

THE AI VALUE CHAIN

FUNDED OBSOLESCENCE

How AI Wrappers Finance Their Own Replacement

NABIL EL MAHYAOUI

Funded Obsolescence

Funded Obsolescence

How AI Wrappers Finance Their Own Replacement

Nabil El Mahyaoui

An essay on structural dependency in the AI value chain

Copyright © 2026 Nabil El Mahyaoui.
All rights reserved.

Contents

1 The Booster **1**

1.1 The Fastest SaaS in History 2

1.2 The Single Brain 4

1.3 The Anomaly 5

2 Funded Obsolescence **8**

2.1 The Double Trigger 8

2.2 The Spotify Precedent 10

2.3 The Choice of Executioner 12

2.4 The Open-Source Mirage 13

3 The Tenant Syndrome **16**

3.1 The Invisible Threat 17

3.2 The Mutual Hostage 18

3.3 The Full Map 19

3.4 Apple Is Not the Counter-Example . . . 21

4 The Synthesis Layer **23**

4.1 The Silent Deployment 24

4.2 The Surrender of the Screen 25

4.3 Two Fronts 27

4.4 The Metric Illusion 28

5 The Collision of Cycles **31**

5.1 Clock One: The Capability Leap 32

5.2 Clock Two: The Enterprise Sales Cycle 32

5.3 Clock Three: The Open-Weight Catch-Up 34

5.4 Clock Four: Infrastructure ROI 35

5.5 The Crossover Point 35

5.6 The Downstream Casualties 37

6 The Substrate Test **39**

6.1 Question One: Who Provides the Model? 40

6.2 Question Two: What Share of Revenue Flows to the Model Provider? . . . 41

6.3 Question Three: What Happens When the Model Provider Launches a Competing Product? 42

6.4 Question Four: Is This Tool Another Silo, or Is It Moving Toward the Synthesis Layer? 43

6.5 The Verdict. 44

6.6 The European Exposure 45

7 Epilogue: The Escape Velocity 47

7.1 Cursor's Gambit 47

7.2 Perplexity's Card 50

7.3 The Automation Paradox. 52

7.4 The Common Imperative 53

7.5 The Ultimate Scenario. 55

7.6 The Booster's Fate. 56

References 60

1

The Booster

When a space shuttle launches, the solid rocket boosters provide seventy percent of the initial thrust. They carry the structure through the zone of maximum aerodynamic pressure, burning their propellant at a rate that would terrify any accountant, and then they detach. The explosive bolts fire. The boosters fall in a graceful arc toward the ocean, their function complete, their future irrelevant. They existed

to exhaust themselves so that something else could reach orbit.

The technology industry is now celebrating startups that are not spacecraft. They are boosters.

And the perfect example is called Cursor.

1.1 The Fastest SaaS in History

If you are not a developer, you may never have heard of Cursor. But reducing it to a mere code editor would be a mistake. Thanks to its orchestration of autonomous agents and its intuitive interface, Cursor has become something else entirely. It is the tool that lets a financial analyst code his own prediction algorithm, an investor automate his entire market surveillance, an entrepreneur launch a functional ap-

plication from A to Z without writing a single line of code himself. It is the most fashionable vibe-coding tool on the market. The AI executes; the human pilots through natural language.

And that frictionless execution is what explains what happened next.

Cursor became the fastest-growing subscription software in history, generating $2 billion in annual recurring revenue. Three hundred employees. Roughly $6.7 million in revenue per head—eleven times the industry average. The narrative is perfect. The growth curve is vertical. The press coverage is rapturous.

Until you look at who built the engine.

1.2 The Single Brain

Officially, Cursor sells itself as a multi-model platform. You have choices. Claude, GPT, Gemini—take your pick. But talk to anyone in the industry, or scrape the recent leaks of their infrastructure invoices, and the narrative collapses quickly.

In reality, the overwhelming majority of Cursor users run on a single brain: the Claude model, manufactured by Anthropic.

The dependency is so complete that when Anthropic's servers go down, the entire Cursor community stops coding and screams on the forums. This is not a partnership. It is a life-support system.

And it is in the leaked internal financial analyses that the anomaly reveals itself. A sub-

scription sold at $200 per month can cost up to $5,000 in compute. The mechanics are clinically efficient: raise millions, incinerate them at maximum velocity to subsidise usage, make an entire generation viscerally addicted to AI-generated code. And the day they can no longer work without you, the jaw snaps shut and the real invoice arrives.

Let us be clear. Today, Cursor's users are not customers. They are the population being locked into a dependency. The $200 price tag is temporary. The habit they are building in you is permanent.

1.3 The Anomaly

The dominant narrative is seductive. Cursor is the fastest subscription software in history. Perplexity is reinventing online search. N8N

is democratising automation. The tech press celebrates these successes as proof that innovation remains possible in the face of the giants.

On this point, the press is right. These products are excellent. Cursor offers a better development environment than what the model makers propose alone. Perplexity structures answers better than ChatGPT for research. N8N lets you connect tools without writing a line of code.

But this reading is an anaesthetic. Behind every product, there is a financial flow. And those flows tell an entirely different story.

Here is what is happening concretely. You pay a Cursor subscription. A portion of that subscription is remitted to Anthropic for every call to the Claude model. Anthropic collects, reinvests in its research, and with that re-

search develops Claude Code—a product that does *exactly* what Cursor does.

The result: the more successful Cursor becomes, the more money it sends to Anthropic, the more Anthropic has the means to develop the product that competes with it.

It is a circle. But not a virtuous one. It is a structural trap. And if you think Cursor can simply switch providers—pivot to GPT or Gemini—you are confusing *changing landlords* with *becoming an owner*.

Every model that Cursor supports is manufactured by a company that is building its own coding tool. Claude Code. Codex. Gemini Code Assist. The multi-model strategy does not provide multiple escape routes. It provides three competitors.

2

Funded Obsolescence

I do not use the term “planned obsolescence,” because nobody planned this. It is worse. Every actor accelerates it rationally by paying their own invoices.

2.1 The Double Trigger

Cursor has built a brilliant product on two pillars that do not belong to them. The skeleton

belongs to Microsoft: it is VS Code. The brain belongs to other actors, principally Anthropic. This is the essence of what I call the *tenant syndrome*—building a dream villa on your neighbour's land. Every brushstroke on the interface proves that the value comes from elsewhere.

And it is here that funded obsolescence delivers the coup de grâce. You are not merely paying for terrain you will never own. Your money is paying for the bricks of the building that will crush you.

Consider the financial loop in its naked simplicity. Cursor's users pay Cursor. Cursor pays Anthropic. Anthropic reinvests in research. That research produces Claude Code, which already generates $2.5 billion in revenue—a product that surpasses the uncontested star of the market. And because

Anthropic owns the models directly, they can subsidise usage even more aggressively to asphyxiate Cursor.

This is funded obsolescence: each actor accelerates their own replacement by paying their bills on time.

2.2 The Spotify Precedent

Those who follow the technology industry will recognise the mechanism. The Spotify–Apple analogy has circulated for years. Spotify built the best music streaming service in the world—but it ran on the App Store. Apple took its commission, then launched Apple Music. Every Spotify subscription financed the competitor that wanted to destroy it.

But Spotify's reality is even darker, and it is exactly what awaits the AI wrappers.

For a long time, the conventional wisdom held that Spotify would survive thanks to Android—the perfect escape hatch. Except Android is Google. And Google pushes YouTube Music. The result: regardless of the operating system, Spotify pays rent to a giant that owns a competing service.

They tried to escape this trap by going vertical. That is why they invested hundreds of millions in exclusive podcast creators—to create value that Apple and Google could not tax. But the distribution trap was too strong. It was not enough.

Today, Spotify is condemned to raise its prices to survive. Quality stagnates. And it remains

at the mercy of the platforms. Cursor is the Spotify of AI.

2.3 The Choice of Executioner

And if you think Cursor can simply flee between GPT and Claude to switch to an alternative provider, look carefully at who the alternatives are.

If they flee to Google, they run straight into Gemini Code Assist. If they flee to OpenAI, they face Codex and Copilot. There is no neutral provider among the frontier labs. Every entity that owns the intelligence is now building the platform to exploit it.

The multi-model strategy does not offer Cursor three escape routes. It simply gives them

the choice of which executioner they will finance.

2.4 The Open-Source Mirage

This is generally where someone produces the miracle argument. The escape hatch that everyone hopes for: open source and Chinese models. The lifebuoy nobody dares discuss frankly.

Let us puncture the abscess immediately.

The open-source miracle—particularly the Chinese ecosystem with models like DeepSeek or Qwen—is, on paper, the perfect escape route. Their giant models innovate on architecture and demolish performance benchmarks for a fraction of the price.

But there is a formidable trap that those benchmarks never measure: distillation and real-world behaviour in production.

The problem with these models is that an immense portion of their training is distilled on the backs of the American giants. To put it simply: these models feed themselves by observing and cloning the responses of Claude. Let us be cynical for thirty seconds. No morality here—it is a perfect economic calculation, and it is what they *should* do. Spending a few million to capture a technology that cost billions to invent is intelligent. It is the software piracy of the 2000s, back with a vengeance. And everyone plays this game. We saw it concretely in February with the massive siphoning operation documented by Anthropic, where competing laboratories used

millions of automated requests to literally extract Claude's brain.

But here is where the mirage collapses. A distillation—even one augmented with logical reasoning—will always remain a lossy compression. It is the student who memorised the answer key instead of understanding the lesson. On a short task, or for demolishing the scores of a standardised test, the trompe-l'œil is absolute. But the value of artificial intelligence is migrating today toward autonomous agent work. When you ask the AI to reason autonomously for hours, to navigate unforeseen obstacles on a complex codebase, the distilled model eventually collapses. It simply does not have the depth of the original.

Switching to open source does not save the tenant. The tenant syndrome merely descends one floor in the stack.

3

The Tenant Syndrome

Let us widen the aperture. Cursor is not the only target in Anthropic's sights. It is simply the most visible. In their case, the assassin has a name and a face: Claude Code. The obsolescence is direct, a frontal replacement.

But the real danger is the second form of this trap. Furtive. Faceless.

3.1 The Invisible Threat

Here, the AI does not launch a competing application against you. It simply becomes capable of connecting to your tools directly. This is the poison now attacking the plumbing of N8N, Zapier, Make—all the way to Jira.

For those unfamiliar: these automation platforms do not have a model provider launching a head-on competitor. Their danger is not a product. It is a paradigm shift. The layer below them—the agents—is progressively absorbing their function.

When an agent knows how to connect two APIs, read a database, and trigger an action without passing through a visual platform, the visual platform loses its reason for existing. When the agent can lay the pipes itself, the plumber loses his job.

Cursor finances its competitor. N8N finances its own irrelevance. And this is harder to fight, because there is nobody on the other side from whom to reclaim market share. The market itself disappears.

3.2 The Mutual Hostage

But let us not paint Anthropic as an omnipotent predator without examining the constraints it faces. Today, the hostage-taking is mutual.

Of Anthropic's $14 billion in annualised revenue, only $2.5 billion comes from their own product, Claude Code. Everything else comes from the tenants paying to use their API. And Cursor is one of their largest tenants.

To put it simply: Cursor needs Anthropic's brain to exist. But Anthropic has a vital need for the millions of Cursors to pay for its own servers. They hold each other by the wallet—for now.

Except that this reciprocity has an expiration date. Claude Code doubles its revenue approximately every six months. The day Anthropic's direct revenues surpass its third-party API revenues, the incentive to preserve the wrapper ecosystem will vanish. On that day, funded obsolescence will graduate from structural pressure to verdict.

3.3 The Full Map

Let us trace the complete chain. The user pays Cursor. Cursor pays Anthropic. Anthropic runs on the servers of Google Cloud and Ama-

zon Web Services. Google has invested over $3 billion in Anthropic, giving it 14% of the company. And the chain does not stop there, because Anthropic is itself a tenant of Google Cloud and Amazon.

The model provider is caught in the same trap as its own customers, one floor down.

Google finances Anthropic, hosts Anthropic, and competes with Anthropic through Gemini. Amazon hosts Anthropic, invests in Anthropic, and pushes its own models with Nova. The pressure comes from everywhere. Every layer is simultaneously supplier, investor, and competitor of the layer above.

And this is where the picture becomes vertiginous. Google is not content to invest in Anthropic. Google pushes Gemini, which surpassed 750 million monthly active users

in February 2026—up from 650 million four months earlier. One hundred million new users in a single quarter.

Google is playing both sides simultaneously. If Anthropic wins, Google captures 14%. If Gemini wins, Google captures 100%. In every scenario, the house wins.

The illusion that a neutral actor exists in the world of AI shatters definitively. There are only owners and tenants on borrowed time.

3.4 Apple Is Not the Counter-Example

Someone will inevitably object: Apple does exactly the same thing by wrapping Gemini, and it works fine.

But Apple uses artificial intelligence as a loss leader. Their financial survival does not de-

pend on the price of AI. For Cursor, it is fundamentally different. At its core, Cursor is a token reseller. Artificial intelligence is 100% of their business, and the best user experience in the world will never save you if the engine you must rent to exist suddenly becomes prohibitively expensive against an Anthropic that owns the factory outright.

Their only contingency plan to save their margin would be to flee toward the distilled open-source ecosystem. But by accepting a downgrade in capability, they lose their edge.

The booster is condemned to never fly as high as the spacecraft.

4

The Synthesis Layer

One might think the trap stops here. But there is a final storey to this architecture that most analysts have not yet mapped: the synthesis layer.

Until now, we have watched value flow vertically—from the wrappers up to the model providers. What is now emerging is a horizontal concentration of value. An actor that plugs into all your silos—tickets, documents, CRM,

data—and becomes the default way to understand what is happening inside the enterprise.

4.1 The Silent Deployment

While everyone's eyes are riveted on Claude Code, watch what Anthropic is deploying in silence.

They are not content with an agent framework. They have launched the Claude Marketplace—the venue where large enterprises pre-approve tools and where any independent developer can sell their agents. But the real coup de force is elsewhere.

It is infiltrating the spinal column of your teams: Google Workspace, your data warehouses, and Atlassian—the publisher of Jira and Confluence.

The mechanics are surgical. Via an official Atlassian connector, Claude reads your tickets, summarises your documents, updates your tasks, and publishes pages. No need to open Jira or Confluence. Atlassian has even created a standardised protocol to deliver all of this data to Claude on a silver platter.

The result: from any interface, Claude surveys your tickets, your pages, and your projects as a single, unified territory.

4.2 The Surrender of the Screen

This is the implacable proof of a thesis I have articulated before. Atlassian has conceded defeat in the war for the screen—a major tactical sacrifice. They are ceding the user experience to Anthropic in order to lock in the monopoly on enterprise memory, now that the value is

no longer in the human clicking buttons on a workstation.

The battle is no longer about capturing your attention or training you to navigate interfaces. The stakes have shifted to feeding the AI. And this pattern is contaminating the entire market.

Claude already integrates with data giants like Snowflake, Kibana, and Datasol to query your metrics and hierarchies directly. Make no mistake: Claude is not trying to rebuild a new Salesforce or a new Atlassian. Its ambition is infinitely vaster. It grafts itself *above* them—above your messaging, above your data—to become the ultimate synthesis layer.

4.3 Two Fronts

To summarise: the jaws are closing along two distinct axes.

Frontal replacement. Claude replaces Cursor. A direct, visible, named competitor launches and captures the market from above.

Silent encirclement. AI comes to absorb all your software from within, imposing itself as the sole control tower of the enterprise. Value must therefore migrate to another layer.

The systems of record—Jira, Confluence, the CRM, the IT helpdesk—remain in place. But the question of *how do I know what is happening, how do I decide, how do I act*—that question can change owners. The first battle is about who writes the code and executes the tasks. The second is about who owns the bird's-eye view.

And if tomorrow a single interface—Claude or another—becomes the default way to read, cross-reference, and pilot all your data, your metrics, your tickets, and your documents, then that interface will be worth more than any isolated silo.

This is what I call the synthesis layer. And for now, most of the wrappers that the market celebrates are fighting to be one more silo—not to be *that* layer.

4.4 The Metric Illusion

Which reveals a deeper problem in how we evaluate these companies.

$2 billion in annual recurring revenue. When that number appears on LinkedIn, the reaction is instantaneous: the fastest subscription soft-

ware in history. But does this figure measure Cursor's success, or the size of the trap they have just entered?

Annual recurring revenue has always been an approximation of a company's solidity. This approximation worked in traditional SaaS because marginal costs were negligible. Every incremental dollar of revenue cost a few cents in server time. In AI, the mechanics invert. Every user query triggers a call to the model provider. The more users you have, the more you pay your provider—and that provider can raise prices, change terms, or launch a competing product. You have zero leverage.

Take the numbers. Cursor generates $2 billion in revenue with roughly 300 employees—$6.7 million per head. That is 11× the industry average. It looks like a micro-multinational, one

of those tiny teams generating the revenues of a major corporation.

But if the net margin after model costs is 30–40%, real revenue per employee falls to $2.7 million. Still impressive, but it is the model provider that captures the difference. When you read "$2 billion ARR," do not look at the gross number. Look at the margin after model cost. That is where the real solidity lives—and for now, that margin is compressed and structurally threatened.

5

The Collision of Cycles

Funded obsolescence is not merely a question of financial flows. It is an implacable mechanism that I call the collision of cycles. The drama of the AI wrappers is that they are stretched between four temporalities that do not turn at the same speed.

Listen carefully to how these four clocks collide.

5.1 Clock One: The Capability Leap

Every three to six months, the model provider releases an update. That is the precise amount of time it needs to integrate your flagship feature directly into its own system—and thereby render you obsolete.

This is not speculative. Claude Code did not exist eighteen months ago. Today it generates $2.5 billion. The feature set that made Cursor indispensable in January is a checkbox in the model provider's product by July.

5.2 Clock Two: The Enterprise Sales Cycle

Signing a large enterprise account takes six to eighteen months. The procurement commit-

tee meets quarterly. Legal reviews the contract. Security audits the infrastructure. The pilot runs for ninety days. The budget is approved in the next fiscal cycle.

The trap is now purely mechanical. At the precise moment the enterprise contract is finally signed, the product you just sold is already surpassed—because the raw model can do it all by itself.

You spend a year convincing the customer that your orchestration layer is essential. By the time the ink dries, the model has absorbed the orchestration.

5.3 Clock Three: The Open-Weight Catch-Up

Free and open-weight models—Llama, Mistral, and their descendants—take eight to twelve months to reach the level of the proprietary model you are using. For nearly a year, you have no escape route. You cannot flee to an alternative without degrading your product below market expectations.

This is your window of maximum vulnerability. You are locked into the provider, and the provider knows it. It can raise prices, change terms, or announce a competing product, and you have no credible threat to leave.

5.4 Clock Four: Infrastructure ROI

At the top of the chain, the cloud hyperscalers—Google, Amazon, Microsoft—need two to three years to recoup their billions in GPU and data-centre investments. This very long cycle forces them to apply monumental financial pressure on the model providers. And by domino effect, those providers press you in return—often by raising their tariffs or launching competing products.

You are trying to fit an eighteen-month sales cycle onto a substrate that mutates every three months. The tension is unsustainable.

5.5 The Crossover Point

Claude Code doubles its revenue approximately every six months. At this trajectory,

the day when Anthropic's direct consumer revenues surpass its third-party API revenues is not a distant abstraction. It is a date on a calendar.

On that day, the incentive to preserve the wrapper ecosystem disappears. The economics flip. Why nurture a thousand tenants when your own product generates more revenue, at higher margin, with fewer support tickets?

Funded obsolescence will pass from structural pressure to sentence. And the collision of cycles ensures that the wrappers cannot move fast enough to escape. They are trapped between a sales cycle that moves in years and a substrate that moves in months.

5.6 The Downstream Casualties

This mechanism contaminates the rest of the value chain.

Perplexity is caught in a pincer. On the consumer side, how do you compete with Google, which has 750 million users and global distribution? This battle is not technological. It is a question of acquisition cost. Google owns Chrome and Android. In the long run, distribution always defeats the model.

On the enterprise side, Perplexity's Search API—its research service for businesses—ranked seventh out of eight in independent evaluations, with eleven seconds of latency. The product is good, but it is squeezed between Google's distribution and the foundation models' raw performance.

As for the automation layer—N8N, Make, Zapier—the pressure comes from above and below. From above, agents are learning to orchestrate tasks directly without passing through visual automation platforms. From below, cloud platforms are natively integrating these capabilities.

On paper, N8N is valued at $2.5 billion on $40 million in revenue. Zapier books $400 million. But this duel is an illusion. The market they are targeting is being ground to powder by the two layers that bracket them.

6

The Substrate Test

Everything we have examined—funded obsolescence, the collision of cycles, the synthesis layer—constitutes the map of pressures. But these pressures shift every three months, and the technology stack you use today will not be the one you use in September.

The work we do here, essay after essay, is to trace these fault lines before they give way. And for that, you need a practical instrument.

Here is a tool you can apply on Monday morning to any AI tool you use or invest in. I call it the substrate test. Four questions.

6.1 Question One: Who Provides the Model?

If the tool you use is built on a third-party model—Claude, GPT, Gemini—it is structurally exposed to funded obsolescence. Note the provider.

This is not a disqualification. It is a risk factor. The question is whether the company has built enough value *above* the model to survive the day the model provider verticalises.

6.2 Question Two: What Share of Revenue Flows to the Model Provider?

The higher this share, the more fragile the company. If you cannot find this information, it is probably because it is not flattering.

When Cursor sells a $200 subscription and pays $5,000 in compute for power users, the gross margin is not merely compressed—it is inverted. The company is spending venture capital to purchase your loyalty, and the model provider collects the difference regardless.

6.3 Question Three: What Happens When the Model Provider Launches a Competing Product?

If the answer is "our user experience is superior," remember Spotify. User experience is not sufficient when the other party controls the platform.

The correct answer—the only defensible answer—is one of two things. Either: "we own proprietary data or capabilities that the model provider cannot replicate." Or: "our integration into the customer's workflow is so deep that switching costs exceed the benefit of the competitor."

Anything less is a prayer, not a strategy.

6.4 Question Four: Is This Tool Another Silo, or Is It Moving Toward the Synthesis Layer?

A tool that does one thing—even very well—is replaceable. An agent can learn to do that one thing.

A tool that connects your silos and becomes your default way of understanding what is happening in your enterprise—reading, cross-referencing, and piloting your data, metrics, tickets, and documents—that is a different strategic position entirely.

Most of the wrappers the market celebrates today are fighting to be one more silo. They are not fighting to be the synthesis layer.

6.5 The Verdict

If a tool you use fails all four questions—third-party model, compressed margin, no defence if the provider verticalises, just another silo—you are not a customer. You are a contributor. Your subscription finances the product that will replace it.

If you lead a team, audit your technology stack with these four questions. Identify the tools that rest entirely on a third-party model and prepare a contingency plan.

If you invest, before financing the next "fastest SaaS in history," ask: what is the margin after model cost? Gross revenue tells you nothing about durability.

If you build, ask yourself the question that Cursor and Perplexity are asking themselves

right now: does your advantage exist independently of the model you use? If the answer is no, you are a wrapper. And wrappers have a limited life expectancy.

6.6 The European Exposure

And for Europe, the situation is doubly exposed.

European enterprises use American models—Claude, GPT, Gemini—inside American wrappers—Cursor, Perplexity—running on American clouds. Three layers of dependency, not one of them under European control.

The Digital Markets Act regulates app stores and search engines, but nobody yet regulates the model layer. And that is where power is migrating.

When a European company pays an American wrapper, which pays an American model provider, on an American cloud, it is not merely financing a commercial competitor. It is financing a form of cultural interference and its own geopolitical downgrading. The tenant syndrome becomes a matter of state.

Even the actors at the heart of the European effort see this as an existential danger. Arthur Mensch, the CEO of Mistral, said it plainly before the students of École Polytechnique: his own generation of engineers had managed to commoditise their own profession. And he redefined American technology not as a simple efficiency tool, but as a vector of power and colonisation.

7

Epilogue: The Escape Velocity

We have traced the map. Now the concrete question: if I were at the helm of these companies, what would I do?

7.1 Cursor's Gambit

Cursor has already begun to move. They launched Composer, their own model—a mix-

ture of experts, meaning several specialised mini-models that cooperate, specifically designed to write code autonomously at high speed. Cursor positions it as four times faster than models of equivalent intelligence. It is their concrete attempt to break the dependency on Anthropic, to exit funded obsolescence.

And Cursor has real assets to succeed.

$2 billion in revenue to finance development. A weapon Anthropic does not possess: microscopic telemetry. Cursor sees every keystroke, every cancellation, every millisecond of developer hesitation. One million daily developers generating an interface feedback loop that a pure API cannot capture. Anthropic owns the general intelligence. Cursor owns the intimate map of how that intelligence is used inside the editor.

There is a second advantage, more technical, that most analysts underestimate. Cursor is installed on the developer's machine. It has direct access to the local environment—files, project context, real-time edit prediction, code merging. Claude Code has access to the same local files, but its terminal remains a visual bottleneck.

Cursor's true defence is to be the only environment capable of making autonomous execution readable and controllable by a human. But I will take the bet: within weeks, Anthropic will deploy its own solution to correct that problem.

The countdown is launched. The only real question—the quitte ou double that will define this industry's future—is who will move fastest. Will Composer be powerful enough to

break the dependency before the dependency breaks them?

If the answer is yes, Cursor graduates from booster to spacecraft. It would be the first case of a wrapper escaping funded obsolescence. If the answer is no, the explosive bolts fire.

By December 2026, either Composer reaches a level that significantly reduces the dependency on Claude, or Cursor begins losing share to Claude Code. There is no middle ground.

7.2 Perplexity's Card

Perplexity has a card to play, but it is not where the media are looking.

The consumer product—the search interface—cannot compete with Google's distribution

and its three-quarters of a billion users. That battle is not technological. It is a question of acquisition cost. Google owns Chrome and Android. In the long run, distribution always defeats the model.

But the enterprise Search API is a terrain where Perplexity can differentiate. Integrate its search into other companies' processes. Become the search provider for agents rather than for humans. The most revealing signal: in February, Perplexity killed its advertising business. When a company voluntarily eliminates a revenue stream, it is because it is betting on another.

My hypothesis: by December, either the Search API becomes the primary business, or the consumer product stagnates against Gemini.

7.3 The Automation Paradox

For N8N and Zapier, the situation is the most tense. Agents are learning to orchestrate tasks without passing through visual automation platforms. And cloud platforms are natively integrating these capabilities.

But there is a paradox that few people see. Enterprises despise agentic black boxes when it comes to critical processes—payments, CRM, sensitive data. When an agent hallucinates an infinite email loop, the human wants a visual graph to pull the plug. N8N could survive not as a tool for *creating* automations, but as a dashboard for *monitoring and constraining* agents.

A radical pivot—from constructor to controller. But it may be the only terrain the agents cannot absorb: the terrain of human trust.

My hypothesis: by December, at least one of the three announces an absorption partnership or a major strategic pivot.

7.4 The Common Imperative

In all three cases, the imperative is the same. Accept that your current product is temporary. Cannibalise your own revenue to migrate toward a layer that the model provider cannot reach.

This is exactly what Spotify eventually did: invest massively in podcasts and exclusive content—assets that Apple does not control. Cursor, Perplexity, and N8N must find their podcast equivalent. For now, only Cursor has begun searching, with Composer.

There is one other exit: open-weight models. But this is where the eight-to-twelve-month catch-up cycle becomes vital. This escape route exists only if you survive long enough for Llama or Mistral to match Anthropic's level on your use case. If you hold out that long and switch your orchestration to self-hosted models, you instantly stop financing a direct competitor.

That is the structural exit from funded obsolescence. But survival has a price. On the most complex tasks, open-weight models still lag. And running them demands colossal compute. Moving from Anthropic to a self-hosted Mistral is refusing to pay rent on your apartment to buy a house—and discovering you are now hostage to the electric company.

The tenant syndrome does not disappear. It merely descends one floor in the stack.

7.5 The Ultimate Scenario

There is a final scenario this analysis must account for: the commoditisation of intelligence itself. If models all become excellent and the price war intensifies—which is already visible in API costs—the cost of intelligence will trend toward zero.

Classical tech theory says that at that point, value rises to the interface. On paper, Cursor would become the great winner of this price war.

But this reading is incomplete and potentially false. If agentic work becomes truly perfect—if the machine plans, codes, and self-corrects for hours—the Cursor interface itself becomes obsolete. Why would you need a complex code editor with file trees and thousands of colour-

coded lines if the AI does all the work in the background?

In a world where the agent is perfectly autonomous, you do not need a tool to write code. You need a messaging interface—or better, a terminal—to give the order, and a dashboard to validate the result.

Even if intelligence becomes free, the expert wrapper is not saved. It will simply be erased by a much simpler supervision interface.

7.6 The Booster's Fate

Most AI startups that you see celebrated today—the record fundraising rounds, the exponential revenue curves, the fastest-growing SaaS in history—are solid rocket boosters. They provide the initial thrust. And when the

spacecraft reaches orbit, the explosive bolts detonate.

They exist in the gap between what the model providers *can* do and what they *choose* to do. The day the provider decides to compete—as Anthropic has done with Claude Code—the gap closes.

Let us be lucid about the limits of this thesis. A product that offers a superior user experience, deep enterprise integration, or a loyal community *can* survive even if its provider verticalises. Spotify is the proof. Funded obsolescence is not an automatic death sentence. But it is a structural pressure, and the pressure intensifies when the model providers no longer need to force anything. The invoices do the work for them.

What is true for Cursor facing Anthropic is true for you facing your tools. The question is not which tool has the best interface. It is: in which layer do you sit? Are you the spacecraft, or the booster?

And there is a contradiction you should notice. If Anthropic is the predator devouring everything, why would I push you to use precisely their tools?

The answer is purely tactical. When value migrates violently downward, clinging to wrappers is applying the tenant syndrome to your own career. In this industry, you are either the prey, or you climb on the predator's back.

The only viable move today is to learn to work directly on the layer that is absorbing the entire market. No intermediaries.

The diagnostic is yours. The tool to stop being the booster is right there. The only question left is how many of your current subscriptions are financing the product that will replace them—and for how many of those do you have a real contingency plan the day the provider decides to verticalise.

Take the inventory. And look around you. In your circle, there is a founder, an investor, or a developer paying a monthly subscription to a tool without knowing that their money is directly financing the product that will make it obsolete.

Sending them this analysis is the best wake-up call you can give. The best service you can render before the explosive bolts fire.

References & Further Reading

Thompson, B. (ongoing). *Stratechery*. *Essential analysis of platform economics and the AI value chain.*

Anthropic (2026). "Detecting and Countering Automated Model Extraction." *Documentation of the massive siphoning operation targeting Claude's capabilities.*

Christensen, C. M. (1997). *The Innovator's Dilemma*. Harvard Business Review Press. *The foundational text on why incumbents fail—and why wrappers are not incumbents.*

Mensch, A. (2025). Address at École Polytechnique.
On the commoditisation of engineering and the geopolitics of AI dependency.

Rifkin, J. (2014). *The Zero Marginal Cost Society*. Palgrave Macmillan.
On what happens when the cost of production trends toward zero—now applied to intelligence.

Ek, D. (various). Spotify earnings calls, 2019–2026.
The original case study of a wrapper trapped by its platform providers.

a16z (2024–2026). Research on the AI application layer and "Service-as-Software."
Venture capital perspective on value migration from wrappers to models.

Cursor (2026). Composer model announcement.

The first serious attempt by a wrapper to break model dependency through in-house intelligence.

About the Author

Nabil El Mahyaoui

https://nabil-el-mahyaoui.com/

https://www.linkedin.com/in/elmahyaoui