GTM ENGINEER PORTFOLIO

Artur Rakhimullin

AI Automation • CRM Workflows • Revenue Operations • Business Systems

PORTFOLIO V0.2 Evidence-updated edition. Confirmed facts are separated from unverified claims; technical artifacts are sanitized for public use.

POSITIONING

GTM Automation Engineer building practical systems that connect customer acquisition, CRM data, service delivery, follow-up, and AI-assisted operations.

I combine owner-operator experience with workflow design: translating a commercial process into clear stages, reliable records, reusable communications, and automations that keep work moving without losing human oversight.

Portfolio at a glance

- 3+ years operating a customer-facing service business (Surf Moving, March 2023–present).
- Built and coordinated a six-person service team and delegated daily delivery through repeatable procedures.
- Managed the customer lifecycle from inquiry and qualification through estimate, scheduling, delivery, payment, feedback, and follow-up.
- Administered Wilcart CRM, customer and referral records, website content, reviews, local listings, and digital marketing.
- Built public trust signals: 5.0 on Google across 17 reviews and 4.9 on Yelp across 8 reviews (SEO audit snapshot, July 2026).
- Built Python-based retail sales and device-data workflows covering ZIP/JSON ingestion, normalization, reporting, Foreman facts, and operational follow-through.

Core stack

Confirmed in accessible materials: Python, asyncio, Pydantic, pandas, XlsxWriter, ZIP/JSON, CSV, SQLAlchemy, aiohttp, Foreman API, Google Sheets API, Telegram aiogram, Artix POS, Wilcart CRM, WordPress, Microsoft 365, Google Workspace, ChatGPT, and Codex.

Evidence boundary: Puppet is supported only as part of the Foreman/facts architecture; standalone Puppet manifests were not found. Competitor-price parsing, LTV, renewal segmentation, and measured time-savings remain unverified and are not claimed.

CASE STUDY 01

Lead-to-Service Revenue System

Surf Moving — intake, qualification, CRM, delivery, reputation, and follow-up

CONFIRMED A live commercial workflow supported by resume materials, a documented service-workflow sample, and the July 2026 SEO / listings audit.

Business challenge

New inquiries could arrive at any hour through different channels. Manual monitoring made response quality inconsistent and forced the team to reconstruct information customers had already provided. The GTM challenge was to turn inbound demand into a qualified, scheduled service while preserving context, ownership, and a visible next action.

Role: Founder & Operations Manager; owner of the customer journey and its supporting operating system.

Revenue workflow

Website / Google / Yelp → intake form → Wilcart CRM record → structured notification → qualification → estimate → scheduling → service delivery → payment → review request → follow-up

System components

- An online inquiry creates or updates one Wilcart CRM lead record; a structured notification exposes the essential details and next action.
- Statuses make ownership visible: new, awaiting information, quoted, scheduled, or closed.
- Standard response templates accelerate acknowledgment and follow-up; each message is reviewed and adapted before sending.
- Documented procedures support team handoffs, while the website, Google Business Profile, Yelp, listings, and reviews support acquisition and trust.

Outcome & evidence

The workflow improved response consistency, reduced administrative reconstruction, and made follow-through visible. Customer context stayed attached to the correct record, and daily service delivery was delegated through repeatable procedures to a six-person team.

- Scale and public evidence: six-person team; Google 5.0/17 reviews; Yelp 4.9/8 reviews; 10 Yelp photos (July 2026 audit).
- Process evidence: intake-to-CRM update, notification, status model, communication templates, and follow-up reminders documented in the service-workflow sample.
- Control design: automation handles routing and reminders; people retain responsibility for qualification, tone, exceptions, and service quality.

Evidence boundary: the available materials support the workflow, team scale, and public trust signals. They do not establish a precise conversion lift, revenue contribution, or number of hours saved, so those metrics are intentionally not claimed.

CASE STUDY 02

Retail Sales & Customer Operations Automation

ZIP/JSON sales archives, normalized reporting, store metadata, and loyalty workflows

CONFIRMED Retail sales/customer operations automation supported by sanitized code, an anonymized input/output pair, and a source-level audit.

Business challenge

Retail sales data arrived as distributed ZIP archives containing shift, receipt, payment, and item-level JSON. Operations needed a reliable way to extract these records, distinguish sales and returns, enrich them with store metadata, and produce a reviewable tabular report.

What I built

- Implemented asynchronous ZIP ingestion that separates embedded JSON sections, removes service headers, deserializes records, and validates the resulting sales objects.
- Flattened shifts, checks, payments, and product positions into an Excel-ready dataset with receipt, cashier, item, quantity, price, total, payment method, timestamp, and document type.
- Enriched output with store and legal-entity metadata from Foreman and connected the operational platform to a Telegram loyalty-card workflow.
- Kept a clear evidence boundary: the repository confirms sales and customer-operations automation, not email identity resolution, LTV, renewal campaigns, or measured conversion impact.

GTM relevance

This case demonstrates GTM implementation at the data-to-action layer: converting fragmented transaction records into consistent operational reporting and connecting customer-facing loyalty steps to the same business platform.

Evidence package

- Selected code: a sanitized 28-line parser from sales.py shows ZIP inspection, JSON-section separation, service-header removal, and deserialization.
- Input example: anonymized shift JSON with shop, cash register, receipt, item, and payment objects; it is explicitly labeled as a structural example rather than a production record.
- Output example: anonymized CSV with receipt, shift, cashier, product, quantity, price, total, payment, timestamp, and SALE/RETURN document type.
- Confirmed technologies: Python, asyncio, Pydantic, pandas, XlsxWriter, ZIP/JSON, SQLAlchemy, Foreman API, aiogram, and Artix POS. Exact receipt/customer volume and time saved remain unmeasured.

CASE STUDY 03

Infrastructure Operations Automation

Multi-vendor POS inventory, Foreman facts, snapshots, and consolidated reporting

CONFIRMED Infrastructure operations automation supported by multi-vendor parsers, Foreman integration, anonymized data, and aggregate scale from real local outputs.

Business challenge

Device state was distributed across vendor-specific JSON for VikiPrint, ATOL 5, and ATOL 2 cash registers. Operations needed one normalized view of fiscal-storage dates, service status, firmware, document backlog, models, stores, and timestamps—without manually opening hundreds of device files.

What I built

- Built separate Python parsers for VikiPrint, ATOL 5, and ATOL 2 and mapped vendor-specific JSON into a shared CSV schema of more than 30 operational fields.
- Added resilient directory traversal, per-device error handling, normalized status/date fields, and minute-level snapshots with rotation of the latest 15 files per device type.
- Implemented Foreman API retrieval for fleet facts, freshness filtering, and expiration reports, plus a consolidated handoff to Google Sheets reporting.
- Separated publishable evidence from sensitive infrastructure data: all examples redact tax IDs, serials, addresses, registration numbers, hostnames, internal endpoints, and credentials.

Architecture

Vendor JSON and Foreman facts → parser-specific validation → common device schema → consolidated CSV → timestamped snapshots → Google Sheets / operational reports.

Scale & evidence

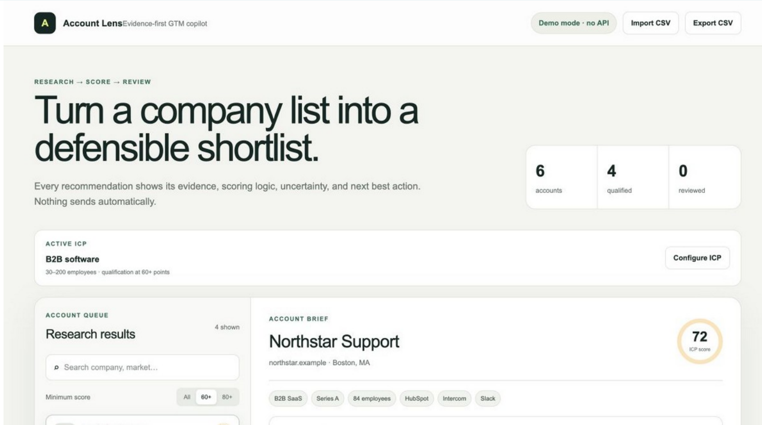
- Selected code: a sanitized 31-line ETL excerpt shows cash-* discovery, store/device parsing, vendor-file processing, exception handling, and unified CSV output.
- Anonymized input/output pair documents the transformation from nested fiscal-device JSON to a shared CSV row without exposing production identifiers.
- Verified local outputs: 702 VikiPrint records, 728 ATOL 5 records, and 20 ATOL 2 records—1,450 device-records total. This is record count, not a claim of 1,450 unique cash registers.
- Store coverage bounds: at least 571 and at most 1,005 unique stores across independent files; cross-file deduplication was intentionally not performed on sensitive identifiers.
- Confirmed outcome: automated normalization, consolidated reporting, and 15-snapshot history per device type. SLA, MTTR, error reduction, and hours saved were not measured and are not claimed.

CASE STUDY 04

Account Lens — GTM Qualification Copilot

Evidence-first account research, deterministic scoring, and human review

CONFIRMED Working portfolio application built and deployed in July 2026; no external API or automated outreach.



Live prototype: lens.arturrakhimullin.com

Product problem

AI-assisted sales research can produce plausible recommendations without showing why an account was selected. Small teams need a reviewable shortlist—not another black-box sender.

What I built

- Responsive account queue with search, filters, configurable ICP rules, and deterministic scoring.
- CSV import preview, evidence labels, uncertainty flags, explicit risk penalties, and recommended account angles.
- Human Approve/Hold decisions, message drafting, and CSV export—with no automatic sending.

Architecture & safeguards

Fictional dataset or validated CSV → configurable rules → transparent scoring with risk penalties → evidence-aware brief → human review → CSV export. No external API, personal-contact collection, or automatic sending. The project demonstrates ICP design, business rules, data modeling, explainability, uncertainty handling, workflow UX, approval controls, and integration boundaries.

Validation

- Six fictional accounts; four qualify at the default 60-point threshold; configurable scoring and review state verified.
- CSV schema validation, preview, import, filtering, and export were exercised in the working interface.
- The evaluation panel separates verified prototype behavior from unproven predictive claims.

TECHNICAL EVIDENCE APPENDIX

Sanitized artifacts, confirmed scale, publication boundaries, and remaining gaps

Evidence inventory

Claim	Status	Current source
Surf Moving tenure / role	Confirmed	Existing resume source
Six-person team	Confirmed	Existing resume source
Wilcart CRM + full customer lifecycle	Confirmed	Existing resume source
Website, reviews, listings, marketing	Confirmed	Existing resume source
Google 5.0 / 17 reviews; Yelp 4.9 / 8	Confirmed	July 2026 SEO audit
SEO / listings audit + correction template	Confirmed	Prior Codex SEO project
Surf Moving intake-to-CRM workflow	Confirmed	Documented service-workflow sample
Account Lens GTM Copilot	Confirmed	Working deployed application
AI-assisted forms/templates/workflows	Confirmed	Existing resume source
Python / APIs / data pipelines	Confirmed	Sanitized source audit + excerpts
Retail sales archive parser	Confirmed	Code + anonymized I/O
LTV / renewal workflow	Missing	No supporting artifact found
Foreman / device fleet	Confirmed	Code + real aggregate counts
Competitor-price parsing	Missing	Conversation only

Project 02 — implementation proof

- Code artifact: project2_cash_sales_parser.txt — 28 lines from the production parser, sanitized for portfolio use.
- Input artifact: project2_input_anonymized.json — structural shift, receipt, item, and payment example; no production values.
- Output artifact: project2_output_anonymized.csv — normalized sales-report schema with SALE document type.
- Publication rule: describe this as Retail Sales & Customer Operations Automation, not e-commerce LTV or competitor-pricing automation.
- Missing metric: exact receipts, customers, rows processed, and measured time savings require a defensible baseline before publication.

Project 03 — implementation proof

The evidence package includes a sanitized ETL excerpt and an anonymized before/after pair. Aggregates below were computed from local results without copying sensitive rows.

- Scale: 1,450 records across three device-output files; 571–1,005 estimated unique-store range; 30+ normalized fields.
- Resilience: multi-vendor parsers, per-file exception handling, consistent UTF-8 CSV output, and timestamped snapshots.
- History: retention of the latest 15 snapshots per device type supports comparison and operational review.
- Security boundary: production tax IDs, addresses, serials, registration numbers, hostnames, URLs, and Google credentials are excluded.
- Missing metric: manual baseline, hours saved, SLA/MTTR, and error-rate reduction were not recorded.

Remaining evidence gaps

- Add one safe interface screenshot for the sales workflow and one for Foreman/device reporting if they can be redacted reliably.
- Confirm a defensible project-02 processing volume: receipts, shifts, rows, stores, or reporting cadence.
- Measure one before/after operational baseline rather than estimating hours saved after the fact.
- Keep competitor pricing, LTV, renewal automation, standalone Puppet manifests, and deployment scripts out of public claims until supporting artifacts are found.

Source note

Updated July 22, 2026 from the referenced conversation, resume sources, Surf Moving SEO audit, the working Account Lens application, and a read-only audit of the original technical projects. Only sanitized code excerpts, anonymized structural examples, and aggregate counts are used; production records, credentials, customer data, device identifiers, tax IDs, addresses, and internal infrastructure details are excluded.