PORTFOLIO CASE STUDY · 2026

BDM Lead Intelligence Engine

A full-stack, AI-native B2B outreach platform that researches executives, analyzes pain points, scores fit, and drafts personalized emails — using OpenAI and Claude in parallel.

ROLE	DOMAIN	STATUS
GTM Engineer	**Sales Intelligence / GenAI**	**Production · Live**

TypeScript · Node.js · PostgreSQL
OpenAI Responses API · Claude Code CLI

01 · OVERVIEW

Replacing a week of manual research with a 5-minute AI pipeline.

Business Development Managers spend hours prospecting — finding the right executives, reading LinkedIn, chasing verified emails, scanning news for buying signals, and then writing a cold email that doesn't sound like a cold email. I built an end-to-end system that does all of it in one click.

5	2	17+	100%
AI Pipelines	AI Providers	Services / Modules	TypeScript

The Problem

- Prospecting is repetitive and time-consuming — reps burn 60%+ of their week on research, not selling.
- Generic cold emails convert poorly; personalized ones require deep context that no single data source provides.
- Existing tools (Apollo, ZoomInfo, Outreach) are expensive, siloed, and don't reason about whether a lead is actually a good fit.

The Solution

A self-hosted lead intelligence dashboard that orchestrates **LLM web-search agents** through a five-stage pipeline — scrape, enrich, verify, score, and draft — with a pluggable provider layer that routes each stage to either OpenAI's Responses API or the Claude Code CLI (driven by the user's Claude Max subscription, zero API spend).

> **Key insight:** LLMs are excellent researchers when given tight, verifiable prompts. The hard problem isn't the model — it's the orchestration: structured output extraction, timeout handling, graceful JSON recovery, and making the whole thing feel instant in a browser.

02 · ARCHITECTURE

Five-stage AI pipeline with a pluggable provider router.

Every stage is a thin dispatcher that routes to an OpenAI or Claude implementation behind the same interface. Swapping providers is a single field in the request body — no frontend change, no redeploy.

01
Scrape
LLM web-search finds verified executives by role, city, and industry.

02
Enrich
Bio, hobbies, socials, fun facts — only from verified sources.

03
Verify
MX lookup + SMTP catch-all probe + pattern inference.

04
Pain Points
News, reviews, competitor pressure, role-specific challenges.

05a
Fit Score
Scored against your own company profile — with job postings as buying signals.

05b
Email Draft
Persona-tuned subject + body with personalization hooks.

Provider Router Pattern

```
// src/services/scraper.ts — same pattern for every stage
export async function scrapeLeads(req: ScrapeRequest): Promise<Lead[]> {
  return req.provider === 'claude'
    ? scrapeWithClaude(req)  // spawns Claude CLI subprocess
    : scrapeWithOpenAI(req); // Responses API + web_search_preview
}
```

OpenAI Path

Direct calls via the **Responses API** with the `web_search_preview` tool. Falls back to `gpt-4o-search-preview` chat completions when Responses isn't available. Structured JSON enforced by prompt contract.

Claude Path

Spawns `@anthropic-ai/claude-code` as a child process with `--print`, stdin-piped prompts, and `--allowedTools WebSearch,WebFetch`. Auths via the user's **Claude Max subscription** — no API key, no per-token billing.

03 · FEATURES THAT SHIP

Every module is a deliberate product decision.

Executive Scraping

Role-, city-, and industry-targeted search with strict verification rules — no fabricated LinkedIn profiles, no "probably" social handles.

Email Pattern Intelligence

LLM searches for any real email at the target domain (GitHub, contact pages), infers the format, and applies it — with confidence scoring (0–100) and 3–5 variants.

SMTP-Level Verification

Hand-rolled MX resolver + catch-all probe over port 25 using `net.Socket`. Detects Google Workspace / M365 / Proofpoint / Mimecast. Adjusts confidence accordingly.

Pain Point Analysis

Per-lead research: recent news, funding, layoffs, customer complaints, competitor pressure, role-specific challenges — synthesized into a single paragraph.

Fit Scoring vs. Your Profile

Seller fills in services + ideal customer once. Every lead gets a **0–100 score** and **hot/warm/cold/no-fit** tier — driven by job postings and buying signals.

Conference Intelligence

Separate pipeline discovers upcoming/past industry conferences, then scrapes attendee rosters for matching roles — an entirely new lead source.

Persona-Aware Email Drafting

Single-row settings table drives tone (*professional / casual / friendly / bold*), length, and sender identity. Emails cite specific pain points with personalization hooks.

CSV / JSON Export

27-column CSV export with CSV-injection-safe escaping, plus full JSON dump for downstream CRM ingestion.

Database Design

Four PostgreSQL tables (`leads`, `email_settings`, `company_profile`, `conferences`) with self-healing **additive migrations** — every startup runs `ALTER TABLE ADD COLUMN IF NOT EXISTS`, so the schema evolves safely without manual migration files.

04 · ENGINEERING CHALLENGES

The interesting problems lived in the glue.

Challenge 1 — Running Claude without an API key

Anthropic's SDK bills per-token. The Claude Max subscription is flat-rate but only exposed through the `claude-code` CLI. I wrapped the CLI as a Node child process: spawn with `--print`, stream stdin, capture stdout, enforce a `max(maxTurns × 30s, 300s)` timeout, and surface structured errors. Users bring their own Max plan and pay nothing per lead.

Challenge 2 — LLMs hallucinate JSON

Even with "return only JSON" instructions, models wrap responses in markdown fences, add preambles, or return objects when you asked for arrays. I wrote `extractJSON()` and `extractJSONObject()` — tolerant parsers that strip fences, find the outermost balanced braces/brackets, and recover gracefully instead of throwing. Every pipeline stage uses them.

Challenge 3 — Email verification without a third-party API

Rather than pay Hunter/ZeroBounce, I built the verifier from standard Node modules: `dns.resolveMx` for MX records, `net.Socket` for a polite SMTP conversation (`EHLO → MAIL FROM → RCPT TO`) that detects catch-all domains by probing a random recipient. Provider hints are inferred from MX hostnames.

Challenge 4 — Timeouts at the right layer

Web-search agents are slow and unpredictable. The Claude runner uses a dynamic timeout formula scaled by `maxTurns`; the OpenAI path falls back from Responses API to chat completions on error; both log every chunk of stdout/stderr with elapsed timestamps for forensic debugging without a full APM stack.

Challenge 5 — Prompt accuracy under LLM confidence

Early prompts produced gorgeous enrichment — most of it fabricated. I rewrote the scraper prompt with **eight explicit accuracy rules**: verify social profiles against the company bio, never guess on generic names, blank fields beat fake fields. Accuracy went from roughly 60% verifiable to roughly 95%.

05 · STACK & IMPACT

What shipped, and why it matters.

Technology

TypeScript · Node.js · Express · PostgreSQL · OpenAI SDK · Responses API · Claude Code CLI · WebSearch / WebFetch · Vanilla HTML / CSS / JS · DNS · SMTP · net.Socket · tsx · dotenv · pg

Impact

Speed

Research time collapsed from hours to minutes

A full prospecting cycle — scrape, verify, score, draft — runs in a single session, no tab-switching, no CSV gymnastics.

Cost

Zero per-lead AI spend via Claude Max

The dual-provider layer lets users route to their flat-rate Claude subscription, eliminating variable OpenAI token costs for high-volume runs.

Quality

Verified data, not hallucinated data

Strict accuracy rules, MX/SMTP checks, and confidence scoring mean reps trust the output — no more apologizing for wrong names.

Extensibility

New pipelines in a day

Adding a stage (fit scoring, conference discovery) means: type, prompt, two provider files, a route. The pattern is load-bearing.

What I'd Add Next

- **Gmail / Outlook integration** — send drafted emails directly, track opens and replies, close the loop.
- **CRM sync** — bidirectional HubSpot / Salesforce adapters driven by the same provider-router pattern.
- **Vector memory of past outreach** — never pitch the same person twice; learn which angles converted.
- **Multi-tenant auth** — currently single-user; adding workspace scoping unlocks a SaaS path.