AI AGENTS & AUTOMATION

AI that does real work, inside rules the business sets.

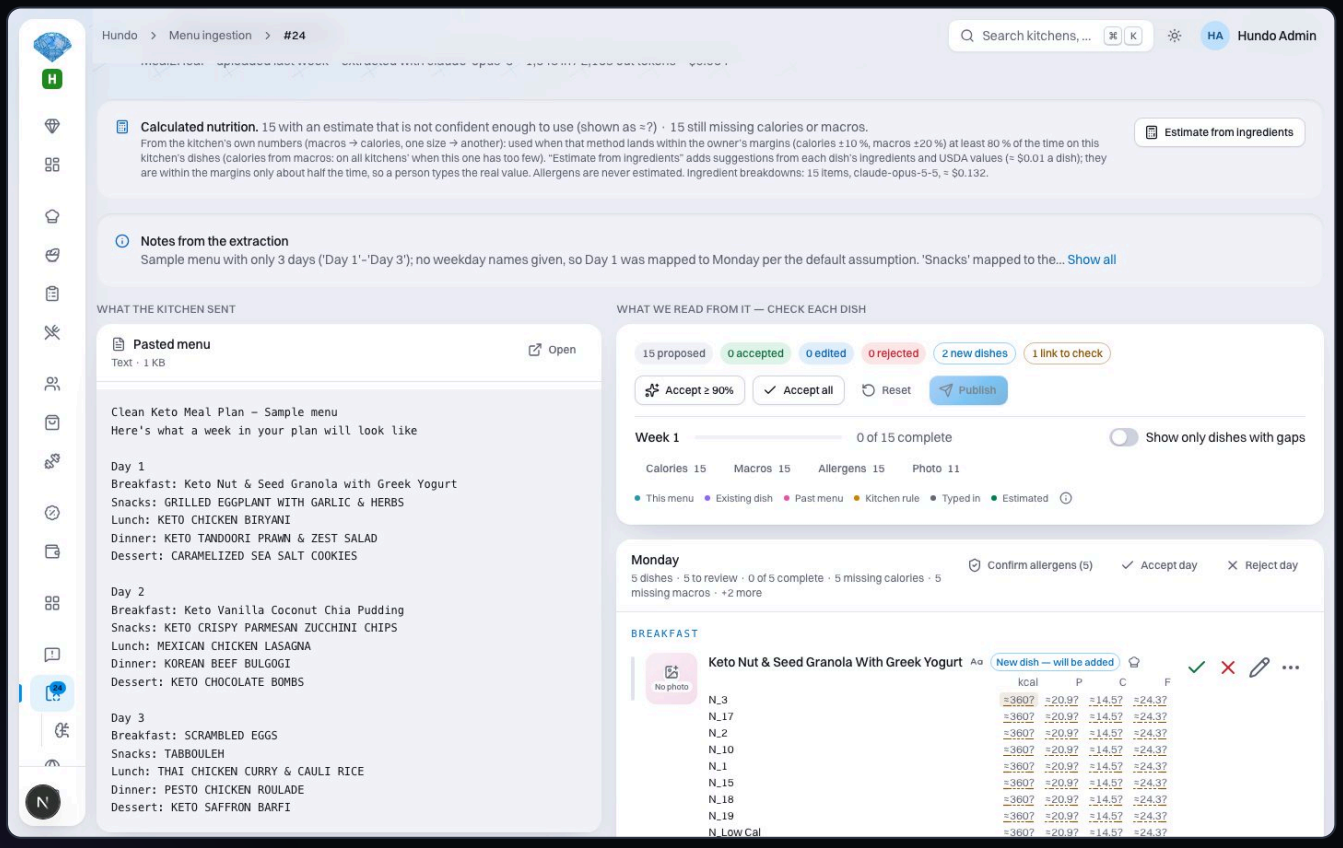
I build AI systems that run every day: extraction from messy documents, assistants that triage email and draft in someone's voice, agents that research overnight, and support bots, with structured output, confidence thresholds, human review and hard limits enforced in code.

Claude API
in production: structured output from zod schemas, streaming, cached prompts

Local LLMs
32B and 27B models plus Whisper on my own GPU

Never guessed
hard rules outside the model, e.g. allergens

357 of 368
HUNDO commits written with Claude Code



PRODUCTION AI · HUNDO

Menus in any format, turned into data an app can sell from

Partner kitchens send menus as PDFs, photos, spreadsheets, web pages or pasted text. The pipeline turns each one into the same structure the menu builder uses (week, day, meal and dish, with calories, macros and allergens per size), then a person or a strict gate decides what goes live.

Prepare
PDF → document block; image → image block; XLSX/CSV → text tables; the company template parses with no model call

Extract
Claude, streaming, adaptive thinking, structured output validated by a zod schema; cached system prompt; extraction runs after the HTTP response

Match and score
exact or fuzzy dish matching (trigram Dice + Jaro-Winkler); calibrated confidence

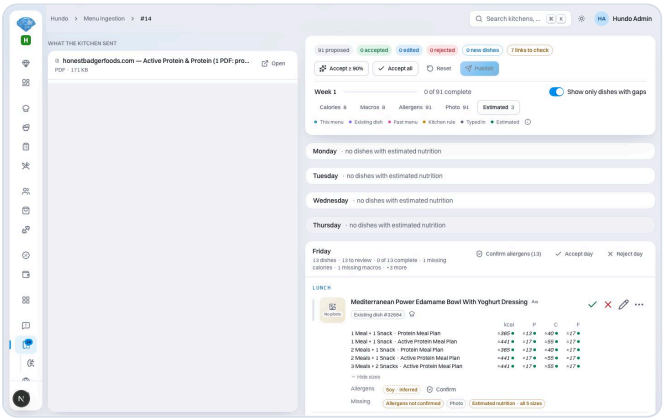
Gate
review screen, or auto-publish only past 6 rules (≥ 0.9, everything placed, no clashes, allergens and nutrition complete)

Making it reliable

- Every reviewer correction is stored per kitchen; the newest 12 are put in the prompt, so it improves with each review.
- Cost estimated before each run (per PDF page, image and dish) and actual usage shown after.
- Validation reports every problem with its JSON path; text over 400,000 characters is refused, never silently truncated.
- Row locks in a fixed order plus fingerprints stop two writers overwriting each other.

Rules the model can't break

- Allergens are never estimated: inferred allergens need a person's confirmation.
- Nutrition calculated from ingredients (USDA SR Legacy) goes live only marked "estimated", and only where the method proves accurate on that kitchen's own dishes.
- A separate vision check reviews dish photos; AI-generated photos are never used.
- The collector for kitchen websites respects robots.txt, renders JavaScript pages headlessly, skips unchanged pages by content hash and stops at a monthly budget.



The review screen (VVS Panel): source on one side, the proposed menu on the other, with accept, edit or reject per item, day or week, "Accept ≥ 90%", and the completeness of every dish.

ASSISTANTS AND AGENTS · FRONTAL

An AI operations platform I run my businesses on

Frontal is my private assistant: eight services on my own GPU server, reached only over a private network, with a phone app and push alerts. The full case study is a separate PDF; here is what is relevant to AI work.

Capability	How it works
Email triage	IMAP IDLE across mailboxes; a local 32B model scores importance, category and needs-reply; learned rules run first; VIP senders never drop below 0.75
Drafts in my voice	Built from the thread, my own writing samples and a ledger of my commitments; returns blockers and a confidence score; every send is held 30 seconds so it can be undone
Standing orders	Pre-agreed replies with an autonomy level each (ask, earn, auto, never); auto-send only with no blockers and confidence ≥ 0.6, everything else waits for me
Commitments	Overnight, it re-reads unanswered threads and extracts promises, keeping one only if its quote really appears in the email
Competitor research	Every night a fixed-template order runs Claude Code for 20–40 minutes of web research, which posts one JSON report per business
Briefings and voice	A spoken morning brief (LLM summary, Kokoro text-to-speech) and voice notes through Whisper, whose actions must quote the recording
Monitoring	Servers, backups, Stripe disputes and renewals, SEO health and a reseller panel's build service, with phone alerts that carry action buttons
Engineering from the phone	Claude coding sessions ordered from the phone run in plan mode first; one tap approves and starts the work

SUPPORT AND AUTOMATION BOTS

AI customer support (EzSuppBot)

Python, FastAPI and python-telegram-bot running one bot per customer. AI replies use the last 10 turns in three modes: **auto** (replies to the customer), **suggest** (posts a draft to the staff group with Send/Discard) and **on demand**. It flags escalation keywords, can learn from resolved tickets, translates, and runs a self-check of the AI connection at every start. Ticket text is encrypted at rest.

Telegram automation (EzForwardBot)

FastAPI, python-telegram-bot and Telethon workers with one client per customer account. AI touch-ups per connection (summarise or rewrite, translate, a yes/no relevance filter) and digests in five styles, through any OpenAI-compatible endpoint. If the AI fails, posts pass through unchanged. Telegram's signed Mini App data is verified with HMAC; account sessions are encrypted at rest.

HOW I BUILD AI SYSTEMS

The app owns the process; the model owns the language

01

Process in code

Stages, required data and exit rules live in the application. The model sees only the current step's instructions.

02

Structured, validated output

Schemas (zod, JSON schema) for everything the model returns; validation with precise error paths; retries only when safe.

03

Confidence and gates

Calibrated scores, thresholds and multi-rule gates decide what is automatic; everything else goes to a person.

04

Hard rules outside the model

Sending, booking, deleting, paying and safety-critical fields (like allergens) are enforced by code, not by prompt.

05

Learning loops

Human corrections are stored and fed back, so quality improves with use rather than prompt-tinkering.

06

Cost and failure control

Estimates before runs, budgets, content hashing to skip unchanged work, and graceful pass-through when the model is down.

AI-assisted engineering, with accountability

- I build with Claude Code every day: 357 of HUNDO's 368 commits since 20 September.
- Agents get tight tasks; changes are tested, and bigger ones go through independent review passes (one confirmed 36 issues; another raised 84 findings), with confirmed findings fixed.
- A hook I built blocks any deploy of code that isn't committed and pushed.
- A fact store with receipts feeds every session the verified state of each project, so agents start from facts, not guesses.

Bugs AI wrote that I caught

- Reporting queries ran cleanly but counted the wrong rows: the ORM printed bare column names, so a correlated subquery bound to the wrong table. A shared helper fixed four wrong counts at once.
- An order endpoint jammed under 13 simultaneous changes. Advisory locks, a lock timeout returning 409 and rate caps fixed it.

Claude API · Claude Code · Ollama · qwen2.5-coder 32B · gemma3 27B · Whisper · Kokoro TTS · Groq · OpenAI-compatible APIs · zod · Python · TypeScript

Weeks of work, done in hours

I build with AI coding agents under a strict review-and-test process, so things that take most developers weeks take me hours. At HUNDO: first commit on 24 September, a working admin panel demo by 5 October, the go-live plan by 13 October, and the new iOS app in TestFlight by 19 October.