

CASCADE

Operator Guide

A confirmatory-test allocation and cascade-completion engine for tuberculosis. Everything needed to stand it up and run it, in order.

Read this before anything else. This is a pre-calibration evaluation release. The software is built and its security model is tested against an adversarial suite; the clinical model is built from established tuberculosis risk factors and awaits refitting on Nigerian data, which is Phase 0.

What that does and does not limit. Calibration affects the probability values, not necessarily the ordering — a model built from known risk factors ranks people usefully before its coefficients are fitted locally. CASCADE's output is an ordering, so the interface shows a priority rank and withholds the percentage until Phase 0 justifies it. What cannot be claimed yet is any specific probability, any sensitivity or specificity figure, or any statement about subgroup performance. Use within an approved evaluation protocol.

Contents

Section	Section
1 What you need first	7 Load your real facilities
2 Set up the database	8 Day-to-day use, by role
3 Create the storage bucket	9 Wire CASCADE into PulmIQ
4 Deploy to Netlify	10 Test your security
5 Environment variables	11 When things break
6 Create your first users	12 Before any pilot

Why this product exists

Nigeria notified 458,534 tuberculosis cases in 2025 against an estimated incidence of roughly 510,000. The detection gap that dominated the last decade has narrowed sharply. What has not improved is the economics — domestic financing has historically covered 6 to 8 per cent of national programme need, and international support contracted in 2025.

So the binding constraint is no longer how many people are screened. It is **which of the people screened receive a confirmatory molecular test**, given a finite number of cartridges, instrument-hours and transport runs. CASCADE answers that question per facility, per day, and then tracks every authorised test through to treatment or a documented reason for loss.

Its headline measure is **number needed to test**: how many presumptive tests it takes to confirm one case.

Number needed to test is gameable, and you must know this before you report it. It improves if you test fewer, higher-probability people — so a facility optimising its dashboard could raise its score while finding fewer cases, which is the opposite of what the programme wants. Never report it without absolute case-finding alongside.

1 What you need first

- A **Supabase project**, in a Nigerian or nearest-region data centre — the Nigeria Data Protection Act 2023 expects data residency
- A **Netlify account** connected to the repository
- **Node 20 or later**, if you are building locally
- An **Anthropic API key** — optional. The AI features are assistive; everything works without one

You do not need a CAD vendor, a telehealth account or a messaging provider to get running. Add those later.

2 Set up the database

In the Supabase SQL editor, run the three files **in this order**:

```
db/cascade_schema.sql    tables, enums, views, seed facilities
db/cascade_rls.sql       roles and row level security
db/cascade_storage.sql   bucket policies (after step 3)
```

All three are idempotent. Re-running is how you apply a change, so re-running must never error — if it does, that is a bug worth reporting.

Order matters. `cascade_rls.sql` references tables created by the schema, and `cascade_storage.sql` references helper functions created by the RLS file. Running them out of order produces confusing “does not exist” errors that look like corruption and are not.

3 Create the storage bucket

Storage → New bucket, named exactly `cascade-documents`, and **not public**. Then run `db/cascade_storage.sql`, which re-asserts the private flag, sets the 12 MB size limit and the allowed file types, and installs the facility-scoping policies.

Never make this bucket public. It holds chest X-rays and laboratory results belonging to identifiable people in small communities, where an age band and an LGA together come close to naming someone. The SQL re-asserts `public = false` on every run precisely so that an accidental toggle in the dashboard is undone the next time you deploy.

4 Deploy to Netlify

`netlify.toml` is already configured — build command, publish directory, functions folder and redirects. Connect the repository and Netlify picks it all up. The landing page is served at `/` and the console at `/app/`.

Every file must sit at the repository root. Both build entry points (`index.html` and `app.html`), `netlify.toml` and `netlify/` live at the top level. The console is served at the `/app/` URL by a redirect, not by living in an `app/` folder — a one-file subdirectory is exactly the thing that goes missing in a zip upload, and its absence fails the build with `Could not resolve entry module`.

Do not set `BASE_PATH`. That variable exists only for GitHub Pages project sites. Netlify serves from the domain root, and setting it will break every asset path on the site.

5 Environment variables

Site configuration → Environment variables. Nothing here is optional except where marked.

Variable	Value	Notes
NODE_VERSION	20	
VITE_SUPABASE_URL	Project URL	Public by design
VITE_SUPABASE_ANON_KEY	Anon key	Public by design — RLS is what protects the data
SUPABASE_URL	Same URL	Server side
SUPABASE_ANON_KEY	Same anon key	Used to verify caller tokens
SUPABASE_SERVICE_ROLE_KEY	Service role key	Never with a VITE_ prefix
ALLOWED_ORIGIN	Your site URL	
ANTHROPIC_API_KEY	Your key	Optional — enables AI intake and briefing
VITE_AI_PROXY_URL	/api/ai	Optional, with the above
ENGINE_SHARED_SECRET	Mesh secret	Needed for the PulmIQ bridge

The VITE_ prefix means “bake this into the browser bundle.” Anything carrying it is readable by anyone who loads the page. That is correct for the project URL and the anon key, and catastrophic for the service role key. There is one Anthropic key, not two — ANTHROPIC_API_KEY and VITE_ANTHROPIC_API_KEY hold the same value, and only the first is safe in a deployment anyone else can reach.

Redeploy after adding variables. Vite reads VITE_* at build time, not runtime. A variable added after a build will not appear until the next one — which looks exactly like a wrong key and wastes an afternoon.

6 Create your first users

Signing in is not enough. A user with no app_user row sees an explicit “no role assigned” screen and nothing else — which is the correct default, not a fault. Create the account in Supabase Auth, then insert the role. The first must be an admin:

```
insert into app_user (id, role, display_name)
values ('<auth-user-uuid>', 'admin', 'Programme admin');
```

Then the facility-scoped staff:

```
insert into app_user (id, role, facility_id, display_name)
values ('<uuid>', 'chew',
      (select id from facility where code = 'LA-PHC-001'),
      'Adebayo O.');
```

Role	Scope	Cannot see
chew	One facility	Any other facility
facility_officer	One facility, plus cartridge allocation and closing losses	Any other facility
lab_tech	One testing site	The assessment — no triage answers, no HIV status, no risk factors
dots_officer	One treatment centre	The assessment
state_officer	One state, read only	Assessments and individual documents; cannot change anything
admin	Everything	—

A facility-scoped role saved without a facility would silently see nothing, so the database refuses the row outright rather than creating a user who quietly cannot work.

7 Load your real facilities

The schema seeds six placeholder facilities across Lagos and Kano so the system is usable immediately. Replace them with your own before anything real happens:

```
insert into facility
  (code, name, state, lga, tier, has_genexpert, has_xray, accepts_swabs)
values
  ('LA-PHC-014', 'Ikorodu PHC', 'Lagos', 'Ikorodu', 'phc', false, false, true)
on conflict (code) do nothing;
```

tier is one of phc, general, tertiary, lab_hub, dots_centre. Get has_genexpert right — the referral logic depends on it.

8 Day-to-day use, by role

Community health worker

1. Open the **triage channel**. Work through it with the person, one question per screen.
2. Read them the outcome card. It gives one action, what it costs, and what to bring.
3. If they are authorised for a test, collect the specimen and mark it in the **tracker**.
4. When a result slip comes back, record it as a document. It advances the cascade automatically.

A red flag — coughing blood, breathless at rest, chest pain, too unwell to eat or drink — stops the interview immediately and sends the person for same-day care. Above the testing line it adds a clinician as *well* as the specimen; below the line it overrides the deferral entirely. The queue optimises yield, and it does not get to overrule clinical urgency.

Facility officer

1. Set today's cartridge count in the **console**. The authorisation line moves and the queue reorders.
2. Check the overdue count in the tracker each morning — anything past its window needs a human today.
3. Close anything genuinely lost, **with the real reason**.

The loss reason is the most valuable field in the system. “Lost to follow-up” tells a programme manager nothing they can act on. “Could not afford transport” tells them to fund transport. Most reasons on that list name something a programme can actually pay for — which is exactly why it is a fixed list and not a free-text box.

Laboratory

Work from the lab worklist: what arrived, under which voucher, what to test. Enter the result including the semi-quantitative grade and the rifampicin outcome. An invalid or error result does not close anything — a cartridge was consumed and the person still needs a test, so the tracker asks for a re-collection.

DOTS officer

Work from the DOTS worklist. Enrol confirmed cases the same day, screen the household, and start daily check-ins. Upload liver function tests during treatment; the system flags transaminases at roughly five times the upper limit of normal, or three times with symptoms, **for a clinician to review**. It never tells anyone to stop the drugs.

State programme officer

Read-only dashboards across your state: number needed to test, where the cascade leaks and why, time to treatment, and what is overdue. You cannot see individual triage answers or clinical documents, by design — a programme is run from counts, not from reading someone's chest X-ray.

9 Wire CASCADE into PulmiQ

CASCADE reaches the DoktorConnect app only through PulmiQ — it is deliberately not on Core's roster.

```
CASCADE -> PulmiQ -> Core -> app
```

Copy `integrations/pulmiq/cascadeBridge.js` into the `pulmiq` project's `functions/` folder, set `CASCADE_ENGINE_URL` and `ENGINE_SHARED_SECRET`, and add one line to the existing export handler:

```
const { withCascade } = require('./cascadeBridge');
payload = await withCascade(payload, { dcid, pseudonym });
```

Four fields cross: whether an episode is open, when the next contact is due, the handoff, and whether the person has opted out. **TB status, regimen, drug resistance, the risk score, the recorded risk factors, HIV status and the callback number do not.** A tuberculosis diagnosis costs people housing and work, and the far end of that pipe is a consumer app a family may share a phone to use.

That file is CommonJS, because Firebase functions are. It will not load inside a Vite repository whose `package.json` has `"type": "module"` — you get `module is not defined in ES module scope`, which surfaces as a 502 with no other clue.

10 Test your security

Against a scratch database, never production:

```
createdb cascade_test
psql -d cascade_test -f db/cascade_schema.sql
psql -d cascade_test -f db/cascade_rls.sql
psql -d cascade_test -f db/rls_test_seed.sql
bash db/rls_test.sh
bash db/storage_test.sh
```

The suites create a Lagos CHEW, a Kano CHEW, a lab, a DOTS officer, a state officer and an admin, then try to break the boundaries: cross-facility reads and writes, audit-trail tampering, path traversal, unauthenticated access, and pseudonym enumeration.

Run this after **any** policy change. Two real bugs have already been caught this way, and both looked like working security: one made every user see nothing at all, and the other made the append-only audit trail impossible to append to. Neither would have shown up in a build or a syntax check.

11 When things break

Symptom	Cause	Fix
type "cascade_state" already exists	An older, non-idempotent copy of the schema	Use the current cascade_schema.sql; it guards every enum
permission denied for schema app	Grants did not run — usually the RLS file was truncated or run out of order	Re-run cascade_rls.sql whole
permission denied for sequence	Missing USAGE on a bigserial sequence	Re-run cascade_rls.sql; the grant is in it
A save reports success but nothing changed	RLS refused it. A denied UPDATE returns zero rows, not an error	Every write must go through writeGuarded(). Never call .update() directly
AI panels say "AI off" with a key set	Vite reads VITE_* at build time	Redeploy
Function returns 502	CommonJS .js in an ESM repo, or a call over the 10-second ceiling	Check module syntax first, then look for a slow upstream call
Uploads fail on a large X-ray	Bytes routed through a function	They must not be — the browser PUTs straight to storage from a signed URL
Site-wide 404, no functions run	Files nested one folder deep in the repository	index.html, netlify.toml and netlify/ must sit at the repository root
Could not resolve entry module "app/index.html"	A subdirectory did not survive the upload to GitHub	Already fixed — the console entry is app.html at the root. Seeing this means an older copy

12 Before any pilot

None of these is optional, and none is currently done.

- **Field test with CHEWS.** Every design rule in the triage channel — danger questions first, one question per screen, “not sure” as a real answer, a skippable HIV question — is a plausible hypothesis tested by nobody. This outranks everything else on the roadmap.
- **Translate.** Hausa, Yoruba, Igbo and Nigerian Pidgin, by clinicians, with back-translation. The question wording is the instrument; a bad translation is a bad test, and machine translation is not adequate for “have your clothes become loose?”
- **Phase 0 calibration.** Until the model is refitted on Nigerian register and GxAlert data, every probability on every screen is illustrative.
- **Replace the handoff log sink** with a real datastore. A lost handoff is a lost patient, and logs rotate.
- **Set a retention period** with NTBLCP. NDPA 2023 expects one, and tuberculosis has a long clinical tail — outcome at six months, relapse for years. This is not a developer's decision.
- **Engage NAFDAC.** A tool that allocates diagnostic tests and issues clinical dispositions is software as a medical device. Seek a pre-submission meeting before the Phase 1 protocol is finalised, not after.
- **Check policy performance** at a realistic row count. `app.sees_pseudonym()` runs on every storage access and contains two joins.

Nigeria only. CASCADE must never surface in the United States — see SCOPE.md. Obligations are the Nigeria Data Protection Act 2023, NAFDAC, NTBLCP and the WHO target product profile for a tuberculosis triage test.

Every figure this system produces comes from a provisional, uncalibrated model. It exists to make one mechanism concrete: that allocation order, not screening volume, is the lever on cases confirmed per cartridge consumed. Whether it actually moves the number needed to test is the primary endpoint of a trial that has not been run.

Evidence for every factual claim: docs/EVIDENCE.md. Adversarial review, including the three objections that would kill the project: docs/RED_TEAM.md. Neither is optional reading before a partner meeting.