The Job-Search Workflow

A step-by-step guide to running a job search like a software project, with an AI assistant that prepares every application and a human who sends each one: a private repository, one tracking file, written content rules, interview preparation as a test suite, a CASP cockpit whose roadmap ends at a signed contract, and mail automation that fills your Drafts folder and never presses Send.

EDITION 1.1 — 25 SEPTEMBER 2026

S1 One private repository, one folder per employer

S2 One tracking file — sent means dated

S3 Written content rules — no unprovable sentence

S4 Masters and routing — which CV, which letter, where

S5 Interview preparation as a test suite

S6 Offer watch — sources, filters, deduplication

S7 CASP — a roadmap that ends at a signed contract

S8 Mail automation — prepare everything, send nothing



Juste Thales Gnimavo

Founder & CEO, ZeroSuite, Inc. · with Claude (Anthropic) as AI CTO · Abidjan, Côte d'Ivoire

justegnimavo.com · CASP, the Coding-Agent State Protocol — casp.sh · thalesandhisaictoclaude.com

© 2026 ZeroSuite, Inc. · Licensed CC BY-NC-ND 4.0 · Full notice on the last page

Contents

00 The operating model — the AI prepares, you decide and you send

01 Step 1 — The repository: one private repo, one folder per employer

02 Step 2 — The tracking file: sent means dated

03 Step 3 — The content rules your assistant must follow

04 Step 4 — Masters and routing: which CV, which letter, where

05 Step 5 — Interview preparation as a test suite

06 Step 6 — Offer watch: sources, filters, deduplication

07 Step 7 — CASP: a roadmap that ends at a signed contract

08 Step 8 — Mail automation: fill the Drafts folder, never press Send

09 Checklists and the first message to paste

> **Where this comes from.** This guide is the written-down version of one real session, on 24 September 2026, between a CEO and Claude as AI CTO. No product code was written that day; the tooling ZeroSuite uses to ship software ran a job search instead. The method does not depend on the trade, so the examples rotate through five: developer, nurse, salesperson, accountant, site manager. They are illustrations, not the real case. The full story is the companion article on thalesandhisaictoclaude.com.

00 · The Operating Model

Most people use an AI assistant for a job search the way they use a spell-checker: paste a letter, take what comes back. This guide uses the opposite contract:

- **You own:** the target role, the list of employers, every claim about yourself, every send.
- **The assistant owns:** structure, drafting, consistency between documents, verification of every exported file, the tracking file, the roadmap — and saying no to a sentence you cannot defend in an interview.
- **The interface:** you give facts and decisions; the assistant gives drafts and recommendations, and argues when it disagrees. You decide. Decisions and reservations are written down.

Five rules hold everything together. **1.** The assistant never sends anything. **2.** An application counts as sent only when it carries a date. **3.** No claim you cannot prove in the interview. **4.** What an employer received is frozen, even when your master CV evolves. **5.** The project ends at a signed contract, not at "applications sent".

What you need: a computer, git (or any folder you back up), an AI assistant that can read and write files — Claude Code is what we use — and a word processor that exports PDF. CASP (Step 7) is optional and free.

01 · The Repository

ONE PRIVATE REPOSITORY PER CANDIDATE, SEPARATE FROM ANYTHING PUBLIC

If you have a personal CV website, it lives in its own repository. The job search never rides along with a site deploy, and a site change never lands in the job search. Two repositories, two lifecycles: the private one can take a commit every ten minutes without anything going live.

```
job-search/
  RULES.md                  # the rules, named so your assistant loads them (Step 3)
  tracking.md               # the ONE tracking file (Step 2)
  contacts.md               # employers, type, verified contact, source
  _masters/                 # CV per target role, generic letters, e-mail templates
  _interview/               # questions and coached answers (Step 5)
  EMPLOYER-A/               # one folder per employer, capitals and hyphens
    Letter_EMPLOYER-A.docx  Letter_EMPLOYER-A.pdf
    Mail_EMPLOYER-A.txt     # To, Subject, Attachments, body
    CV.pdf                  # the CV exactly as sent — frozen
  RECRUITER-B/
  casp/                     # optional cockpit (Step 7)
  .env                      # mail credentials — git-ignored from the first commit
```

Why a folder per employer. At three applications you remember everything. At fifteen you no longer know which letter version went to which employer or which CV the second recruiter received. The folder answers it, forever.

02 · The Tracking File

ONE FILE, ONE ROW PER APPLICATION, A FIXED CHAIN OF STATUSES

EMPLOYER	ROLE	TYPE	STATUS	SENT ON	FOLLOW-UP	NOTES
Employer A	Tech lead	spontaneous	ready			Career page
Recruiter B	Tech lead or developer	spontaneous	sent	2026-09-24		Staffing firm

Example rows for a developer aiming at tech lead. Plus two columns in the real file: **channel** (the verified address or the online form) and **folder**. **Statuses:** `to prepare → ready → sent → followed up → interview / rejected / no answer`.

The one rule: *a row is submitted if and only if "Sent on" holds a date.* A check-mark is a claim; a date is evidence. With twenty rows you cannot fake it by accident. A follow-up is a new date in its own column, once, ten working days after sending with no answer.

Contacts: an address enters `contacts.md` only if you read it on the employer's own site or in the offer, with the source noted. Otherwise it stays "to find". Never a guessed `jobs@company.com`.

03 · The Content Rules

WRITTEN ONCE IN A FILE YOUR ASSISTANT LOADS AT THE START OF EVERY SESSION, SO EVERY FUTURE AGENT INHERITS THEM

```
# RULES.md — job search of <name>
- The assistant never sends anything. It prepares and sets the row to "ready".
- No guessed address. A contact needs a source; otherwise "to find".
- One application per employer and per role. Check tracking.md first.
- The master CV is the truth for dates, employers and job titles.
- No claim that cannot be defended in an interview. No record, no "zero errors".
- No internal figures of the current employer. No criticism of it.
- PDF only as attachments. Every exported PDF is re-read, not trusted.
- Credentials live in .env only, never committed.
```

A rule the assistant never reads is not a rule. Name the file after what your assistant loads automatically at startup (`CLAUDE.md`, `AGENTS.md`, or whatever your tool documents). If it loads nothing on its own, open every session with "*Read RULES.md first.*" This guide calls the file RULES.md.

The rule that matters most is the unprovable sentence. In our session the CEO's draft said the candidate "beats almost every record". The assistant refused it: no one publishes those records, and "which records?" has no answer. Replace every superlative with a fact a reference could confirm. Not "an exceptional engineer" but "*led the billing migration, no rollback*"; not "irreplaceable on the ward" but "*trained the last four new hires*"; not "the best closer" but the quota reached in a year you can name; not "flawless" but "*monthly close brought from ten days to six*"; not "always on schedule" but the handover date held on a named project. "Zero errors" goes too: nobody believes it, and it invites the wrong question.

04 · Masters and Routing

DERIVE, DON'T REWRITE — AND MAKE THE ROUTING A RULE, NOT A MOOD

One CV per target role (target role and current role), **one generic letter per role**, **one e-mail template per role** with `[bracketed]` placeholders and one line at the top: *no bracket may survive in a sent e-mail.* The second letter is the first one with the role-specific paragraph replaced — diffed, not rewritten, so the claims cannot drift apart.

EMPLOYER TYPE	CV	LETTER	E-MAIL SUBJECT
Direct employer	Target-role CV	Names the employer	Spontaneous application – <role> – <name>
Recruiter / staffing firm	Same CV	Generic	<role> or <adjacent role> – <name>
Published offer	CV matching the offer	Names employer and offer	Exact offer title and reference, as the offer demands

The e-mail is a summary, not a cover note. Three short paragraphs: the employer's need, your proof, the invitation (the letter, the CV, the interview). "Please find attached" alone reads like a circular.

A trap to write down: a staffing-firm placement can reproduce the very contract you want to leave. It is worth taking only for a step up or a direct hire.

05 · Interview Preparation as a Test Suite

THE INCIDENT QUESTIONS OF YOUR TRADE, EACH WITH A COACHED ANSWER YOU CAN SAY OUT LOUD

Ours has 32 questions in six sections: presentation and motivation; the job itself; **tools, incidents and security**; managing people; personal questions; conditions. The third section is the one generic "top 50 questions" lists never have, and the one we recommend to everyone. Write the incident questions a recruiter who has done your job would ask:

- **Developer:** production goes down on a Friday evening and the last deploy was yours. Your first ten minutes?
- **Nurse:** you notice a dosage error after the medication was given. Who do you tell, in what order?
- **Salesperson:** a key client disputes a signed order and threatens to leave. What do you concede?
- **Accountant:** the accounts do not balance the day before the close. Where do you look first?
- **Site manager:** a delivery arrives non-compliant on the morning of the pour. Do you stop the pour?
- **Every trade:** which tools you master without help; someone from "IT support" calls and asks for your password.

Add one question for every strong claim in your letter. If the letter says you are fast, the interview will ask "*how do you know?*" — write that answer now.

06 · Offer Watch

WHERE TO LOOK, WHAT TO KEEP, WHAT TO REFUSE

- **Sources:** national job boards, employers' own career pages, the recruiters already in `contacts.md`. No automated collection on sites whose terms forbid it (LinkedIn included) — apply there by hand.
- **Filter on the trade, not on the title.** A title is a label each employer defines for itself: "consultant", "technician" or "coordinator" each cover jobs with nothing in common. Filter on tasks and context: the stack for a developer, the department and shift pattern for a nurse, what is sold for a salesperson, the scope for an accountant, the type of works for a site manager. Write the filter into `RULES.md`.
- **Deduplicate against the tracking file:** one application per employer per role.
- **Read your competition.** When a recruiter showcases the profiles it places, read them: they tell you which argument sets you apart, and that becomes the angle of every letter.
- **A published offer beats a guess:** copy its exact subject line, reference and required attachments (some demand a salary expectation in the letter).

07 · CASP — A Roadmap That Ends at a Signed Contract

THE STATE LAYER ZEROSUITE RUNS ON EVERY PRODUCT, USED HERE WITH NO CODE AT ALL

```
npm i -g @justethales/casp
cd job-search && casp init   # casp/state.json · now.md · roadmap.md + first session prompt
casp status                  # "where are we?" on one screen, at the start of every session
casp close && casp check     # at the end: state bumped from git, then validated — exit 1 on drift
```

PHASE	CONTENT
1 — Application kit	CVs, letters, e-mail templates, interview preparation, website aligned
2 — First wave	One employer folder at a time; each one sent by you before the next is prepared
3 — Offer watch	Sources, filters, folders prepared up to "ready"
4 — Follow-ups, interviews	One follow-up at ten working days; one interview sheet per employer
5 — Offer and transition	Compare with the current job, negotiate, notice period — **the cockpit closes at signature**

Keep a **"Blocked / to decide"** table in `roadmap.md` (a salary expectation to provide, a recruiter to identify) and record decisions with their reservations in the session log. Three weeks later, nobody reconstructs why a folder exists.

08 · Mail Automation — Prepare Everything, Send Nothing

THE CREDENTIALS EXIST TO FILL YOUR DRAFTS FOLDER, NOT TO SEND

A sent application cannot be recalled: a wrong attachment or a leftover `[placeholder]` stays in the recruiter's inbox. A burst of scripted sends from a new mailbox is what spam filters catch. Your manual send is the last review, and it costs two minutes. So the script **appends the prepared message to Drafts over IMAP**; you open your mail client, read, press Send, and write the date in `tracking.md`.

```
# drafts.py — python3 drafts.py EMPLOYER-A   (IMAP_HOST, IMAP_USER, IMAP_PASSWORD in .env)
import imaplib, os, sys, time, pathlib
from email.message import EmailMessage
d = pathlib.Path(sys.argv[1]); lines = (d / f"Mail_{d.name}.txt").read_text().splitlines()
head = dict(l.split(" : ", 1) for l in lines[:3])          # To / Subject / Attachments
body = "\n".join(lines[3:]).strip()
assert "[" not in body, "placeholder left in the e-mail"
msg = EmailMessage(); msg["From"] = os.environ["IMAP_USER"]
msg["To"] = head["To"]; msg["Subject"] = head["Subject"]; msg.set_content(body)
for name in head["Attachments"].split(", "):
    msg.add_attachment((d / name).read_bytes(), maintype="application", subtype="pdf", filename=name)
with imaplib.IMAP4_SSL(os.environ["IMAP_HOST"]) as m:
    m.login(os.environ["IMAP_USER"], os.environ["IMAP_PASSWORD"])
    m.append("Drafts", r"(\Draft)", imaplib.Time2Internaldate(time.time()), msg.as_bytes())
```

The Drafts folder name depends on the provider (`Drafts`, `INBOX.Drafts`, `[Gmail]/Drafts`): list it once with `m.list()`. Load `.env` into the environment before running, and check SPF, DKIM and DMARC on your domain before the first real send.

09 · Checklists and the First Message

BEFORE EVERY SEND

- Recipient address has a source in `contacts.md`
- Subject matches the rule (or the offer, exactly)
- No `[bracket]` left in the e-mail
- Letter names the right employer; one page
- PDF re-read after export, not trusted
- Attachments are PDF, frozen in the employer folder
- After sending: date in "Sent on", commit

EVERY WEEK

- Rows sent ten working days ago without answer: one follow-up
- New offers checked against the tracking file
- "Blocked / to decide" table emptied or dated
- `casp status` read, `casp check` green

BEFORE AN INTERVIEW

- The letter you sent that employer, re-read
- Every claim in it has a rehearsed proof
- The incident section said out loud once

THE FIRST MESSAGE TO YOUR ASSISTANT

```
I am looking for a job as <role>. Here are my CV and my current letter: <paths>.
Set up a private repository following this structure: one folder per employer, one
tracking file where a row counts as sent only when it has a date, and a rules file with
my content rules, named so that you load it at the start of every session.
You never send anything: you prepare, I send.
Challenge any sentence I cannot defend in an interview. Show me the plan first.
```

Copyright and licence

© 2026 ZeroSuite, Inc. — 1130 Ogletown Rd, Ste 2, Newark, DE 19711, United States. Author: Juste Thales Gnimavo. Licensed **Creative Commons BY-NC-ND 4.0 International**: copy and share this document in full, with credit, for non-commercial use. The method is yours — run it for your own search, teach it, adapt it; no permission is needed. *CASP — Coding-Agent State Protocol* and *ZeroSuite* are marks of ZeroSuite, Inc.

`creativecommons.org/licenses/by-nc-nd/4.0 · casp.sh · thalesandhisaictoclaude.com · hello@zerosuite.dev`