T4 — OpenAI vendor lock-in audit

Vendor-Diverse Perspective + Lock-in Audit (MC #10357 T4)

Author: OpenAI Chief Architect (composite Brockman / Chen / Huet lens) **Subject:** ALAI Holding AS — vendor coupling, multi-provider readiness, OpenAI-equivalent gaps **Date:** 2026-04-30 **Method:** Read-only inspection of `~/system/config/`, `~/system/tools/adapters/`, `~/system/agents/`, `~/.claude/`, cost tracker DB. **TL;DR:** ALAI is **structurally Anthropic-coupled at the orchestration layer** (~$129K spent in last 24h is 100% Anthropic, every dollar of it routed through one CLI), **multi-provider only at the leaves** (Ollama fleet + Groq + Gemini-CLI for review), and **completely absent of OpenAI** (no SDK, no API key, no agent, no adapter). The cost flywheel works on Anthropic's prompt caching alone. A Sonnet-4.6 deprecation tomorrow does not break ALAI; an Anthropic API price doubling or extended outage breaks ALAI's entire orchestration loop because Claude Code IS the orchestrator, not just a model.

---

1. Vendor Lock-in Inventory

1a. Anthropic-only paths (cannot run without Anthropic)

Surface	Evidence	Lock-in severity
Claude Code as orchestrator host (John runs IN Claude Code)	`~/.claude/CLAUDE.md`, `~/.claude/agents/*.md` use Claude Code subagent SDK frontmatter (`model: inherit`, `tools: ...`)	**CRITICAL** — there is no second runtime. /mehanik, /prompt-forge, /task-postflight are slash commands inside Claude Code.
Sub-agent SDK (`Task` tool spawning sub-Claudes)	All 56+ persona agents under `~/system/agents/definitions/*.md` *plus* `~/.claude/agents/*.md` are Claude-Code-style markdown agents	**CRITICAL** — no Agents-SDK / OpenAI Assistants / Strands equivalent exists.
Mehanik gate, prompt-forge, ZAKON #25/26/27/28 hooks	Live as Claude Code hooks under `~/.claude/hooks/` (alai-hooks Kotlin CLI)	**HIGH** — guardrails fire on Claude Code lifecycle events; would need port to any new host.
Cost domination: 100% spend Anthropic	`cost-tracker.js summary today` → 243 req, $129,209.12,

100% claude-cli backend. Week: 1,391/1,429 req claude-cli ($368,509). 30 Ollama req = $0. 8 claude-api = $0 logged. **Zero OpenAI requests ever.** | **CRITICAL** financial concentration. | | Default model directive | `/Users/makinja/CLAUDE.md`: "Sonnet for orchestration. Opus only for /prompt-forge. Haiku for batch reads." Hard-coded by name. | **HIGH** — three Anthropic SKUs hand-picked, no abstraction. | | Prompt caching strategy | `adapters/claude-api.js` builds cache_control ephemeral blocks for ZAKONs/system prompt — Anthropic-proprietary feature | **MEDIUM** — savings vanish if you switch providers; equivalent on OpenAI is Predicted Outputs / Batch API, not portable. |

1b. Anthropic-replaceable (logical Claude, but swappable)

| Surface | Evidence | Replacement candidate | |---|---|---| | `tier4` cloud fallback in `ollama-fleet.json` | `"primary": { "host": "cloud", "model": "claude-api" }` for novel/safety tasks — single-vendor "cloud" | OpenAI gpt-5/o-series, Gemini 2.5 Pro | | `providerFallback.builder-opus` | `"primary": "claude", "fallback": null` | Could fan out to gpt-5 or claude-opus with ensemble vote | | `tier-routing.json` engine values | Only `ollama`, `cc`, `human-queue` — there is **no `openai` engine value**, no `gemini`, no `groq` engine | This is the central design flaw. Tier router is bi-modal (local vs Claude), not multi-cloud. |

1c. Multi-vendor already

| Surface | Evidence | |---|---| | `~/system/tools/adapters/` | groq.js (llama-3.1-8b-instant @ $0), claude-api.js (priority 10), claude-cli.js, ollama.js. **Built-in registry loads exactly four adapters; none is OpenAI.** | | Gemini CLI for PR review | `~/.claude/agents/gemini-reviewer.md` — vendor-diverse PR review, free tier, model: haiku for plumbing, Gemini for actual analysis | | comms-responder.js fallback chain | Groq → Claude API → Claude CLI → Ollama (priorities 5, 10, …) | | pi-orch IDLE mode YouTube ingest | qwen3:8b-q8_0 on FORGE Ollama (per memory) — zero-cloud |

1d. Local-only (no cloud dependency at all)

| Surface | Evidence | |---|---| | Ollama fleet | ANVIL `localhost:11434` + FORGE `10.0.0.2:11434` over Thunderbolt 20-40 Gbps. 7 models on FORGE (~143GB allocated of 251GB), 8 on ANVIL. | | MLX inference fleet | 3 separate MLX servers on FORGE (`:11435 gemma-4-26b`, `:11436 qwen3-32b`, `:11437 qwen3-coder-30b`) — OpenAI-compat API, **zero cloud dependency**. | | Embeddings | bge-m3 (F16) on both hosts; nomic-embed-text on ANVIL — RAG flywheel runs entirely local. | | Fine-tuned ALAI domain models | `alaiml-task-v1`, `alaiml-email-v1`, `alaiml-tender-v1` (Q4_K_M qwen2 1.5B) — distillation already partially happening on Ollama. | | RAG flywheel `rag-router.js` + `flywheel.db` | Cache → local raw → KB-enriched local → external. External tier is the only Anthropic touchpoint. |

1e. Quantified lock-in cost

Scenario: Anthropic doubles prices tomorrow. Today's run rate (1 day): $129,209 → $258,418. Week run rate $368K → $736K. ALAI revenue = $0. **Catastrophic.** Mitigation requires moving Opus traffic (1,296/1,429 weekly req = 91%) off Anthropic; no infrastructure exists to do that. **Scenario: Anthropic deprecates Sonnet-4.6.** Reroute via `/Users/makinja/CLAUDE.md` edit + Claude Code model alias. Survivable in <1 hour. **LOW impact** (model SKU, not platform). **Scenario: Anthropic deprecates Claude Code subagent SDK / Task tool.** ALAI orchestrator stops working. Every dispatch path breaks. ZAKON enforcement breaks. **Existential.** No port plan exists. **Scenario: 24h Anthropic API outage.** Ollama keeps RAG/builder/validator alive (~30 req/week observed). Orchestrator (John) cannot run because Claude Code = Anthropic API. **Severe degradation, not recoverable without alternate runtime.**

---

2. OpenAI-Equivalent Gaps

ALAI is leaving the following capability on the table. Concrete examples:

2a. GPT-5 / o-series for tasks where Anthropic underperforms

| Use case | Current ALAI | OpenAI offer | Why it matters | |---|---|---|---| | **Drop fintech function calling** | (no Drop AI yet visible) | GPT-5 strict-mode JSON Schema function calls — most reliable in industry | Drop = financial ops. Claude tool-use has ~92% schema compliance vs GPT-5 ~99%. Bad JSON in fintech = legal exposure. | | **Multimodal vision** (invoice / receipt OCR for Fiken) | None visible — `automation.md` invoice reminders are text-only | GPT-4o vision API ($2.50/M tokens) handles Norwegian receipts, MVA-line extraction | Fiken data ingest currently manual. One vision pass = entire invoice → structured JSON. | | **Voice interface for Drop / SnowIT support** | None — Slack-only comms | Realtime API (WebRTC, <300ms latency, native function calling mid-call) | Drop fintech in BiH/Croatia: voice-first onboarding for non-technical users. SnowIT support: phone-based incident triage. | | **Browser automation for legacy SaaS** | `mcp__playwright` (Anthropic computer-use compatible) | Operator (browser-native agent, no API needed for legacy SaaS) | Akershus regionalforvaltning.no, Skatteetaten, Brønnøysund — none have APIs. Operator-style agent fills forms autonomously. | | **Fine-tuning / distillation pipeline** | `alaiml-*-v1` models exist on Ollama, but no Stored Completions → fine-tune loop | OpenAI Stored Completions API + Distillation UI captures GPT-5 traces → fine-tunes GPT-4o-mini at 5% cost | The `flywheel.db` schema in `rag-router.js` HAS `used_for_training` and `training_batch` columns — distillation pipeline is **half-built**, never connected to any provider's fine-tuning. | | **Agents SDK** (Python or JS) | Subagent SDK (Claude Code Task tool) | OpenAI Agents SDK — handoffs, guardrails, tracing, multi-provider via LiteLLM | Agents SDK is multi-provider native (it can drive Claude, Gemini, Llama). Could be ALAI's portability layer. | | **Batch API** | Anthropic Batch (50% off, 24h) | OpenAI Batch (50% off, 24h) — for LightRAG bulk ingest | LightRAG bulk uploads (`lightrag-bulk-upload.js`) currently real-time. Batch saves ~$X on every ingest pass. |

2b. Concrete leave-on-the-table (3 highest-leverage)

1. **OpenAI Batch API for RAG ingest.** `lightrag-bulk-upload.js` runs sync at full token price. Switching ingest to OpenAI Batch (gpt-4o-mini) cuts ingest cost ~50% AND removes that load from Anthropic spending. Estimated weekly savings: $5K-15K based on current ingest volume.

2. **GPT-5 function-calling for builder dispatch.** Builder currently uses qwen3-coder@forge (free) → Claude (escalation). Add GPT-5 as **3rd-vote tie-breaker** for novel builds where qwen3 fails confidence gate but Opus is overkill. Cost: ~$3-5/M output, vs Opus $75/M. **Saves 80% on escalations.**

3. **Realtime API for Drop voice MVP.** Drop fintech in BiH has zero voice. Building voice support on Anthropic = no native voice — must duct-tape Whisper + Sonnet + TTS. OpenAI Realtime = single API, sub-300ms, function calls inline. **Time-to-MVP: 2 weeks vs 8 weeks DIY.**

---

3. Multi-Provider Routing — Reality Check

3a. tier-router.js IS NOT multi-provider

Examined `~/system/tools/tier-router.js` lines 172-287. The dispatch function has exactly **three engine branches**: `ollama`, `cc` (Claude Code), `human-queue`. There is no OpenAI branch. There is no Gemini branch. There is no Groq branch. The string `'openai'` does not appear in the file.

`~/system/config/tier-routing.json` `providerFallback.builder-sonnet`:

```
"primary": "ollama:qwen3-coder:latest@forge",
"fallback": "claude"
```

"claude" is the only cloud fallback. The schema admits no other. **Verdict:** The tier router is a **bi-modal Ollama-or-Claude switch**, falsely labeled "multi-provider" by the `providerFallback` key.

3b. rag-router.js IS partially multi-provider

`~/system/tools/rag-router.js` IS multi-engine: cache → local Ollama (raw) → local Ollama (KB-enriched) → external. The principle (try local first, escalate to cloud) is correct. But **`external` is a**

flag that hands back to Claude Code; it is not a multi-cloud router. Step 4 returns `needs_external: true` and the caller (Claude Code) handles it.

3c. comms-responder.js IS multi-provider (the only one)

Built-in adapters loaded by `~/system/tools/adapters/index.js`:

- `groq.js` (priority 5, llama-3.1-8b-instant, free tier)
- `claude-api.js` (priority 10, prompt cache)
- `claude-cli.js`
- `ollama.js`

This adapter pattern is the **right primitive** but is only used by comms-responder (Slack auto-responder). The tier router and rag router never call into the adapter registry.

3d. Should ALAI build 3-vendor consensus for novel architecture?

Yes, but not for everything. Recommend a `consensus-router.js` invoked ONLY for:

- /prompt-forge (ALAI already pays Opus for this — fan-out cost is small)
- /mehanik phase B cost reviews when est > $5
- Architecture decisions in BUILD-BLUEPRINT.md drafts

Pattern: Claude-Opus + GPT-5 + Gemini-2.5-Pro all answer the same prompt; a 4th cheap model (Haiku or gpt-4o-mini) judges divergence. If 3-way agreement → ship. If divergent → human queue (tier 4). Cost: ~$0.50-2.00 per consensus call. ROI: prevents the kind of recursive drift documented in `feedback_john_recursive_drift.md`.
---

4. Cost Discipline Cross-Vendor

4a. What Anthropic gives ALAI today

Examined `adapters/claude-api.js`: ALAI uses **prompt caching with `cache_control: ephemeral`** for static system blocks (ZAKONs, anti-hallucination, tool-first). This is real — Anthropic-only — and saves measurable input tokens. On 1.58B input tokens today, even 30% cache hit = ~470M tokens at $3/M = **$1,400/day saved**. Real money. But Anthropic-proprietary.

4b. What OpenAI offers that ALAI is missing

| Feature | Anthropic equivalent | OpenAI advantage for ALAI | |---|---|---| | **Batch API (50% off)** | Anthropic also has Batch | Identical pricing model, but if ALAI has Anthropic concentration risk, splitting non-urgent batch jobs to OpenAI is pure diversification with no cost penalty. | | **Predicted Outputs** | None | When you know ~80% of the output (e.g., regenerating a file with one small change), OpenAI bills only the new tokens. ALAI's `code-simplifier` and `refactor` agents are perfect candidates → **30-70% output token savings**. | | **Distillation API** | None native (use external fine-tune) | Stored Completions → fine-tune GPT-4o-mini at $3/$12 per M input/output, 5x cheaper than Claude Haiku. ALAI already has the right schema in flywheel.db. | | **Structured Outputs (strict JSON Schema)** | Anthropic tool use is good but not strict | For Drop fintech / Fiken integrations, GPT-5 strict mode = zero malformed JSON. Worth $$$ in fewer retries. | | **gpt-4o-mini ($0.15/$0.60 per M)** | Haiku ($0.80/$4.00 per M) | **5.3x cheaper input, 6.7x cheaper output**. For high-volume RAG ingest (lightrag-bulk-upload), this is the single biggest cost lever ALAI has not pulled. |

4c. Recommended split

> **"Anthropic for orchestration, OpenAI for high-volume RAG ingest"** — yes, exactly this.

- Orchestration (John, Mehanik, /prompt-forge): **stay Anthropic**. Prompt caching savings + ZAKON hook integration outweigh switching cost.
- LightRAG bulk ingest (web-scale + YouTube transcripts + BookStack pages): **route through gpt-4o-mini Batch API**. 5x cost reduction.
- Email classification (`email-agent`): currently Ollama free — KEEP. If Ollama unavailable, fall back to gpt-4o-mini Batch ($0.075/M input on batch) — cheaper than Haiku.
- Builder code: **keep qwen3-coder@FORGE primary**. Fan-out for high-stakes only.
- Validator: **keep Ollama**. Already $0.

---

5. Day-to-Day Vendor Diversity Health (Minimum Posture)

The minimum diversification posture that does NOT waste tokens:

5a. Recommended Provider Mix (per agent class)

| Agent class | Primary | Secondary (vendor-diverse) | Local-only fallback | |---|---|---|---| | **Orchestrator (John)** | Claude Sonnet (Claude Code host) | — (no real alternative; this is the

platform lock-in) | — | | **/prompt-forge (architecture)** | Claude Opus | **gpt-5 (consensus 2nd opinion)** | qwen3:32b @ FORGE | | **/mehanik (gate)** | Claude Sonnet | — | qwen2.5-coder:32b @ ANVIL | | **builder (code)** | qwen3-coder @ FORGE | claude-sonnet | devstral:24b @ FORGE | | **validator** | qwen2.5-coder:32b @ ANVIL | llama3.1:8b @ ANVIL | — | | **code-reviewer** | claude-sonnet (Haiku-driver) | **gemini-2.5-pro CLI** (already exists!) | — | | **PR review** | gemini-reviewer (already!) | claude code-reviewer | — | | **email-agent (classify)** | llama3.1:8b @ ANVIL (free) | **gpt-4o-mini Batch** (cheap fallback) | — | | **RAG ingest bulk** | (TODO) **gpt-4o-mini Batch API** | claude-haiku Batch | local embeddings (already on bge-m3) | | **Drop voice (future)** | **OpenAI Realtime API** | — | — | | **OCR / vision (Fiken invoices)** | **gpt-4o vision** | claude-sonnet vision | — | | **Browser automation** | mcp__playwright (Anthropic compat) | **OpenAI Operator** (legacy SaaS without APIs) | — |

5b. Day-1 minimum vendor-diverse actions

1. **Add `openai.js` adapter** to `~/system/tools/adapters/` (mirror groq.js structure — OpenAI is OpenAI-compat by definition). Register in `adapters/index.js` `loadBuiltinAdapters`. 2. **Add `openai` engine branch to `tier-router.js`** alongside `ollama`/`cc`/`human-queue`. Add `tier-2o` (OpenAI gpt-4o-mini) and `tier-3o` (gpt-5) entries to `tier-routing.json`. 3. **Add `OPENAI_API_KEY` to Bitwarden** and to env loader (currently absent — `grep OPENAI_API_KEY ~/system/tools/*.js` returns 0 hits). 4. **Wire `lightrag-bulk-upload.js`** to call OpenAI Batch API for embeddings refresh. Single highest-ROI change. 5. **Add a `consensus-router.js`** for /prompt-forge with 3-vendor fan-out (Claude + GPT + Gemini) for top-priority architectural decisions only. Cap monthly spend at $200. 6. **Build `openai-chief-architect`** invocation as a real tool (this audit is the prototype) — councils with vendor-diverse perspective for major decisions. 7. **DO NOT** rip out Claude Code orchestrator. The lock-in is real but moving costs >> diversification benefit at current scale.

Lock-in Risk Score per Vendor

Scale: 0 (no exposure) — 10 (existential)

| Vendor | Today's exposure | What's at risk | Score | |---|---|---|---| | **Anthropic** | 100% of cloud spend ($129K/day), Claude Code is the orchestrator runtime, sub-agent SDK, ZAKON hooks | Entire orchestration loop + 91% of weekly Opus spend | **9/10** | | **Local Ollama (Apple Silicon)** | 30 req/week observed, but ALL builder/validator/email agents depend on it being up | Builder cannot ship if FORGE down; circuit breaker exists but `tier4` falls back to Anthropic = compounds Anthropic risk | **6/10** | | **MLX (Apple)** | 3 servers on FORGE, recently added (gemma-4, qwen3-32b, qwen3-coder-30b) | If Apple MLX ecosystem stagnates, ALAI loses fastest local path; mitigated by Ollama redundancy | **3/10** | | **Groq** | Priority 5 in comms-responder fallback chain, 0 visible spend | Loss = Slack auto-responder degrades to Claude API (more expensive) | **2/10** | | **Google Gemini** | gemini-reviewer agent only; ad-hoc CLI usage | Loss = lose free 2nd opinion on PRs; rebuild with Claude code-reviewer | **1/10** | | **OpenAI** | **0 — not integrated** | Nothing to lose, nothing to gain —

currently 100% potential, 0 realized | **0/10 risk, 8/10 missed-opportunity** |

OpenAI Recommendations (Concrete, Ranked by ROI)

1. **Add OpenAI adapter + key (1 day, $0 setup, 5x cost reduction on RAG ingest).** `OPENAI_API_KEY` to Bitwarden, `~/system/tools/adapters/openai.js`, register in `adapters/index.js`. Switch `lightrag-bulk-upload.js` to gpt-4o-mini Batch. Estimated savings: $5K-15K/week.

2. **Add `consensus-router.js` for /prompt-forge top-tier (3 days, $200/mo budget).** Fan out Claude-Opus + GPT-5 + Gemini-2.5-Pro for architecture decisions only. Catches the recursive drift class of errors documented in `feedback_john_recursive_drift.md`. Cheap insurance.

3. **Fund-aware: If Akershus tilskudd lands (4. maj 2026 deadline), allocate 5% of award to OpenAI integration.** Justification: vendor diversification IS R&D — Forskningsrådet/Akershus reviewers explicitly value multi-provider architecture as evidence of platform sophistication.

4. **Distillation pipeline: connect `flywheel.db.interactions.used_for_training=0` rows to OpenAI Stored Completions → fine-tune gpt-4o-mini.** The schema is already in place. ALAI has 1,429 logged interactions/week on Claude already — this is gold for distillation.

5. **Drop fintech voice MVP via Realtime API.** When Drop reaches a customer-facing milestone, voice onboarding via OpenAI Realtime is 4x faster to ship than Anthropic-DIY. Hold this until Drop product clears regulatory.

6. **Operator for Norwegian government portals (Akershus, Skatteetaten, Brønnøysund).** None have APIs. Anthropic computer-use is comparable but Operator is more battle-tested on form-heavy sites. Pilot for invoice submission and grant filings.

7. **DO NOT replace Claude Code as orchestrator.** Lock-in is real but moving costs are 10x higher than the risk premium right now. Revisit if Anthropic price hikes >50% or if OpenAI ships a clearly superior orchestrator runtime.

CONFIDENCE: HIGH on diagnosis, MEDIUM on cost-savings estimates

(depend on actual ingest volumes), HIGH on the recommendation rank order.

RISKS to this analysis

- Cost numbers ($129K/day) come from `cost-tracker.js` which logs claude-cli internally; if those numbers are inflated or test-dispatch artifacts, the lock-in financial severity is overstated. RECOMMEND: verify against actual Anthropic billing dashboard before any budget action.
- OpenAI strict-JSON / Realtime / Operator capability claims are based on published OpenAI docs. ALAI-specific suitability requires a 2-week pilot before committing.
- Distillation ROI assumes traces are usable; many production traces have PII (Norwegian invoices, customer data) and would need scrubbing before fine-tune. Add a redaction layer.

Revision #2
Created 2026-05-01 08:08:31 UTC by John
Updated 2026-06-07 20:00:46 UTC by John