32 341 $ de dépenses AWS, 10,5 millions de pages produit, 9 % de fiabilité : le verdict du pari qui a appris à SHORA pourquoi l'automatisation web déterministe devait exister

Ceci est la Partie 2 d'une histoire en trois parties. La Partie 1 expliquait notre pari. La Partie 3 nomme ce que nous avons construit. Cette partie est ce qui s'est passé entre les deux.

Dans la Partie 1, nous avons expliqué la couche de mesure manquante dans l'e-commerce (l'**Expérience**). Nous avons parié sur l'instrumentation de l'Expérience sur le web (surface publique) à l'échelle, par un modèle de langage suffisamment bien outillé. Cette partie est le verdict de ce pari. Nous rapportons ce que le modèle de langage a produit lorsqu'il a rencontré les 10,5 millions de pages produit dont nous lui avons demandé d'instrumenter les parcours d'achat, et pourquoi les données nous ont forcés à construire autre chose à la place.

Le verdict

Nous avons dépensé 32 341 $ sur AWS pour cette expérience : un agent de pointe fondé sur Bedrock Claude Opus 4.6, conçu pour lire des pages web d'enseignes en direct et reconstruire le parcours d'achat complet, de la page produit (PDP) au panier. L'agent a tourné sur 10,5 millions de pages produit couvrant 1 056 enseignes françaises depuis 480 localisations — un parcours par page produit, chaque parcours traversant au minimum la PDP et le panier, l'agent étant censé extraire un relevé cohérent à chaque étape. Le taux de complétion était de 9 %, c'est-à-dire le taux auquel l'agent a navigué avec succès un parcours de bout en bout sur le site en direct, avant même de classifier ce qui était capturé au panier (le résultat attendu où le produit était ajouté avec le prix PDP intact, ou l'une des deux anomalies qui comptaient : une rupture de stock, ou un écart de prix où le prix du panier dépassait celui de la PDP).

Les lecteurs qui veulent reproduire l'expérience peuvent lire le post compagnon technique (en anglais), qui documente le code open-source `pji-agent` sous licence MIT. Ceux qui veulent la décomposition ligne par ligne de la répartition des 32 341 $ entre services AWS peuvent lire le post compagnon financier (en anglais).

Vous pourriez penser que le premier mode d'échec a forcément été les défenses anti-bot. Ce n'était pas le cas. La pile de navigateurs Camoufox sur laquelle l'expérience tournait a contourné environ 92 % de toutes les défenses anti-bot rencontrées, d'Akamai à Cloudflare et DataDome. L'agent a atteint les pages web qu'il était chargé d'atteindre. **Les 9 % représentent le taux de réussite de l'agent à instrumenter le parcours d'achat.**

Pourquoi l'architecture a atteint un plafond

Le pipeline ne donnait pas au modèle des pixels bruts en lui demandant de trouver des choses. Chaque capture d'écran était annotée de cadres de délimitation, et chaque cadre portait un identifiant unique traçable jusqu'à l'élément du DOM qu'il encadrait. Par conséquent, l'identité était déjà résolue avant que le modèle ne commence à raisonner. Ce qui restait, c'était l'ancrage sémantique : regarder chaque cadre adressable de la page et décider lequel était le prix que l'acheteur paierait, lequel était le déclencheur d'ajout au panier, et lequel était le total du panier après ajout, etc.

Pour décider du sens de chaque cadre, le modèle n'a rien d'autre sur quoi s'appuyer que son a priori d'entraînement, appliqué une inférence à la fois. C'est ce qu'est un LLM : un prédicteur de jeton suivant, sans état, aux poids figés. Le pipeline de SHORA a utilisé tous les mécanismes d'adaptation que la pile de production offre par-dessus cette base — apprentissage en contexte, mémoire augmentée par récupération, et échafaudage d'agent. Aucun d'eux ne transforme des poids figés en un système qui s'améliore à mesure qu'il voit plus de pages en production. La facture croît linéairement avec le volume ; la précision, non.

Ce n'est pas une découverte propre à SHORA. C'est la forme publiée de toute la littérature sur les agents web. Le benchmark académique canonique est **WebArena** [Zhou et al., ICLR 2024], un environnement de 812 tâches web à long horizon. À la publication, le meilleur agent fondé sur GPT-4 atteignait 14,41 % de succès de bout en bout, contre une base humaine de 78,24 % — un écart que les auteurs ont nommé explicitement. Deux ans et des dizaines de soumissions plus tard, le classement WebArena en direct montre l'état de l'art à 74,3 % (Deepseek v3.2, février 2026), toujours sous la base humaine.

Le point plus profond est que même ces chiffres proviennent d'un environnement contrôlé, instrumenté, déterministe. Les mêmes agents échouent bien davantage sur le web ouvert. L'article *An Illusion of Progress?* [Tao et al., COLM 2025] a introduit un benchmark réaliste de 300 tâches sur 136 sites en direct précisément parce que les chiffres affichés sur les anciens benchmarks ne se généralisaient pas. Sur la version réaliste, la frontière 2025 chute à 61 %. Un article de suivi, *WAREX* [2025], a soumis des agents publiés à des conditions proches de la production et a conclu qu'ils « présentent une dégradation sévère en conditions réalistes, exposant des lacunes fondamentales de robustesse ». Sur **OSWorld** [Xie et al., NeurIPS 2024], le benchmark plus large d'usage informatique, l'article original rapporte une base humaine de 72,36 % contre un meilleur score modèle de 12,24 %.

Et la réponse n'est pas d'embaucher des humains

Le substitut instinctif à un modèle peu fiable est une personne fiable. Les preuves académiques contre cette substitution sont plus anciennes et plus reproduites que n'importe quel benchmark de la section précédente. En 1948, Norman Mackworth — mandaté par la Royal Air Force pour comprendre pourquoi les opérateurs radar manquaient les signatures de sous-marins allemands

lors de longues vacations — a mené l'expérience qui a fondé le champ de la recherche sur l'attention soutenue. Les opérateurs effectuant une tâche de vigilance de deux heures montraient une chute marquée de la précision de détection dès les 30 premières minutes, suivie d'un déclin régulier. Le phénomène, appelé aujourd'hui *décrément de vigilance*, a été reproduit continûment pendant soixante-quinze ans [« The vigilance decrement : its first 75 years », Frontiers in Cognition, 2025]. Le constat propre de Mackworth a tenu : dire aux sujets d'être plus attentifs n'avait aucun effet sur le déclin. Le décrément est une propriété du système attentionnel humain sur les tâches de surveillance répétitive, pas une propriété de la motivation.

La même architecture de dégradation apparaît dans la recherche sur l'étiquetage de données, où elle est mesurée plutôt qu'observée. Le kappa de Cohen [Cohen, 1960] est la métrique tierce standard de fiabilité inter-juges — le taux auquel deux annotateurs formés appliquant la même tâche catégorielle aux mêmes éléments sont réellement d'accord. Les études multi-annotateurs sur l'étiquetage à grande échelle trouvent systématiquement que les valeurs de kappa tombent sous les seuils de fiabilité acceptables sur une part significative des catégories, même avec des annotateurs formés.

> Prenez les trois modes d'échec ensemble. **Le modèle échoue parce que des poids figés ne peuvent pas s'adapter d'une page à l'autre.** Le relecteur humain échoue parce que le système de vigilance décline dès la première demi-heure de toute tâche de surveillance répétitive, et que l'accord inter-juges entre humains sur la même tâche n'est pas assez élevé pour substituer la fiabilité à la mémoire. L'ingénieur-avec-un-scraper — la véritable alternative de production — n'échoue pas sur l'attention mais sur les erreurs silencieuses qu'un outil dérivant renvoie entre les réparations, ce qui est en soi la tâche de vigilance dont les humains se dégradent. Un acheteur qui a besoin que les mêmes pages web soient lues correctement des dizaines de milliers de fois par mois, chaque mois, pendant des années, ne choisit pas entre une IA défaillante et un humain fiable. L'acheteur choisit entre trois plafonds, tous réels, tous mesurés — et aucun ne fournit une couche de mesure qu'un acheteur peut signer.

L'aveu

L'architecture d'agent LLM avait été défendue, en interne comme en externe, pendant près d'un an avant l'arrivée de ces données. La conviction était mathématique : avec assez de prompting, assez de vision, assez d'échafaudage et assez de budget, le modèle franchirait la barre. La barre était de 80 % de complétion. Les données disaient 9 %.

Le pivot n'était pas une question de réglage. Re-prompter déplaçait le chiffre de quelques unités, et resserrer le schéma aussi. Ajouter la couche de capture d'écran annotée, qui était le test le plus fort possible que l'ancrage visuel n'était pas la contrainte limitante, a produit le même plafond que la base non annotée. La propriété structurelle que la littérature prédit est la propriété

structurelle que l'expérience a confirmée. Le fondateur, qui avait signé pour la dépense et le calendrier, a arrêté l'expérience en février 2026.

De quoi parle la Partie 3

La Partie 1 était le pari. La Partie 2 était le verdict. La Partie 3 parle du moteur que nous avons construit parce que le verdict de notre pari est revenu tel qu'il est revenu.

Nous avons construit un moteur de capture web déterministe qui lit la même page de la même manière à chaque fois, à travers les refontes et à travers les années, sans modèle de langage dans le chemin de données et sans relecteur humain. Notre moteur réalise des tâches web à un dixième du coût LLM par tâche et à environ cent pour cent de fiabilité. C'est ce que shora.ai vend aujourd'hui. La Partie 3 explique comment il fonctionne et comment l'architecture contourne chacun des modes d'échec que le LLM a rencontrés.