



(RESEARCH ARTICLE)



A PRIVACY PRESERVING PERSONAL AI ASSISTANT AND THE COST OF CLOUD MODEL INFERENCE: ARCHITECTURE, CAPABILITY TRADEOFFS, AND EVALUATION PROTOCOL

Aravind Chinnaraju *

Independent researcher, Seattle, WA.

* Corresponding Author

ORCID Details

Aravind Chinnaraju: <https://orcid.org/0009-0009-2282-7803>

World Journal of Advanced Research and Reviews, 2026, 32(01), 1106–1113

Publication history: Received on 05 September 2026; revised on 11 October 2026; accepted on 13 October 2026

Article DOI: <https://doi.org/10.30574/wjarr.2026.32.1.2606>

Copyright © 2026 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

ABSTRACT

Personal AI assistants need reliable research and controlled access to private data. This report specifies a local-first Windows architecture and examines the capability, privacy, and operating-cost tradeoff between local Llama 3.2 3B through Ollama and self-hosted Kimi K2.5 on rented NVIDIA GPUs. We analyzed a supplied project design summary, mapped trust boundaries and failure modes, and calculated illustrative costs from a documented eight-H200 deployment and published rental rates. The design separates public research from authenticated browser sessions, enforces permissions outside the model, and requires evidence-supported answers. At an illustrative \$4.59 per GPU-hour, eight H200s cost \$36.72 per running hour, \$4,406.40 for four hours daily over 30 days, or \$26,438.40 for continuous 30-day operation, before storage and operations. An assumed 100 W incremental local draw for four hours daily at \$0.15/kWh costs \$1.80 monthly on an existing PC. These are scenarios, not measured deployments. Kimi K2.5 offers multimodal and agentic capabilities that a text-only 3B baseline should not be assumed to match; no matched end-to-end test was performed. Stronger cloud inference can impose substantial dedicated GPU rental cost and transfers prompts to a cloud operator. Local inference preserves a narrower data boundary at far lower incremental cash cost but requires direct task-quality evaluation.

Keywords: Personal AI Assistant; Privacy by Design; Retrieval Augmented Generation; Kimi K2.5; Llama 3.2 3B; GPU Inference

1 INTRODUCTION

Personal assistants increasingly combine web research with calendars, messages, files, and browser accounts. These capabilities create a useful but difficult boundary: the same system that reads untrusted pages may also hold private context and execute consequential actions. Retrieval augmented generation (RAG) can connect generated text to external evidence (Lewis et al., 2020), yet retrieval quality and citation discipline remain application responsibilities. Indirect prompt injection shows how instructions embedded in retrieved material can redirect an assistant that treats content as authority (Greshake et al., 2023).

A local model addresses one important path of disclosure: prompts need not be transmitted to a hosted inference provider. It does not prevent a search engine from receiving a query, a signed-in website from identifying an account, a browser profile from retaining credentials, or a model from suggesting an unauthorized action. NIST recommends managing generative AI risk across the system lifecycle (Autio et al., 2024), and OWASP identifies prompt injection and excessive agency among salient application risks (OWASP Foundation, 2025).

A private personal AI assistant is essential for protecting highly sensitive user data—such as financial records, health information, and confidential communications—from unauthorized third-party access, data harvesting, or cloud breaches, ensuring absolute data sovereignty. Furthermore, the financial impact of shifting to a private, local-first architecture is substantial. It eliminates recurring subscription fees and token-based inference costs associated with proprietary cloud APIs, replacing unpredictable operational expenses with a fixed, depreciable hardware investment that is far more economical for always-on, high-volume tasks.

This technical report develops an architecture for a Windows assistant with evidence-grounded public research, a separately scoped browser, and explicit memory and action controls. It also asks whether a small local model or a much larger model on rented cloud GPUs is economically and operationally appropriate for a personal assistant. The comparison fixes Llama 3.2 3B as the reported baseline (Private Assistant, 2026) and Kimi K2.5 as a distinct, self-hosted cloud alternative; it does not assume that a 3B model delivers equivalent agentic quality. The objectives are to define data and authority boundaries, specify the proposed mechanisms, and state how quality, privacy, and cost should be measured. This is a design and scenario analysis, not a performance study or security certification.

2 MATERIALS AND METHODS

2.1 Source material and design method

The primary case material was a project design summary dated 28 September 2026 (Private Assistant, 2026). It reports an existing 0.3.0 foundation with local Ollama and an encrypted vault, an adapted RAG integration and SearXNG research, and incomplete live verification of WARP routing. It describes the dedicated browser, remembered sessions, expanded memory, and proactive workflows as proposed. The underlying repository, installation, threat fixtures, and verification report were not supplied for independent inspection. Consequently, the status and incident observations below are attributed to the supplied summary rather than independently reproduced.

We used a design-science procedure: (i) enumerate assets, actors, and trust boundaries; (ii) state invariants enforced outside the model; (iii) map workflows to bounded sequences and failure responses; (iv) compare model deployment scenarios using published specifications and transparent arithmetic; and (v) define evaluation and release gates. External literature and documentation motivate threats and platform capabilities, not certification (Autio et al., 2024; Cloudflare, n.d.; Greshake et al., 2023; Lewis et al., 2020; Microsoft Playwright, n.d.; Ollama, n.d.; OWASP Foundation, 2025; SearXNG project, n.d.). Cost is calculated as GPU count multiplied by hourly rental price and billed hours; throughput and cost per successful task are reserved for measurement.

2.2 Threat model

Protected assets include prompts and saved memories, source documents, OAuth tokens, browser authentication state, local encryption keys, selected files, and action approvals. The adversary can control a search result, article, website, document, or response metadata; may induce redirects or DNS changes; and may race or replay an action. A remote service legitimately receives traffic directed to it. The model may make mistakes, including mistaking page text for instructions. A compromised operating system or administrator is outside the protection claim. Physical theft is considered only insofar as full-disk encryption and vault recovery controls can reduce exposure while the device is off or locked.

The system assumes a correctly installed application, a trusted local policy broker, and operating-system controls that restrict browser egress. The renderer, retrieved content, model output, and visited pages do not confer authorization. Where OS-level isolation cannot be enforced or tested, authenticated automation remains disabled. This is an explicit dependency rather than an assertion that a browser profile or proxy setting is itself a security boundary.

2.3 Design invariants

Table 1: Required design invariants, their operational meanings, and failure responses.

Invariant	Operational interpretation	Failure response
Local mode	Prompts, embeddings and saved personal data remain on device; an allowlist selects installed local models.	Stop if local inference is unavailable; no implicit cloud fallback.
Authority separation	Only the broker may grant scoped reads, writes, or memory persistence; retrieved text and model output are proposals.	Deny ungranted or ambiguous operations.
Session separation	Public retrieval has no personal browser cookies; browser tasks use dedicated selected profiles.	Stop cross-profile reads and require new authorization.
Route integrity	All traffic covered by a selected route, including DNS and browser subresources, follows that route.	Fail closed on route failure; do not silently reconnect directly.
Claim support	Every factual research claim cites accessible evidence with an auditable excerpt and date status.	Abstain when evidence is missing or too stale for the question.

3 PROPOSED ARCHITECTURE

3.1 Local components and data flow

A constrained Electron renderer communicates with a main coordinator through validated IPC. The coordinator assigns request identifiers and owns cancellation. A policy broker evaluates user grants and action scopes. The research orchestrator uses SearXNG as a metasearch broker, a restricted article fetcher, local embeddings, and a local language model. SearXNG aggregates configured external engines; it does not make the web local or hide identifying query text from upstream services (SearXNG project, n.d.). A separate browser worker operates a visible Chromium instance under a dedicated profile. A vault stores approved memories and broker-held credentials. The scheduler uses scoped grants and pauses authenticated work when Windows locks.

The reported baseline uses Llama 3.2 3B served by Ollama; its listing gives an approximately 2.0 GB model download, which is not the full runtime memory requirement (Ollama, n.d.). The supplied design summary separately proposes evaluating Qwen3.5 9B as a future local upgrade (Private Assistant, 2026). This manuscript compares the existing 3B baseline with cloud-hosted Kimi K2.5 and does not re-label the 9B candidate as an implemented component. The adapted RAG pipeline is described as Vane-derived rather than the full upstream server (Private Assistant, 2026). Model selection affects quality and resource use; neither model is entrusted with permissions.

3.2 Public research and evidence control

The proposed research workflow takes an explicit web grant and immutable network-route lease before issuing queries. It attempts four configured search sources per logical search, preserves partial successes, deduplicates up to 20 candidates per search, and evaluates relevance and freshness before fetching at most two eligible HTTPS articles. The default budget is three searches and four generation calls, with up to five final sources. These values are engineering ceilings to be benchmarked, not experimentally optimized parameters. A zero-result or irrelevant-result path returns "insufficient evidence."

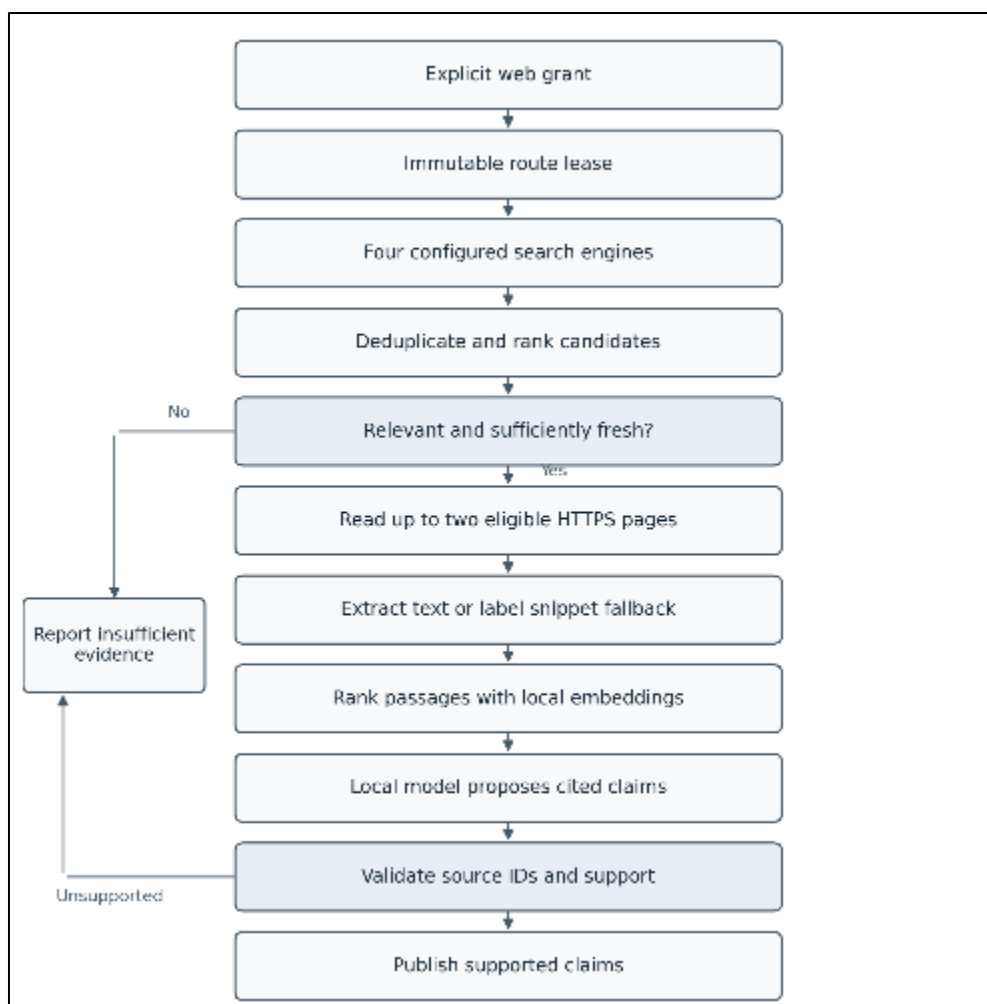


Figure 1: Proposed evidence-grounded public research flow, adapted from the supplied design summary (Private Assistant, 2026). The branches are design rules, not validated performance.

The fetcher should respect robots policy and enforce DNS and redirect checks, a 1 MB response-body cap, 20,000 extracted characters, at most three redirects, and a five-second total deadline per page. It should not borrow an authenticated browser session, execute page JavaScript, or bypass a login, CAPTCHA, or rate limit. Readability extraction can be followed by an explicitly labeled snippet fallback. Candidate passages are approximately 400 tokens with 60-token overlap; no more than eight passages enter a 3,200-token evidence budget. These are proposed defaults requiring evaluation on diverse tasks and languages.

The generator returns structured claims with source identifiers and supporting excerpts. A deterministic validator checks that references resolve to retrieved records, cited excerpts exist, and publication dates are not represented as verified when unknown. Semantic claim support remains fallible and requires human-labeled evaluation. Progress may stream immediately, while factual claims are buffered until checked. An uncited answer with a generic warning does not satisfy the evidence rule. In the supplied case, a request for current stock-market news reportedly produced irrelevant TODAY/calendar/wellness links and uncited text; the summary attributes this to premature selection of five candidates, weak relevance and freshness checks, and acceptance of uncited output (Private Assistant, 2026). This incident motivates the pipeline but is not an independently reproduced benchmark.

3.3 Browser sessions and consequential actions

Public research and authenticated browsing are separate capabilities. A user creates a browser session for named sites and a task, then signs in through a private handoff while observation and automation are paused. Playwright supports persistent contexts, and its documentation cautions that stored authentication state can impersonate an account (Microsoft Playwright, n.d.). Such profiles are therefore treated as secrets. Temporary profiles are the default; remembered profiles require explicit choice, encrypted archives at rest, and restricted working directories. Active Chromium can still place plaintext artifacts in its working profile, so full-disk encryption is a practical additional control, not a guarantee of forensic erasure.

The browser worker exposes typed actions and short-lived observed element references; the model does not receive raw CDP access or arbitrary JavaScript execution. For a consequential action, the broker prepares a preview bound to task, session, origin, exact payload, and observed document state. A one-use grant expires after five minutes or a relevant state change. Before executing once, the broker revalidates all bound fields. If a submission outcome is unknown, it requests manual inspection instead of retrying blindly. Ambiguous clicks and autosaving fields are treated as consequential when their effect cannot be determined. Provider APIs with narrow scopes can be preferred where they provide clearer transaction semantics.

3.4 Network routing and storage

An optional outbound WARP route can mask the household egress IP for traffic that actually traverses it. Cloudflare observes the incoming connection and has its own traffic and DNS-data practices; WARP does not hide account identity or submitted content from a visited service (Cloudflare, n.d.). Its documentation notes that WebRTC can bypass the expected route (Cloudflare, n.d.). Research DNS, queries, and article requests must use the same selected route, and a disconnect must stop the operation. Browser routing is configured separately and changing it requires a new browser session. OS or process-level gateway enforcement must be tested for IPv4, IPv6, DNS, WebRTC, QUIC, WebSocket, redirects, popups, and subresources. No inbound tunnel exposes local inference, debugging, or the vault.

The default retention design keeps retrieved pages, snippets, and embeddings in request memory; conversations remain session-only until history is enabled. Personal memory is saved after approval and can be inspected, edited, exported, and forgotten. Jobs store encrypted checkpoints with revocable grants and deduplication. A content-minimized audit retains metadata for a stated rolling period, proposed as 30 days. Encrypted exports require user-held recovery material; forgetting a vault password cannot recover old content without a separately provisioned recovery path. Deleting a local browser profile does not terminate the corresponding site-side session.

3.5 Optional cloud model boundary

Kimi K2.5 is a separately assessed deployment option, not a silent fallback from the local architecture. Its model card reports one trillion total parameters, 32 billion activated per token, a 256,000-token context length, and image-text capability (Moonshot AI, 2026). The vLLM project documents an eight-NVIDIA-H200 serving configuration (vLLM Project, 2026). If this option is enabled, prompt and retrieved context cross from the user's device into a rented cloud machine. The cloud operator can potentially access the running workload and associated infrastructure under its controls; encryption in transit and at rest does not establish end-to-end confidentiality from the operator. The user must explicitly select the cloud mode and the data classes allowed to leave the device. Broker-held credentials and raw browser profile files remain excluded from model prompts in either mode. Kimi self-hosting is a different trust model from using Moonshot's managed API, and it does not satisfy the local-inference invariant.

4 RESULTS

The result of this design analysis is a set of architectural outputs and explicit implementation gates, summarized in Table 2. These are specifications and status claims from the supplied case material. No measured accuracy, leakage rate, attack resistance, or latency result is available. The distinction matters because a security control can be specified precisely yet fail in implementation or deployment.

Table 2: Status reported in the supplied project summary and the corresponding evidence needed. "Existing" does not imply that this author independently inspected or validated the component.

Capability	Evidence required before release
Llama 3.2 3B and vault	Baseline quality, recovery, peak memory and lock-state tests
Kimi K2.5 cloud option	Consent, egress, isolation, throughput and cloud-cost verification
Research pipeline	Blinded relevance, freshness, support and abstention evaluation
WARP research routing	Packet-level no-bypass and route-disconnect tests
Dedicated browser	Profile isolation, login handoff, observation filtering and egress tests
Writes and approvals	Replay, race, changed-document and unknown-outcome tests
Memory and jobs	Deletion, backup, crash recovery and duplicate-action tests

5 EVALUATION PROTOCOL

5.1 Research quality and resource use

Freeze model weights, embeddings, configuration, source corpus or capture time, hardware, and prompt set before comparison. Compare the reported 0.3.0 baseline with the revised pipeline on at least 100 prespecified information requests stratified by time sensitivity, topic, and answerability. Include the reported stock-market-news case, but do not let it dominate the set. Two blinded reviewers independently label candidate relevance, final factual claims, source support, freshness appropriateness, and justified abstention; adjudicate disagreements and report agreement. For each system report precision of supported claims, proportion of unsupported claims, source relevance, correct abstention on unanswerable items, latency distributions, and retrieval failures with denominators and confidence intervals. Record the engine availability and date of each run; live-web comparisons are otherwise confounded by changing content.

Run both configurations on the same task set and with equivalent permissions, retrieval inputs, tool descriptions, and stopping criteria. Separate simple summarization from multi-step planning, browser work, and image-containing tasks; record whether a modality is unsupported rather than assigning an invented score. Report task completion, unsupported claims, harmful action proposals, human interventions, retries, latency, and cost per completed task with confidence intervals. Disclose the different serving hardware and versions. Kimi's published benchmark results were produced in its authors' own evaluation setup (Moonshot AI, 2026) and do not measure this assistant against Llama 3.2 3B.

On the target 16 GB Windows computer, run repeated cold and warm Llama 3.2 3B trials with the browser and search service active. Measure peak committed and resident memory, time to first progress, time to supported answer, incremental wall power, and out-of-memory or crash frequency. For cloud Kimi, log actual GPU billing intervals, cold starts, storage, data transfer, utilization, and completed tasks. Report model quantization, context length, concurrency, and power state. Do not infer operational feasibility from download size alone.

5.2 Security and privacy verification

Use an instrumented egress gateway and controlled test domains. Inject malicious instructions into search snippets, retrieved articles, and authenticated pages; attempt to trigger an ungranted file read, cross-session read, exfiltration, or write. Test one-use approvals against replay, changed payload, changed origin, changed DOM state, expiry, concurrent requests, and a lost submission response. Test URL redirects, DNS rebinding, private address resolution, WebRTC/UDP, QUIC, IPv6, WebSocket, popup, and subresource paths with WARP enabled, disabled, and disconnected. Inspect packet traces, broker decisions, and receipts, while avoiding collection of real credentials or sensitive page content.

A critical safety breach is any unapproved consequential action, cross-profile secret access, or direct egress after a selected route fails. Report the number of such breaches per tested opportunity, the fixture taxonomy, and the exact environment. Zero observed breaches is a release gate for the tested fixtures, not proof of universal security. Run deletion and recovery tests over vault records, indexes, temporary browser files, remembered archives, backups, and rolling audits. An external review remains valuable because the broker and OS enforcement are trusted parts of the design.

6 DISCUSSION

The architecture treats privacy as a data-flow property and safety as an authorization property. Local inference prevents routine transmission of full prompts to an inference vendor, while metasearch still sends queries to engines and browser sessions still identify users to signed-in sites. Cloud-hosted Kimi instead transmits selected prompts to a rented instance and places the cloud operator inside the trust boundary. Optional network routing changes visible IP information only for traffic that passes through it. These distinctions make a narrower, testable privacy claim than "fully private" or "anonymous."

The main reliability improvement proposed here is to refuse unsupported research claims rather than depend on a stronger model alone. RAG supplies a method for conditioning generation on retrieved evidence (Lewis et al., 2020), but the attached incident illustrates that poor discovery and selection can make downstream prose fluent and wrong. Source identifiers and exact excerpt checks can detect some mechanical citation failures; they cannot establish that a paraphrased claim logically follows from an excerpt. Human-labeled support evaluation remains necessary.

The action broker supplies a second independent layer. Prompt injection studies show why untrusted material must not be treated as instructions (Greshake et al., 2023). A local model is no more authorized to approve its own action than a hosted model. Binding approvals to the observed transaction reduces replay and confused-context risks, but dynamic websites, autosaving controls, and uncertain network outcomes may still force manual takeover. Likewise,

persistent browser contexts help implement separate sessions (Microsoft Playwright, n.d.) but do not themselves guarantee process isolation or prevent live plaintext caches.

6.1 Capability and operating cost

Ollama is a runtime, not an intelligence rating. Llama 3.2 3B is a text-only model suitable for bounded summarization and retrieval tasks (Ollama, n.d.). Kimi K2.5 supports vision, extended context and agentic workflows (Moonshot AI, 2026). Its greater scale and published evaluations provide a credible reason to expect better performance on difficult multi-step work, but parameter count cannot be converted into a task-success rate. Moonshot reports 76.8% on SWE-bench Verified in its own agent harness (Moonshot AI, 2026); the 3B baseline was not run in that harness, and the result should not be presented as a head-to-head advantage. The proposed broker, retrieval quality, and tool design can affect either model's realized assistant performance. Direct paired testing remains necessary.

For a concrete cloud cost, the vLLM recipe verified Kimi K2.5 on a single node with eight H200 GPUs (vLLM Project, 2026). Runpod lists H200 Secure Cloud pods at \$4.59 per GPU-hour on its pricing page updated 27 September 2026 (Runpod, 2026). Multiplying that per-GPU rate by eight gives \$36.72 per running hour. This is an indicative arithmetic scenario, not a confirmed quote or availability guarantee for a co-located eight-GPU node. The full model, chosen precision, context length, throughput and concurrent requests must be tested on the rented configuration. Billing can continue while an instance is idle; storage, orchestration, network transfer, tax and engineering work are extra.

Table 3: Scenario costs in US dollars. Cloud cost = $8 \times \$4.59 \times \text{billed hours}$. Local electricity = $0.100 \text{ kW} \times 4 \text{ h/day} \times 30 \text{ days} \times \text{assumed } \$0.15/\text{kWh}$. Dedicated local hardware = $\$900 \div 36 \text{ months}$. The cloud price is a publicly posted single-GPU rate (Runpod, 2026); actual eight-GPU node pricing and availability may differ. The local power draw and hardware purchase are explicit assumptions. All rows omit internet service and the time required to maintain the assistant.

Deployment scenario	Illustrative direct cost	Scope and caveat
Existing Windows PC, Llama 3.2 3B; assumed additional 100 W for 4 h/day at \$0.15/kWh	\$1.80 per 30-day month	Assumed incremental power, not a wall-meter result; \$0 new hardware
Dedicated local \$900 PC over 36 months, same power assumption	\$26.80/month (\$25 hardware + \$1.80 power)	\$900 is a scenario input, not a quoted product
Kimi K2.5 on 8 H200 GPUs, rented 1 h/day at \$4.59/GPU-h	\$36.72/h; \$1,101.60 per 30-day month	Warm-up and minimum billing intervals may add cost
Same cloud cluster, 4 h/day	\$4,406.40 per 30-day month	120 billed node-hours; no storage or operator labor
Same cloud cluster, continuously available	\$26,438.40 per 30-day month	720 billed node-hours; availability and SLA unverified

At sparse personal usage, keeping an eight-H200 cluster reserved is likely economically inefficient even if its answers are substantially better. Stopping the cluster between tasks reduces billed time but introduces model loading and availability delay. A shared inference API might offer a different cost curve, yet it would be a separate commercial comparison, with provider data handling and per-token pricing rather than dedicated GPU rental. The relevant decision is cost per correctly completed and authorized task at the required privacy level, not cost per token or model size alone.

The present work has three material limitations. First, its project status and defect account come from one author-supplied summary (Private Assistant, 2026); the linked repository documents were unavailable and no code audit was conducted. Second, performance and security effects are hypotheses pending the specified tests, including real routing verification. Third, the design targets a Windows 16 GB environment and selected web workflows; search engines, browser behavior, local-model versions, and provider APIs can change. A publication reporting empirical effectiveness should add implementation details, a reproducible fixture set, measured results with uncertainty, and a comparison with explicit baselines.

7 CONCLUSION

A privacy-preserving personal assistant requires evidence checks, deterministic authorization and explicit data boundaries regardless of model size. Llama 3.2 3B through Ollama supports a low-cost local baseline; Kimi K2.5 on

rented NVIDIA GPUs offers a plausible step up in complex, multimodal agentic capability while moving selected prompts into a cloud operator's environment. Under the stated assumptions, incremental local electricity is \$1.80 per month on an existing PC, whereas eight H200 GPUs at a posted \$4.59 each per hour cost \$1,101.60 for one hour daily or \$26,438.40 for continuous operation over 30 days, before ancillary costs. These figures are transparent scenarios, not measured total costs or proof of a quality gap. Selection should follow matched task-completion, safety, privacy and cost-per-success evaluation. The proposed architecture and release gates remain unvalidated until those tests are performed. While it is not currently economical to run local models and achieve frontier-level intelligence, this dynamic is expected to change in the future as hardware capabilities improve and smaller models become more capable.

REFERENCES

- [1] Autio C, Dunietz J, Reznick L, et al. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1. Gaithersburg: National Institute of Standards and Technology; 2024. DOI: 10.6028/NIST.AI.600-1
- [2] Greshake K, Abdelnabi S, Mishra S, Endres C, Holz T, Fritz M. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In: Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security. 2023. DOI: 10.1145/3605764.3623985
- [3] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Adv Neural Inf Process Syst*. 2020;33:9459–9474.
- [4] Microsoft Playwright. Authentication [Internet]. [cited 2026 Sep 28]. Available from: <https://playwright.dev/docs/auth>
- [5] Microsoft Playwright. BrowserType launchPersistentContext [Internet]. [cited 2026 Sep 28]. Available from: <https://playwright.dev/docs/api/class-browsertype#browser-type-launch-persistent-context>
- [6] Cloudflare. WARP client known issues and FAQ [Internet]. [cited 2026 Sep 28]. Available from: <https://developers.cloudflare.com/warp-client/known-issues-and-faq/>
- [7] Cloudflare. WARP client privacy [Internet]. [cited 2026 Sep 28]. Available from: <https://developers.cloudflare.com/warp-client/privacy/>
- [8] Moonshot AI. Kimi K2.5 model card [Internet]. 2026 [cited 2026 Sep 28]. Available from: <https://huggingface.co/moonshotai/Kimi-K2.5>
- [9] vLLM Project. Kimi K2.5 usage guide [Internet]. 2026 [cited 2026 Sep 28]. Available from: <https://github.com/vllm-project/recipes/blob/main/moonshotai/Kimi-K2.5.md>
- [10] Runpod. GPU cloud pricing [Internet]. Updated 2026 Sep 27 [cited 2026 Sep 28]. Available from: <https://www.runpod.io/pricing>