES

1. This also suggests something about how these products get priced. When the same underlying model can produce wildly different results depending on orchestration quality, pricing based on access (SaaS seats) or raw usage (tokens) becomes misaligned with value. What customers actually want to pay for is whether the job gets done. In a way, the rise of model harnesses (and thus more powerful agents) will give rise to outcome-based pricing.
2. Its strong performance in scaffold-resistant benchmarks like Humanity’s Last Exam (HLE) and ARC-AGI-2 is evidence of a fundamental improvement in the transformer’s internal world model. And the improvements in scientific reasoning without access to external tools also suggests a deeper “latent knowledge” density.
3. OpenAI researcher Noam Brown used chain-of-thought harnesses to get GPT-4-level models to perform reasoning, now that’s native and these harnesses can actually slow down reasoning processes. More recently, tool-use required scaffolding but now models are trained on it directly.