

Ruan Grobler

Cape Town, South Africa | ruan@receva.co.za | receva.co.za

PROFESSIONAL SUMMARY

AI and Automation builder with production Microsoft Power Platform delivery at TrueProp Property Services (True Group), where I had five automation systems live in my first month, three I built and two I took over and hardened, plus a sixth built and in business walkthrough. I build Power Automate approval flows, AI Builder GPT email intake, SharePoint list architectures and document OCR pipelines, and I quantify the hours and Rand each one saves. I also founded Receva, an AI voice receptionist product built on Retell, n8n and Twilio, handling real customer calls. Microsoft PL-900 exam booked, with a documented certification path toward PL-400.

EXPERIENCE

AI and Automation, TrueProp Property Services (True Group), Western Cape

July 2026 to present

- Took five automation systems live in my first month, entirely on the Microsoft stack (Power Automate, SharePoint Online, AI Builder, Microsoft Forms): three built from scratch, two taken over from a departed developer and hardened
- Designed an AI email intake system using AI Builder GPT prompts that reads inbound business email, extracts requestor, department, request and priority, creates numbered tickets and replies to the requestor automatically with a pre-filled correction form
- Built an AI broker deal intake pipeline that reads property deal emails, captures property, broker, sector and location into SharePoint, files attachments into dated folders and deduplicates on message ID. Captured the pipeline's largest deals unaided in its first weeks, with no template and no sender training
- Took over an orphaned document OCR sorting system for a retail branch, repaired it and hardened it: scans from the branch copier are classified by AI into eight categories and filed into SharePoint automatically
- Built a bulk ordering platform end to end in three days, ranked the number one priority by management: QR code and form intake, order references, mobile approvals for the branch manager, automatic buyer confirmations and prep sheets for the branch. In business walkthrough before go live
- Built a meeting-transcript-to-handover system: AI reads recorded meeting transcripts, drafts the handover register entry and checks delivery against the original request, with human sign off mandatory
- Recovered 6 orphaned production systems from a departed developer's inaccessible account and returned them to service with documentation, including five HR probation workflows that had been down for over a month
- Migrated the historic deal backlog with a machine reconciliation of every record against the source, rather than a spot check
- Ran a POPIA review of a lease application system, then designed and shipped POPIA and credit-check consent capture on the application form, with a hard gate that stops the affordability step when consent is not recorded
- Own the AI and Automation environments end to end (DEV, TEST, PROD): set the build-in-DEV, run-in-PROD standard, created and retired developer environments, and manage solution ownership and co-owner handover so no flow depends on one person's account
- Standards applied on every build: failure alerting on every flow, no data on personal drives, full spec with every arrival path enumerated before building, test cases written before building, human approval on all decisions

Founder, Receva

2026 to present

- Founded an AI voice receptionist business for hospitality and healthcare SMEs, built on Retell AI, n8n, Twilio and REST APIs, taking real customer calls end to end

- Designed and operate the full stack: call flows, webhook integrations, booking capture, CRM writeback and follow-up automation

SELECTED PERSONAL BUILDS

- ApplyForge: an AI job-matching engine in Next.js and TypeScript on the Claude API, with structured outputs, live job-board scraping, match scoring and automated CV and cover letter tailoring
- ApplyForge CI/CD: continuous integration in GitHub Actions. Every push to main and every pull request installs from the lockfile with npm ci, then gates on TypeScript typecheck, ESLint, the 121 test Vitest suite and a production build. A dependent job builds the multi-stage Docker image (build only, never pushed) with Buildx layer caching and boots the container to confirm it serves HTTP 200 as a non-root user. Every stage verified locally
- ApplyForge testing: 121 Vitest unit tests across 8 files over the ingestion, scoring, validation and storage logic, with the network and the LLM API mocked at the module boundary so the suite runs offline in under a second. It locks in score clamping, URL canonicalisation, validation of model-supplied list indices, and safe handling of a malformed data file
- Agent Pipeline: multi-agent document triage in Python on LangGraph. A supervisor orchestrates four agents (classifier, extractor, validator, router) and owns retry with exponential backoff, escalation and graceful degradation, over typed shared state, schema-constrained structured output from Claude via the Anthropic SDK, and a JSONL audit log of every decision. 267 unit tests, no network calls. Verified live run over 26 synthetic property and finance documents: 26 of 26 routing decisions and 25 of 25 document types correct. LangGraph and the Anthropic SDK directly, with no LangChain, CrewAI or AutoGen
- DealFlow SQL: a property deals database in PostgreSQL 16 running in Docker, with schema design, CSV staging loads and 10 analysis queries
- DealFlow CRM: Dataverse proof of concept in build (spec, data model, seed data complete)
- Claude Code automation rig: custom subagents, hooks, MCP servers and nightly scheduled agents that run research, monitoring and build tasks unattended
- receva.co.za and client demo microsites: Next.js and Astro, deployed on Vercel

SKILLS

Microsoft stack: Power Automate (cloud flows, approvals, error handling, premium license and premium connectors), SharePoint Online (lists, libraries, views, permissions), AI Builder (GPT prompts, OCR), Microsoft Forms, Power Platform environment and solution management (DEV, TEST, PROD, solution ownership and co-owner handover)

AI engineering: designing and operating an AI agent rig on Claude Code and the Claude API (custom subagents, MCP servers, hooks, scheduled autonomous agents, multi-agent workflows); LangGraph multi-agent orchestration (supervisor and worker agents, typed state, structured output, retry and graceful degradation); OpenAI API, Retell AI voice agents, prompt engineering, structured outputs

Development and data: TypeScript and Next.js (deployed on Vercel), Python (type hints, PEP 8, pytest), SQL, Docker, REST APIs, JSON, webhooks, n8n, Twilio, Supabase (Postgres), Airtable, Git

Data and delivery evidence: PostgreSQL schema design with primary and foreign keys, CSV staging loads and 10 SQL analysis queries spanning joins, GROUP BY with HAVING, subqueries and window functions; multi-stage Docker containerization of a Next.js 16 app (standalone output, non-root user, HTTP 200 smoke test); CI/CD in GitHub Actions; unit testing with Vitest and pytest, network and LLM API mocked so suites run offline

Practice: POPIA-aware design, spec-first builds, tests before code, documentation and runbooks

CERTIFICATIONS AND EDUCATION

Microsoft PL-900 Power Platform Fundamentals (exam scheduled), with a documented certification path toward PL-400 Power Platform Developer

Education and references: available on request, with written stakeholder confirmation of delivered systems.