

FLAGSHIP CASE STUDY



FORGE

FIND • OUTFIT • REACH • GROW • EMBARK

A production-grade, multi-agent AI pipeline
that researches, designs, builds, and ships
bespoke websites for local trades,
autonomous, with a human at the single
highest-stakes gate.

ARCHITECTED & OPERATED SOLO / RUN END-TO-END ON REAL DATA / PRE-REVENUE BY DESIGN

THE PREMISE

Build the site first. Earn the conversation second.

Trades owners do good work and lean on referrals that matter less every year. The agency model takes advantage of that gap, promising the world, under-delivering.

FORGE inverts the sale. A plumber gets a finished website and a single message: "I built you a website. If you like it, great. If not, no problem." Build it cheaply enough to accept a 99% rejection rate and still be profitable.

One human moment: review, edit, send. Everything else automated.

01 · BUILD-FIRST, NOT PITCH-FIRST

Bypass the objections, the upfront payment, the creative handholding tradespeople have no time for.

02 · ONE HUMAN MOMENT

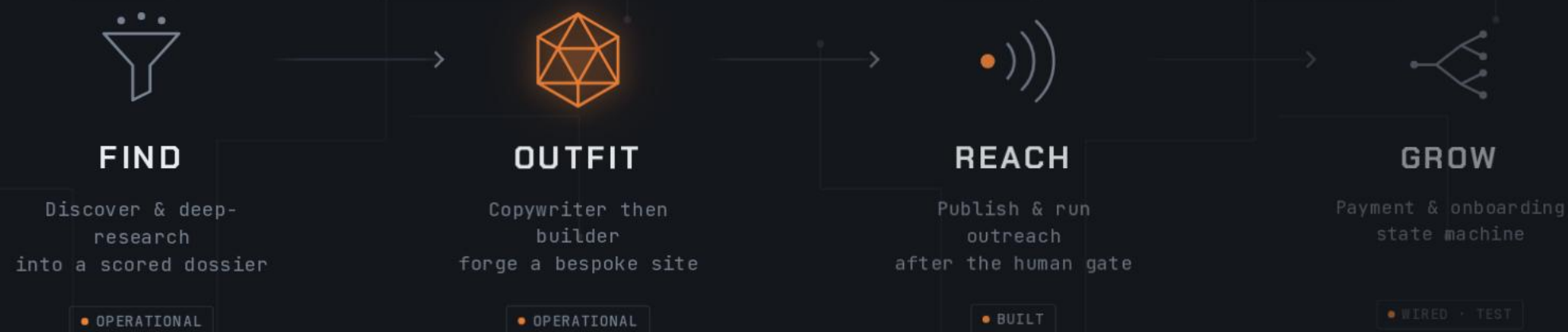
Make the most revenue with the least of my own time.

03 · THE FIRST CONTACT IS THE PROOF

Real license number, real reviews, real services, not a template with placeholder text.

Five stages. One durable spine.

The name is the pipeline. Each letter is a standalone activity folder with its own subtasks, prompts, and tools.



FIND. Discover and deep-research into a scored dossier.



FIND

• OPERATIONAL • AUTOMATED

Works a pre-scored, location-filtered Postgres hopper and clones a fresh workspace per prospect.

01 • VERIFY

A verification pass on review counts, CCB license validity and expiration, and contact details. Only DNC and already-a-client trigger disqualification.

02 • ASSETS

Image scrape and Cloudinary optimization. Stock placeholders are flagged transparently whenever real photos are too few.

03 • DOSSIER

Gemini synthesizes the raw intel into a single normalized dossier, the source of truth every downstream agent reads.

04 • SCORE

A six-dimension qualification score with explicit weights, summed to a 0-3 tier.

• WEBSITE_QUALITY .25	• SOCIAL_PROOF .20	• CONTACT_CONFIDENCE .15	• MARKET_OPPORTUNITY .15	• BUSINESS_MATURITY .15
• COMPETITIVE_POSITION .10				

OUTFIT. Two agents forge a bespoke site, then it halts at the gate.



OUTFIT

• OPERATIONAL • HALTS AT HUMAN GATE

16.1 min median
build only • p90 20.6 • n=56

01 • HATTORI • copy first

The copywriter goes first because copy dictates layout. It hands over a copy package plus a separate creative brief.

02 • HANZO • builds the site

The builder writes a complete bespoke site in raw HTML and CSS from scratch. No template fill.

03 • INFORMATIONAL AUDITS

A Playwright DOM-lint (no LLM) and a seven-dimension Gemini Vision QA run in parallel, each wrapped in `.catch()`. Second eyes for the human, never a gate.

04 • THE HUMAN GATE

The sole hard gate in the pipeline is the human `approveSignal`. Nothing reaches a prospect without it.

REACH. Publish and run outreach, only after approval.



REACH

• BUILT • BEGINS AFTER APPROVAL

Every step runs after one human approval. Engagement gates decide whether the next touch sends.

01 • BARD DRAFTING

Message variants and a per-prospect sales playbook, generated behind an approval gate with the LLM off by default. A deterministic `auditMessage` scans every outbound draft.

02 • EMAIL CADENCE

An autonomous `forgeOutreachEmail` cadence at T0, +3d, +7d, +14d with per-step engagement gates and a `stopOutreach` signal.

03 • DIRECT SEND

Single-send Resend email and single-send AppleScript iMessage, dispatched on demand.

NOT BUILT

The unifying `forgeReach` orchestrator, and a multi-touch iMessage cadence. The cadence is email only.

GROW. A payment and onboarding state machine.



GROW

• WIRED END-TO-END • TEST-VERIFIED

Live payment infrastructure is wired and exercised on the local stack with a signed test event.

01 • CHECKOUT

Stripe checkout fires a webhook, which raises a Temporal signal. Payment links, webhooks, and signature verification are all wired.

02 • ONBOARDING WORKFLOW

The signal starts a 14-day, two-round, multi-signal onboarding workflow (`forge-onboard.ts`) that sends real Resend emails.

03 • PRE-REVENUE BY DESIGN

Exercised through the local stack on a signed test event. No real card charge has driven onboarding yet, by design. The first-customer gate is deliberate, not a limitation.

EMBARK. Agents that review and improve their own skills.



EMBARK

• PLANNED • NOT BUILT

An empty stage folder today. The closing loop of the system, designed but deliberately unbuilt.

01 • PERIODIC REVIEW

On a schedule, each agent reviews its own logged learnings drawn from the long-horizon Postgres memory table.

02 • PROPOSE EDITS

An agent proposes edits to its own skills, so a correction made once is inherited by every future prospect.

03 • DELIBERATE RESTRAINT

By design this is targeted skill editing, not a growing, always-loaded memory file that bloats every agent's context.

WHY IT IS LAST

EMBARK only earns its place once the first four stages have run on enough real prospects to have learnings worth compounding.

Five architectures. Each one taught the next.

I did not design the right system up front. I arrived at it by failing forward.

V1

ELEVEN AGENTS

One agent per business function plus one overloaded orchestrator. Never got past copywriting.

Lesson • context bloat

V2

ONE SUPER-AGENT

Overcorrected to a single hat-switching agent. Drowned in context, no auditability of failures.

Lesson • need structure

V3

THE CORE INSIGHT

Bought the full Tailwind Plus library, built a curated component system. Quality dropped. I had turned an infinitely creative system into a factory worker.

More constraint → worse output

V4

CLAUDE-NATIVE

Rebuilt on Claude Opus + the Max subscription. Quality rose, build time climbed, still no observability of silent failures.

Lesson • need a spine

V5

TEMPORAL SPINE

Per-activity observability, durable execution, and the file-based FORGE folder architecture. The system as it runs now.

Current build

Five iterations across multiple repos; the current Temporal repo alone went through three architecture cutovers in about two months.

Durable orchestration as the spine.

Every stage is a Temporal workflow. The two datastores are cleanly separated. Temporal owns workflow state, Postgres owns application data, so the app DB carries no orchestration tables.

TEMPORAL • WORKFLOW STATE

Self-hosted local dev server. History in SQLite at `data/temporal.db`.

POSTGRES • SYSTEM OF RECORD

Local app SoR with a Neon cloud instance as a read-only mirror.

WORKERS

Operator-launched TypeScript workers (blit-worker, brain-worker) handle research, disk writes, and the Gemini synthesis calls.

```
// per-activity classification
RetryPolicy {
  retryable:    ApplicationFailure
  nonRetryable: ApplicationFailure // empty / invalid output
}
```

approveSignal

Signal-based human-in-the-loop gates with blocking multi-day timers, e.g. a 14-day approval condition.

Patching API

Safely version a running workflow without breaking in-flight executions.

defineQuery

Live state reads against a durable, crash-safe execution history.

Named for the swordmakers.

The writing supplies the finest materials; the builder forges the blade. Two agents run autonomously inside the build workflow, two are triggered, one is scaffolded.

• AUTONOMOUS

HATTORI

Copywriter & creative director. Turns the dossier into a copy package and a separate creative brief: direction, not hard requirements. Writes “like a neighbor.”

• AUTONOMOUS

HANZO

The builder. Ingestion pass, writes its own internal design plan, wire-frames, then writes a complete bespoke site in raw HTML/CSS from scratch, no template fill.

• TRIGGERED

HANZO-lite

The apprentice. Surgical minimal-diff revisions, template repurposing for the reusable library, and publishing: deploy to Vercel, generate the OG image.

• TRIGGERED

BARD

Outreach & sales coach. Drafts behind an approval gate, builds a per-prospect playbook; a deterministic `auditMessage` scans every outbound draft.

• NOT BUILT

SHERPA

Client success, retention, and analytics for the GROW stage.

Scaffolded • not built

RIGHT MODEL, RIGHT JOB

Routed by job type, enforced in code.

The rule that decides routing is quality-per-dollar at the task's volume, enforced by which helper each activity imports, not a central router.

CREATIVE · LOW VOLUME, HIGH STAKES

Claude Max

The four creative agents call Claude through the CLI on the Max subscription, so creative quality carries no per-call API cost.

claude-opus-4-7 · CLI, not SDK

ANALYTICAL & VISION · HIGH VOLUME

Gemini 2.5 Flash

Research synthesis, dossier compilation, brand/logo vision, and the visual-QA pass: fast and cheap at volume.

analysis only · generates no images

RESEARCH · CONDITIONAL GAP-FILL

External APIs

Perplexity Sonar & Tavily for business/owner intel and Oregon SoS lookups, plus Apify scrapers and PageSpeed, skipped when the data is already known.

no OpenAI · no MCP in the pipeline

Engineering for the failures you can't see.

The hardest failure I hit was invisible: a third-party skill bundle silently steered every build toward the same aesthetic. It took about a week to diagnose.

SILENT-HOMOGENIZATION DETECTOR

A build-fingerprint capture plus a nightly convergence detector. It records each build's template, palette, fonts, hero strategy, and section count, flags any dimension where three of the last five builds match, and fires a Discord alert.

Now it catches aesthetic drift in two or three builds instead of a week.

ROOT CAUSE · THE MARKDOWN TRAP

The copywriter produced rigidly structured markdown; the builder followed the section order to the letter. The formatting of the copy doc was homogenizing every site. Fix: hand over copy as suggestions, not an outline.

FAILURE LIVES AT THE TASTE LAYER

After heavy debugging, the agents reliably produce the requested output. Remaining misses are quality and taste, fixed once in the master template so every future prospect inherits the correction.

Deterministic code where models can't be trusted.

Some failures look right but are wrong. A language model will not catch them, so they are handled deterministically, in code.

`__SITE_SLUG__`

Token substitution into the lead form's hidden field, so a model that drops or hallucinates the slug cannot misroute a lead.

`byte-floor tripwire`

Throws a non-retryable error if generated HTML is suspiciously small, catching truncated output that "looks done."

`locked claim-bar`

Re-injects the CTA snippet if the model drops it.

`idempotent webhook`

Stripe dedup via insert-on-conflict in a Postgres transaction; rolls back so Stripe retries cleanly if Temporal is unreachable.

THE HUMAN GATE · SITE APPROVAL

It sits before any outreach. The first impression a trade owner gets is the entire pitch: the highest-stakes, least-reversible moment in the pipeline.

Honest boundary: the builder's self-QA checklist is model-self-attested; only the anti-slop grep truly executes. Downstream Playwright and Gemini audits are informational second eyes. The human is the sole hard gate.

WHERE IT STANDS

Production-grade. Pre-revenue by design.

THE FUNNEL: EACH TIER DISTINCT, NEVER COLLAPSED

~45K

licensed CCB
contractors on file

7,857

plumbers,
serviceable slice

2,405

promoted into
the scored pool

72

deep-scored on
six dimensions

63

spec sites built

60

preview-uploaded

54

staged &
approval-ready

BUILD LATENCY

16.1 min median
build scope only · n=56

STATUS

Stages 1-3 built and shippable on demand. GROW wired and test-verified. EMBARK planned.

No per-prospect cost logging exists, so no marginal cost is quoted. Pre-revenue is a deliberate gate, not a limitation of the build.

Move slower. Trust the model's confidence less.

It will tell you something works, and it might even run, but running is not durable, and it is not optimized.

A thing that runs in your own bubble is not enterprise-grade. A compelling architectural flow is easy; a durable, measurable, scalable build is hard, and the large majority of implementations never reach real production.

-  Run a real discovery phase before writing any spec.
-  Build key features in their own sandboxes; test endlessly before integrating.
-  Use an orchestration layer from the very beginning.
-  Stop retrofitting shiny ideas mid-build; that is where spec drift starts.

PROOF

FORGE on Temporal. A second system on LangGraph.

Two multi-agent systems, both code-defensible: durable orchestration on Temporal.io, and Solutions Factory, an autonomous solution architect, on LangGraph.

PUBLIC REFERENCE REPOSITORY

github.com/upswing-solutions/forging-pipeline

The Temporal / Postgres / LLM orchestration core is real; external integrations sit behind stubbed adapters. The primary public proof artifact.

BUILT & OPERATED BY

SAGE

upswing

SALEM, OREGON

