

CASE STUDY · 2026

# MitchAI

## Vertical Operational AI for Hospitality

*A multi-agent operational suite — compliance, stock, marketing, and ops copilots — that runs on hybrid local + cloud LLM infrastructure to keep per-outlet economics viable for the businesses that need it most.*

---

**Author:** Vasile Bogdan Godja

**Role:** Founder, sole engineer, first customer

**Stack:** Python · FastAPI · n8n · Docker · PostgreSQL · Redis

**LLM layer:** Hybrid local (Ollama) + cloud (Claude / OpenAI / Gemini)

**Status:** In active development · grounded in 10+ years of hospitality ops

**Contact:** [godjabogdan@gmail.com](mailto:godjabogdan@gmail.com) · [github.com/Bogdan0708](https://github.com/Bogdan0708)

---

# 01 The Problem

*The owner-operator of a small or mid-sized hospitality business does three jobs at once: customer-facing service, financial and compliance plumbing, and supplier and marketing operations. The last two consume nights and weekends, run on spreadsheets and WhatsApp, and don't scale.*

## A week in the life of a hospitality operator

- **HACCP & compliance records.** Daily temperature logs, allergen sheets, supplier verifications, and accident reports — most still completed on paper or in screenshots.
- **Stock and purchasing.** A weekly count, a fortnightly supplier reconciliation, and a constant low-grade anxiety about what's about to run out and what's been over-ordered.
- **Marketing & social.** Daily Instagram posts, weekly newsletter, Google reviews, ad campaigns. Each takes 30 minutes and no two are quite the same.
- **Menu changes.** Seasonal swaps, supplier substitutions, dietary tagging, price recalculations, and the re-print/re-upload of menus across POS, website, and printed cards.
- **Legal & admin.** Licensing renewals, PLH paperwork, fire-safety records, employment compliance. Each is rare enough to be confusing and frequent enough to be expensive.

Multiply by 200+ working days, divide by an owner who is already running a service floor, and the answer is: the work doesn't get done well. It gets done minimum-viable.

## Why existing tools don't fix this

### Enterprise hospitality platforms

Toast, Lightspeed, Otter — strong for chains, priced for chains. The independent operator subsidises features they don't need and still does the work the platform doesn't cover.

### Generic SaaS bundles

Notion + Canva + Zapier + Buffer can do most of this. The cost is integration work the operator does not have time for.

### Compliance-only vendors

HACCP-software vendors digitise the binder and stop. They don't touch the operational workflow that produces the data.

### Generic LLM chat

Useful for drafting one social post. Not useful when the operator needs the post drafted, scheduled, image-generated, brand-consistent, and tied to the menu — every day.

## Why I'm building this

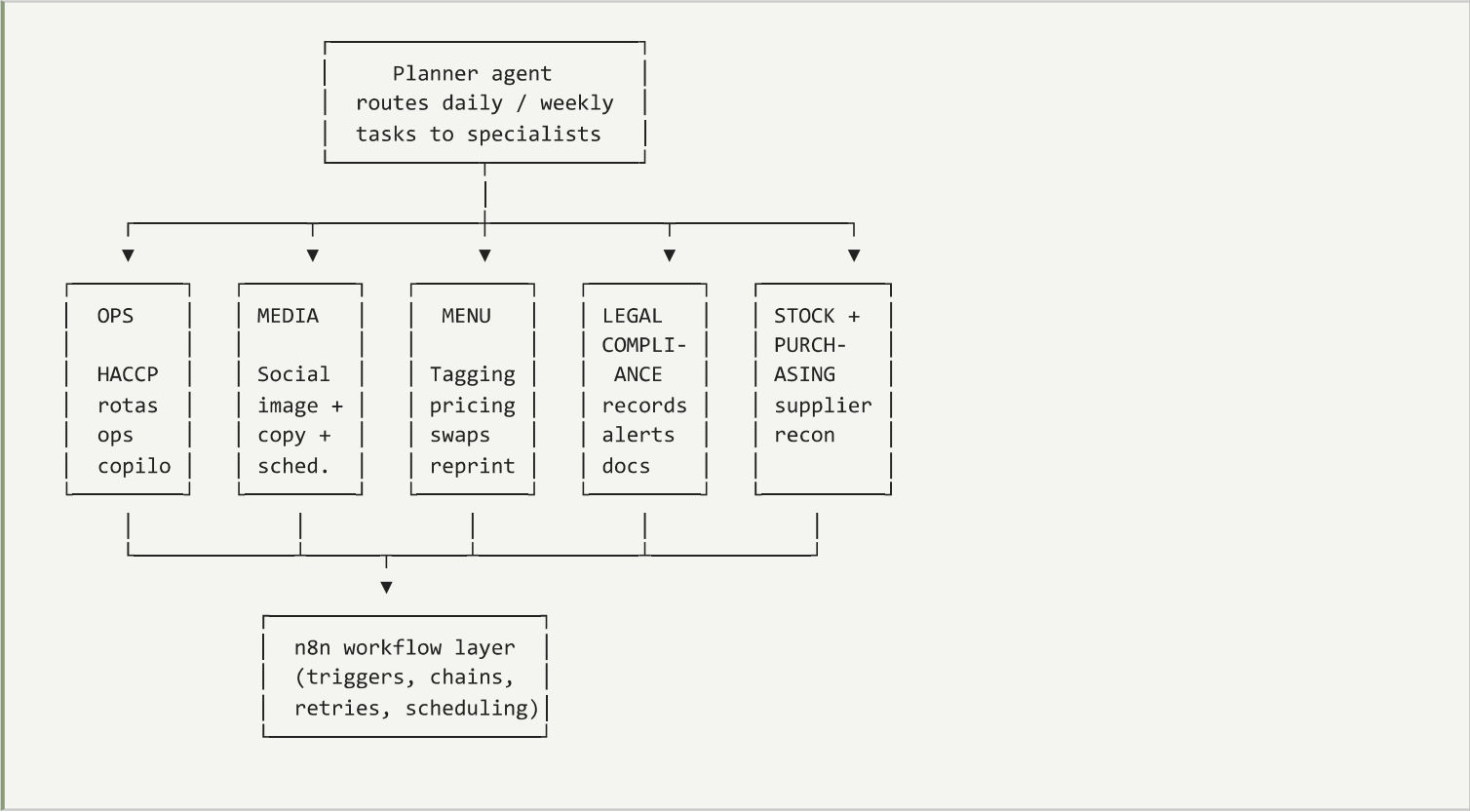
I've spent 10+ years on the operator side of this problem: General Manager at The Crown (Mosaic Pub & Dining), Assistant GM at Head of the River (Fuller's, Oxford), Hospitality Manager at London Business School, and Restaurant Manager at 108 Brasserie (Marylebone Hotel). I've also founded *Mitch from Transylvania*, a London food-market business — meaning I'm not building AI for a customer I read about. I'm building it for myself and for every operator I have ever worked alongside.

*MitchAI's design is constrained by one rule: every feature must save a real operator more than five minutes a day, or it doesn't ship.*

## 02 The Solution

*MitchAI is a multi-agent operational suite — Ops, Media, Menu, Legal — coordinated by a planner agent and wired together with n8n workflows. The system runs on hybrid local + cloud LLM infrastructure so per-outlet cost stays in the same order of magnitude as the operator’s coffee budget.*

### The agent suite



### The specialist agents

**Ops agent**

Daily HACCP and compliance records (temperature logs, allergen checks, opening/closing checklists), rota scaffolding from the planner agent, ops copilot for the shift manager.

**Media agent**

Branded social posts (image + copy + scheduling), Google review replies, newsletter drafts. Voice and visual identity are constrained by per-outlet brand profiles, not free-roam.

**Menu agent**

Seasonal swaps, allergen and dietary tagging, supplier substitutions, price recalculations, reprint/repush across POS / website / printed menus.

**Legal & compliance agent**

Renewal alerts (PLH, licensing, fire-safety), document drafts for routine filings, classification of incoming legal correspondence, audit-log of every compliance action.

**Stock + purchasing agent**

Weekly stock variance, fortnightly supplier reconciliation, low-stock alerts tied to historical consumption, reorder drafts approved by the operator before sending.

**Planner agent**

The orchestrator. Holds the operator's daily/weekly schedule, routes recurring tasks to the right specialist, surfaces what needs human attention, suppresses what doesn't.

### Why n8n at the workflow layer

n8n is the operator-visible substrate for triggers, schedules, retries, and integrations with POS, booking, supplier, and social APIs. The agents are the AI surface; n8n is the rest of the plumbing. This split matters because most of the work in a hospitality

automation suite isn't AI — it's scheduling, retry, and integration with messy APIs. Using a mature workflow engine for that part of the stack is a five-figure decision in delivered time per year.

## Hybrid local + cloud LLMs

Most agent operations are high-volume and low-stakes (classification, drafting, summarisation). These run on local LLMs via Ollama, on a single mid-range workstation per multi-outlet operator or a cheap cloud VM. The long-tail of difficult tasks — image generation, complex drafting, nuanced classification — routes to cloud providers. The unit economics of the whole product depend on this split.

## Key product decisions

- **Operator-first defaults.** Nothing posts, sends, or files without explicit operator approval. The system drafts, surfaces, and queues; the human commits.
- **Branded constraints by default.** Per-outlet brand profile (voice, visual style, do-not-mention list, signature dishes) is a first-class object that constrains every Media output.
- **Operator-visible workflow.** n8n flows are inspectable. When something goes wrong, the operator can see exactly which step failed and why.
- **Daily ops digest.** Every morning the operator receives one message: today's tasks, overnight alerts, drafts awaiting approval. One inbox, not eight.
- **Cost transparency.** Per-outlet cost is reported back to the operator. AI that costs more than it saves is AI that gets uninstalled.

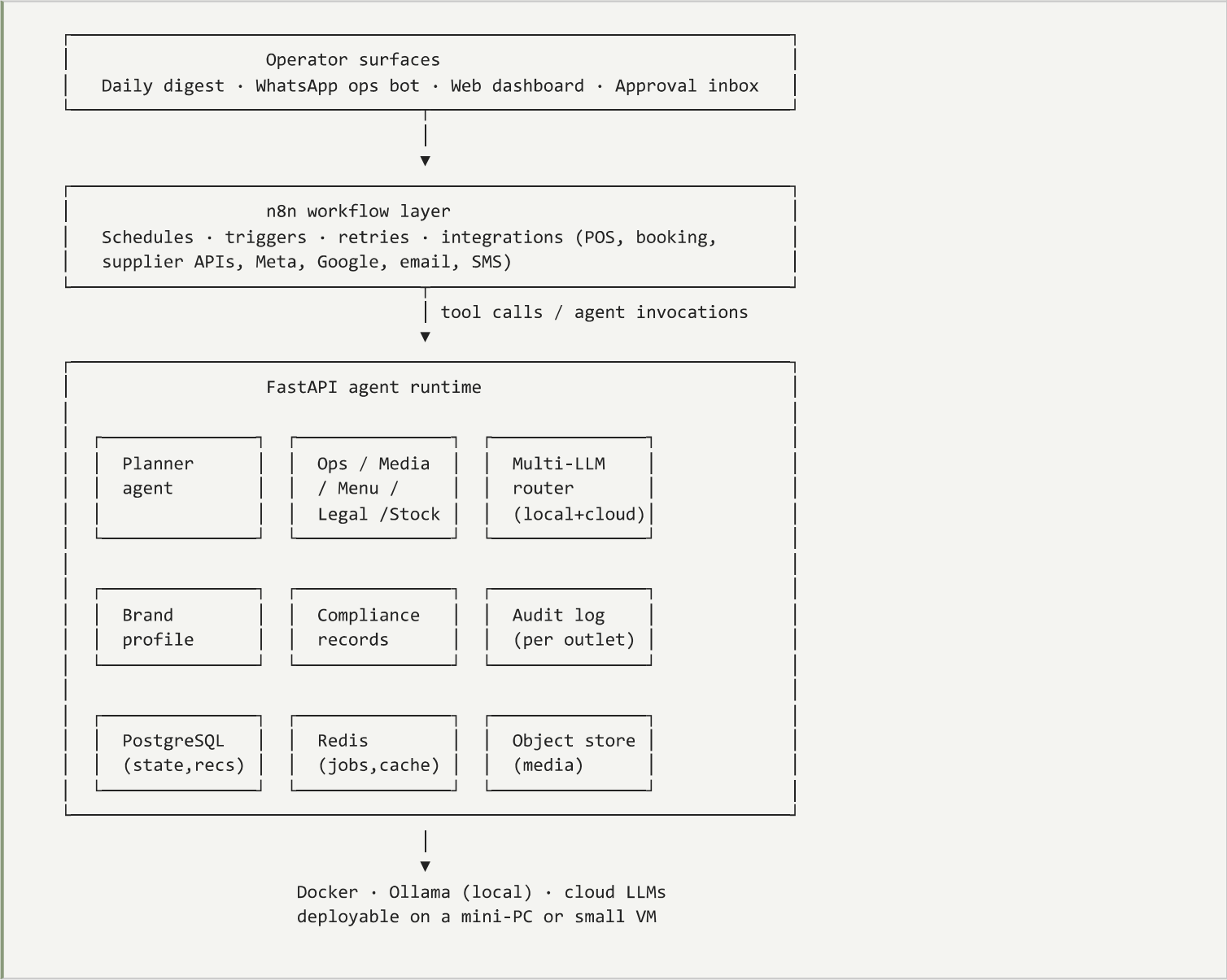
## What we deliberately did *not* build

- No reservation-system replacement. MitchAI integrates with the operator's existing PMS / booking system, never replaces it.
- No autonomous purchasing. Stock and purchasing agents draft orders; humans approve.
- No general-purpose chat. Every surface is task-specific. Generic chat in hospitality produces volume, not value.

# 03 Architecture

MitchAI is a FastAPI agent layer behind an n8n workflow plane, with hybrid local + cloud LLM routing. Deployment is Docker-first because the target operator either runs on a small cloud VM or a dedicated mini-PC behind the bar.

## High-level system diagram



## Stack

Python 3.12

FastAPI

n8n

PostgreSQL

Redis

Docker

Ollama (local LLMs)

Claude (Anthropic)

OpenAI

Gemini

image-gen models

Pydantic

SQLAlchemy

WhatsApp Business API

## Hybrid LLM routing by task profile

| Task                          | Default provider   | Why                                         | Fallback              |
|-------------------------------|--------------------|---------------------------------------------|-----------------------|
| HACCP / record classification | Local LLM (Ollama) | High volume, low stakes; cheap and private  | Gemini Flash          |
| Social copy drafting          | Claude Sonnet      | Brand-voice consistency over short prose    | OpenAI mini           |
| Image generation              | Cloud image model  | No viable local equivalent at quality bar   | Cross-provider        |
| Supplier reconciliation       | Local LLM + rules  | Mostly deterministic; LLM only on ambiguity | OpenAI structured-out |
| Legal correspondence triage   | Claude Sonnet      | Quality and citation on regulatory text     | OpenAI o-series       |

| Task                             | Default provider | Why                               | Fallback     |
|----------------------------------|------------------|-----------------------------------|--------------|
| Ops copilot (shift manager chat) | Local LLM        | Latency + cost; private to outlet | Gemini Flash |

## Brand profile as a first-class object

Every Media agent invocation is conditioned on a typed brand profile: tone descriptors, visual style references, signature dishes and ingredients, banned terms, recurring seasonal motifs, and a few example posts. This is structured data, not a system prompt paragraph. Outputs that drift from the profile are caught at validation; the operator sees a draft that already looks like their place, not a generic "restaurant Instagram" template.

## Compliance records pipeline

1. **Capture.** WhatsApp photo, web form, or scheduled prompt — the floor staff submit through whichever surface is fastest.
2. **Classification.** Local LLM tags the record (temperature log, allergen check, accident, supplier verification, etc.).
3. **Validation.** Structured-output schema enforces required fields. Missing fields trigger a follow-up prompt rather than a silent gap.
4. **Storage.** Append-only record store with operator + outlet provenance. Hash-chained for audit.
5. **Reporting.** Weekly compliance digest auto-generated; gaps surfaced; ready for an EHO visit without a paper-binder scramble.

# 04 Technical Challenges

*The interesting engineering in MitchAI is not the agents — it's the unit economics, the integrations, and the discipline of building software for users who don't have five minutes to spare.*

## 1. Per-outlet unit economics

A hospitality operator running on 12–18% margins will not pay £200/month for software that automates jobs they're currently doing for free at 11pm. The whole architecture is shaped by this: local LLMs for high-volume tasks, cloud only where quality dominates, aggressive caching of brand-profile and menu state, and per-outlet cost visibility surfaced to the operator. If MitchAI costs more than it saves, it correctly gets uninstalled.

## 2. Integrations with messy hospitality APIs

POS systems, booking platforms, supplier portals, and social APIs each have their own auth, rate limits, schemas, and quirks. The discipline that worked: every external integration is wrapped in an adapter with a normalised internal contract, observable retries, and a circuit breaker. When a supplier portal changes its HTML overnight, one adapter breaks — not the whole stock workflow.

## 3. Brand drift in generative media

Image generation without strong constraints produces generic, AI-looking restaurant posts that operators correctly hate. The fix is the brand-profile object plus a post-generation validation pass: a smaller model checks the output against the brand profile's banned terms, voice descriptors, and visual references. Drafts that fail are re-generated with the validator feedback included.

## 4. Local LLM ops on operator hardware

Running Ollama on a mini-PC behind the bar is operationally different from running it on a fleet of cloud GPUs. The discipline was: a single supported hardware profile (e.g. modern consumer CPU + 32GB RAM + a small dedicated GPU), pinned model versions, automated health checks, and a graceful fallback to cloud when local fails. When the chef accidentally unplugs the box on a Friday night, the operator never knows.

## 5. Approval inbox UX

Hospitality operators do not have time to log into a dashboard. The approval surface is WhatsApp by default: a single message per draft, two-button approve/edit, optional voice note for changes. The Media draft turns into an Instagram post with two thumb taps. Every UX decision is graded by "could the operator do this between two service rounds?"

## 6. Audit trail for compliance work

Hash-chained append-only records (same discipline as PrimarIA) for every compliance action. Environmental Health Officers will eventually ask the question and the operator needs a one-tap export, not a stack of paper.

## 7. Multi-outlet scale

A multi-site operator (3–10 outlets) is the sweet spot for value. The architecture is per-outlet by default with shared brand profiles, shared supplier lists, and operator-level dashboards rolling up per-outlet activity. Adding a new outlet is a config + integration step, not a deployment.

| Failure mode                            | Mitigation                                                                                              |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------|
| Per-outlet cost exceeds value           | Local LLMs for high-volume tasks; cloud only where quality demands; cost dashboard surfaced to operator |
| Supplier API / portal change            | Adapter pattern with normalised internal contract; circuit breaker isolates failures                    |
| Brand drift in generative media         | Brand-profile object + post-gen validator with retry on failure                                         |
| Local LLM downtime on operator hardware | Pinned models, health checks, transparent fallback to cloud provider                                    |

| Failure mode               | Mitigation                                                                            |
|----------------------------|---------------------------------------------------------------------------------------|
| Operator approval friction | WhatsApp-first approval surface; thumb-tap approve/edit; voice-note edits             |
| Compliance record gaps     | Schema-validated records; missing fields trigger follow-up prompts; weekly gap digest |
| Multi-site sprawl          | Per-outlet config, shared brand/supplier profiles, operator-level rollups             |



# 05 Business Impact & Lessons

---

## Where MitchAI creates value

6

Specialist agents + planner

Hybrid

Local + cloud LLMs by task  
profile

WhatsApp

First-class approval surface

10+ yrs

Operator experience baked into  
design

## Who MitchAI is for

- **Independent restaurants, gastropubs, and small hotels** (1–10 outlets) who run lean and need an AI ops layer that doesn't have a chain-scale price tag.
- **Multi-site operators** (3–10 outlets) where the rollup view across compliance, marketing, and stock is itself a product.
- **Hospitality groups** piloting AI without committing to a six-figure enterprise platform.

## Scale-out story

The product is built first for the UK market where the regulatory and operational patterns I know best apply. The architecture is country-agnostic; the localisation surface is brand profiles, supplier integrations, and per-country compliance schemas (HACCP for the EU/UK, FDA for the US, etc.). Country expansion is a content and integration project, not a re-architecture.

## Lessons learned

### 1. The operator-customer is the only honest design partner.

Every feature that survived contact with reality was designed against the question "would I, as a GM at the end of a 14-hour Saturday, actually use this?" Every feature that failed was designed against "wouldn't this be cool?"

### 2. Workflow plumbing matters more than AI.

n8n + adapters + retries + scheduling are 70% of the delivered value. The AI is what makes the headlines; the workflow layer is what makes the system not collapse the first time a supplier portal goes down.

### 3. Local LLMs unlock the price point.

Without local inference, the cost curve makes MitchAI an enterprise product. With it, MitchAI is affordable for the operator who needs it most. This is not a research statement — it's a unit-economics one.

### 4. WhatsApp is a feature, not a fallback.

Every hospitality operator I have ever met lives in WhatsApp. Making it the primary approval surface — instead of a dashboard the operator has to remember to open — is the single biggest UX decision in the product.

### 5. Conservatism wins twice.

Never posting, sending, or filing without operator approval slows the demo and wins the real customer. Hospitality operators trust their judgement; they don't trust autonomous software with their brand or compliance records, and they shouldn't.

## What I'd do differently from day one

- Build the brand-profile schema before the first Media draft, not after the third "this doesn't sound like us."
- Treat WhatsApp as the primary surface from commit #1, not after building a web dashboard nobody opens.
- Stand up the per-outlet cost dashboard alongside the first paying customer, not after.
- Adapter pattern from the second integration, not the fifth.

## What this case study demonstrates

- Vertical AI design from inside the customer's reality, not abstracted from it.
- Engineering discipline around unit economics, integrations, and the operator-grade UX that AI startups routinely undervalue.

- Multi-agent orchestration combined with mature workflow tooling (n8n) — the right way to ship operational AI without rebuilding scheduling for the tenth time.

*MitchAI is the only product I have ever built where I am simultaneously the engineer, the founder, the first customer, and the friend the customers complain to after a bad Saturday. That isn't a positioning line — it's the design constraint.*