



THE SALES LEVER

# Command Center

Instruction Manual

Every tab. Every endpoint. Every cron. Every Cipher tool.  
The complete operator runbook for the client command center system.

Locked baseline — May 22, 2026  
Commit cd57e25 • Tag: baseline-2026-05-22

# What this is

The Client Command Center is a single, no-login URL per client that holds **everything between The Sales Lever and them** — live sales performance, verification reports, invoices, recap reports, call recordings, training videos, internal lead intake, and Cipher AI. Each client gets their own subdomain (e.g. [raffiti.the-sales-lever.com](#)), their own data files, their own POCs, their own Stripe + GHSL wiring. The reference build is Raffiti Media.

This manual documents the locked baseline as of **commit cd57e25, May 22, 2026** (git tag **baseline-2026-05-22**). If anything stops behaving as described here, you've drifted from baseline — revert or audit changes.

## Stack at a glance

|              |                                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Frontend     | Vanilla HTML + CSS + JS, Three.js galaxy, Chart.js, no framework                                                                                |
| Backend      | Python FastAPI (aa_server.py) on Railway, ~2300 LOC                                                                                             |
| AI           | Cipher = Claude Sonnet 4.6 via LLM API, tool-calling enabled                                                                                    |
| Data         | Google Sheets (source of truth for sales data) + local JSON files (recaps, verification, invoices, leads, sequences)                            |
| Integrations | Stripe (invoicing), GoHighLevel (CRM tags + workflows), tldv (call recordings), Slack (verification-ready trigger), Melio Pay (invoice payment) |
| Hosting      | Production: Railway @ subdomain.the-sales-lever.com. Mirror: Perplexity pplx asset URL                                                          |
| Auth         | Public read; sensitive actions gated by TATENDA_PW env var (default 'sales@')                                                                   |

# The tabs (sidebar nav)

The sidebar is organized into two zones: the **onboarding zone** (Roadmap, Build Out, Launch Call, Goals — wrapped in a glowing green box that disappears after graduation), and the **main zone** (everything else). Active tab gets a green left bar + brighter text.

| Tab                         | data-tab        | What it does                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Roadmap                     | onboarding      | The 'Start Here' page. Walks new clients through the entire program flow: Launch Call → Build Out → first sales → verification → invoice. The Roadmap navlink pulses green when the client hasn't graduated yet.                                                                                                                                                     |
| Build Out                   | buildout        | Implementation checklist. Tracks the systems The Sales Lever is building for the client (funnels, sequences, AI agents, dashboards). Each task has status + a target date.                                                                                                                                                                                           |
| Launch Call                 | launchcall      | The recording + notes from the kickoff call. Embedded video player, key takeaways, action items.                                                                                                                                                                                                                                                                     |
| Goals                       | goals           | Client revenue goals + progress bars. Editable via Cipher or via the main Goals navlink (which appears post-graduation).                                                                                                                                                                                                                                             |
| Performance Panel           | performance     | The DEFAULT landing tab. KPI grid (MRR, conductor/SDR/committer rates, refunds), date range picker (7D/30D/90D/MTD/YTD/All), the rotating 3D Raffiti logo galaxy with money-planet orbits, the LIVE performance card with refresh, and the welcome hero. Data pulled live from the Google Sheet via the recap engine cache.                                          |
| Sales Lever Training Videos | training        | Video library of TSL training content. Sourced from training-videos-source.md. Cipher can search this list.                                                                                                                                                                                                                                                          |
| Call Recordings             | call-recordings | tldv-synced recording library. Each card shows the call title, date, attendees, duration, and links to the recording + transcript. Synced via tldv webhook POST /api/recordings/webhook or manual sync via POST /api/recordings/sync.                                                                                                                                |
| Recap Reports               | recaps          | Weekly / Monthly / Yearly recaps generated by Cipher. Each recap reads the raw sheet snapshot, runs Claude Sonnet 4.6 on the window, and produces narrative + numbers. Rebuilt automatically every Monday 7am MT via cron f5031b1a.                                                                                                                                  |
| Verification Reports        | verification    | Month-end sales verification flow. Tatenda flips it to 'ready' in Slack → cron 8a13f045 rebuilds the data + deploys + starts the POC sequence (Day 1 'Raffiti Verification' tag pushed to GHL). Client reviews, clicks Approve & Generate Invoice → verification sequence stops, invoice sequence starts. Tracker UI at top shows current step. POC editable inline. |

|                       |                 |                                                                                                                                                                                                                                                                                         |
|-----------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Invoices</b>       | <b>invoices</b> | Invoice list with status (sent, paid, overdue). Each invoice shows amount + Melio Pay button + Stripe link. Auto-generated from Approve & Generate Invoice on a verification. Paid via Stripe → webhook stops invoice sequence + strips lingering tags.                                 |
| <b>Support</b>        | <b>support</b>  | Single contact card for raising support tickets. Goes to ops@thesaleslever.com.                                                                                                                                                                                                         |
| <b>Internal Leads</b> | <b>leads</b>    | Client-side lead intake. Paste a screenshot of a lead's profile → Cipher OCRs name + email + phone + notes → preview → submit. Submitted leads push to GHL contact + the kanban board (New / Contacted / Qualified / Won / Lost). Webhook POST /api/leads/webhook for inbound from GHL. |
| <b>Alerts</b>         | <b>alerts</b>   | Unread notifications. Badge counter shows pending. Click an alert to mark read.                                                                                                                                                                                                         |

# POC follow-up sequence engine

The sequence engine is the most operationally important part of the command center. It controls when GHL tags get pushed to the verification POC + invoice POC, drives the email/SMS reminder cadence via GHL workflows, and escalates to Tatenda on Day 10.

## Two cadences

**Verification:** Day 1, 4, 7, 10 — fires when Tatenda flips a verification to 'ready' in Slack `#ai-verification-report-push`.

**Invoice:** Day 1, 3, 6, 9, 10 — fires when the client clicks 'Approve & Generate Invoice' on a verification.

## Day → Tag map

| Day    | Verification tag                     | Recipient        |
|--------|--------------------------------------|------------------|
| Day 1  | <Prefix> Verification                | Verification POC |
| Day 4  | <Prefix> Verification Follow Up      | Verification POC |
| Day 7  | <Prefix> Verification Last Follow Up | Verification POC |
| Day 10 | <Prefix> Verification Escalation     | <b>Tatenda</b>   |

  

| Day    | Invoice tag                          | Recipient      |
|--------|--------------------------------------|----------------|
| Day 1  | <Prefix> Invoice                     | Invoice POC    |
| Day 3  | <Prefix> Invoice Follow Up           | Invoice POC    |
| Day 6  | <Prefix> Invoice Last Follow Up      | Invoice POC    |
| Day 9  | <Prefix> Invoice Past Due            | Invoice POC    |
| Day 10 | <Prefix> Invoice Past Due Escalation | <b>Tatenda</b> |

<Prefix> is the client's tag prefix from `client-config.json` (Raffiti's is 'Raffiti').

## Replace semantics

On every advance, we **remove every other day-tag for that sequence** before adding the new one. So when Day 4 fires, the Day 1 tag is stripped from the GHL contact. This means GHL workflows can be keyed to the tag itself without worrying about state — at any moment a contact has exactly ZERO or ONE tag for that sequence.

## Escalation routing

Day 10 (the final escalation step for both kinds) routes to **Tatenda**, not the client POC:

- Email: `tatenda@thesaleslever.com`

- Name: Tatenda
- Phone: none
- Extra tags added: Source: Command Center + Escalation Recipient
- The client contact still gets its prior day-tag stripped (clean state)
- Tatenda's other tags are NEVER touched (she may have escalations from other clients)

If Tatenda ever changes, it's a single constant at the top of `aa_server.py` — `ESCALATION_POC`.

## How sequences start / stop

**Start verification:** Tatenda posts 'verification ready' (or similar) in Slack `#ai-verification-report-push`. Cron `8a13f045` (active days 25–31) reads the Slack channel, rebuilds `verification-data.json` from the sheet, deploys, then calls `POST /api/sequences/start-verification`. Auto-stops any prior active verification, pushes Day 1 tag.

**Stop verification + start invoice:** Client clicks Approve & Generate Invoice in the Verification Reports tab. Backend calls `stop_sequence('verification', 'approved')`, then `start_sequence('invoice', ...)`, then pushes Day 1 invoice tag to the invoice POC.

**Stop invoice:** Stripe sends `invoice.paid` webhook → backend calls `stop_sequence('invoice', 'paid')` and strips any lingering tags from the client contact.

# Scheduled crons

Three recurring tasks run on Perplexity Computer's cron scheduler. Each runs in its own sandbox session and uses the same workspace files.

| ID       | Schedule (UTC) | What it does                                                                                                                                                                                                                                                                                                                                  |
|----------|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8a13f045 | 7 * 25-31 *    | <b>Verification report watcher.</b> Active days 25–31 each month. Reads Slack #ai-verification-report-push (channel C0B5EJT57L2) for 'verification ready' messages. If found: rebuilds verification-data.json from the Google Sheet, deploys, calls /api/sequences/start-verification, sends in-app notification. Dedup by last_processed_ts. |
| f5031b1a | 0 13 * * 1     | <b>Weekly recap rebuild.</b> Mondays 7am MT. Pulls the 6 R/R sheet tabs into raw-sheet-snapshot.json, calls /api/recaps/build (which runs Cipher on weekly/monthly/YTD windows), redeploys, sends notification.                                                                                                                               |
| faca04c3 | 0 14 * * *     | <b>Daily sequence advance.</b> Every day 8am MT. Calls /api/sequences/advance which walks all active sequences, finds any whose day has elapsed, pushes the new day-tag with REPLACE semantics. Sends notification if anything was pushed.                                                                                                    |

## Failure handling

- Each cron sends an in-app notification on failure with the step that broke.
- Crons never retry within the same run — if a deploy fails, the next scheduled run will retry from scratch.
- Slack messages are deduped by ts (timestamp) in cron\_tracking/<cron\_id>/last\_processed\_ts.txt so a message processed once never fires twice.

# Backend API endpoints

All endpoints are served by `aa_server.py` (FastAPI). Production:  
<https://raffiti.the-sales-lever.com/api/<path>>.

| Method + path                          | Purpose                                                                                                                            |
|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| GET /api/health                        | Liveness probe.                                                                                                                    |
| GET /api/stripe/ping                   | Verifies Stripe API key + customer can be looked up.                                                                               |
| POST /api/chat                         | Cipher chat endpoint. Body: { messages, mode: 'client'   'team', context }. Streams Claude Sonnet 4.6 with tool-calling.           |
| POST /api/cipher-grade-call            | Send a call transcript → Cipher returns a structured grade (rapport, discovery, objection handling, close).                        |
| GET /api/invoices                      | Returns invoices-data.json contents.                                                                                               |
| POST /api/invoices/from-verification   | Generate a Stripe invoice from a verification row. Stops verification sequence, starts invoice sequence, pushes Day 1 invoice tag. |
| POST /api/stripe/webhook               | Stripe webhook receiver. Handles invoice.paid → stops invoice sequence, strips tags. Verified via STRIPE_WEBHOOK_SECRET.           |
| POST /api/recordings/sync              | Manually pull latest recordings from tldv.                                                                                         |
| GET /api/recordings                    | Returns call-recordings-data.json.                                                                                                 |
| POST /api/recordings/webhook           | tldv webhook receiver — fires on new recording.                                                                                    |
| GET /api/recordings/{id}/media         | Proxies the recording media stream.                                                                                                |
| GET /api/recaps                        | Returns recaps-data.json.                                                                                                          |
| POST /api/recaps/build                 | Synchronously rebuilds all recaps from raw-sheet-snapshot.json. Takes ~30s. Used by the weekly cron.                               |
| GET /api/sequences/current             | Returns the active verification + invoice sequences for the tracker UI.                                                            |
| POST /api/sequences/start-verification | Starts a verification sequence. Body: { period }. Used by cron 8a13f045.                                                           |
| POST /api/sequences/advance            | Walks all active sequences, pushes any due day-tags. Idempotent. Used by cron faca04c3.                                            |
| POST /api/sequences/stop               | Manually stop a sequence. Body: { kind, reason }.                                                                                  |
| POST /api/leads/preview                | OCR a lead screenshot. Returns parsed fields without saving.                                                                       |

|                            |                                                                                    |
|----------------------------|------------------------------------------------------------------------------------|
| POST /api/leads/submit     | Submit a parsed lead → pushes to GHL + adds to leads-data.json kanban.             |
| GET /api/leads             | Returns leads-data.json.                                                           |
| POST /api/leads/webhook    | GHL webhook receiver — for leads created in GHL flowing back to the kanban.        |
| POST /api/leads/{id}/stage | Move a lead to a different kanban column.                                          |
| GET /api/recap-watcher     | Returns recap-watcher-data.json (which weekly/monthly windows are pending review). |

# Cipher — the in-app AI agent

Cipher is the floating green chat dock in the bottom-right. Powered by Claude Sonnet 4.6 via the LLM API. Two modes: **client mode** (default — what the client sees) and **team mode** (gated by Tatenda password — exposes admin tools).

## Voice rules

- **No exclamation points.** Ever.
- **No emojis.**
- Sign-off: — Decoded by Cipher
- Direct, calm, slightly mysterious. Never marketing-speak.

## Client tools (always available)

| Tool                   | Purpose                                           |
|------------------------|---------------------------------------------------|
| push_internal_lead     | Push a new lead to GHL + leads-data.json kanban.  |
| lookup_lead_status     | Look up a lead's current stage + GHL info.        |
| read_recap_data        | Read the current recap data (weekly/monthly/YTD). |
| read_verification_data | Read the active verification report.              |
| get_pocs               | Return verification + invoice POC info.           |

## Team tools (Tatenda password required)

| Tool                  | Purpose                                                                     |
|-----------------------|-----------------------------------------------------------------------------|
| create_ghl_task       | Create a GHL task on a contact.                                             |
| add_ghl_note          | Append a note to a GHL contact.                                             |
| move_pipeline_stage   | Move a GHL opportunity to a different pipeline stage.                       |
| create_custom_invoice | Bypass verification flow — create a one-off Stripe invoice for any amount.  |
| set_verification_poc  | Update the verification POC name/email/phone (saved to client-config.json). |
| set_invoice_poc       | Update the invoice POC. Independent of the verification POC.                |

# Data model — what lives where

Source of truth is the client's Google Sheet (Raffiti's: 1weS-GPvUBAXkPBoTfIh3RD2XrcxqZx6jzT1THpoxK1Q). Crons pull from it and cache locally. The UI reads only the local cache files — never the Sheet directly.

| File                      | Contents                                                                                                                                         |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| client-config.json        | Client metadata: name, brand colors, prefix, verification POC, invoice POC, GHL location id, Stripe customer id, ghlReminders counters.          |
| raw-sheet-snapshot.json   | Last pull from the 6 R/R Google Sheet tabs (Conductor, SDR, Committer, Commissions, Refunds_Churn, Follow Up). Used as input for recap rebuilds. |
| recaps-data.json          | Output of recap builds — weekly + monthly + YTD narratives + numbers, each with generated_at timestamp.                                          |
| verification-data.json    | Current month's verification report — period, status, newSales[], recurring[]. Rebuilt by cron 8a13f045.                                         |
| invoices-data.json        | All invoices for this client — number, amount, customer, status, Stripe id, dates, notes.                                                        |
| call-recordings-data.json | Synced tldv recordings. Each has id, title, date, attendees, duration, video url, transcript url.                                                |
| leads-data.json           | Internal Leads kanban — lead objects with stage, GHL contact id, source, created_at.                                                             |
| performance-data.json     | Live KPIs for the Performance Panel — MRR, conductor rate, SDR rate, etc. Refreshed from sheet on demand.                                        |
| churn-data.json           | Refunds and churn detail.                                                                                                                        |
| active-sequences.json     | Active + completed sequence state. The brain of the follow-up engine.                                                                            |
| recap-watcher-data.json   | Which weekly/monthly windows are awaiting review.                                                                                                |

## client-config.json shape

```
{ "client": "Raffiti Media", "tagPrefix": "Raffiti", "brand": { "primary": "#A4C436", "logo": "assets/raffiti-logo-3d.png" }, "ghl": { "locationId": "...", "pipelineId": "..." }, "stripe": { "customerId": "cus_...", "defaultPriceId": "price_..." }, "pocs": { "verification": { "name": "...", "email": "...", "phone": "..." }, "invoice": { "name": "...", "email": "...", "phone": "..." } }, "ghlReminders": { "verification": 0, "invoice": 0 } }
```

# Environment variables

---

Set on Railway. Never check into git.

| Variable              | Purpose                                                                                    |
|-----------------------|--------------------------------------------------------------------------------------------|
| LLM_API_KEY           | Perplexity LLM API token. Powers Cipher (claude_sonnet_4_6) + recap generation + lead OCR. |
| STRIPE_API_KEY        | Stripe secret key. Used to create + read invoices.                                         |
| STRIPE_WEBHOOK_SECRET | Stripe webhook signing secret (whsec_...). Required to validate incoming Stripe webhooks.  |
| GHL_API_BASE          | GoHighLevel API base URL.                                                                  |
| GHL_API_VERSION       | GoHighLevel API version header.                                                            |
| GHL_API_KEY           | GoHighLevel API key (Bearer token).                                                        |
| GHL_LOCATION_ID       | GHL location/sub-account id for tag + contact operations.                                  |
| TATENDA_PW            | Password gating team-mode Cipher tools. Default: 'sales@'.                                 |
| TLDV_API_KEY          | tldv API key for call recording sync (optional).                                           |

# End-to-end workflows

---

## Workflow A — Month-end verification → payment

- **Day 25-31:** Tatenda completes the verification rows in the Google Sheet, then posts 'verification ready' in Slack #ai-verification-report-push.
- Within the next hour, cron 8a13f045 fires. It reads Slack, finds the message, rebuilds verification-data.json from the sheet, deploys, calls /api/sequences/start-verification, and sends the client an in-app notification.
- Day 1 tag (<Prefix> Verification) is pushed to the verification POC in GHL. GHL workflow 'Raffiti Verification' fires — sends the first 'please review' email/SMS.
- Client opens the command center, lands on Performance Panel, sees Cipher's notification or Verification Reports tab highlighted.
- Client reviews newSales[] + recurring[] in the Verification Reports tab.
- Client clicks **Approve & Generate Invoice**. Backend stops verification sequence, creates the Stripe invoice, starts invoice sequence, pushes Day 1 invoice tag.
- GHL workflow 'Raffiti Invoice' fires — sends the invoice email with Melio Pay button.
- **Days 3, 6, 9:** If still unpaid, daily cron faca04c3 pushes Day 3, Day 6, Day 9 tags in sequence — Follow Up → Last Follow Up → Past Due. Each pushes its own GHL workflow.
- **Day 10:** Cron pushes 'Past Due Escalation' tag to **Tatenda**, not the client. Tatenda's GHL workflow alerts her to step in.
- When client pays, Stripe sends invoice.paid webhook → backend stops invoice sequence + strips all lingering invoice tags from the client contact. Done.

## Workflow B — Internal lead intake (client side)

- Client opens Internal Leads tab.
- Pastes a screenshot of a lead profile (LinkedIn, Instagram, anywhere).
- Frontend uploads to /api/leads/preview. Cipher OCRs the image, extracts name + email + phone + notes, returns a preview card.
- Client confirms or edits → submit. Frontend POSTs /api/leads/submit.
- Backend creates a GHL contact, tags it with 'Internal Lead', adds it to leads-data.json under the 'New' kanban column.
- Lead now appears on the live kanban. Drag to update stage (Contacted / Qualified / Won / Lost).

## Workflow C — Weekly recap refresh

- Monday 7am MT: cron f5031b1a fires.
- Pulls 6 R/R tabs from the Google Sheet → raw-sheet-snapshot.json.
- Calls POST /api/recaps/build. Backend reads the snapshot, runs Cipher on each window (weekly / monthly / YTD), writes recaps-data.json.

- Redeploys the site.
- Sends in-app notification: 'Weekly recap is ready'.

# Spinning up a new client

---

To stand up a new client command center, use the **client-command-center** user skill. The skill points to `/home/user/workspace/raffiti-command-center/` as the canonical reference build. Bootstrap copies that directory, swaps the client-specific assets (name, prefix, brand color, logo, POCs, Stripe + GHL ids), and deploys to a fresh subdomain.

## Manual checklist if you're doing it by hand

- Copy `raffiti-command-center/` → `<newclient>-command-center/`.
- Edit `client-config.json`: client name, tagPrefix, brand.primary, brand.logo path, GHL location id, Stripe customer id, both POCs.
- Generate a 3D logo for them via `asi-generate-image` (use `raffiti-logo-3d.png` as the style reference, swap shape/color to match their logo).
- Save the new logo as `assets/<client>-logo-3d.png` and update `galaxy.js` to load it.
- Run their first sheet sync: paste their sheet id into the `recap_engine` helpers + run the manual recap build endpoint.
- Deploy with `deploy_website` + Railway. Bind the subdomain in Cloudflare.
- Set up the three crons for them (clone the Raffiti crons, swap sheet id + Slack channel + asset\_id).
- Build their GHL workflows: one workflow per day-tag (Verification, Verification Follow Up, etc).
- Test the loop with a fake verification on a test contact before flipping live.

# UI design rules (locked)

---

## Color

- Sales Lever brand is **GREEN**. 'Golden' means green (#A4C436), never yellow/gold.
- Primary: #A4C436 (Raffiti lime). Brighter: #C5E04F. Pale highlight: #E8FF7A.
- Backgrounds: near-black with slight green tint. Never pure #000.

## Mobile vs Desktop

- Mobile breakpoint: max-width 820px.
- On mobile, the sidebar collapses to a horizontal icon strip at the top. The onboarding zone box is hidden (display: contents), START HERE pill banner hidden, children flow inline.
- Discipline: **fix one viewport at a time**. Verify before touching the other. Mobile and desktop fixes have historically broken each other.

## Galaxy centerpiece

- The client's 3D logo is the centerpiece (size 3.0 in three.js scene). Bigger than the original PNG.
- Halos sized 1.35 → 3.0 sphere radii layered with increasing opacity inward.
- 5 tilted elliptical orbits with money-sign planet textures. Each orbit precesses slowly through 3D space.
- Logo PNG must have transparent alpha background. Square canvas (object centered with padding).

## Cipher voice

- No exclamation points. No emojis. Sign-off '— Decoded by Cipher'.
- Mode auto-switches based on Tatenda password unlock.

# Gotchas + tribal knowledge

---

- **Tag replace semantics live in `_push_sequence_step`.** If you change `INVOICE_STEPS` or `VERIFICATION_STEPS` in `sequences.py` without thinking about cleanup, you'll leave orphan tags on contacts. The `all_tags_for()` helper is what does the cleanup — make sure new steps make it into that list.
- **Sequence cron is idempotent.** Calling `/api/sequences/advance` twice the same day is a no-op. Don't add retry logic — the next day's run picks up anything that genuinely failed.
- **Day counting is by date, not 24-hour windows.** A sequence started at 11:59pm UTC is on Day 2 one minute later. This is intentional — keeps GHIL workflows aligned with calendar days.
- **Escalation pushes do NOT update `seq.ghlContactId`.** The cached contact id stays pointed at the client so tag cleanup keeps working. Tatenda's contact id is stored in a separate field (`escalationGhlContactId`).
- **Slack watcher dedup is by ts.** If you delete `cron_tracking/8a13f045/last_processed_ts.txt`, the next run will re-process every matching message in the last 90 minutes.
- **media-cache must be moved out of the project before `deploy_website`.** It's huge (tldv video cache). The pre-deploy ritual stashes it to `/tmp` and restores it after. If you forget, the deploy tarball is gigabytes.
- **`__pycache__` must be deleted before deploy.** Same reason.
- **The Perplexity `pplx` asset and the Railway production host are different deployments.** `deploy_website` refreshes the `pplx` mirror. Railway redeploys automatically on git push to main.
- **If Tatenda's email changes, update `ESCALATION_POC` at the top of `aa_server.py`.** One spot. Both verification + invoice escalations route through it.
- **Sales Lever brand is GREEN.** If you see yellow/gold anywhere in the UI, that's a bug.

# Quick reference card

---

## Verification sequence schedule

Day 1 → <Prefix> Verification → POC Day 4 → <Prefix> Verification Follow Up → POC Day 7 → <Prefix> Verification Last Follow Up → POC Day 10 → <Prefix> Verification Escalation → Tatenda

## Invoice sequence schedule

Day 1 → <Prefix> Invoice → POC Day 3 → <Prefix> Invoice Follow Up → POC Day 6 → <Prefix> Invoice Last Follow Up → POC Day 9 → <Prefix> Invoice Past Due → POC Day 10 → <Prefix> Invoice Past Due Escalation → Tatenda

## Cron schedule (UTC)

8a13f045 Verification watcher 7 \* 25-31 \* \* f5031b1a Weekly recap rebuild 0 13 \* \* 1 (Mon 7am MT)  
faca04c3 Sequence advance 0 14 \* \* \* (Daily 8am MT)

## Endpoints to know by heart

POST /api/sequences/start-verification # manual fire POST /api/sequences/advance # walk + push due  
POST /api/sequences/stop # emergency stop GET /api/sequences/current # tracker state POST  
/api/recaps/build # rebuild recaps POST /api/invoices/from-verification # generate invoice

## Production URL

<https://raffiti.the-sales-lever.com>

Locked at: baseline-2026-05-22 · commit cd57e25