

FREIGHT HERO · AI ENGINEER TAKE-HOME

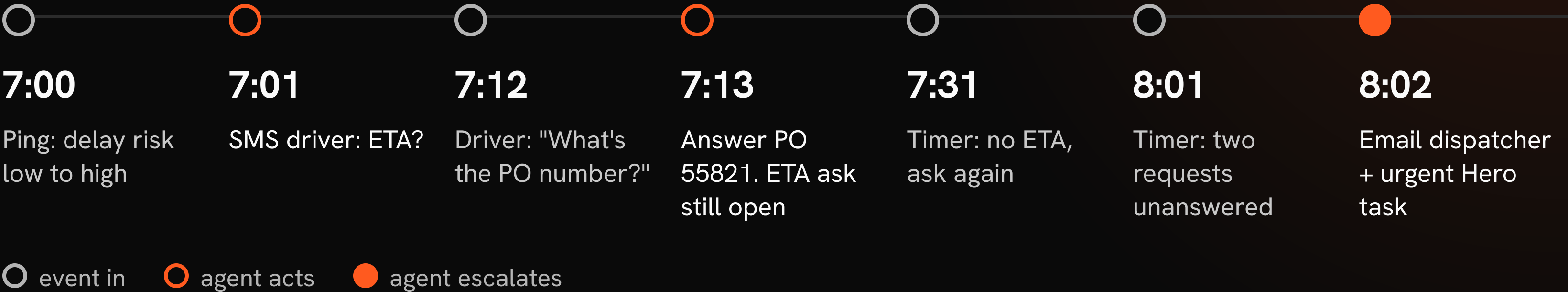
# A load agent that knows when **to do nothing**.

Follows each client's SOP, acts when safe, escalates when not, and does nothing when nothing is needed.

**Marcus Caum**

THE PROBLEM

# Load #481207 in one picture



**100,000**

loads a month

**50 to 100**

messages per load

**\$1**

LLM budget per load

**5 min**

event to last action

## PRINCIPLES

# The model is the exception. Doing nothing is a result.

### LLM as the exception

Code owns filters, state, timers, attempt counts and escalation chains. A model reads language and handles what no SOP covers.

**1 model call in the whole 481207 timeline**

### State from the log

Every input and every decision is appended to a per-load log. State, including open asks, is a projection of it. Replay is exact.

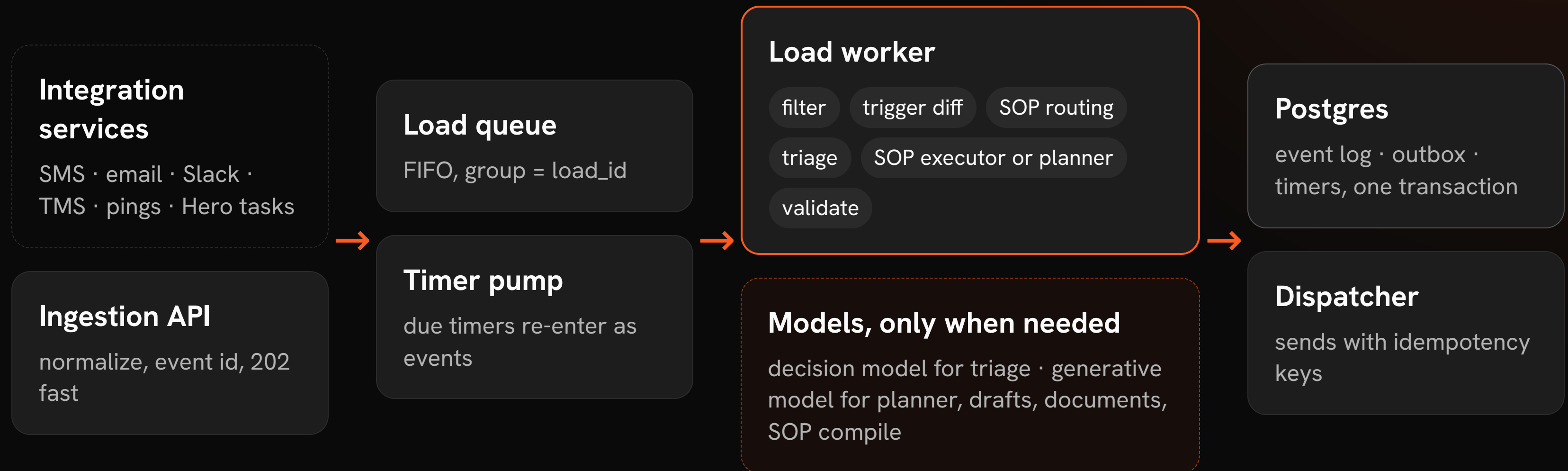
### Noop is valid and tested

"ok" with no open question is dropped by the filter, with no model call. A timer whose ask is closed does nothing.

**15 of 30 scenarios assert a do-nothing step**

## ARCHITECTURE

# Hybrid, event-driven, serial per load



MEMORY

# At 7:13, "is the ETA request open?" is a lookup

Event log, load 481207

- 07:00 ping, risk high
- 07:01 decision: ask opened, SMS 1, timer 07:31
- 07:12 SMS "What's the PO number?"
- 07:13 decision: SMS "PO 55821", no ask change

Open ask, projected from the log

- id ask:eta\_request:ping\_0700
- target driver, SMS
- attempts 1 of 2
- next timer 7:31 AM
- SOP client\_a/delay\_risk\_high@v1
- resolved\_by none

No model re-reads chat history. The PO answer takes no step on the ask, so the 7:31 follow-up stays on schedule.

## TIMERS AND THE ESCALATION CHAIN

# A timer carries three fields. Everything else is state.

**7:01**

SMS 1

timer 7:31



**7:31**

ask open, attempt 1 = step 1

SMS 2, timer 8:01



**8:01**

max attempts reached

email dispatcher, urgent Hero task

### Payload

`{load_id, ask_id, attempt}`. On fire, the agent projects state and checks the ask is open at that attempt.

### Stale or duplicate

Ask closed, or already past that attempt: noop, no model call. A second delivery has the same event id and is dropped.

### Durable

Timers are rows next to the log, set through the outbox, fired by a pump within a minute. Evals drive the same code with a fake clock.

## SOPS

# Layered per topic, compiled at edit time, previewed in plain English

### client + load type

sops/clients/client\_b/reefer/departed\_pickup.md

### client

sops/clients/client\_a/delay\_risk\_high.md

### standard

sops/standard/delay\_risk\_high.md

Most specific file wins the whole topic. Open asks stay pinned to the version they started under.

### Client A vs standard, from **sop preview**

Attempts at the request: 2 (was 1).

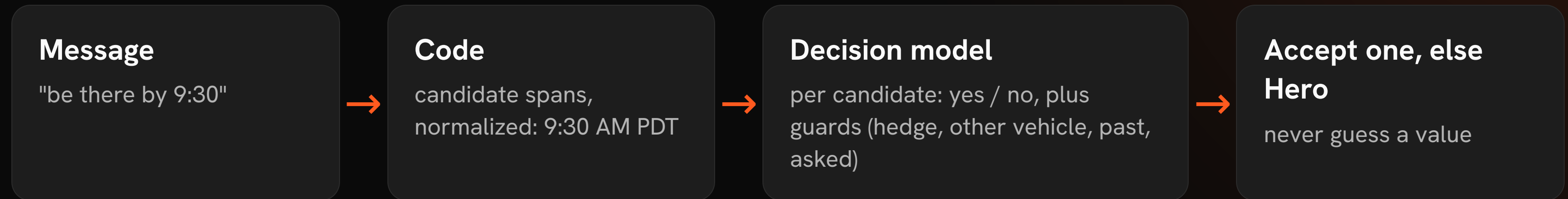
Escalation: now also does: email the dispatcher in the main thread (say the driver does not answer and ask for the ETA).

Escalation: the Hero task is now urgent.

Rendered by code from the compiled spec, not by a model.

## TRIAGE

# Code finds values. The decision model confirms.



---

**0**

wrong values on 116 adversarial negatives

---

**82 to 84%**

of 74 positives accepted across two recordings (62/74, 61/74); held-out 19/23 and 17/23. The rest go to a Hero

---

**~0.2 s**

per triage call, measured

Calibration on 190 phrasings in two recordings of one model version. ADR 0005. A labelled production set is required before launch.

## SAFETY

# Every action is data, validated before it is committed

### Validation

Only contacts of the load, only channels they support, never a role the client forbids, never a used key. A rejected action becomes a Hero task.

### Idempotency

`481207/ask:eta_request:ping_0700/2/send_sms`. Retries and re-decides produce the same key; the service dedupes.

### Outbox

Event, decision and actions in one transaction at an expected version. A crash after commit still sends; a concurrent run re-decides.

Model timeout, invalid output, an unsafe draft or an unknown situation: urgent Hero task carrying the event. The run never crashes.

## EVALS

# Every step asserts the exact actions, or an explicit nothing

---

# 30

scenarios replayed through the real runtime with a fake clock

### Scenario files

Timeline plus expectations per step: kind, target, channel, ask, attempt, key, or noop; and the open asks after it.

---

# 15

scenarios with a do-nothing step, asserted with zero model calls

### Replay gate

Recorded decision-model answers for 190 phrasings replayed offline: no phrasing may yield a wrong value.

---

# 1,455

tests: 1,354 run offline with no API key; 101 Postgres contract tests run in CI

### Live tests

Real providers, daily and on demand, outside the per-push CI so an outage never blocks a merge.

## COST AND LATENCY, MEASURED

# Within budget. Here is what the measurement taught us.

## \$0.110

LLM per load against \$1; the decision model at an assumed \$0.002 per call

## \$0.0218

per decision-model call is the break-even: above it a load passes \$1

## 21 s

slowest run against 5 min: one planner call on the strong model

### Triage agreed on 30 of 45

Every disagreement but one fails toward a Hero task or a TMS note. The exception: one unhelpful reply to a driver.

### Chatter is off-distribution

Off-topic driver messages became TMS notes or planner calls. Proposed lever: send topic-less questions straight to a Hero.

### Borderline goes to a Hero

A dispatcher's ETA by email sat near two thresholds and escalated. No wrong value was accepted.

51.7% of 87 synthetic inputs reached a model. Real providers, one seed, 2026-10-08. Try other prices and volumes: [the interactive benchmark](#).

# Exposure follows the cost of a mistake

## Stages, each with a go / no-go gate

1. Offline replay of the client's history
2. Shadow: the agent decides, Heroes act
3. Canary: internal actions (Hero task, TMS note)
4. Canary: driver SMS
5. Canary: dispatcher email, broker Slack
6. Full rollout; kill switch per client and action type

## Metrics, all queries over the log

- **Silent-miss rate**: highest priority, from audits
- **Wrong-action rate**: actions a Hero reverses
- **Hero minutes per load**: the value claim
- **Escalation rate by reason**, per client and topic
- **Time to detect a delay, cost per load, p95 latency**

Gate thresholds are proposed targets to agree with Freight Hero. The kill switch and Hero feedback tooling are designed, not built.

## SCOPE

# What is built, what is designed

### Built and tested

- Event log, projection, open asks
- SQLite and Postgres stores, outbox, dispatcher, durable timers
- Ingestion API, Postgres queue, worker, docker compose, proven in CI
- Layered SOPs, compiler, routing, preview with behavior diff
- Pipeline, validation, templates, run deadline
- Real model clients behind one protocol; offline fake for evals
- 30 scenario evals, CI, daily live tests, a cost and latency benchmark

### Designed

- SQS adapter, provider signature verification, autoscaling
- Real SMS, email, Slack, TMS and Hero integrations
- SOP editor UI, shadow mode, publish service
- Metrics aggregation, dashboards, alerts
- Per-type document schemas and accuracy on real BOLs and PODs

## NEXT STEPS

# Before launch

### 1 Production replay set

Label real driver and dispatcher replies; rerun the triage gate; pin the model version.

### 3 Hero feedback tooling

"Agent was wrong" marks, structured reasons and audits: the inputs to the quality metrics and the rollout gates.

### 2 Fewer avoidable Hero tasks

FYI news and chatter from the benchmark: tune the gray band, route topic-less questions to a Hero without the planner.

### 4 Integrations

Real channel services behind the typed clients, provider signature verification, an SQS adapter and autoscaling.

[freighthero.kodama.solutions](https://freighthero.kodama.solutions) · [github.com/marcuscaum/freighthero-load-agent](https://github.com/marcuscaum/freighthero-load-agent)