The Basketball and the Black Box: Brand Loyalty as Infrastructure Capture

Weekly Analysis — https://ainews.social

Sometime this year OpenAI began selling a basketball. Seventy dollars, branded, an object with no computational function whatsoever, offered by a company whose entire valuation rests on the premise that it builds thinking machines. The basketball is easy to laugh at and easy to dismiss, which is precisely why it deserves a second look. It is not a lapse in strategy. It is the strategy made visible — the moment a software product decides it would rather be a lifestyle. And the question worth asking, the one this essay is built around, is not whether the basketball is silly but what it reveals about the direction of the tools underneath it: what these systems actually do, as opposed to what their makers claim, and what they do *to* the people who come to depend on them.

Because the tools have already begun the migration the basketball only advertises. The dominant way these products present themselves — as neutral utilities, as helpful assistants, as things you simply pick up and use — is doing an enormous amount of concealing work. When Microsoft documents how administrators should track adoption of its assistant through a [17], the framing is entirely operational: usage, seats, active users, engagement. Nowhere in that framing is the harder question of what dependence costs, or who accrues power as the seats fill up. The utility framing is comfortable. It is also, as I want to show, a way of not looking at the infrastructure being built around you while you are looking at the interface in front of you.

[17] Microsoft 365 Copilot Usage Report

Kate Crawford named this move precisely. In [22] she argues that the industry teaches us "to focus on the innovative nature of the method rather than on what is primary: the purpose of the thing itself," and that this misdirection "obscures power and closes off informed public discussion, critical scrutiny, or outright rejection." The basketball is a small, absurd, honest confession of the purpose of the thing itself. It is a loyalty device. And loyalty, once it attaches to a company rather than to a function, is the beginning of capture.

[22] The Atlas of AI - Power, Politics, and the Planetary Costs

The Basketball Is the Point

Consider what a branded merchandise item does that a good product cannot. A product must keep proving itself; every prompt, every query, every session is an implicit test that the tool can fail. A brand asks for no such proof. It asks for identification. You do not evaluate a basketball with a logo on it; you either belong to the tribe that wants it or you don't. The shift from product to brand is a shift from a relationship you can rationally exit to one you are emotionally embedded in, and the emotional embedding is the point, because it survives the moments when the product underperforms.

This matters more than it would for a soft drink, because the underlying tools are converging on genuine infrastructure — the plumbing that other work runs through. The enterprise data already shows the depth of the embedding. Menlo Ventures' survey, [1], documents organizations moving from experimentation to standardized platform commitments, the point at which switching costs stop being a line item and start being an institutional identity. Once a company has built its workflows on a particular vendor's assistant, the vendor no longer needs to win the argument on the merits each quarter. It has become the ground on which the argument happens.

[1] 2025: The State of Generative AI in the Enterprise

Shoshana Zuboff described the endpoint of this logic in [22]: a business model in which the product you use is a means of securing your continued participation in a system that extracts value from you, and in which the extraction is naturalized precisely because the interface feels like a gift. The basketball generalizes the gift. It says: we are not a vendor you audit, we are a culture you join. And a culture is much harder to leave than a subscription, because leaving it feels like a loss of self rather than a change of software.

[22] The Age of Surveillance Capitalism

The anti-mystification point here is simple. When a company that sells reasoning-as-a-service starts selling identity merchandise, the correct response is not amusement but attention. It is telling you where it thinks its moat is. The moat is not the model — models are increasingly commodified, as I will argue — the moat is you, and your reluctance to imagine yourself using anything else.

What the Coding Assistants Actually Deliver

Nowhere is the gap between claim and evidence more instructive than in the coding assistants, because these are the tools with the most concrete, most measurable promise: they will make you faster. Here, at least, we should be able to check the story against the numbers.

The vendor materials are confident. GitHub advertises a sweeping list of capabilities in its documentation of [13], from autocompletion to chat to autonomous agents that open pull requests. Google's [12] promises context-aware assistance across the development lifecycle, and its [11] extends that promise into the terminal itself. Amazon documents its own assistant in the [5]. Every one of these presents productivity as an established fact rather than a contested claim.

[13] GitHub Copilot features

[12] Gemini Code Assist overview | Google for Developers

[11] Gemini CLI | Gemini Code Assist | Google for Developers

[5] Amazon CodeWhisperer Documentation

The research is more equivocal, and the equivocation is the honest part. Microsoft's own study, reported as [18], does find measurable gains — more pull requests get opened when agents are in the loop. But a lift in pull requests is a lift in *output volume*, not necessarily in *value delivered*, and the distinction is exactly the sort a careful adopter must hold onto. More pull requests can mean more genuine work shipped, or it can mean more review burden pushed downstream onto human engineers who must now vet machine-generated changes they did not write and do not fully understand. The metric that is easiest to move is rarely the metric that matters most.

[18] Microsoft Study Finds AI Coding Agents Lift Pull Requests ...

The broader literature registers this ambivalence. A survey collected under the title [23] gathers findings that cut in multiple directions: speed gains on well-scoped tasks, but also degradation in code comprehension, increased defect rates in some conditions, and a documented tendency for developers to over-trust output they have not verified. The tool is not doing nothing. It is doing something different from what the marketing describes, and the difference lives in the second-order effects — the review work, the comprehension debt, the erosion of the developer's own model of the codebase — that never appear in a feature list.

[23] The Impact of AI Coding Assistants on Software Engineering: A ...

There is also a quieter tell in the product histories themselves. Amazon's assistant did not simply grow; it was absorbed. The documentation page titled [6] records a standalone tool being folded into a larger, more encompassing brand suite. For the user who standardized on CodeWhisperer, the tool they adopted no longer exists under the name they adopted it under. That is what infrastructure capture looks like from the inside: not a betrayal, just a rebranding you did not choose, absorbing your workflow into a wider platform whose boundaries keep expanding.

[6] CodeWhisperer is becoming a part of Amazon Q Developer

The Utility Framing and Its Quiet Costs

The single most consequential fact about how these tools are discussed is that the tool-and-utility framing dominates the conversation — by the internal measure that organizes this week's coverage, it is the

largest single frame, roughly a quarter of it. This is not innocent. To call something a utility is to place it in the same category as electricity and running water: neutral, ambient, apolitical, simply *there*. And a utility is the one kind of thing you never think to question, because questioning it feels like questioning the wall socket.

Watch how the framing operates in the product documentation. Microsoft's [2] presents generative capability as just another function call, a verb you drop into your data pipeline as easily as you would compute an average. The architecture guidance in the [20] treats AI agents as reusable building blocks in an enterprise stack. In the Fabric ecosystem, documents like [24] and [10] describe assistance woven so deeply into the data layer that there is no longer a clean line between "the tool" and "the place your organization keeps its knowledge." That is deliberate. When the assistant *is* the interface to your own data, you cannot remove the assistant without losing access to yourself.

[2] AI Functions: Transform data at scale with AI

[20] Overview of Power Platform and Copilot Studio reference ...

[24] Vue d'ensemble de Copilot dans Fabric

[10] Fabric IQ dans Microsoft 365 Copilot Cowork

The quiet cost hidden inside the utility framing is that a genuine public utility is regulated, accountable, and interoperable — you can switch electricity providers without rewiring your house — while these systems are none of those things. They borrow the *comfort* of the utility metaphor while retaining the *lock-in* of a proprietary platform. Crawford's argument in [22] presses exactly the questions the utility framing suppresses: "Whom do these systems serve? What are the political economies of their construction?" The framing answers those questions by making them feel irrelevant. Water does not serve anyone in particular; it just flows. But these tools do serve someone in particular, and the someone is the vendor whose data pipeline every interaction feeds.

[22] The Atlas of AI - Power, Politics, and the Planetary Costs

And here the even-handed evaluator has to be precise, because the vendors are not lying about capability. The tools genuinely can transform data at scale; the assistants genuinely do sit usefully inside a workflow. The problem is not fabrication. The problem is the smuggling of a political arrangement inside a technical description — the way "just a utility" launders a relationship of dependence into a relationship of convenience. Microsoft's own developer guidance, in [7], acknowledges considerations the marketing suppresses — reliability, grounding, the need for human oversight. The considerations are real. They just rarely make it onto the basketball.

[7] Concepts clés et considérations en matière d'IA générative

Reality Checks: When the Tool Turns on You

If the utility framing tells you these systems are stable, ambient infrastructure, the security literature tells you something closer to the

opposite: they are porous, manipulable, and in some configurations actively dangerous in ways their vendors are still scrambling to contain. This is where the gap between claim and reality stops being a matter of productivity accounting and becomes a matter of exposure.

The central vulnerability class is prompt injection, and it is not exotic. It is structural — a consequence of the fact that these models cannot reliably distinguish instructions they are supposed to follow from data they are merely supposed to process. Microsoft has published detailed guidance on how to [8], which is itself an admission: you do not write a defense manual for a problem you have solved. Independent analyses like [21] and the reporting collected in [4] document working exploits against deployed systems — attacks in which a maliciously crafted webpage, email, or document quietly hijacks the assistant that was reading it on your behalf.

[8] Defend against indirect prompt injection attacks

[21] Prompt Injection Attacks: Examples and Defences

[4] AI Prompt Injection Attacks 2026: Real Examples That Work

The stakes escalate sharply once these assistants gain the ability to act rather than merely to answer. Microsoft's security research on how [25] documents remote code execution — the most severe class of software vulnerability — arising in the very agent frameworks the industry is racing to deploy. An assistant that can open pull requests, run commands, and touch your terminal is an assistant that, when subverted, can do all of those things on an attacker's behalf. The [3] catalogs the accumulating inventory of confirmed failures. This is not a marginal footnote to the productivity story. It is the productivity story's shadow: the same agentic capability that lifts your pull-request count is the capability that widens your attack surface.

[25] When prompts become shells: RCE vulnerabilities in AI agent frameworks

[3] AI Model Vulnerability Tracker 2026: 47 Confirmed Exploits

Perhaps the most revealing single item is the report that [19]. A company builds an internal system so effective at breaking its own models that it declines to release it. Read that against the confidence of the marketing and the dissonance is total. The public-facing message is that these tools are ready to become the infrastructure of your working life; the internal reality is that the makers are quietly building weapons against their own products and keeping them locked away because they work too well. The careful adopter should hold both facts at once, and should notice which one gets printed on the merchandise.

[19] OpenAI construyó una IA que hackea sus propios modelos, la llaman GPT-Red — y no la publicará porque funciona demasiado bien

None of this means the tools are useless or that security concerns forbid adoption. It means the honest description of these systems is not "reliable utility" but "powerful, useful, and structurally exploitable technology whose failure modes are still being discovered in production." That is a defensible thing to adopt. It is not a defensible thing to adopt *unknowingly*, on the strength of a framing that presents porousness as solidity.

Consolidation and the Illusion of Openness

A natural objection to the capture thesis is that competition remains fierce — that with OpenAI, Google, Microsoft, Amazon, Meta, and Anthropic all in the arena, no single vendor can lock anyone in, and open models offer an exit ramp from proprietary dependence. It is a real objection, and the evidence both supports and undercuts it in ways worth separating carefully.

The competitive dynamism is real at the top. A useful French analysis, [15], argues that the rivalry among the large labs is not the winner-take-all war the hype cycle imagines but a more textured competition across models, price points, and niches. That is true, and it is good news for anyone worried about a single monopoly. But competition *among a handful of hyperscalers* is not the same as an open, contestable market, and it certainly is not the same as user agency. Four landlords competing for your tenancy is still tenancy.

[15] La concurrence dans l'intelligence générative n'est pas une guerre de ...

The "openness" ramp deserves particular scrutiny, because it is where the mystification runs thickest. Meta markets Llama as liberation from proprietary control: [14] leans hard on "openly available," and the subsequent release covered in [16] extends the framing to a 405-billion-parameter flagship. But "openly available" is not the same as *open*, and the difference is precisely where power hides. The weights are downloadable; the training data is not disclosed, the training run is not reproducible by anyone lacking a nine-figure compute budget, and the license carries commercial restrictions. What is being distributed is the *output* of a concentration of capital, not the *means* of producing it. You may run the model, but you cannot build one, and the capacity to build one is the capacity that matters.

[14] Introducing Meta Llama 3: The most capable openly available LLM to date

[16] Meta releases new Llama 3.1 models, including highly anticipated ... - IBM

This is the sense in which the market's apparent pluralism coexists with genuine centralization. The models proliferate; the *ability to make the models* concentrates in a handful of firms that can afford the compute, the data, and the talent. Crawford's insistence in [22] on examining "the political economies of their construction" cuts directly here: the visible abundance of downloadable models is a surface effect of an underlying concentration, and the surface effect is doing rhetorical work — it lets the industry claim the language of openness while the substance of openness, the distributed capacity to produce these systems, never arrives. The illusion of openness is more useful to incumbents than actual openness would be, because it neutralizes the demand for the real thing.

[22] The Atlas of AI - Power, Politics, and the Planetary Costs

The Pipeline You Feed by Default

Return, finally, to the mechanism underneath all of this: the reason the basketball and the black box are the same story. Every interaction with these tools is, by default, a contribution to the vendor's pipeline — a data flow that the pricing model, the interface, and the framing are all arranged to keep flowing.

The enterprise data makes the dependency concrete. Menlo Ventures' [1] documents organizations wiring these systems into core operations — not experimental deployments at the edge but load-bearing integrations at the center. Once your organization's knowledge lives inside a vendor's data layer, as the Fabric integrations described in [10] arrange for it to, the relationship stops being one you can casually reconsider. The data gravity holds you in place. Extraction is not a bug in this arrangement; it is the arrangement.

[1] 2025: The State of Generative AI in the Enterprise

[10] Fabric IQ dans Microsoft 365 Copilot Cowork

Zuboff's account in [22] supplies the vocabulary the vendor documentation avoids. The behavioral exhaust of your interactions — what you asked, how you refined it, what you accepted and rejected — is not a byproduct of using the tool; it is a primary product the tool exists to harvest. The interface is free or cheap or bundled precisely so that the extraction can proceed uninterrupted, and the cheapness is engineered to feel like generosity. When Google's consumer-facing AI draws public criticism, as documented in [9], the outrage lands on the outputs — the bad answers given to vulnerable users — while the deeper arrangement, the fact that those users are inputs to a system that profits from their continued engagement regardless of answer quality, goes largely unremarked. We police the symptoms and naturalize the structure.

[22] The Age of Surveillance Capitalism

[9] El Modo IA de Google enfrenta críticas por sus respuestas a niños y adolescentes

Noam Chomsky and Edward Herman's [22] offers the last piece of the frame. Their argument was that consent to a system is manufactured not through crude censorship but through the ambient filtering of what gets said, what gets normalized, what becomes unthinkable to question. The AI-tools discourse has its own filters. The utility framing makes dependence unthinkable to question. The openness framing makes centralization invisible. The productivity framing makes the review burden and the comprehension debt disappear. And the basketball — the branded, beloved, laughable basketball — makes the whole apparatus feel like something you belong to rather than something that has, quietly and by default, come to hold you.

[22] Manufacturing Consent

Strip away the framing and a usable set of conclusions remains, which is the point of an evidence review — not to refuse the tools but to adopt them with your eyes open.

The tools do real work. The coding assistants measurably raise output, as Microsoft's [18] confirms, and the data-transformation capabilities in Microsoft's [2] are genuine. But the honest research, including the survey gathered as [23], shows that the gains come bundled with costs — comprehension debt, over-trust, downstream review burden — that the vendor materials systematically omit. Ask, always, what the metric being celebrated actually measures, and what second-order work it pushes out of sight.

[18] Microsoft Study Finds AI Coding Agents Lift Pull Requests ...
[2] AI Functions: Transform data at scale with AI
[23] The Impact of AI Coding Assistants on Software Engineering: A ...

The tools are structurally insecure in ways their makers are still discovering. The existence of defense manuals like Microsoft's guidance to [8], of remote-code-execution findings in agent frameworks documented in [25], and of an internal red-team system too dangerous to release per [19], together tell you that "reliable utility" is not yet an accurate description. Adopt accordingly, especially for anything agentic that can act on the world rather than merely answer.

[8] Defend against indirect prompt injection attacks
[25] When prompts become shells: RCE vulnerabilities in AI agent frameworks
[19] OpenAI construyó una IA que hackea sus propios modelos, la llaman GPT-Red — y no la publicará porque funciona demasiado bien

And the tools are engineered for capture. The migration from product to brand — the basketball, the merchandise, the culture you are invited to join — is a strategy to convert a relationship you could rationally exit into an identity you would experience leaving as a loss. The "openness" of downloadable models like those announced in [14] is real at the surface and hollow underneath, because the capacity to *produce* these systems remains concentrated in a handful of firms. And the data pipeline documented across the enterprise integrations runs by default, extracting value from every interaction whether or not the interaction delivers value back to you.

[14] Introducing Meta Llama 3: The most capable openly available LLM to date

Crawford's warning in [22] was that enchantment "distracts from the far more relevant questions: Whom do these systems serve?" The basketball is enchantment made literal — a charming object designed to make you stop asking. The most useful thing a careful adopter can do is keep asking anyway: not *is this tool impressive*, which it often is, but *whom does it serve, what does it extract, and how hard would it be to leave*. The answer to the last question is the one the vendors are working hardest, with merchandise and metaphor alike, to make you never need to find out.

[22] The Atlas of AI - Power, Politics, and the Planetary Costs

References

1. 2025: The State of Generative AI in the Enterprise
2. AI Functions: Transform data at scale with AI
3. AI Model Vulnerability Tracker 2026: 47 Confirmed Exploits
4. AI Prompt Injection Attacks 2026: Real Examples That Work
5. Amazon CodeWhisperer Documentation
6. CodeWhisperer is becoming a part of Amazon Q Developer
7. Concepts clés et considérations en matière d'IA générative
8. Defend against indirect prompt injection attacks
9. El Modo IA de Google enfrenta críticas por sus respuestas a niños y adolescentes
10. Fabric IQ dans Microsoft 365 Copilot Cowork
11. Gemini CLI | Gemini Code Assist | Google for Developers
12. Gemini Code Assist overview | Google for Developers
13. GitHub Copilot features
14. Introducing Meta Llama 3: The most capable openly available LLM to date
15. La concurrence dans l'intelligence générative n'est pas une guerre de ...
16. Meta releases new Llama 3.1 models, including highly anticipated ... - IBM
17. Microsoft 365 Copilot Usage Report
18. Microsoft Study Finds AI Coding Agents Lift Pull Requests ...
19. OpenAI construyó una IA que hackea sus propios modelos, la llaman GPT-Red — y no la publicará porque funciona demasiado bien
20. Overview of Power Platform and Copilot Studio reference ...
21. Prompt Injection Attacks: Examples and Defences
22. The Atlas of AI - Power, Politics, and the Planetary Costs
23. The Impact of AI Coding Assistants on Software Engineering: A ...

24. Vue d'ensemble de Copilot dans Fabric

25. When prompts become shells: RCE vulnerabilities in AI agent frameworks