$32,341 of AWS spend, 10.5 million product pages, 9% reliability: the receipt for the bet that taught SHORA why deterministic web task automation had to exist

This is Part 2 of a three-part story. Part 1 explained our bet. Part 3 names what we built. This part is what happened in between.

In Part 1, we explained the missing measurement layer in e-commerce (**Experience**). We bet on instrumenting the Experience on the web (public surface) at scale, by a sufficiently well-scaffolded language model. This part is the receipt for that bet. We report on what the language model produced when it met the 10.5 million product pages whose purchase journeys we asked it to instrument, and why the data forced us to build something else instead.

The receipt

We spent $32,341 on AWS for this experiment: a state-of-the-art agent based on Bedrock Claude Opus 4.6, built to read live retailer web pages and reconstruct the full purchase journey from product detail page (PDP) to cart. The agent ran against 10.5 million product pages spanning 1,056 French retailers from 480 locations — one journey per product page, every journey traversing at minimum the PDP and the cart, with the agent expected to extract a consistent record at every step. The completion rate was 9%, meaning the rate at which the agent successfully navigated a journey end-to-end on the live retailer site, before we even classified what was captured at the cart (the expected outcome where the product was added with the PDP price intact, or one of the two anomalies that mattered: a stockout, or a price mismatch where the cart price exceeded the PDP).

Readers who want to reproduce the experiment can read the technical companion post, which documents the open-source `pji-agent` codebase, the journey-orchestrator loop, and the prompt scaffolding under MIT license. Readers who want the line-by-line breakdown of where the $32,341 went across AWS services and the February 2026 burn curve can read the financial detail companion post.

You may wonder that the first failure mode must have been anti-bot defenses. It was not. The Camoufox browser stack the experiment ran on bypassed roughly 92% of every bot defense it encountered, from Akamai to Cloudflare and DataDome. The agent reached the web pages it was tasked to reach. **The 9% is the agent's success rate at instrumenting the purchase journey.**

Why the architecture hit a ceiling

The pipeline did not hand the model raw pixels and ask it to find things. Every screenshot was annotated with bounding boxes, and every bounding box carried a unique identifier traceable back to the DOM element it framed. As a result, identity was already solved before the model started reasoning. What remained was semantic grounding: looking at every addressable box on the page and deciding which one was the price the buyer would pay, which was the trigger to add the item to the cart, and which was the cart total after adding it, etc.

To decide what each box means, the model has nothing to fall back on except its training prior, applied one inference at a time. That is what an LLM is: a stateless next-token predictor with frozen weights. SHORA's pipeline used every adaptation mechanism the production stack offers on top of that base — in-context learning, retrieval-augmented memory, and agent scaffolding. None of them turn frozen weights into a system that gets better the more pages it sees in production. The bill grows linearly with volume; the accuracy does not.

This is not a SHORA-specific finding. It is the published shape of the entire web-agent literature. The canonical academic benchmark is **WebArena** [Zhou et al., ICLR 2024], an environment of 812 long-horizon web tasks that emulate the work humans routinely perform on the internet. When the paper was published, the best GPT-4-based agent reached 14.41% end-to-end success, against a human baseline of 78.24% — a gap the authors named explicitly. Two years and dozens of submissions later, the live WebArena leaderboard shows the SOTA at 74.3% (Deepseek v3.2, February 2026), still below the human baseline. Anthropic's reported 77.5% for Claude Opus 4.5 [November 2025 System Card] sits within that leaderboard, not above it.

The deeper point is that even those numbers come from a curated, instrumented, deterministic environment. The same agents fail much worse on the open web. The paper *An Illusion of Progress? Assessing the Current State of Web Agents* [Tao et al., COLM 2025] introduced a realistic 300-task benchmark spanning 136 live websites precisely because the headline numbers on the older benchmarks were not generalizing. On the realistic version, the 2025 frontier dropped to 61%. A follow-up paper, *WAREX* [2025], stress-tested released agents on WebArena, REAL, and WebVoyager under production-like conditions and concluded that agents "exhibit severe degradation under realistic conditions, exposing fundamental robustness gaps." On **OSWorld** [Xie et al., NeurIPS 2024], the broader computer-use benchmark, the original paper reported a human baseline of 72.36% against a best-model score of 12.24% — the gap the authors named in their abstract.

SHORA's experiment ran on 10.5 million product pages across 1,056 different retailer websites from 480 French locations, with one full PDP-to-cart journey attempted per product. Every per-step error compounded across the chain. The 9% is well below the curated-benchmark SOTA (74.3% on WebArena) and well below the realistic-benchmark frontier (61% on Online-Mind2Web), which is the exact direction *An Illusion of Progress?* and *WAREX* predict for any

deployment in which the agent meets the open web, deeper task chains, and DOMs that change without warning.

And the answer is not to hire humans

The instinctive substitute for an unreliable model is a reliable person. The academic evidence against that substitution is older and more reproduced than any benchmark in the previous section. In 1948, Norman Mackworth — commissioned by the Royal Air Force to understand why radar operators missed German submarine signatures during long shifts — ran the experiment that founded the field of sustained-attention research. Operators on a two-hour vigilance task showed a steep drop in detection accuracy within the first 30 minutes, followed by a steady decline. The phenomenon, now called the *vigilance decrement*, has been replicated continuously for seventy-five years across radar, screening, content moderation, and data-labeling tasks ["The vigilance decrement: its first 75 years," Frontiers in Cognition, 2025]. Mackworth's own finding stood: telling subjects to be more attentive had no effect on the decline. The decrement is a property of the human attention system on repetitive monitoring tasks, not a property of motivation.

The same architecture of degradation appears in data-labeling research, where it is measured rather than observed. Cohen's kappa [Cohen, 1960] is the standard third-party metric for inter-rater reliability — the rate at which two trained annotators applying the same categorical task to the same items actually agree. Multi-annotator studies on large-scale labeling consistently find that kappa values drop below acceptable reliability thresholds on a meaningful share of categories, even with trained annotators, and the industry standard response is to pay three to five annotators per record and aggregate, rather than to expect a single human to deliver consistent labels at scale ["Assessing Inter-Annotator Agreement for Medical Image Segmentation," PMC, 2023].

The page-by-page reviewer is a strawman; nobody runs catalogs of tens of thousands of products that way. The production substitute for the LLM agent — wherever it runs today — is a piece of automation software that someone has to write, maintain, and repair whenever the page it reads changes: a custom scraper, a vendor tool, or, increasingly, code an AI wrote once. Having an AI write it does not change the architecture. The code is still frozen judgment, and keeping it alive against a web that shifts under it is still a repetitive monitoring task — the exact kind of vigilance work the Mackworth literature above says humans degrade at. The breaks you catch are cheap; the silent ones are the ones that bleed money while everyone trusts the dashboard. That is the third ceiling, and it is the same shape as the first two.

> Take the three failure modes together. **The model fails because frozen weights cannot adapt across pages.** The human reviewer fails because the vigilance system declines within the first half-hour of any repetitive monitoring task, and the inter-rater agreement across humans on the same task is not high enough to substitute reliability for memory. The

> engineer-with-a-scraper — the real production alternative — does not fail on attention but on the silent errors a drifting tool returns between repairs, which is itself the vigilance task humans degrade at. A buyer who needs the same web pages read correctly tens of thousands of times a month, every month, for years, is not choosing between a flawed AI and a reliable human. The buyer is choosing between three ceilings, all real, all measured — and none of them delivers a measurement layer the buyer can sign for.

The admission

The agentic LLM architecture had been pitched, internally and externally, for the better part of a year before this data came in. The conviction was mathematical: with enough prompting, enough vision, enough scaffolding, and enough budget, the model would clear the bar. The bar was 80% completion. The data said 9%.

The pivot was not a tuning question. Re-prompting moved the number by single digits, and so did tightening the schema. Adding the annotated-screenshot layer, which was the strongest possible test that visual grounding was not the binding constraint, produced the same ceiling as the unannotated baseline. The structural property the literature predicts is the structural property the experiment confirmed. The founder, who had signed for the spend and the calendar, called the experiment in February 2026.

What Part 3 is about

Part 1 was the bet. Part 2 was the receipt. Part 3 is about the engine that we built because the receipt of our bet came back the way it did.

We built a deterministic web capture engine that reads the same page the same way every time, across redesigns, and across years, with no language model in the data path and no human reviewer. Our engine completes web tasks at one-tenth the LLM cost per task and at roughly one hundred percent reliability. It is what shora.ai sells today. Part 3 explains how it works and how the architecture sidesteps each failure mode the LLM hit.