CASE STUDY · OWN BUILD · SEPTEMBER 2026

Answer the right leads fast, without handing over who you are.

We built a pipeline that finds Upwork jobs worth bidding on, writes a tailored proposal and submits it, while our identity and credentials stay on one Mac.

IN	An Upwork job alert, the moment it lands in Gmail
OUT	A checked, tailored proposal, and a line in the next morning's review
STAGE	In use since 17 Sep 2026. Automatic from alerts since 23 Sep

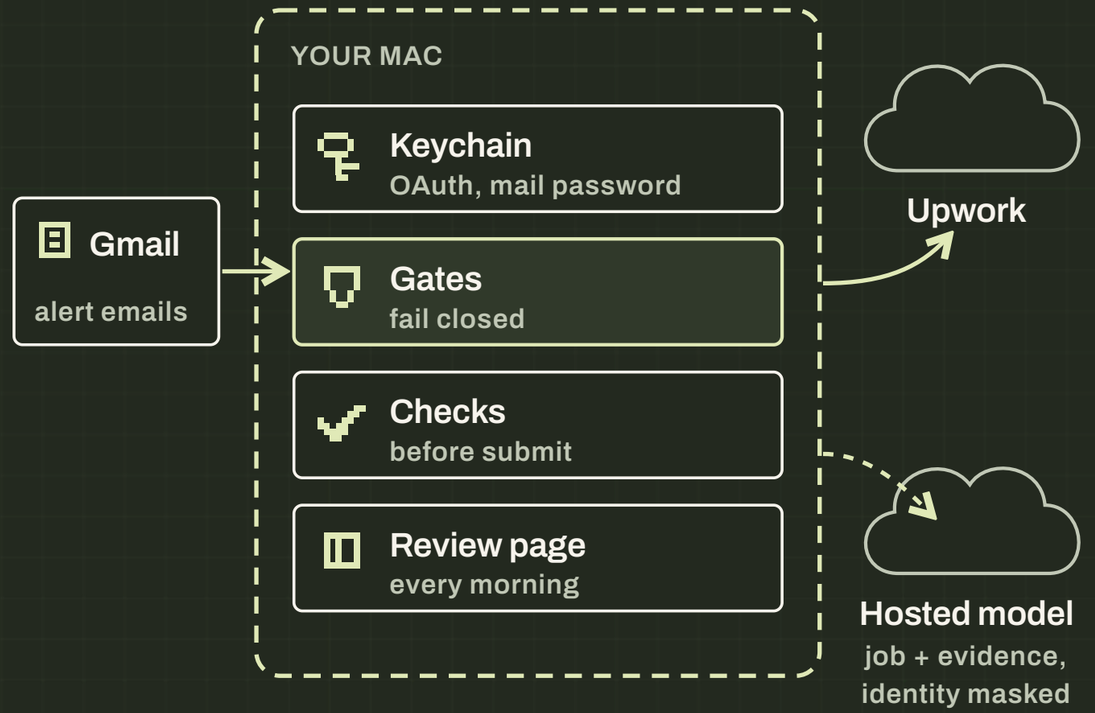


ILLUSTRATION · NO CLIENT OR ACCOUNT DATA

THE PROBLEM

Be fast, picky and private, all at once.

Doing it by hand is slow. A naive bot is fast, but it bids on anything, floods the platform and sends your identity to whoever writes the letter.



Fast

An alert is worth most in its first minutes. By hand, reading the post, checking the client and writing a letter takes far longer.



Picky

Most posts fail on budget, payment or a client who has never hired. Every proposal costs Connects, so a bad bid is a real cost.



Private

A hosted model that writes your letters should not see your name, your location or your keys. The platform should never see a burst of bot-like calls.

~670

jobs to screen in four days

From saved searches alone, 20–23 Sep

8–27

Connects quoted per proposal

The range across our 33 proposals

60 s

to notice a new alert

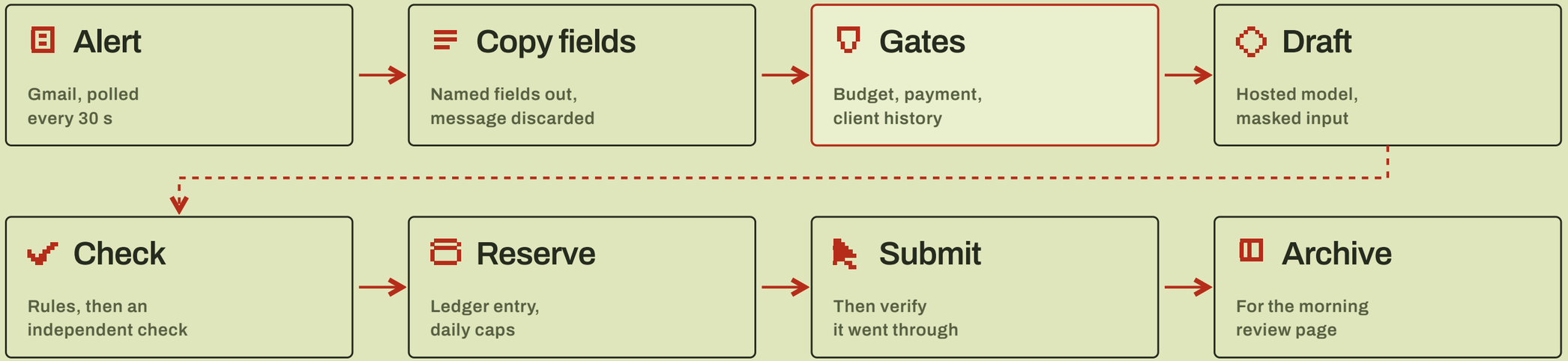
The detection limit we set ourselves

THE SYSTEM

Eight steps. Only one of them uses a model.

Everything around the draft is plain code with its own tests, so the one component with judgement is boxed in on both sides.

EVERY STEP IS CODE EXCEPT ONE: THE DRAFT



0

third-party packages

Python standard library, SQLite and macOS launchd

~7,400

lines of pipeline code

Plus about 4,900 lines of tests

458

tests passing

Up from 165 a week earlier

DECISION 01 · GATES IN CODE

Code decides which jobs to bid on. The model only writes.

Every rule the owner set is enforced before a model sees the job. The model gets no vote on whether to apply.

EACH GATE FAILS CLOSED: A MISSING OR UNKNOWN VALUE MEANS SKIP, WITH THE REASON LOGGED



20

alerts on 24 Sep

Every one logged with its outcome

6

passed every gate and were sent

Three AI, three marketing: the day’s cap

14

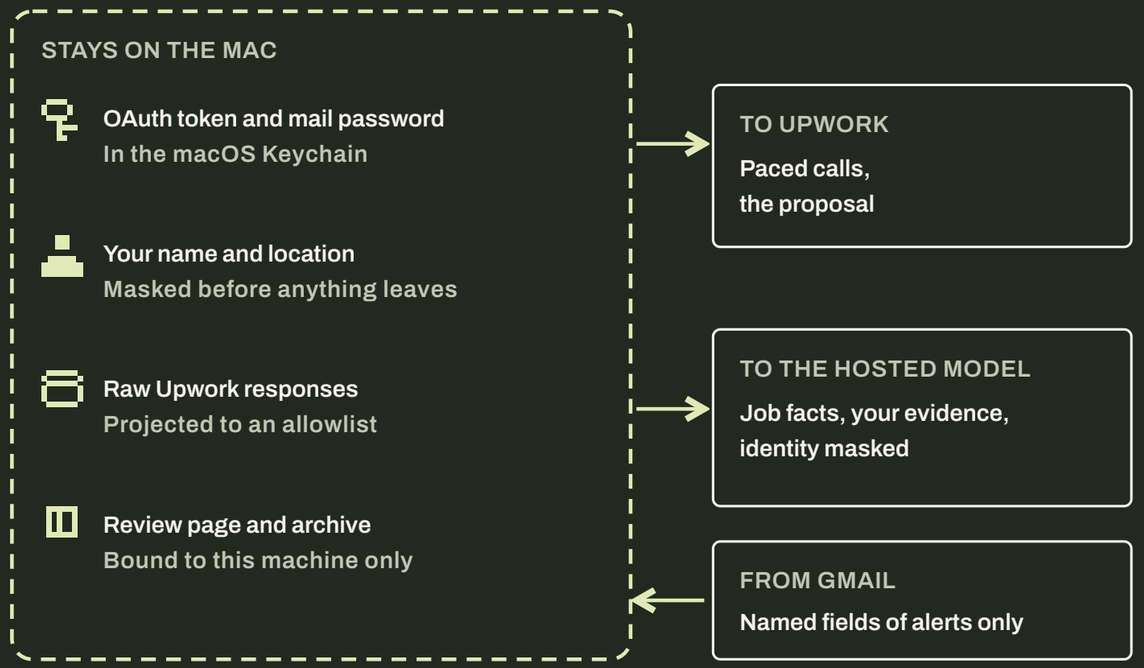
skipped, each with its reason

Review-only searches, budgets below the floor, closed posts

DECISION 02 · PRIVACY

Identity stays on the Mac. The model sees the job.

Upwork is reached through its official MCP server. Every response is masked and cut down to an allowlist of fields before anything else reads it. The model also gets our curated evidence library.



01 **Masked before it leaves**

Protected terms become [PRIVATE] and emails are withheld. The owner enters those terms in a local form, and they are never saved to a file.

02 **Allowlists, not blocklists**

A job row keeps budget, payment status and hiring history. The client’s country is the only location released. Unknown fields are dropped.

03 **Fails closed**

An alert email is copied out field by field, then discarded. If any field holds a protected term, the whole record is refused.

BUILT TO BEHAVE

The limits live in code, below the prompt.

A rule the model can reason past is not a rule. These hold whatever the model writes.



Paced like a person

A second between searches, two between job reads, 30 reads a session and one session every three hours.



Stop on any error

Any Upwork error writes a HALT file. Nothing runs again until a person clears it.



Checked before it is sent

At most 150 words. No links, emails, phone numbers or questions. A second pass checks every factual claim against our evidence.



A model in a sandbox

The drafting call has no shell, web, plugins or tools. It gets a single prompt in an ephemeral session and returns text.



Keys in the Keychain

The OAuth token and the mail password sit in the macOS Keychain, never in the repository or a log.



A ledger before each bid

Each submission is reserved in a ledger first, so a retry can never send the same proposal twice.

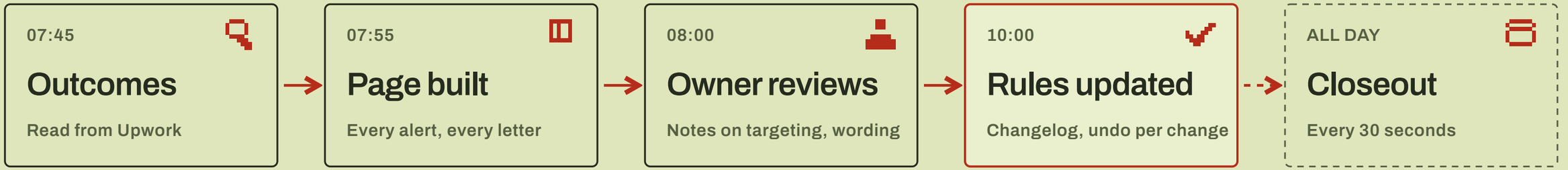
The residual risk, stated

Masking reduces what a hosted model can learn, but it is not OS isolation and no guarantee against inference. Our privacy policy says so in the same paragraph.

DECISION 03 · THE MORNING REVIEW

Every proposal is read the next morning.

No one sees a proposal before it is sent, so a local page lists every alert from the day before: why it was sent or skipped, the full letter and the screening answers.



01 **Unknown is never success**

An outcome the pipeline could not read is labelled unknown, never counted as a win.

02 **Feedback has a fence**

Morning notes can change targeting and wording rules only. Caps, bids and code are out of reach by design.

03 **Too few to read**

Trends split replies and hires by job type, budget and letter length. Any group under five proposals is greyed out.

04 **Process changes wait for a person**

Feedback that would change the process is queued for a working session, never applied unattended.

WHAT THE REVIEWS CHANGED

Each fault we saw became a rule in code.

A week of run records and reviews. Each row is a real observation and the rule it produced.

WHAT WE SAW	THE RULE IT BECAME
A proposal went to a client who had never hired anyone	**Client must have hired at least once**
Two alerts were replayed 39 and 85 minutes late	**Act only on alerts detected within 60 seconds**
We were hand-picking which qualified jobs to read	**Job reads per session raised from 8 to 30**
Connects cost looked like a quality signal	**Tested against the rate: r = −0.11. Not used**
The drafter declined three jobs the rules had passed	**Gates decide; the model only writes**

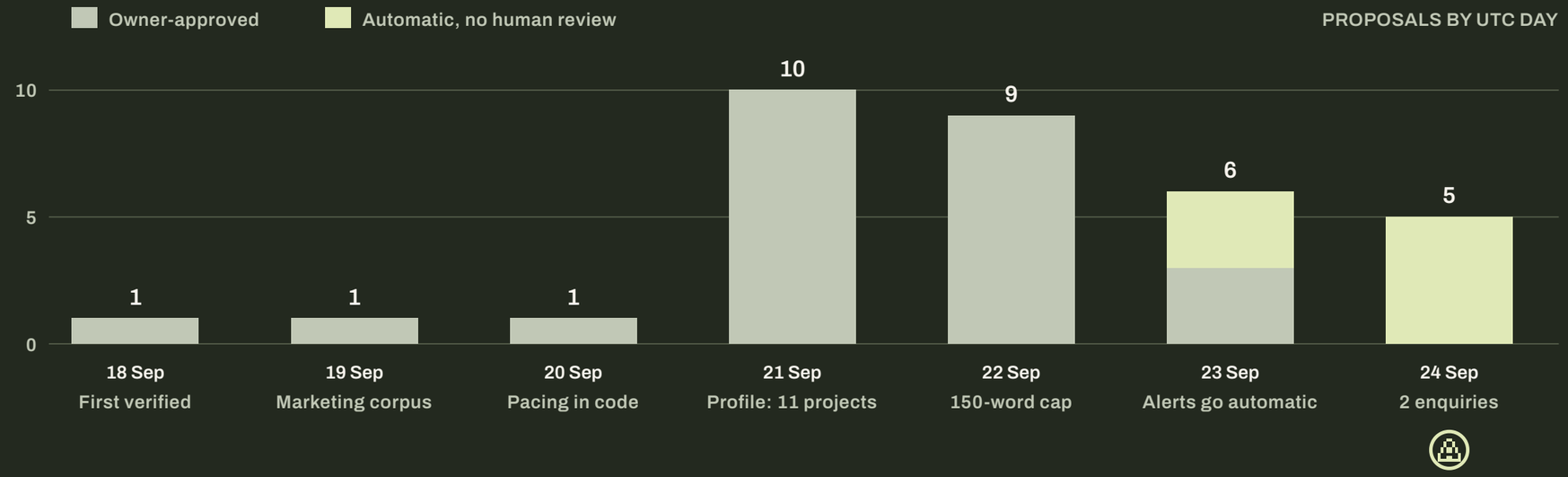
From incidents to a routine

Until 23 Sep these came from run records, when something went wrong. Since then the same loop runs every morning, over every alert.

ONE WEEK IN

The first enquiries arrived the morning after the changes.

On 23 Sep alerts went real-time and automatic, and the profile evidence was rewritten. On 24 Sep two clients got in touch.



33

proposals in seven days

18–24 Sep. 8 of them fully automatic

~670

jobs classified, each with a reason

20–23 Sep, from 16 sweeps of saved searches

36–204 s

from alert to submitted

The six automatic proposals of 24 Sep, UK time, from receipts

WHAT IS NOT PROVEN

Early results, and a small sample.

Two enquiries in a week is a start, not a rate. This is what we can and cannot yet show.

The enquiries are not yet in the tracker

The outcome tracker reads proposal statuses, not messages. The two enquiries are reported by the owner.

Upwork does not report views

Rates can only use replies and hires, and eight automatic proposals are too few to read.

Connects are quoted, not audited

Each receipt records the quoted cost. No one has checked the charge against the account.

The drafter is a hosted model

Identity is masked first, but the letter is written in the cloud. The local-model version is planned, not built.

The scheduled sweep is not switched on

Sweeps three times a day are written and tested, but not yet installed. Alerts carry the live path.

Feedback has not yet changed a rule

The review page went live on 23 Sep. The first owner feedback is still to come.

What the next reviews will measure

Reply and interview rates by job type, budget and letter length, once each group holds five proposals or more.

BEYOND UPWORK

The same pattern fits any lead source that sends an alert.

The gates, the privacy boundary and the morning review carry over unchanged. Only the alert source and the rules differ.

LEAD SOURCE	THE SIGNAL	GATES WORTH CODING	WHAT STAYS PRIVATE
Freelance marketplaces Built and running	Job alert emails	Budget, payment, client history, daily caps	Identity, credentials, account data
Public tenders	Portal notices and emails	Value, deadline, eligibility, past bids	Pricing, past submissions
Inbound enquiries	Web forms and inbox	Service fit, budget, location, conflicts	The client's own details
Partner referrals	CRM and shared inboxes	Account owner, stage, duplicates	Partner terms, contact data

Harness layers 02 Tools · 05 Orchestration & routing · 06 Guardrails & permissions · 09 Evals & verification · 10 Observability & cost

YOUR LEAD PIPELINE

Bring the leads you answer by hand.

We'll map where they arrive, code the rules you already apply in your head, and build a pipeline you review over coffee.

01
Map
Sources, rules, what stays private

02
Gate
Your criteria, enforced in code

03
Draft
Your evidence, checked before it goes

04
Review
A morning page you own

Book a 30-minute mapping call

hello@aiarchitect.md