EdgePointe Baseline — State vs. The Frontier

Date: 2026-07-18 · **For:** Bob, Niven, and Paul
Purpose: A one-time baseline that measures where EdgePointe stands *today* against the current AI frontier, so future weekly updates are deltas against a known starting line — and so you have a jump-start map of where to grow. Reading time ~8 min. This document seeds the **Opportunity Register** (`Opportunity-Register.md`), which is the living, ranked backlog that everything after this feeds.

How to read this

Seven domains. Each gets a maturity rating, one line on where we are, where the frontier is, and the gap that becomes an opportunity. Ratings:

- **Leading** — at or ahead of the frontier; defend and extend.
- **Solid** — competitive; close a specific gap to reach Leading.
- **Developing** — real foundation, meaningful gaps.
- **Gap** — mostly absent; highest-leverage place to invest.

The honest headline: **you are Leading on the two hardest things to copy (agentic build pipeline and design-as-a-system) and have a Gap on the one discipline that would make that lead compound and defensible (formal evals). Closing that one gap is the highest-return move on the board.**

Domain scorecard

1. Agentic pipeline & harness — the Genesis Pipeline · LEADING

Where we are: deep-evaluate (nav-first capture → verbatim copy, CTA/NAP/image-role tables), game-plan generator (grounded-or-fail CI), 15 lints, a sharded gate with drift check, mirror-score, preview-smoke, cache self-invalidation with live-verified ENGINE_REV stamping. This is a genuinely sophisticated build harness — most agencies and MSPs have nothing like it.

Where the frontier is: Anthropic's documented method is now generator + *separate skeptical evaluator* (a builder can't grade its own work honestly), plus evals wired into CI as regression gates. (Anthropic harness design)

The gap → opportunity: our mirror-score/critique still leans on the builder's own judgment. Split it into a standalone critic subagent with an adversarial prompt ("assume this page is mediocre; prove it"), scored against `aesthetics.prompt.md` + tenant `DESIGN.md`. This is the taste-layer Phase 2 already on the roadmap; research now confirms the architecture. *(Register #1.)*

2. Design taste as a system · SOLID → LEADING

Where we are: `aesthetics.prompt.md`, per-tenant `DESIGN.md`, taste lints (L-16/L-17), restyle palette packs, and the elite-hospitality style pack — taste is already partly codified, not just in your head. That's rare.

Where the frontier is: Anthropic's `frontend-design` skill is a prescriptive anti-"AI-slop" rulebook (mandatory rejection of commodity fonts, deliberate display+body pairing, intentional type scale and motion), and Opus 4.8 now carries strong default design instincts with less prompting. (Claude blog)

The gap → opportunity: diff our aesthetics prompt against the official skill and absorb any missing rule; add a guard so the admin/dashboard UI (navy+gold) doesn't drift into Opus 4.8's default cream/terracotta "house style." *(Register #2.)*

3. Evals & measurement · GAP — highest leverage

Where we are: we have lints, mirror-score, and a visual-QA ritual — good instincts, but no *formal* evaluation asset. There is no versioned golden dataset, no library of past failures used to design the metrics, no calibrated rubric-judge with position-swapping.

Where the frontier is: the 2026 discipline is **bottom-up** — collect real failures first, then design metrics to catch them; maintain a focused 50–100 case golden set (stratified production sample + adversarial library + edge cases + shipped-failure replays); reserve LLM-judges (temperature 0, chain-of-thought before verdict, position-swap) for subjective dimensions; wire it into CI as a regression gate so quality is *enforced, not hoped for*. (golden-set guide, rubric evals)

The gap → opportunity: stand up an evals discipline for the Genesis Pipeline — a versioned golden set of "elite vs. not" pages seeded from our own historical builds and their bugs, scored by a calibrated critic, gating promotion. **This is the move that turns your taste into a compounding, teachable, defensible asset.** Every build failure caught since the campaign (object-fit distortion, robots-meta leak, CSS-in-wrong-bucket) is already a golden-set case waiting to be filed. *(Register #3 — do this first.)*

4. Platform & multi-tenant architecture (Cloudflare) · SOLID

Where we are: Workers + D1 + KV, GitHub CI/CD deploy discipline, KV-cached publishing, cache self-invalidation, per-tenant isolation, entitlements module, passwordless auth, audit log. Production-grade.

Where the frontier is / near-term levers: R2 media pipeline (pending your dashboard toggle) unlocks srcset/responsive images; Wrangler "Flagship" adds CLI feature flags to flip capabilities *without redeploying*; D1 location-hint changes are now non-destructive; Workers AI added kimi-k2.6 for cheap edge inference. (Workers changelog)

The gap → opportunity: (a) enable R2 (one toggle) to close the media/architecture-paper-v2 loop; (b) evaluate Wrangler Flagship to replace hand-rolled env-var gating for the config-gated capabilities (Stripe/Twilio/Calendar). **Risk to track:** Workflows per-step billing starts ~Aug 10; `workers-types` v5 drops dated entrypoints. *(Register #6, #7.)*

5. AI product surface — Vantage (receptionist, Reach, leads, reputation, SEO, booking, blog) · LEADING vs. SMB peers

Where we are: an AI receptionist grounded in tenant content, Reach analytics with AI commentary, Lead Inbox + alerts, reputation engine, SEO toolkit emitting JSON-LD, booking v2, and a blog engine — a category-first stack; the venue-competitive report found the AI receptionist was 0-of-9 among competitors.

Where the frontier is: the voice market has split into OpenAI Realtime (end-to-end, sub-500ms, premium price) vs. ElevenLabs (composable, best voice, ~50% cheaper at volume); MCP servers now exist for Stripe, Twilio, and Resend — the exact integrations Vantage has config-gated. (voice comparison, Twilio MCP)

The gap → opportunity: most gaps here are *account-blocked, not capability-blocked* (EIN/Stripe/A2P). When those clear, the Stripe/Twilio/Resend MCP servers make the integrations fast to build. Meanwhile, confirm venue tenants ship LocalBusiness+Event+FAQ+Review schema (now make-or-break for AI-Overview visibility). *(Register #4, #5.)*

6. Business model & economics — productization, margin, moat · DEVELOPING (biggest strategic upside)

Where we are: 4-tier Vantage pricing, sellable kits, a GTM plan (1,500 customers / $113.5K MRR by Dec 2026), per-client profitability and billing/licensing skills exist.

Where the frontier is: automation-first MSPs are hitting **18–22% EBITDA vs. 11–14% for traditional MSPs**, serving far more clients per technician via a "zero-touch" model (AI handles 90%+ of tickets/patches/provisioning). Yet **only 13% of MSPs monetize AI as a revenue line** while 48% say clients will demand it — a wide-open gap. The winning play: use AI first to cut delivery cost, *then package the same capability as a client-facing service.* (Kaseya, MSP profit shift)

The gap → opportunity — this is your anti-"one-trick-pony" move: your core competency isn't "cloning venue sites." It's *AI-native operations* — agentic pipelines + measurable taste + edge deployment. Point the same engine at **Managed IT delivery**: automated ticket triage, security posture monitoring, QBR/report generation, billing reconciliation. Same competency, new revenue lines, and it lifts EBITDA on the *existing* book. This is how EdgePointe grows exponentially instead of linearly. *(Register #8 — the big one for company growth.)*

7. Moat & defensibility posture · the synthesis

Where the frontier thinking lands: in 2026, *model access is explicitly not a moat.* The five durable moats are **proprietary data, workflow depth, distribution, brand, and network effects.** Value accrues to products that fit a specific workflow, capture proprietary feedback, and get painful to replace — the "data flywheel." (AI moats 2026, defensibility)

Where EdgePointe already stands: you have four of the five. **Workflow** (the Genesis Pipeline + skill library), **proprietary data** (tenant content, client environments, historical builds), **distribution** (DFW +

Bob's executive relationships), **brand** (EdgePointe/Vantage, navy+gold, flat-pricing positioning). You are *not* betting on staying ahead on the model — which is the correct bet.

The gap → opportunity: make the **data flywheel explicit and instrumented.** Right now every build, every owner edit, every ticket is a proprietary signal that mostly evaporates. Capture it: builds/edits feed the evals golden set and the aesthetics prompt; client-environment data feeds Managed IT automation; owner behavior feeds Vantage product decisions. The moat isn't any one product — it's that *using EdgePointe makes EdgePointe better in ways a competitor starting today cannot catch. (Register #3 and #8 are the flywheel's first two turns.)*

The leapfrog thesis (24–48 months) — for leadership

State it plainly to Bob and Niven: **we are not betting that Claude stays ahead of GPT or Gemini. We are betting that whoever encodes their taste, workflows, and client data into an AI-native operating system first and fastest owns the market — and the model underneath is interchangeable and getting cheaper for everyone.**

That reframes the whole competitive question:

- Website builders (Wix/Squarespace/GoDaddy) are template-generators; even Webflow's customization-first AI can't ship *your measured, critiqued, owner-editable elite site + business dashboard.* Our moat is the pipeline and the taste rubric, not the fact that we use AI.
- Traditional MSPs are getting their help-desk margins compressed by AI. We can be the automation-first MSP capturing 18–22% EBITDA while they defend 11–14% — and sell the automation back as a service the 87% aren't monetizing.
- The model getting cheaper every quarter is *tailwind for us*, because our cost to produce an elite site or resolve a ticket drops while our workflow/data/distribution moat stays.

The de-risked version leadership can approve: invest in the four moats we already hold (workflow, data, distribution, brand), treat the model as a commodity input, and measure the compounding quarterly.

Your personal baseline (Paul)

Strong: systems thinking, design taste, and the discipline to encode both into a pipeline rather than doing it by hand. You already build institutional memory (skills, DESIGN.md, RESUME-HERE). That is exactly the right instinct against being the bottleneck.

Grow into, in priority order: (1) **evals/measurement** — the one discipline that converts your taste into a company asset; (2) **agent/harness architecture** as a first-class engineering practice (read Anthropic engineering like RFCs); (3) **MSP economics** so you can point the engine at new revenue deliberately; (4) **moat thinking** as the lens on every bet.

How you absorb it without becoming the bottleneck: learn just-in-time (deep-dive only what you're about to ship), learn by doing one experiment a week, and encode every lesson into a skill or prompt so the company keeps it. The cadence below is built around that.

Cadence going forward

- **Daily — Game-Changer Watch (reconfigured today):** a lean scan that pings you *only* when something material lands (a major model release, a pricing/deprecation change hitting our stack, a competitor category move, a capability that changes the plan). Silent otherwise. Anything notable is appended to the Opportunity Register.
- **Weekly — Research Update (new, Mondays):** the substantive digest — what changed, re-ranked Opportunity Register, and one suggested lab-hour experiment for you.
- **Monthly — Capability Review:** what got cheaper/newly possible, and the one thing we shipped because of it (doubles as the leadership update).
- **Quarterly — Strategic/Moat Review:** with leadership; re-rank priorities against this baseline, kill dead bets, reset the thesis.

Everything routes through one artifact: `Opportunity-Register.md` — the ranked backlog that turns research into revenue instead of trivia. That file, kept current, is the growth engine; this baseline is its starting state.

Guardrails: Vantage changes ship via GitHub CI/CD only; D1 changes additive/idempotent; account-dependent capabilities (Stripe/Twilio) stay config-gated and fail-soft until the EIN/credentials exist; republish/purge affected tenants after any engine change.