ARCHITECTURE REVIEW · AN UPSWING SYSTEM

FORGE

A production-grade, multi-agent AI pipeline, architected and operated solo.

FIND · OUTFIT · REACH · GROW · EMBARK

TEMPORAL ORCHESTRATION / LOCAL-FIRST RUNTIME / SALEM, OREGON

SYSTEM STATUS

FORGE is a production-grade, multi-agent AI pipeline that I architected and operate solo. It runs end-to-end on real data and is currently pre-revenue, sitting at the manual first-customer approval gate. Stages one through three, Find, Outfit, and Reach, are built and shippable on demand. Grow is wired end-to-end and verified in test mode. Embark is designed and deliberately unbuilt.

This is a self-authored architecture review of a system I designed, built, and run. **Every figure carries a claim-discipline tag, and the deliberate gaps are named as plainly as the strengths.** The intent is to let a technical reviewer trace any claim to a database query, a timed run, or a line of code.

MEASURED | CODE-VERIFIED | SELF-REPORTED | NOT BUILT

Authored by **Sage Dutra** · Architect and operator · sage.dutra@gmail.com

01 · OVERVIEW

What FORGE is, and where it stands

An autonomous-with-a-human-gate web-design-agency pipeline for a Salem, Oregon studio. It finds local-trades businesses, researches each into a scored dossier, has an LLM copywriter and then an LLM builder produce a complete bespoke website, then halts at an operator approval gate before anything goes outbound.

The name is the five-stage acronym, Find, Outfit, Reach, Grow, Embark. Each letter is a standalone activity folder with its own subtasks, prompts, and tools, orchestrated by Temporal so every step is durable and individually observable. The word autonomous always carries the human gate in the middle. The system can run a prospect from discovery to a staged, approval-ready site without intervention, and it dispatches outreach only after a person approves.

16.1
MIN MEDIAN BUILD
P90 20.6 · N=56 ·
BUILD ONLY

63
SPEC SITES
BUILT

54
STAGED &
APPROVAL-READY

2,405
CONTRACTORS IN
THE SCORED POOL

133
COMMITS SINCE
2026-03-30

The discovery funnel, kept uncollapsed

STAGE	COUNT	SOURCE OF TRUTH
Reservoir, CCB licensing registry	45,302	COUNT(*) CCB_LICENSES
Reservoir, BCD licensing registry	48,085	COUNT(*) BCD_LICENSES
Serviceable vertical, plumbers (classified slice)	7,857	CCB WHERE TRADE='PLUMBER'
Promoted into the scored pool	2,405	COUNT(*) PROSPECTS
Deep-scored on the full six-dimension matrix	72	SCORING ? 'WEBSITE_QUALITY'
Built (distinct prospects)	63	ARTIFACTS KIND='BUILT_SITE'
Staged (qa_complete)	54	STATUS='QA_COMPLETE'

READING DISCIPLINE

The reservoir is two overlapping registries and is never summed. The pool figure is read as 2,405 promoted and 72 deep-scored, not one number. The funnel ran as separate manual batches, so it is leaky and non-monotonic by construction. These numbers were pulled from the live database on 2026-06-17 and are reproducible from the queries shown.

Stack at a glance

Orchestration
Temporal.io TypeScript SDK, version ^1.15.0. Local self-hosted dev server with SQLite persistence. Not Temporal Cloud.

Datastores
Local Postgres 18 as system of record, 22 base tables. Neon Postgres is a read-only cloud mirror. Temporal owns workflow state, so the app DB holds no orchestration tables.

Models
Claude Max via CLI (claude-opus-4-7, subscription auth) for creative work. Google Gemini 2.5 Flash for analysis and vision. Perplexity and Tavily as conditional research gap-fillers.

Runtime
One always-on Mac Mini M4. Local-first on the critical path, with edge-only cloud for hosting and a read mirror. No OpenAI, no AWS, GCP, or Azure SDK, no Docker.

02 · SYSTEM CONTEXT

FORGE in its environment

A C4-style context view. One operator, one class of end customer, and a ring of outbound service dependencies. The system is local-first, with a single narrow inbound path.

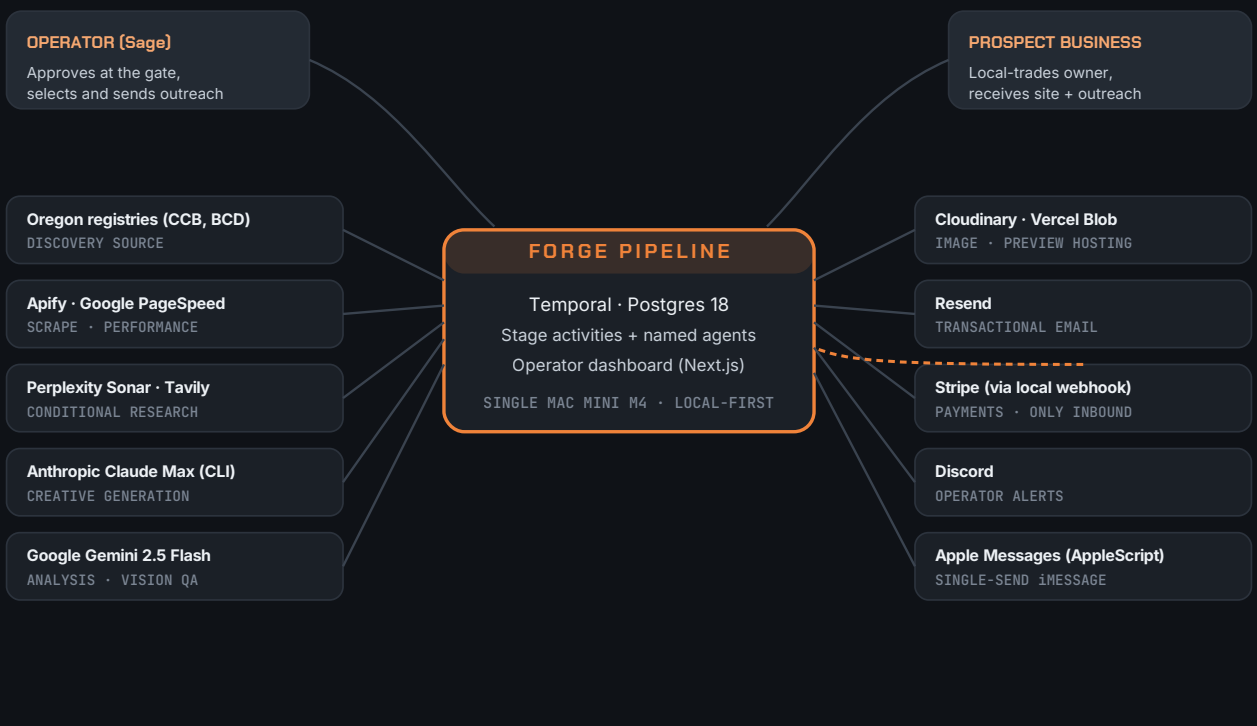

FIGURE 1 · SYSTEM CONTEXT. SOLID LINES ARE OUTBOUND CALLS FORGE INITIATES. THE DASHED EMBER LINE IS THE ONLY INBOUND PATH, A TUNNELED STRIPE WEBHOOK INTO A LOCAL SERVICE.

The trust boundary is unusually tight for an AI system. Almost every dependency is an outbound call the pipeline initiates and controls. The lone inbound path is a Cloudflare tunnel that reaches a local Stripe-webhook service, and that path is signature-verified and idempotent before it can touch any state. There is no OpenAI anywhere in the pipeline, no cloud orchestration plane, and no AI image generation. Image work is enhancement and transformation through Cloudinary plus HTML screenshots through Playwright, and Gemini is used for visual analysis rather than image creation.

03 · COMPONENT ARCHITECTURE

Inside the pipeline

Two datastores cleanly separated, two workers, a small set of durable workflows, five stage activity packages, four built agents, and an operator dashboard that holds the only hard gate.

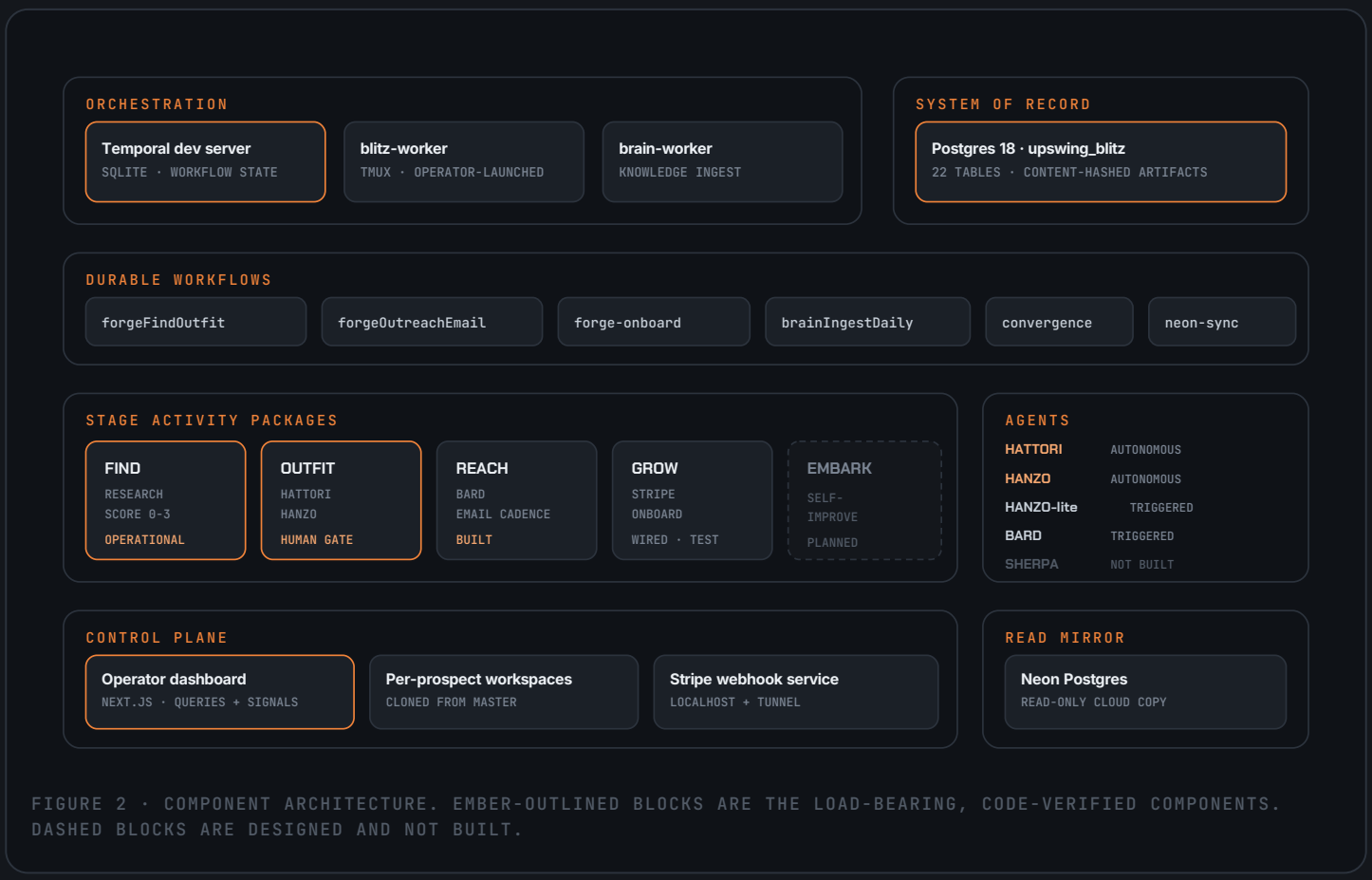

FIGURE 2 · COMPONENT ARCHITECTURE. EMBER-OUTLINED BLOCKS ARE THE LOAD-BEARING, CODE-VERIFIED COMPONENTS. DASHED BLOCKS ARE DESIGNED AND NOT BUILT.

The separation of concerns is the spine of the design. Temporal owns workflow state in its own SQLite store, and Postgres owns application data, so the application database carries no orchestration tables. Each prospect runs in an isolated workspace cloned fresh from a single master template, which means every build uses the current instructions and skills and each cloned directory doubles as a timestamp of the exact system configuration that produced it. Two autonomous agents, HATTORI and HANZO, run inside the build workflow. Two more, HANZO-lite and BARD, are triggered rather than autonomous. A fifth, SHERPA, is scaffolded and deliberately unbuilt.

04 · SEQUENCE, BUILD PATH

Find to Outfit, halting at the human gate

The core build flow. Copy is written first because copy dictates layout, then the builder writes a complete site from scratch. Audits run as informational second eyes, and the only hard gate is a person.

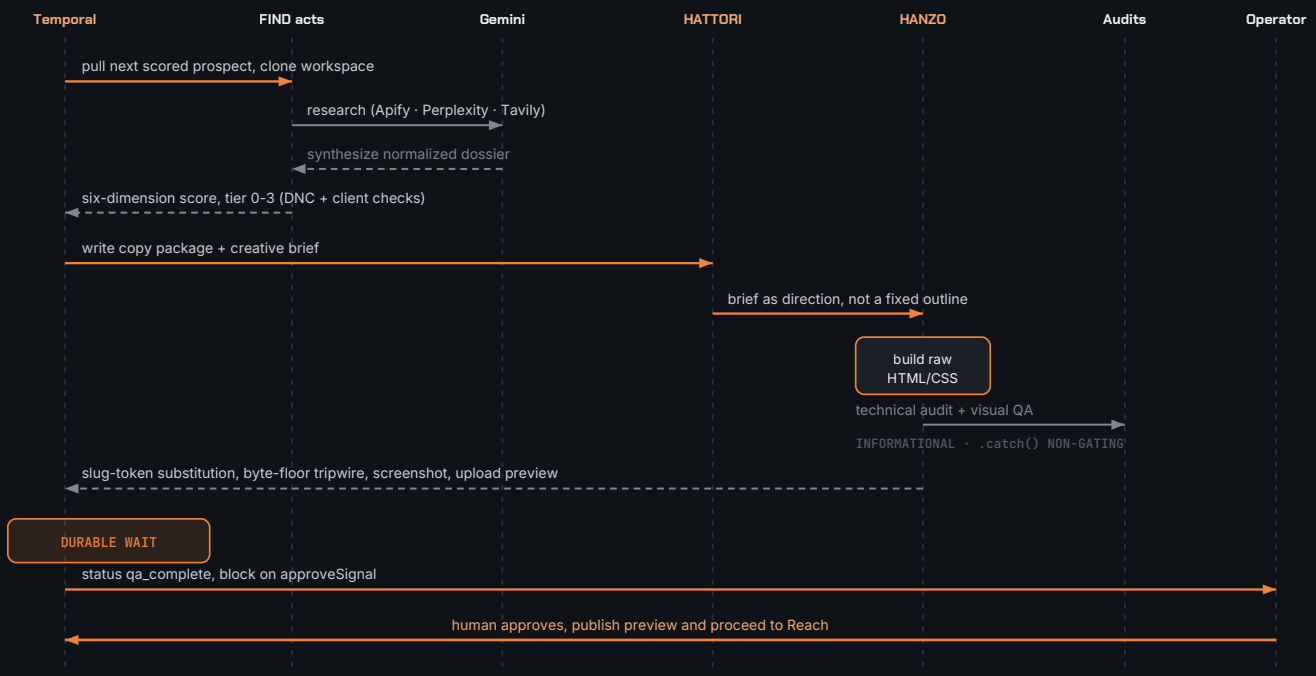

FIGURE 3 · FORGEFINDOUTFIT. EMBER ARROWS ARE CREATIVE GENERATION ON CLAUDE MAX. DASHED RETURNS ARE DURABLE WRITES TO POSTGRES. THE AUDITS NEVER GATE.

Two design choices in this flow are worth a reviewer's attention. First, the technical audit and the visual QA both run after the build, in parallel, and each is wrapped so a failure is logged but cannot stop the workflow. They are second eyes for the human approver, not an automated quality gate. Second, the sole hard gate is the operator approval signal, which Temporal holds as a durable wait. The workflow can sit at that gate for as long as it takes without consuming resources or losing state.

Reach after approval, and a fail-safe payment intake

Outreach begins only after a person approves, and every send is screened against a do-not-contact list. Payment intake is idempotent and rolls back cleanly when the orchestrator is unreachable.

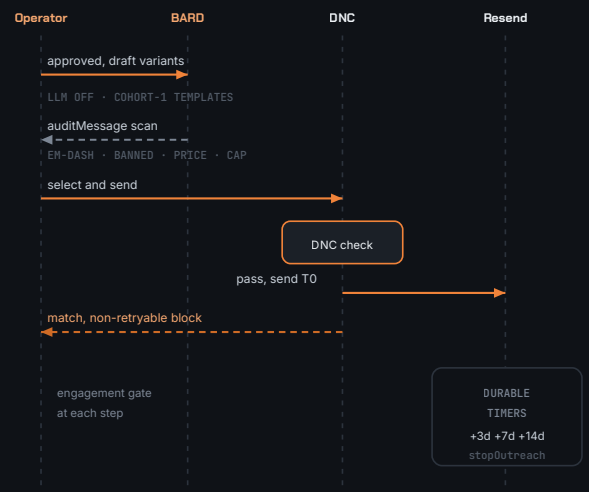

FIGURE 4 · FORGEOUTREACHEMAIL.

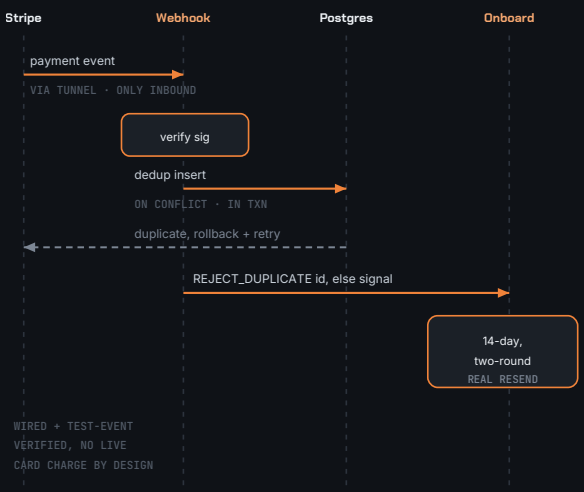

FIGURE 5 · STRIPE TO FORGE-ONBOARD.

Reach ships an autonomous email cadence with engagement gates at each step and a signal that halts the sequence on reply. The creative call inside BARD is off by default, so the cohort runs on locked, reviewed template copy with a deterministic audit on every outbound message. iMessage is a single send on demand through AppleScript, not a multi-touch cadence. The unifying reach orchestrator and a multi-touch iMessage cadence are designed and not yet built. On the payment side, the webhook deduplicates inside a Postgres transaction and rejects duplicate workflow ids, and it rolls back so Stripe retries cleanly when Temporal is unreachable. Grow is wired end-to-end and has been exercised on a signed test event. No real card has driven onboarding yet, by design, because the system is pre-revenue.

06 · TECHNOLOGY DECISIONS

What was chosen, and the honest caveat

Each row is a real decision with a reason and a limitation stated next to it. The caveats are the point, since a senior reviewer trusts the system more when the boundaries are named.

AREA	CHOICE	WHY	HONEST CAVEAT
Orchestration	Temporal.io TS SDK ^1.15.0, local self-hosted dev server, SQLite persistence	Durable execution, retry classification, signal-based gates, durable timers, the Patching API to version running workflows	Not Temporal Cloud. The dev server purges history aggressively, so a recovered run is not retained as a log
System of record	Local Postgres 18, with Neon as a read-only cloud mirror	Clean separation from workflow state, 22 tables, no orchestration tables in the app database	The mirror is read-only and its refresh cadence is not claimed
Model routing	Claude Max via CLI (opus-4-7) for creative, Gemini 2.5 Flash for analysis and vision, Perplexity and Tavily conditional	Job-appropriate routing on cost and quality, enforced by which helper each activity imports	Routed by import, not a central router. It is the Claude CLI, not the SDK. No MCP in the pipeline
Site generation	Builder writes raw HTML and CSS from scratch, no template fill	Bespoke output per business, the lesson learned from an over-constrained earlier version	Pre-ship QA is model self-attested, not independently rendered or programmatically gated
Human-in-the-loop	One approval signal at qa_complete	A money and reputation touching output gets a person before anything goes outbound	Autonomous up to the gate only. Outreach is dispatched after approval, never auto-fired
Runtime	One always-on Mac Mini M4, local-first	Cost, control, and simple observability for a single operator	Workers run in tmux and are operator-launched, not daemonized. No horizontal scale
Images	Cloudinary transform plus Playwright screenshots, no AI image generation	Avoids unreliable asset generation on the critical path	Gemini is used for visual analysis only, never to generate the images

07 · TRADEOFF ANALYSIS

The decisions that cut both ways

Six choices carry a deliberate cost. **Constraint versus creativity**, where an earlier component-and-palette system proved that more constraint produced worse output, so the design now favors freedom plus observability. **Single-pass versus a QA loop**, where an automated re-stylize loop was removed because it homogenized output without improving it, replaced by single-pass generation, informational audits, and the human gate. **Autonomy versus a human gate**, where a load-bearing approval gate was chosen over full autonomy because the output touches a business owner's money and reputation. **Local-first versus cloud**, where single-node local bought cost, control, and observability at the price of horizontal scale and always-on workers. **Flat-rate versus metered**, where creative volume on a subscription removes per-call telemetry and makes a precise marginal cost unverifiable. **Durable history versus a dead-letter queue**, where Temporal retry and failure history is the deliberate substitute for a separate replay store.

THE THROUGH-LINE

Every one of these trades a capability the system does not yet need for clarity, control, and honest observability that it needs right now. That is the judgment a single operator running a real pipeline has to make, and each cost is named rather than hidden.

What goes wrong, and what catches it

This is the strongest territory in the system. Most of these guards were built in response to a real, diagnosed incident rather than designed up front.

FAILURE MODE	MECHANISM THAT CATCHES IT	STATUS
Silent design homogenization	Build-fingerprint capture plus a nightly convergence detector that flags any dimension where three of the last five builds match on template, palette, fonts, hero strategy, or section count, then fires a Discord alert	BUILT FROM A DOCUMENTED INCIDENT
Truncated or empty model output	A byte-floor tripwire throws a non-retryable failure on truncated generation, and empty or invalid model output is classified non-retryable rather than looped	CODE-VERIFIED
Misrouted leads	A literal site-slug token is substituted deterministically after the build, so an inquiry form can never post to the wrong prospect	CODE-VERIFIED
Off-brand or non-compliant outbound	A deterministic message audit scans every outbound message for em-dashes, banned words, internal tier-name leaks, price, and a character cap	CODE-VERIFIED
Duplicate or lost payment webhook	Transactional dedup with insert-on-conflict plus rejected duplicate workflow ids, with rollback so Stripe retries cleanly when the orchestrator is unreachable	CODE-VERIFIED
Spec drift from broad agent edits	A change-control discipline of surgical edits only, a sandbox clone-diff-confirm-commit loop, and a mandatory one-to-one spec update on every code change	BORN FROM A REAL DRIFT INCIDENT
Build-quality misses	Light, targeted guardrails added in the master template at the taste layer, since failure now lives at taste rather than infrastructure	SELF-REPORTED

KNOWN GAPS, NAMED RATHER THAN HIDDEN

No programmatic enforcement that the builder's self-QA actually ran or passed before a site is marked complete, since the checklist is model self-attested with no browser. No offline or regression eval harness, the vendored one is unwired. No PII redaction or prompt-injection defense on ingested web content. No dead-letter queue, Temporal history is the deliberate substitute. These are scoped choices for a pre-revenue single-operator system and are the first hardening items before any multi-tenant production.

A distinction worth naming is instructed versus verified. The builder runs a structured self-QA checklist that includes link-integrity and form-wiring checks, but only the anti-slop grep genuinely executes. The rest are the model reading its own HTML, and nothing programmatically gates completion. Separating what an agent is told to check from what is actually enforced is itself a senior reliability insight, and the cheapest next improvement is to promote the two checks that are verifiable from on-disk HTML, broken navigation anchors and a missing form token, into enforced gates.

09 · COST MODEL

Where the money goes, and what is honestly unknown

- **Creative generation** runs on a flat-rate Claude Max subscription, so the marginal cost of a build for the copywriter and builder is effectively zero.
- **Metered externals per prospect** are Gemini 2.5 Flash for synthesis and vision QA, Perplexity and Tavily when conditionally invoked, Apify scrapers, Google PageSpeed, Cloudinary, Vercel Blob, and Resend.
- **Fixed cost** is one Mac Mini M4 plus the subscriptions. There is no cloud compute bill, no Docker or Kubernetes, and no AWS, GCP, or Azure.

THE HONEST POSITION

No per-prospect token or cost logging exists in the provider wrappers, so a precise marginal cost is unverifiable. Any prior per-prospect cent figure is retired. To make it claimable, token capture is added to the wrappers. Flat-rate creative work is excluded from any marginal figure regardless.

Cost discipline is demonstrated where logging does exist. A sibling system, a second multi-agent pipeline built on LangGraph, routes a high-cardinality critic panel onto the flat-rate subscription transport and meters only the few high-stakes calls to a separately isolated key. Two verified end-to-end runs landed at 0.71 and 0.87 dollars of real metered spend. That is the pattern FORGE would adopt the moment per-call telemetry is wired in.

10 · SECURITY AND PRIVACY MODEL

Boundaries, secrets, and the gaps

Secrets
All credentials flow through a central secrets manager. No secrets are hardcoded, and commands run inside a secrets-injecting wrapper.

Auth isolation
The API key is stripped from the Claude CLI environment so it uses subscription auth. The sibling system keeps a separate metered key that is never written to the global environment, a billing-isolation invariant.

Network posture
Local-first. The only inbound path is a tunnel to the local Stripe-webhook service. Everything else is an outbound call the pipeline initiates.

Payment integrity
Webhook signature verification, idempotent transactional dedup, and rollback for clean retries protect the one inbound surface.

Outbound-contact safety
A do-not-contact blocklist is checked before every send as a non-retryable block, a deterministic audit scrubs banned content, and a person approves before any first contact.

Data separation
The system handles Oregon trades licensing data only, kept cleanly isolated from any other venture data.

SECURITY GAPS, STATED PLAINLY

There is no PII redaction or prompt-injection defense on ingested web content, the model self-attested QA is not independently verified, and the workers are operator-launched rather than hardened and daemonized. These are deliberate scope choices for a pre-revenue single-operator system, and they are the named first items on the path to multi-tenant production.

11 · LESSONS LEARNED

The failing-forward arc

FORGE reached its current shape through roughly five architectural iterations across multiple repositories. The current Temporal repository alone went through three architecture cutovers in about two months, across 133 commits since 2026-03-30.

ITERATION	WHAT IT WAS, AND WHAT IT TAUGHT	LESSON CARRIED FORWARD
V1 · eleven agents	One agent per business function plus a single overloaded orchestrator. Never got past copywriting into design	Context bloat is real, consolidate roles, give agents persistent memory
V2 · one super-agent	An overcorrection to a single hat-switching agent. Drowned in context, with no file structure and no auditability	A single mind is not a structure, failures need to be inspectable
V3 · over-constraint	A full curated component and palette system. Quality dropped because stripping creative freedom made every site look the same	More constraint produced worse output, the core inversion the system now rejects
V4 · Claude-native	Migrated onto the Claude Max subscription for scheduling, custom agents, and mobile. Quality rose, build time climbed, still no observability of silent failures	Quality without observability still hides silent failure
V5 · Temporal spine	The current repository. Adopted Temporal for per-activity observability and durable execution, with a file-based per-activity architecture	Durable, observable orchestration is the foundation everything else sits on

The core inversion

More constraint produced worse output. Creative freedom plus durable observability beat a rigid component system. That single lesson reshaped the whole architecture.

Instructed is not verified

Separating what an agent is told to check from what is programmatically enforced is a senior reliability insight, and it is stated openly rather than papered over.

Failure lives at the taste layer

Once the durable spine was right, the agents reliably produced requested output. Remaining misses were quality and taste, handled with light, targeted guardrails.

Orchestration breadth

A second multi-agent system built on LangGraph and LangChain is evidence the orchestration judgment generalizes beyond a single framework.

PROOF ARTIFACT FOR REVIEWERS

A public, sanitized reference repository is live at github.com/upswing-solutions/forge-pipeline. The Temporal, Postgres, and LLM orchestration core is real, external integrations sit behind stubbed adapters that return deterministic fake data, and it is secret-clean. It is the primary public proof artifact for this review.

Claim discipline. Every number in this document is reproducible from a database query, a timed run, or a line of code. The build-latency figure is build time only, copy handoff to finished HTML, not business name to staged site. Capabilities not yet built are labeled not built. Nothing here is described as live paying production. The system is production-grade and operated end-to-end on real data, and it is pre-revenue at the first-customer gate by design.