MySentinel

Due-diligence pack

For the buyer's technical advisor

100% Cloudflare | POPIA-aligned | en / af / zu | ~R10 / learner / month

MySentinel — NFC-badge learner check-in & school-safety for South African schools. · Chain-owner doc suite · 2026-06-27

How it works for a school group (architecture & tenancy)

This page is written for the buyer's technical advisor. It explains how MySentinel is built, where the trust boundary sits, and — most importantly — exactly how one school's data is kept separate from another's. We have written the tenancy section to be checkable: nothing here is gloss. Where a capability is shipped-and-proven we say so; where it is planned we mark it **Roadmap**.

The short version: MySentinel is a set of small Cloudflare Workers, each owning its own database, sitting behind a single gateway that is the only thing the browser talks to. Tenant isolation today is **logical and code-enforced** — every school-scoped query is filtered by a `school_id` that comes only from the verified login token, never from anything the browser can set. That isolation runs on a single live database shard. The system is built to add more shards without a rewrite, but it is **not physically sharded today**.

The platform: Cloudflare, end to end

MySentinel runs entirely on Cloudflare — Workers (serverless compute), D1 (SQLite databases), KV, R2 (object storage), Queues, Durable Objects, Cron Triggers, and Static Assets. There are no servers to run and no second cloud. This is a deliberate, load-bearing decision (ADR-001): one billing and operations surface, a global low-latency edge, and no local infrastructure for a South African pilot to maintain. The trade-off we accept is living within Cloudflare's primitives (D1 size/SQLite semantics, Workers CPU limits) as design constraints.

Capability services, not a monolith

The backend is roughly 24 small Workers ("capability services"), each doing one job — `identity-service` (learners, badges, guardians), `movement-service` (validate and record check in/out), `tenant-service` (schools, settings, memberships), `notification-service` (delivery fan-out), `auth-service` (login, passkeys, sessions), `operator-service` (onboarding), and so on. Each service:

- **Owns its own D1 database.** No service reads another service's database directly (ADR-003). If service B needs data service A owns, it calls A's typed RPC over a Cloudflare service binding — never a shared table.
- **Ships a typed contract.** Request/response shapes live in `packages/rpc-types` and are shared by both the services and the PWA, so a breaking change surfaces at compile time rather than in production (ADR-014).

The reason for this shape is documented in the project philosophy: smaller services are safer to change, with a smaller blast radius and explicit contracts at every boundary.

The gateway is the only door

The browser (the SvelteKit PWA used by officers, admins, class teachers, and operators) talks to **exactly one** thing: the `gateway` Worker. Every backend service is internal — reachable only via service bindings, never directly from the public internet (ADR-005). The gateway owns:

- **CORS and routing.**

- **Authentication** — it verifies the JWT on every request.
- **RBAC** — every route declares which roles and scope may call it (`requireAuth(roles, scope)`). An officer hitting an `/admin` route, or an admin hitting an `/operator` route, is bounced. This is shipped and verified.
- **Tenant context propagation** — it injects the trusted `X-School-Id` and actor headers downstream from the verified token.
- **Step-up** — sensitive admin actions (safety lockdown, POPIA erasure, school/provider config, staff credential changes) require a WebAuthn passkey assertion. Route completeness here is enforced by a test, not a hand-kept list (ADR-025), so a new mutating route cannot ship ungated by omission.

Because the gateway is the single trust boundary, there is exactly one place to reason about auth, tenancy, and CORS.

☰

PLATFORM OPERATOR

PLATFORM OPERATIONS

Operator Console

Live · updated 08:49:53

Refresh | Onboard school | Health console

Platform access

Active

Schools in platform

1

Messaging setup

Ready

App notifications and email are the live channels. WhatsApp and SMS are in preview and switch on per school once a provider is connected.

Needs you now

ALL CLEAR

LOCAL

Nothing needs you right now

No failed deliveries, stalled imports, or channel issues across the platform.

Platform health

OK

Current snapshot

Last 1 hour delivery success

100%

Failed deliveries needing review

0

Failed logins in last 1 hour

0

Open health console

Generated 2026/06/27, 08:49:53

School onboarding

Manage schools

Schools

Demo Primary School

Africa/Johannesburg - Created 2026/05/01, 02:00:00

Settings | ACTIVE

Showing the most recent schools. Use "Manage schools" to search and page through all 1.

Provider readiness

Ready

Notification channels

WhatsApp READY

Messages can be sent through this channel.

Last checked 2026/06/27, 08:49:53

Email READY

Messages can be sent through this channel.

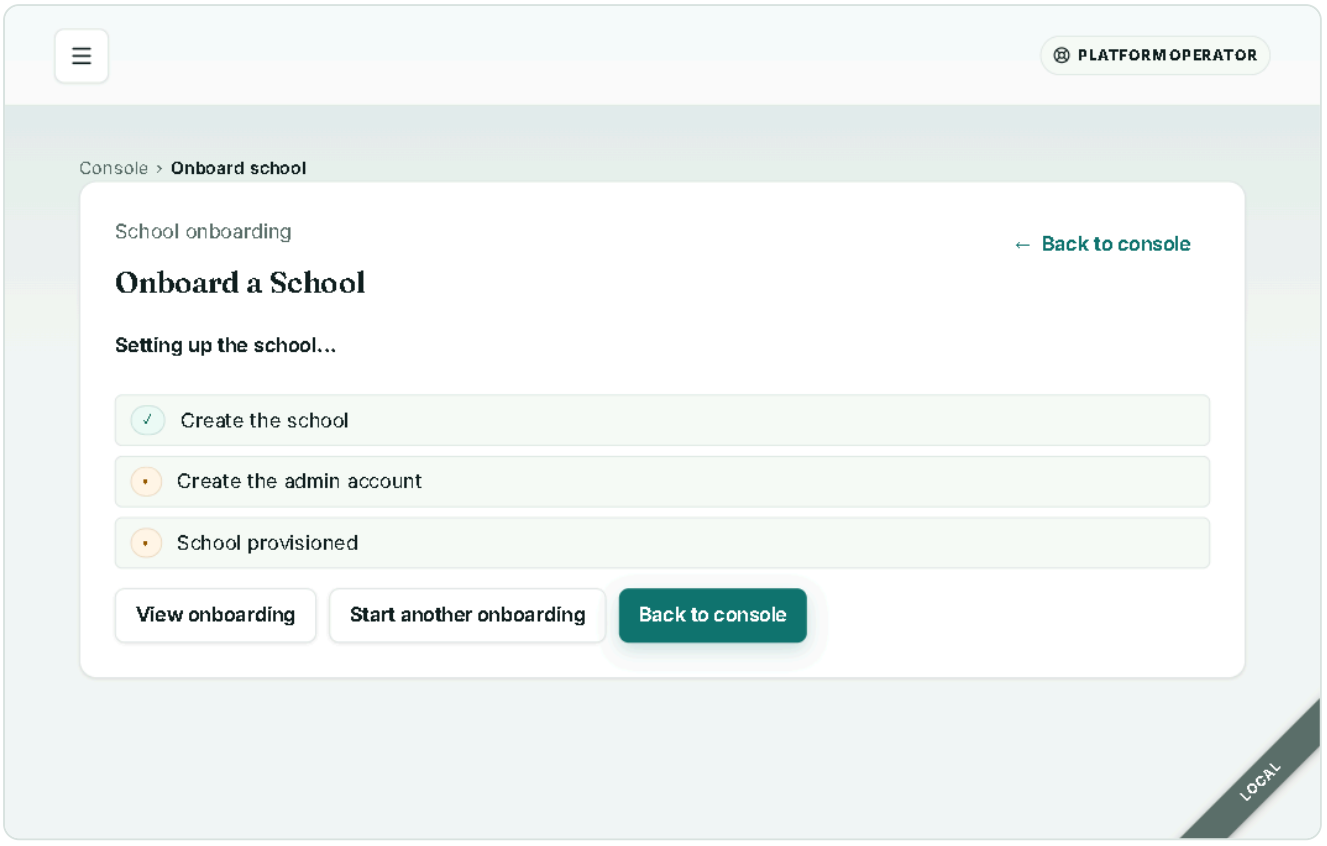
Last checked 2026/06/27, 08:49:53

How work fans out: queues, DLQs, and real-time

A gate scan must return fast. So the officer's confirmation is a synchronous, immutable write plus a handful of queue enqueues — it never waits on downstream bookkeeping. Cross-service fan-out runs on Cloudflare Queues (`movement-events`, `movement-projections`, `movement-attendance`, `movement-postcommit`, `emergency-broadcasts`), each with a dead-letter queue (DLQ) and idempotent consumers (ADR-004). A consumer crash retries; a poison message lands in the DLQ for replay rather than wedging the pipeline.

Real-time dashboard updates use a Durable Object, `SchoolStreamHub`, hosted in the gateway (ADR-010). Services publish per-school "hints" after their own writes; the browser treats those as nudges and refetches the D1 projection for truth, falling back to polling if the stream drops. The stream is never the source of truth — the projections are.

Long-running, multi-step jobs run on Cloudflare Workflows (ADR-020/021): **school onboarding** (provision → drip emails over 30 days → day-30 go-live evaluation), POPIA **erasure** (30-day cool-off then a per-store cascade), scheduled **report runs**, and academic-year **rollover**. These survive Worker restarts, sleep between steps, and retry individual steps durably. You can watch the onboarding steps tick green live in the operator console.



Tenant isolation — the honest detail

This is the section a technical advisor should read most carefully.

How separation works today: logically, in code. Multi-tenancy is enforced by a single hard rule (ADR-006): every school-scoped table carries a `school_id`, and that `school_id` is **always** taken from the verified JWT. A browser-supplied `X-School-Id` header is ignored. Every school-scoped query filters by the token's `school_id`, and the rule is regression-tested in the gateway's golden

suite. A user authenticated to school A cannot read school B — the school identifier is never something the client chooses.

What this is NOT (yet): it is not physical, database-per-tenant isolation. All schools share each service's single D1 database, partitioned logically by `school_id`. Event-heavy services route their D1 through `@mysentinel/shard-router`, which is built to spread schools across multiple shards by hashing `school_id`. **But only one shard is live today** — `shard_0`. The router's own default ring is literally `{ shards: ["shard_0"] }`. So:

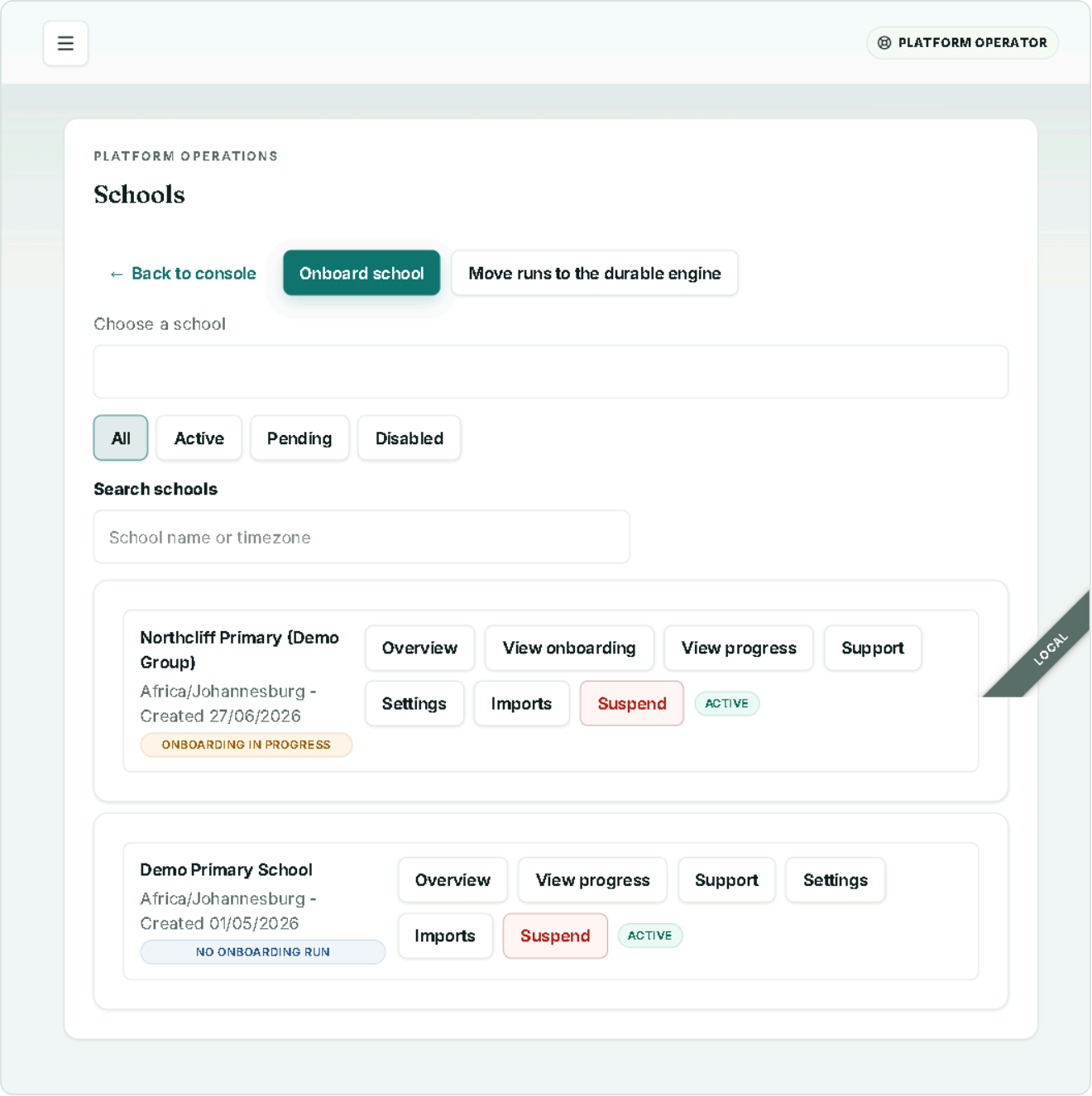
Aspect	Status today
`school_id`-from-token enforcement	**Shipped** — code-enforced, regression-tested
Per-query school scoping	**Shipped** on every school-scoped table
Isolation type	**Logical / code-enforced**, not physical per-tenant DBs
Horizontal sharding	**Shard-ready** — router live, but only `shard_0` provisioned
Cross-school operator/audit reads	Bound to `shard_0` today
Adding `shard_1+`	Possible **without a service rewrite** (config + provisioning), but it is an operational step that has not been taken

The honest summary: the safety guarantee (no cross-tenant reads) is real and tested; the *mechanism* is logical scoping on one shared database, with the path to physical sharding designed in but not yet exercised. A technical advisor should treat "shard-ready" as "no rewrite required," not "already sharded."

What "operator" means — and what a group owner does NOT get yet

It is important to be precise about roles for a chain owner.

- `operator` is **Cybertron platform staff** — the people who run MySentinel. An operator's token carries `platform` scope and **sees every school on the platform**, for cross-school onboarding and health. The operator console lets staff onboard a new campus (a durable Workflow), view platform health, and manage schools.



- There is **no group-scoped owner tenant — Roadmap.** A chain owner cannot today log in to a tenant scoped to *only their own campuses*. The two real scopes are `school` (one school) and `platform` (everything). A multi-school owner role that sees a subset of schools, and a consolidated cross-campus academic report, do not exist yet. The only cross-campus aggregates that exist today are channel-usage and billing-style operational rollups in the operator console — there is no consolidated academic report across campuses.

Other honest boundaries a DD review will check

- **WhatsApp and SMS are Preview / per-school provider-gated** — until a school's provider is connected, the channel silently demotes to email. App push and email are the live channels.
- **Web push** is real and RFC-8291-proven in code, but needs VAPID secrets and a real-device proof per environment before you should call it end-to-end.
- **Branded PDF / SAR rendering** needs the Cloudflare Browser Rendering binding provisioned; local development uses a deterministic stub.
- **Append-only audit** is enforced by a database trigger — it is not a hash-chained / WORM ledger, and audit `appendEvent` is at-least-once (benign, traceable duplicates are accepted; ADR-019).

- **Encryption at rest:** per-school AES-GCM photo keys are shipped; guardian/visitor contact encryption (sealed + hash) is shipped for new writes, with existing-row backfill and plaintext null-out as supervised per-environment steps.
- **Production is currently a hollow shell** (no secrets provisioned). Demo and evaluate on UAT; production is not yet functionally equivalent. POPIA, trilingual en/af/zu, and the South African context are first-class throughout.

The one-paragraph takeaway

MySentinel is a clean, Cloudflare-native, capability-service system with a single enforced trust boundary at the gateway and genuine, tested tenant isolation by `school_id`-from-token. The architecture is built to scale horizontally and to support richer ownership models, but today that isolation is logical (one shared shard, not per-campus databases), and "operator" is platform staff rather than a group owner. A scoped owner tenant, physical sharding, and a cross-campus academic report are all on the roadmap, not in the build — and we would rather you knew that before you signed than after.

Security & POPIA

This is the compliance due-diligence pack for a chain owner's technical or legal advisor. It explains how MySentinel protects learner and guardian personal information, how it enforces the Protection of Personal Information Act (POPIA), and — just as importantly — exactly where a capability is **shipped and proven** versus where it is **Roadmap** or **needs per-environment provisioning**. We have written it so that an advisor who checks our claims against the running system finds them true.

MySentinel runs entirely on Cloudflare (Workers, D1, KV, R2, Queues, Durable Objects, Cron, Static Assets). There is no separate VM fleet, no shared monolith database, and no third-party application runtime in the data path.

Executive summary

Control	Status	Notes
Tenant isolation — `school_id` from the verified token only	Shipped, verified	A school A user can never read school B; cross-role access is bounced
RBAC at the gateway (`requireAuth(role, scope)`)	Shipped, verified	Officer → `/admin` and admin → `/operator` are both rejected
Passkey / WebAuthn step-up for sensitive actions	Shipped, verified	Admin login forces step-up once a passkey is enrolled
Encryption at rest — D1	Shipped	Cloudflare encrypts D1 at rest by default
Per-school AES-256-GCM photo keys + cryptographic shredding	Shipped	Deleting the per-school key renders sealed data unreadable
Guardian/visitor contact sealing + lookup hash	Shipped (v0.2.0), **per-env activation**	New writes in a keyed school are sealed-and-hashed; switching it on per environment is a supervised step
Append-only audit trail	Shipped — **DB-trigger enforced**	Honestly: a storage-layer no-UPDATE trigger, **not** a hash-chain or WORM appliance
POPIA erasure via a durable 30-day cool-off cascade	Shipped	Per-leg visibility; fail-closed; `staff_user` leg still fail-closed (Roadmap)
Time-based retention purge	Shipped	Archive-then-delete, per dataset, audited
Group-scoped owner login	**Roadmap**	A chain owner cannot yet log in scoped to only their campuses
Physical per-campus database isolation	**Roadmap**	Isolation is logical/code-enforced today; only `shard_0` is live
Branded PDF / subject-access-request rendering	**Roadmap**	Needs the Cloudflare Browser Rendering binding

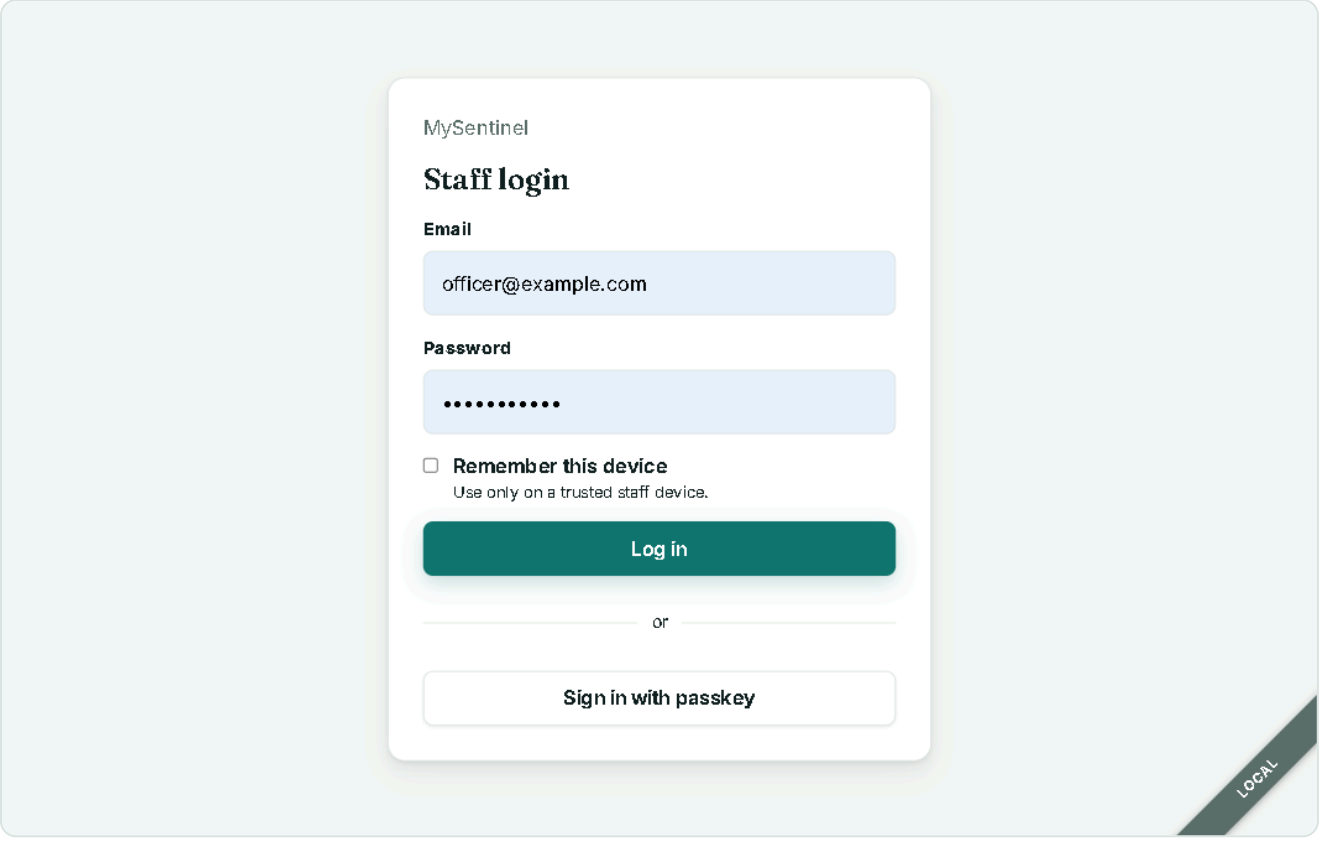
Production is currently a **hollow shell** with no secrets provisioned, so all demonstrations run on UAT. Treat every "verified" claim as verified on the local stack and UAT, not on a live-traffic production tenant.

Tenant isolation — `school_id` comes from the token, never the browser

Every authenticated request carries a JWT that encodes the user's `role` and `scope`. For school-scoped roles the token also carries the `school_id`. The hard rule, enforced at the gateway and never relaxed, is that `school_id` **is read only from the verified token** — never from a request header, query parameter, or body the browser controls.

This closes the most common multi-tenant data-leak class: a logged-in user cannot pivot to another school by editing a header. The same rule governs internal mechanisms. For example, real-time dashboard updates publish over an internal RPC where the tenant is the verified caller, not a request field — an earlier HTTP variant that read the tenant from the body was removed precisely because it allowed forged cross-tenant events.

Backend services compound the protection: each capability service owns its **own** D1 database and may only read its own data. A service that needs another's data calls that service's typed RPC; it never reaches into a sibling database. There is no shared schema for a bug to traverse.



RBAC at the gateway, plus passkey step-up for sensitive actions

The browser talks to exactly one public surface: the gateway. Internal services have no public route and no `workers.dev` exposure — the service binding *is* the trust boundary. The gateway's `requireAuth(role[], scope)` middleware declares, per route, which roles and which scope (`school` or `platform`) may call it, and rejects everything else server-side. The PWA can hide a button, but the gateway rejects the request regardless — the front end is never the security boundary.

Verified behaviour from the live walkthrough: an **officer** who navigates to `/admin` is bounced, and a school **admin** who navigates to `/operator` is bounced. The five roles — `officer`, `admin`, `class_teacher`, `operator`, `guardian` — each see only their authorised surface, and the `class_teacher` is further scoped to only their assigned classes.

WebAuthn / passkey step-up

Sensitive, irreversible, or account/credential-changing actions require a **passkey step-up** (a fresh WebAuthn ceremony) on top of the session. Admin login itself forces step-up once a passkey is enrolled. Step-up gates safety drills and lockdown, POPIA subject-access / erasure / correction / cancellation, school and channel-provider configuration, import commits, staff-invitation create and revoke, session revocation, year rollover confirmation, and photo-key migration, among others.

Two engineering details matter for an auditor:

- **Completeness is structural, not hand-maintained.** A route-completeness test parses every state-mutating gateway route and fails the build unless the route is step-up-gated *or* explicitly listed in a reviewed non-sensitive allow-list. There is no third state, so a new mutating route cannot ship ungated by omission.
- **Discoverable passkey login is genuinely two-factor.** The "Sign in with passkey" flow enforces the authenticator's user verification server-side (the device PIN/biometric is the second factor), preserves a per-environment relying-party ID split and a sign-count replay check, and binds the assertion to the credential owner. Lockout recovery is real: an operator can reset a staff member's passkeys — itself a step-up-gated, audited action — so a user with no working device signs in with their password and re-enrols.

The lockdown override on the gate-scan path shows the design philosophy: the routine scan stays ungated (gate latency is sacrosanct), but the rare admin override that checks a learner out during a lockdown triggers a step-up bound to a distinct action id, asserts the admin role from the token, and is never offline-queued.

Encryption at rest

Cloudflare encrypts D1 at rest by default. On top of that platform baseline, MySentinel adds application-layer encryption for the most sensitive personal information.

Per-school AES-256-GCM envelopes

Learner and visitor **photos** are encrypted with a **per-school** AES-GCM key. Guardian and visitor **contact** fields are sealed under the same per-school envelope:

Store	Fields sealed	Lookup hash
Guardian records (`identity-service`)	`phone`, `email`	Yes — `phone_hash`, `email_hash` (HMAC-SHA256 of the normalised value with a per-environment pepper)
Visitor records (`visitor-service`)	`phone`, host external name	No — never matched on

Each ciphertext's additional authenticated data binds it to its school and field, so a sealed value cannot be transplanted across schools or fields. The lookup hash exists only so inbound parent-message matching keeps working without decrypting anything.

Cryptographic shredding for erasure. Because contacts and photos share the per-school key, deleting that key renders every sealed record for that school unreadable — a true, fast erasure primitive for POPIA. The same sealing pattern protects per-school messaging-provider credentials (WhatsApp/SMS), which carry a `key_version` column for key rotation and are never returned by any read path.

SCHOOL ADMINISTRATOR

ADMIN SETTINGS

School settings

* Required

Display name *

Demo Primary School

Required

Support email

admin@example.com

Support phone

School logo

Upload logo

Primary color

#1f6650

Accent color

#c8923a

Default language *

Late arrival threshold

Required

Save settings

LOCAL

Pickup curfew

Enable late-pickup watch

Pickup time

17:30

Grace minutes

15

Mon Tue Wed Thu Fri Sat Sun

Notification sending window

Limit sending times

Routine check-in and check-out messages are only sent between these times. Emergency, safety and late-pickup alerts are always sent immediately.

End-of-day auto-close

Auto-close on-site learners at a hard campus-close time

After this time, any learner still showing on-site is recorded as a presumed check-out (no parent notification) so attendance is complete and the dashboard doesn't show children at school overnight.

Hard campus-close time

16:00

Going-home policy

Learners may arrive and leave on their own

Most learners walk, take a taxi, or cycle. Turn this off only if your school requires an adult to collect every learner — officers will then always record who collected the learner, and the gate will refuse an on-their-own check-out.

Express scan mode

Auto-confirm scans where the learner goes on their own — no guardian choice needed.

Visitor sign-in

- [x] Record a phone number for every visitor
- [x] Record a reason for every visit
- [ ] Record the visitor's ID number (stored encrypted)
- [ ] Record the visitor's vehicle registration

The officer's sign-in form only asks for what you switch on here. A visitor's name is always recorded.

Who visitors come to see

Reception
Teacher
Front Office
Principal / Management
Learner
Other

One option per line — the destinations the officer picks from at the gate (e.g. Reception, Teacher, Front Office). The officer also adds an optional name. Leave the defaults if you're not sure.

Emergency contact

Emergency contact name

Emergency contact number

Morning digest

One daily summary to every admin: learners not yet in, waiting requests and approvals, unread messages, and channel problems.

- [ ] Send morning digest

Send time

07:00

School-day mornings only. Sent within 5 minutes of this time.

Not sent yet

Weekly family summary

Send guardians one summary message every Friday at 16:00.

- [x] Enabled

Preview

Demo Primary School Accent

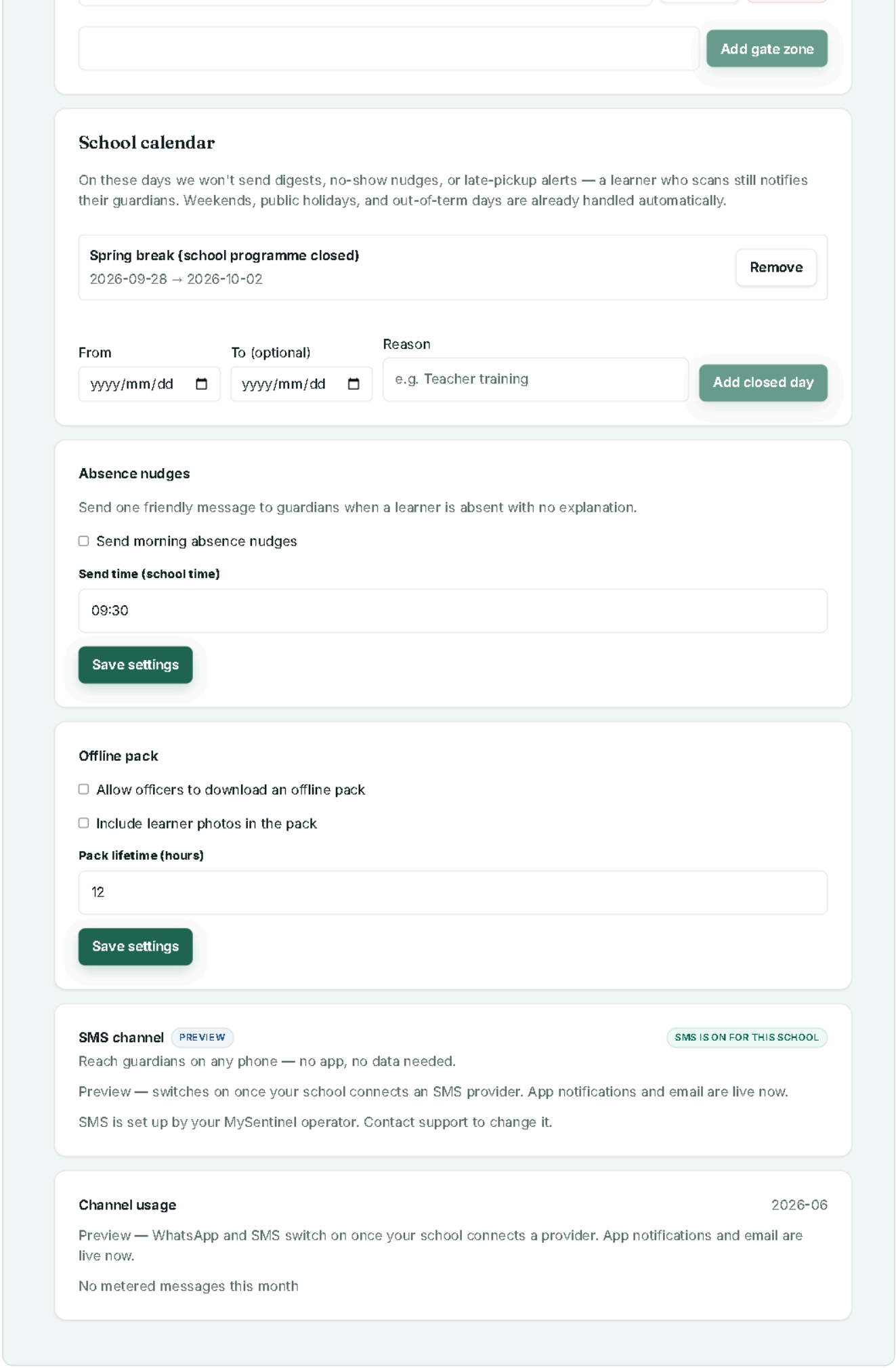
Support admin@example.com

Primary action

Gate zones

Name each gate where officers scan. Officers pick their gate when they start scanning.

Main gate Rename Remove



Honest activation status (per environment)

Contact encryption is deliberately **additive and reversible-safe**, which means it is switched on per environment in a controlled sequence — it is not a single global flip:

1. The decrypted read path is **flag-gated, default OFF**; with the flag off, behaviour is byte-identical to before.
2. A one-time backfill must seal existing rows **before** the read flag flips (otherwise a null lookup hash would silently break inbound matching).
3. Nulling out the legacy plaintext columns is a **separate, later cutover**.

This has been **activated and verified locally** on the demo school (all 11 guardians and 3 visitors sealed and hashed; reads decrypt; new writes seal; the live hash matches the stored hash). Switching it on for UAT or production is an explicit, supervised, per-environment operation. One

honest nuance: a school with no provisioned per-school key stores the contact as **plaintext fallback** (never silently lost) and emits an operator warning until the school is keyed and backfilled — so "encrypted at rest" is per-school-key-gated, not yet universal across an unprovisioned tenant.

The append-only audit trail — what it is, and what it is not

Every security-relevant action writes an immutable audit event: logins (success and failure, with the email **hashed**, never stored in clear), learner and guardian record changes, movement records, emergency drafts and confirmed sends, operator actions, and the POPIA workflows.

Be precise about the immutability guarantee. Append-only is enforced by a **D1** `BEFORE UPDATE` **trigger** that aborts any attempt to modify an existing `audit_events` row (`RAISE(ABORT, 'audit_events is append-only')`). Updates are blocked at the storage layer; deletes are permitted **only** for the sanctioned, audited POPIA retention-purge path. This is honest immutability against tampering-by-update — but it is **not** a cryptographic hash-chain and **not** a WORM storage appliance. An operator with direct database credentials and the ability to drop the trigger could, in principle, delete rows. If your due diligence requires tamper-evident chaining or external WORM, treat that as a **Roadmap** discussion, not a shipped claim.

Two further audit facts:

- **At-least-once, duplicates accepted.** Audit writes mint a fresh id per call with no idempotency key, so a retry can produce a benign duplicate row (same request id, adjacent timestamps). The state-projecting consumers are separately exactly-once; only the audit log is at-least-once, by design.
- **Metadata is redacted.** Audit rows never store plaintext phone numbers, emails, passwords, JWTs, or provider secrets. The redactor also strips value-shaped contact PII (emails and international `+27` / `0027` phone numbers) from metadata, conservatively, so as not to over-redact the immutable ledger.

POPIA data-subject rights

MySentinel implements the data-subject participation rights POPIA grants (access, correction, deletion) and the security-safeguards and minimality conditions.

Right to erasure — a durable, fail-closed cascade

Erasure runs as a **Cloudflare Workflow** with a **30-day cool-off** implemented as a durable sleep. After cool-off, the request cascades across **every** store that holds a copy of the subject's PII:

- Identity (name, contact, photo)
- Movement history (redacts learner and guardian names)
- Inbound parent messages (redacts the raw sender phone/email, names, message body, parsed intent)
- Visitor records (name, phone and its sealed copy, host name, vehicle registration, plus the R2 photo)
- Notification subscriptions (hard-deletes the guardian's push device keys)

A new `erasure_legs` table gives the operator **per-leg visibility** — which store ran and which failed. A request is marked `completed` **only if every leg succeeds**; a partial failure leaves it re-runnable and writes a `popia.erasure_failed` audit event. A cancellation requested during cool-off is

honoured. Unsupported subject types **fail closed** — they throw rather than falsely report completion while PII still exists.

Honest gap: the `staff_user` erasure leg is currently fail-closed (it never falsely completes, but the staff-PII cascade itself is **Roadmap**).

Right of access (SAR) and correction

Subject-access and correction requests are first-class, step-up-gated admin actions. Note the polish gap an advisor will find: **branded PDF rendering for a SAR pack needs the Cloudflare Browser Rendering binding** and is **Roadmap** — today the data is assembled and exportable, but the styled, branded PDF output is not yet wired.

Time-based retention and minimisation

Each school has a retention policy that is now **actually enforced**, not just displayed. A shared per-service purge contract archives-then-deletes the oldest rows past each dataset's cutoff, batched and atomic, writing the real archived rows to an R2 cold archive and auditing `popia.retention_purged`. Defaults:

Dataset	Default retention
Movement events	2555 days (~7 years)
Audit events	2555 days
Notification delivery attempts	90 days
Parent messages	365 days
Visitor records	2555 days

The append-only ledgers keep their no-update invariant; POPIA retention is the one sanctioned deletion path, and it is archive-first.

Network, edge, and secret hygiene

- **CORS is a strict allowlist** per environment — the custom domain plus the workers.dev hostname only, no wildcards and no `*.pages.dev`. Unapproved origins receive no `Access-Control-Allow-Origin`.
- **Edge rate-limiting** fronts the list-heavy admin/operator routes and the unauthenticated portal/login/invitation routes (keyed on the verified token id or client IP, never a browser header), backed by strict app-level limiters on the sensitive login and portal-token paths.
- **Cloudflare Turnstile** is live bot-protection on the tokenised parent portal mutations and the marketing site's demo-request form; the secret is a per-environment worker secret, never committed.
- **Secrets never live in git.** `.dev.vars`, `.env`, generated configs, logs, and exports are all gitignored; production secrets are set via `wrangler secret` or GitHub Environment secrets. Passwords use versioned salted PBKDF2-SHA-256.

What needs per-environment provisioning (state plainly)

A buyer's advisor should know that several controls are **shipped in code but require a deliberate per-environment provisioning step** before they are live in that environment:

- **Production secrets.** Production is a hollow shell today (no JWT secret, etc.). It is **not** functionally equivalent to UAT — demonstrate and pilot on UAT.
- **Contact encryption** — the pepper secret, the read flag, the backfill, and the plaintext null-out, in that order, per environment.
- **Web push** — VAPID secrets plus real-device proof per environment. The cryptography is implemented and RFC-8291-proven in code; end-to-end on a real device is a separate verification per environment.
- **WhatsApp and SMS** — per-school provider credentials. Until connected they show a Preview badge and silently demote to email; the system is honest about which channels are Live.
- **Branded PDF / SAR rendering** — the Cloudflare Browser Rendering binding.

Roadmap and known limitations (no overclaiming)

- **No group-scoped owner tenant.** "Operator" is Cybertron **platform** staff who can see **every** school on the platform. A chain owner cannot today log in scoped to only their own campuses. A group-owner role is **Roadmap**.
- **Logical, not physical, per-campus isolation.** The platform is shard-ready, but only `shard_0` is live, and cross-school operator/audit reads are bound to `shard_0`. Isolation between campuses is code-enforced and token-enforced today, not separate physical databases. Physical per-campus sharding is **Roadmap**.
- **Audit is trigger-enforced, not hash-chained/WORM** (see above).
- `staff_user` **erasure cascade** is fail-closed pending the final leg.
- **No consolidated cross-campus academic report** yet — only channel-usage and billing aggregates exist at the platform level.
- **Polish cracks** an advisor will see: the report-builder still has a free-text "Class ID" box, and the admin attendance page is hardcoded English (the rest of the app is trilingual en/af/zu).

☰

PLATFORM OPERATOR

PLATFORM OPERATIONS

Operator Console

Live · updated 08:49:53

Refresh | Onboard school | Health console

Platform access

Active

Schools in platform

1

Messaging setup

Ready

App notifications and email are the live channels. WhatsApp and SMS are in preview and switch on per school once a provider is connected.

Needs you now

ALL CLEAR

LOCAL

Nothing needs you right now

No failed deliveries, stalled imports, or channel issues across the platform.

Platform health

OK

Current snapshot

Last 1 hour delivery success

100%

Failed deliveries needing review

0

Failed logins in last 1 hour

0

Open health console

Generated 2026/06/27, 08:49:53

School onboarding

Manage schools

Schools

Demo Primary School

Africa/Johannesburg - Created 2026/05/01, 02:00:00

Settings | ACTIVE

Showing the most recent schools. Use "Manage schools" to search and page through all 1.

Provider readiness

Ready

Notification channels

WhatsApp READY

Messages can be sent through this channel.

Last checked 2026/06/27, 08:49:53

Email READY

Messages can be sent through this channel.

Last checked 2026/06/27, 08:49:53

Bottom line

MySentinel ships the load-bearing POPIA controls — token-only tenancy, server-side RBAC with passkey step-up, per-school encryption with cryptographic shredding, an append-only audit trail, a durable fail-closed erasure cascade, and enforced retention — and we have been explicit about the four areas that are Roadmap (group-owner scoping, physical sharding, hash-chained audit, branded SAR PDFs) and the controls that need a supervised per-environment switch-on. That separation is the point of this document: nothing here is a hollow claim, and every "verified" item can be re-checked on the running UAT system.

Onboarding & rollout playbook

This is the hands-on guide for getting a campus from "signed up" to "live at the gate." It walks the operator-console onboarding workflow, the per-campus go-live checklist, roster import, and channel activation — in the order you actually do them.

Read this first — the honest shape of rollout today:

- Onboarding is **one campus at a time**. There is no bulk "import 30 schools" flow yet (**Roadmap**). For a group you repeat the campus loop below per site — each takes minutes, but they are sequential.
- The **operator** role is Cybertron platform staff: it sees *every* school on the platform. There is **no group-scoped owner login** that shows only your campuses yet (**Roadmap**). Plan for Cybertron to drive onboarding with you, not for a self-service owner console.
- All onboarding and demos happen on **UAT**. Production is currently a hollow shell with no secrets — do not point a live campus at production until it is provisioned.

Everything below is shipped and verified live unless explicitly marked **Roadmap**.

The per-campus loop at a glance

Step	Where	Owner	Outcome
1. Provision the campus	Operator console → Onboard school	Operator	School + first admin (+ optional officer) created via durable workflow
2. Import the learner roster	Admin → Imports	School admin	Learners, guardians, badges, classes loaded
3. Issue badges & confirm photo key	Admin	School admin	NFC tags mapped; per-school photo encryption key ready
4. Add staff (officers, class teachers)	Admin → Staff	School admin	Gate + class coverage
5. Connect channels	Operator + Admin → Settings	Operator + admin	Email live; WhatsApp/SMS/push as provisioned
6. Work the go-live checklist to all-green	Admin dashboard	School admin	Campus ready to open the gate

---

Step 1 — Onboard a campus (durable workflow)

From the operator console, choose **Onboard school**.

☰

PLATFORM OPERATOR

PLATFORM OPERATIONS

Operator Console

Live · updated 08:49:53

Refresh | Onboard school | Health console

Platform access

Active

Schools in platform

1

Messaging setup

Ready

App notifications and email are the live channels. WhatsApp and SMS are in preview and switch on per school once a provider is connected.

Needs you now

ALL CLEAR

LOCAL

Nothing needs you right now

No failed deliveries, stalled imports, or channel issues across the platform.

Platform health

OK

Current snapshot

Last 1 hour delivery success

100%

Failed deliveries needing review

0

Failed logins in last 1 hour

0

Open health console

Generated 2026/06/27, 08:49:53

School onboarding

Manage schools

Schools

Demo Primary School

Africa/Johannesburg - Created 2026/05/01, 02:00:00

Settings | ACTIVE

Showing the most recent schools. Use "Manage schools" to search and page through all 1.

Provider readiness

Ready

Notification channels

WhatsApp READY

Messages can be sent through this channel.

Last checked 2026/06/27, 08:49:53

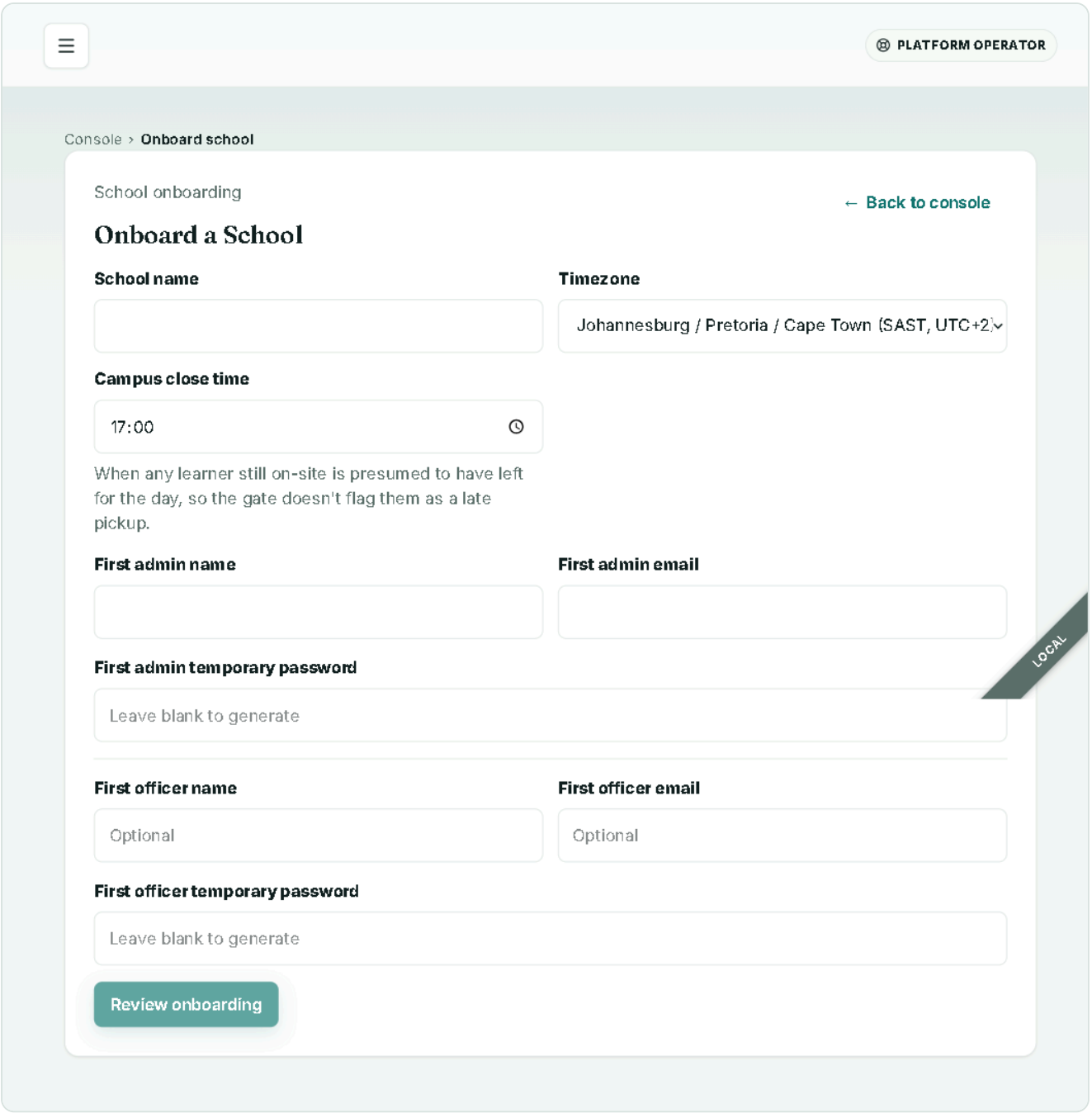
Email READY

Messages can be sent through this channel.

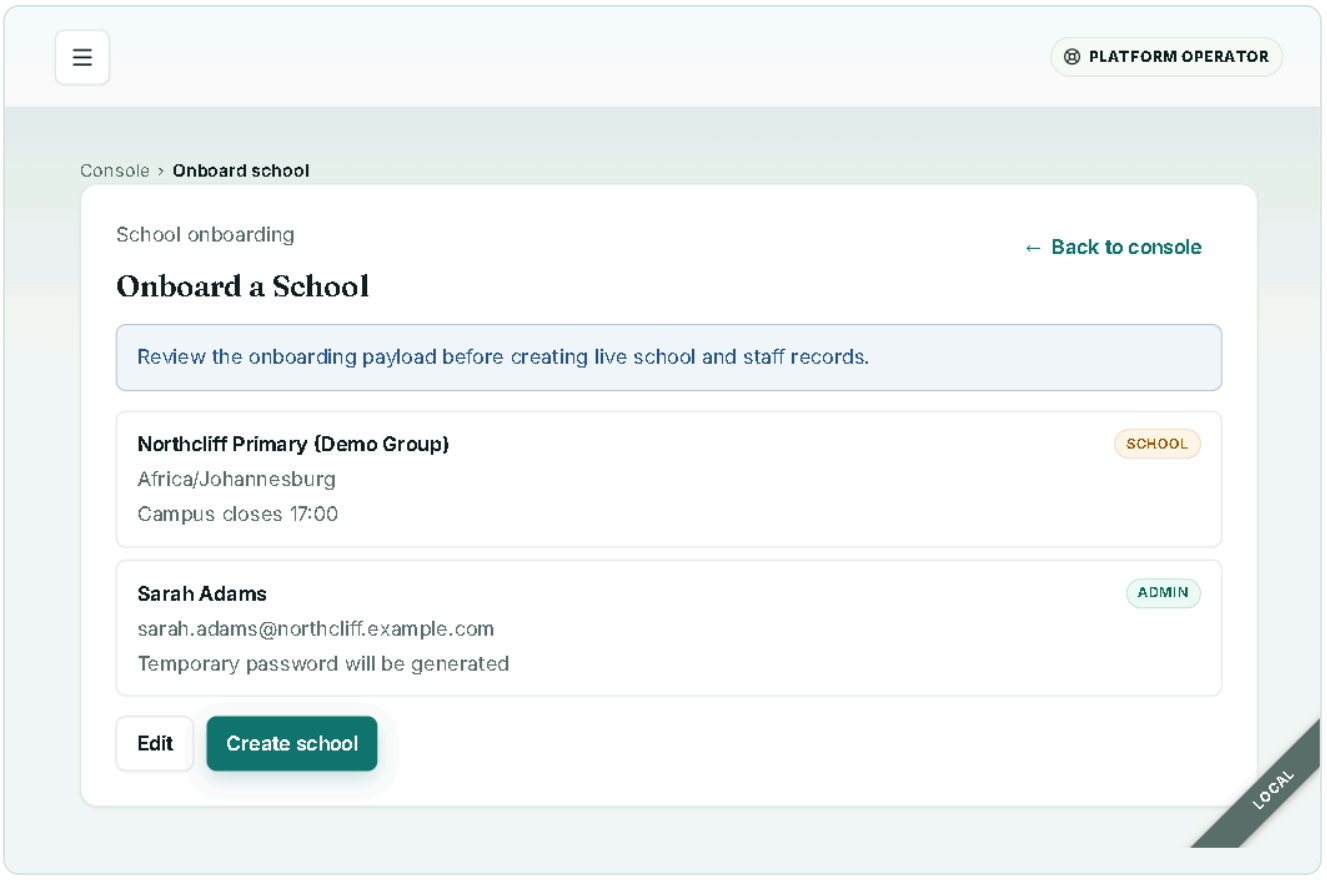
Last checked 2026/06/27, 08:49:53

Fill in the onboarding form (`/operator/schools/new`):

- **School name**
- **Timezone** — defaults to Johannesburg; pan-African and global options are available
- **Campus close time**
- **First admin** — name, email, temp password (leave blank to auto-generate)
- **First officer** — name, email, temp password (optional)

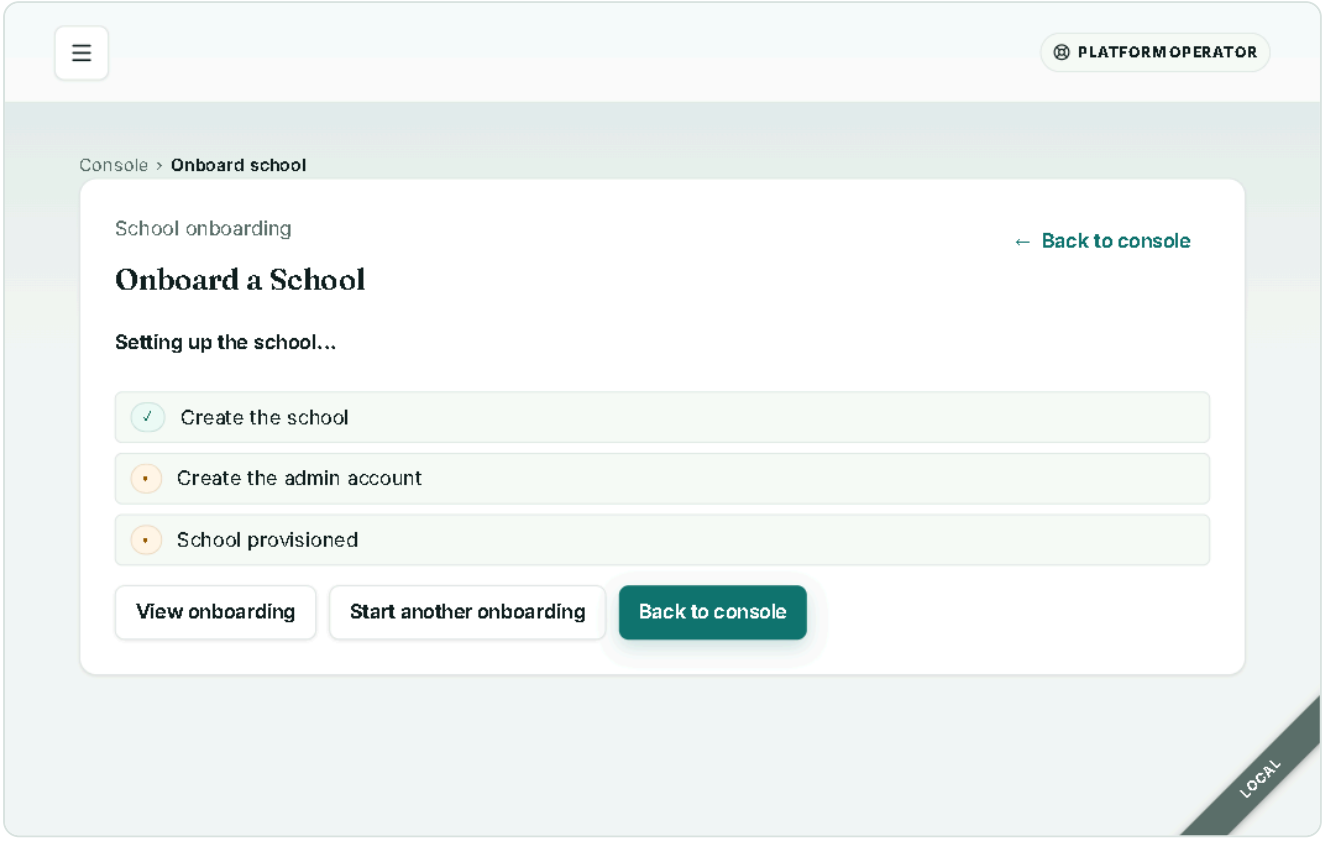


Choose **Review onboarding** to get a review-before-commit panel — confirm every value here, because provisioning runs for real once you proceed.



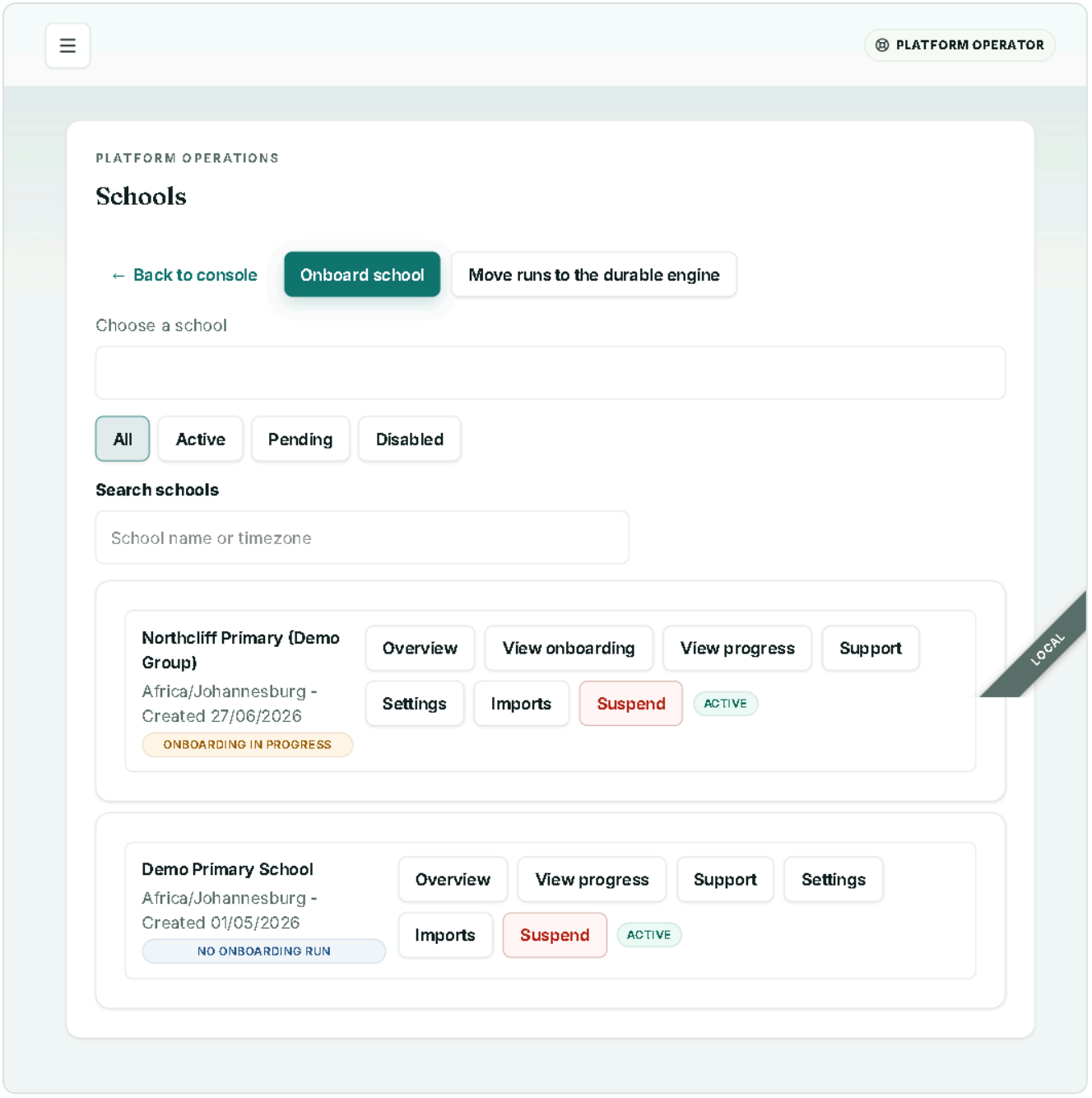
Choose **Create school**. A durable Cloudflare Workflow now runs and the steps tick green live:

1. **Create the school** ✓
2. **Create the admin account**
3. **School provisioned**



The workflow is idempotent with per-step retry/backoff, so a transient hiccup resumes rather than half-creating a campus. When it completes, choose **View onboarding** to track the new school, or **Start another onboarding** for the next campus. The school now appears in the operator **Schools**

list, where you can filter All / Active / Pending / Disabled and open per-school **Overview / View onboarding / View progress / Support / Settings / Imports / Suspend**.



The workflow also schedules onboarding drip emails and a **day-30 go-live evaluation** — keep working the checklist below so that evaluation lands green.

Step 2 — Import the learner roster

The school admin imports learners, guardians, badges, and classes from a CSV. Import is a **two-phase, preview-then-commit** flow — nothing is written until you commit.

1. **Upload** the CSV (or paste). Files are capped at **25 MB**; the file is hashed (SHA-256) for an auditable record.
2. **Preview** — every row is validated and column mappings are suggested. You get `total / valid / invalid` counts plus row-level reasons.
3. **Review and fix** — correct rows inline, then re-preview.
4. **Commit** — choose a commit mode:
   - **All-or-nothing** — commit only if every row is valid.

- **Skip errors** — commit the valid rows, leave the invalid ones.
- **Two-phase** — stage then finalise, with a rollback window.

What validation catches

The preview flags rows by reason, including: missing learner name, missing NFC badge id, missing guardian name or contact, duplicate NFC or guardian within the file, **NFC already exists** in the school, invalid email or phone, and a suggested class link. Spreadsheet formula injection is neutralised automatically. Guardian contact details are stored sealed-and-hashed (POPIA-aligned encryption at rest).

Large-file caveat (important)

For a smooth import, **split rosters above ~1,600 rows into batches** of roughly 1,500 and import them in sequence. The validate-and-commit pass on a very large single file can run long; batching keeps each preview snappy and each commit cleanly auditable. Lifting this so a single large file streams in one pass is **Roadmap**. The 25 MB hard limit is separate and unchanged.

> Tip: import classes first (or in the combined file) so learners link to a class on the way in, which makes the class roll-up and class-teacher views populate immediately.

Step 3 — Badges and the photo encryption key

- **Badges** — map each learner to their NFC tag (via the roster import's NFC column, or per learner). The go-live checklist treats badges as ready at **80% of active learners** issued.
- **Photo encryption key** — each campus uses its own AES-GCM key for learner photos. The checklist's **Photo encryption key ready** item confirms a key is provisioned before any photos are stored. No key, no photo storage — by design.

Step 4 — Add staff

Under **Admin → Staff**, add the people who run the gate and the classrooms:

- **Officers** — gate scanning on a phone.
- **Class teachers** — class roster, daily roll, and pickup approvals, scoped only to their assigned classes. (Reassignments are admin-only.)

The checklist treats staff as ready at **3 or more** active staff members. When admins enrol a passkey, sensitive admin actions then require passkey step-up — expect that prompt and keep a recovery path.

Step 5 — Activate notification channels

Guardians are notified push-first: **web push → email → WhatsApp → SMS**. Channels demote honestly — if a channel is not connected, the UI shows a **Preview** badge and the system silently falls back to email rather than faking a "sent." Activate them per environment:

Channel	What it needs	Status
Email	Verified sending domain (SPF/DKIM/DMARC) for the branded layout	Live channel; the dependable baseline
Web push	VAPID keypair as secrets on notification-service + the matching gateway key, then **real-device delivery proof**	Real + RFC-8291-proven in code; needs per-environment provisioning
WhatsApp	Meta Business number + **template approval**, then connect per school in Settings	**Preview** / provider-gated; demotes to email until connected
SMS	Operator-provisioned provider credentials per school	**Preview** / provider-gated

Operator-side, the console's **Messaging setup** card and **Provider readiness** (WhatsApp / Email) show what is connected. School-side, **Admin → Settings → SMS channel** and the WhatsApp configuration carry the per-school keys. Until a number is connected, the admin go-live checklist correctly reads **"WhatsApp not connected yet [Preview]."**

Two more provisioning notes:

- **Web push** is the headline parent channel — do not claim it end-to-end until you have seen a notification arrive on a real Android device for that environment.
- **Branded PDF / SAR rendering** needs the Cloudflare Browser Rendering binding; without it, documents render title-only. Confirm the binding before promising branded reports.

Step 6 — Drive the go-live checklist to all-green

The admin dashboard shows a live **Go-live checklist** computed on every load — there is no stale snapshot. All six must be ready before opening the gate:

SCHOOL ADMINISTRATOR

ADMIN DASHBOARD

Live

Today's school status

Nothing needs your attention right now

0 on-site · 0 left for the day · 10 no tap yet

Find a learner — is who in or out right now?

10 learners not on site

On-site	Checked in today	Checked out today
0	0	0

LOCAL

Total active learners

10

Daily actions

Use these for the school-day tasks that usually need attention first.

Open attendance | Emergency message | Safety mode | Late pickups | Guardian approvals | Learners

Last updated 08:47:04

Go-live checklist

Refresh checklist

What still needs to happen before this school is fully live.

- ✓ Learner roster imported
- • 4 of 10 learners checked in at least once (40% — 80% needed)
- ✓ 3 active staff members (3 needed)
- • WhatsApp not connected yet PREVIEW
- ✓ 10 of 10 badges issued (100% — 80% needed)
- ✓ Photo encryption key ready — Legacy key

Needs attention

ALL CLEAR

Today's exceptions

No stragglers and no guardian contact issues right now.

By class

Class roll-up

Who is in by class, so you can see at a glance which registers are complete.

Grade R - Sunbird	Grade 1 - Marigold	Grade 2 - Phoenix

Grade 0

0	2	2
IN	OUT	ENROLLED

Grade 1

0	2	2
IN	OUT	ENROLLED

Grade 2

0	1	1
IN	OUT	ENROLLED

Grade 3 - Acacia

Grade 3

0	1	1
IN	OUT	ENROLLED

Unassigned

0	4	4
IN	OUT	ENROLLED

Manage classes

Live

Current learner status

Aluwani Khumalo NO TAP YET
Last check-in at 26 Jun 2026, 09:52

Bongani Smit NO TAP YET
No check-in or pick-up recorded yet

Kgomotso Pillay NO TAP YET
No check-in or pick-up recorded yet

Lerato Nkosi NO TAP YET
Last check-in at 26 Jun 2026, 09:39

Mpho Sithole NO TAP YET
No check-in or pick-up recorded yet

Nomvula Botha NO TAP YET
No check-in or pick-up recorded yet

Refilwe van Wyk NO TAP YET
No check-in or pick-up recorded yet

Sipho Mokoena NO TAP YET
Last check-in at 26 Jun 2026, 09:45

Tendai Naidoo NO TAP YET
No check-in or pick-up recorded yet

Thandi Dlamini NO TAP YET
Last check-in at 25 Jun 2026, 09:51

Patterns to watch

Frequent late arrivals

0

learners late 3 or more times in the last 7 days

Absence risk

10 CHRONIC 10

of school days missed this year

Badge fleet health

10 Active badges

learners awaiting a replacement badge

Guardian contacts failing

0

guardians with 3 or more failed

deliveries in the last 7 days

Checklist item	Ready when
Learner roster imported	At least one active learner exists
Learners checked in (X of Y)	≥ 80% of active learners have ever been scanned
Active staff members	≥ 3 active staff
WhatsApp connected	A provider number is configured (else "not connected yet [Preview]")
Badges issued	≥ 80% of active learners have an active badge
Photo encryption key ready	A per-school photo key is provisioned

Per-campus branding and policy live in **Admin → Settings**: display name, support email/phone, logo, primary colour (#1f6650) and accent (#c8923a), default language (English / Afrikaans / isiZulu), late-arrival threshold, pickup curfew, quiet-hours sending window, end-of-day auto-close, going-home policy and express scan mode, visitor sign-in fields and destinations, gate zones, and the school calendar.

SCHOOL ADMINISTRATOR

ADMIN SETTINGS

School settings

* Required

Display name *

Demo Primary School

Required

Support email

admin@example.com

Support phone

School logo

Upload logo

Primary color

#1f6650

Accent color

#c8923a

Default language *

Late arrival threshold

Required

Save settings

LOCAL

Pickup curfew

Enable late-pickup watch

Pickup time

17:30

Grace minutes

15

Mon Tue Wed Thu Fri Sat Sun

Notification sending window

Limit sending times

Routine check-in and check-out messages are only sent between these times. Emergency, safety and late-pickup alerts are always sent immediately.

End-of-day auto-close

Auto-close on-site learners at a hard campus-close time

After this time, any learner still showing on-site is recorded as a presumed check-out (no parent notification) so attendance is complete and the dashboard doesn't show children at school overnight.

Hard campus-close time

16:00

Going-home policy

Learners may arrive and leave on their own

Most learners walk, take a taxi, or cycle. Turn this off only if your school requires an adult to collect every learner — officers will then always record who collected the learner, and the gate will refuse an on-their-own check-out.

Express scan mode

Auto-confirm scans where the learner goes on their own — no guardian choice needed.

Visitor sign-in

- [x] Record a phone number for every visitor
- [x] Record a reason for every visit
- [ ] Record the visitor's ID number (stored encrypted)
- [ ] Record the visitor's vehicle registration

The officer's sign-in form only asks for what you switch on here. A visitor's name is always recorded.

Who visitors come to see

Reception
Teacher
Front Office
Principal / Management
Learner
Other

One option per line — the destinations the officer picks from at the gate (e.g. Reception, Teacher, Front Office). The officer also adds an optional name. Leave the defaults if you're not sure.

Emergency contact

Emergency contact name

Emergency contact number

Morning digest

One daily summary to every admin: learners not yet in, waiting requests and approvals, unread messages, and channel problems.

- [ ] Send morning digest

Send time

07:00

School-day mornings only. Sent within 5 minutes of this time.

Not sent yet

Weekly family summary

Send guardians one summary message every Friday at 16:00.

- [x] Enabled

Preview

Demo Primary School Accent

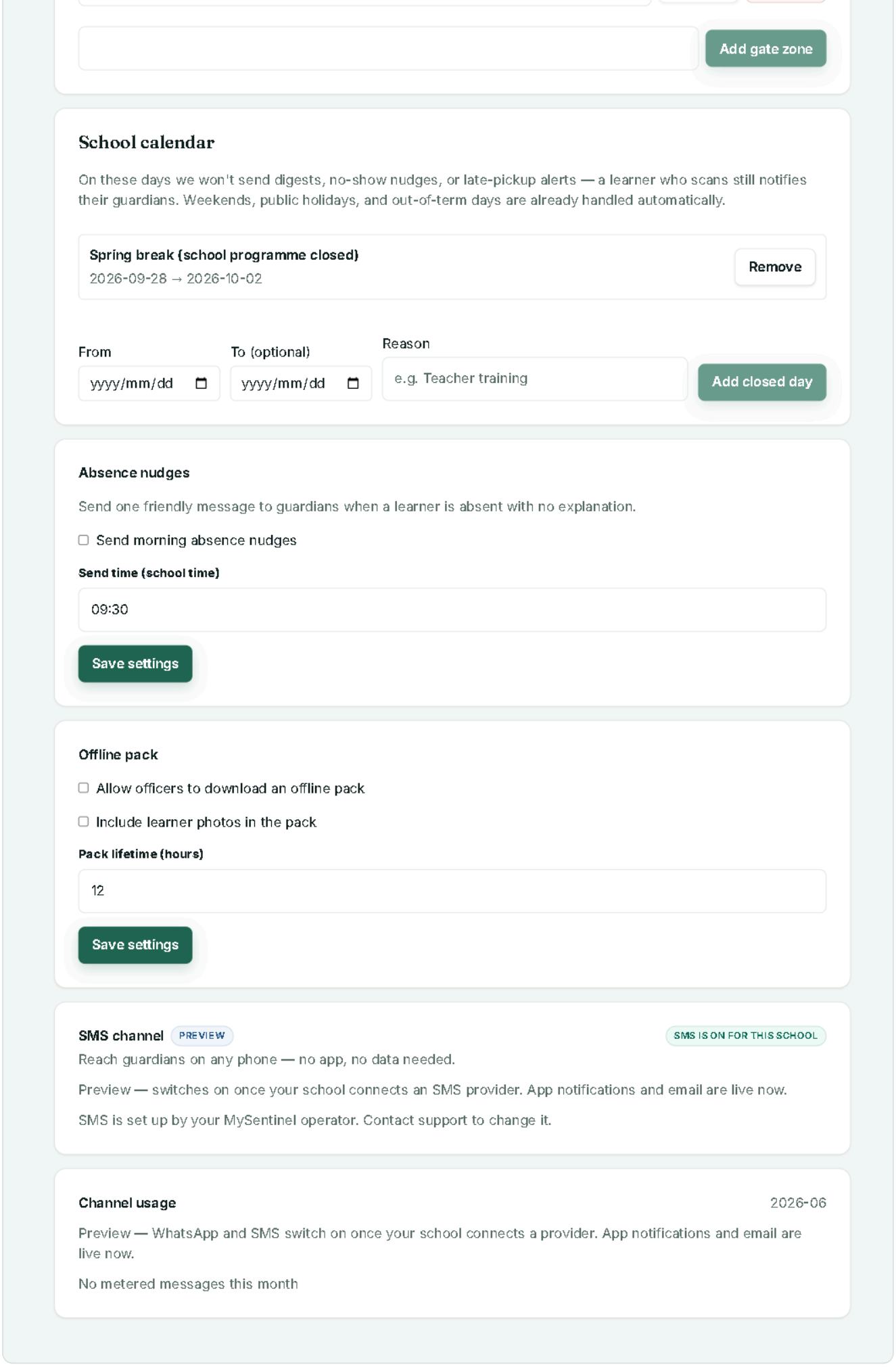
Support admin@example.com

Primary action

Gate zones

Name each gate where officers scan. Officers pick their gate when they start scanning.

Main gate Rename Remove



When the checklist is all-green, the officer can open the gate.

Rolling out across a group

Because onboarding is per-campus today, sequence the group like this:

1. **Pilot 2–3 campuses** through the full loop above so the operator console, platform-health view, and cross-school counts have real data behind them.
2. **Repeat per remaining campus** — each onboarding is a few minutes of form-plus-workflow, plus the roster/staff/channel work.

3. **Watch platform health** in the operator console (last-hour delivery success, failed deliveries needing review, failed logins) as each campus comes online.

☰ PLATFORM OPERATOR

PLATFORM OPERATIONS

Platform health

Read-only view of messaging, school setup, account access, and support items that may need attention.

Console Refresh

Generated 2026/06/27, 08:53:47 OK

School operations

School health

Manage schools

Live
1

Onboarding
1

Went quiet
0

LOCAL

Disabled
0

Northcliff Primary (Demo Group) ONBOARDING View trend Settings
No gate activity yet
0/0 learners - 0 classes - 0 staff - 7d delivery Unavailable

Demo Primary School LIVE View trend Settings
Active today - last gate 2026/06/26, 09:52:00
10/10 learners - 4 classes - 3 staff - 7d delivery 100%

Showing up to the first 2 schools, went-quiet first. Generated 2026/06/27, 08:53:47.

Providers

View deliveries

WhatsApp readiness now
Ready
READY PREVIEW

Email readiness now
Ready
READY

Web Push readiness now
Ready
READY

Deliveries

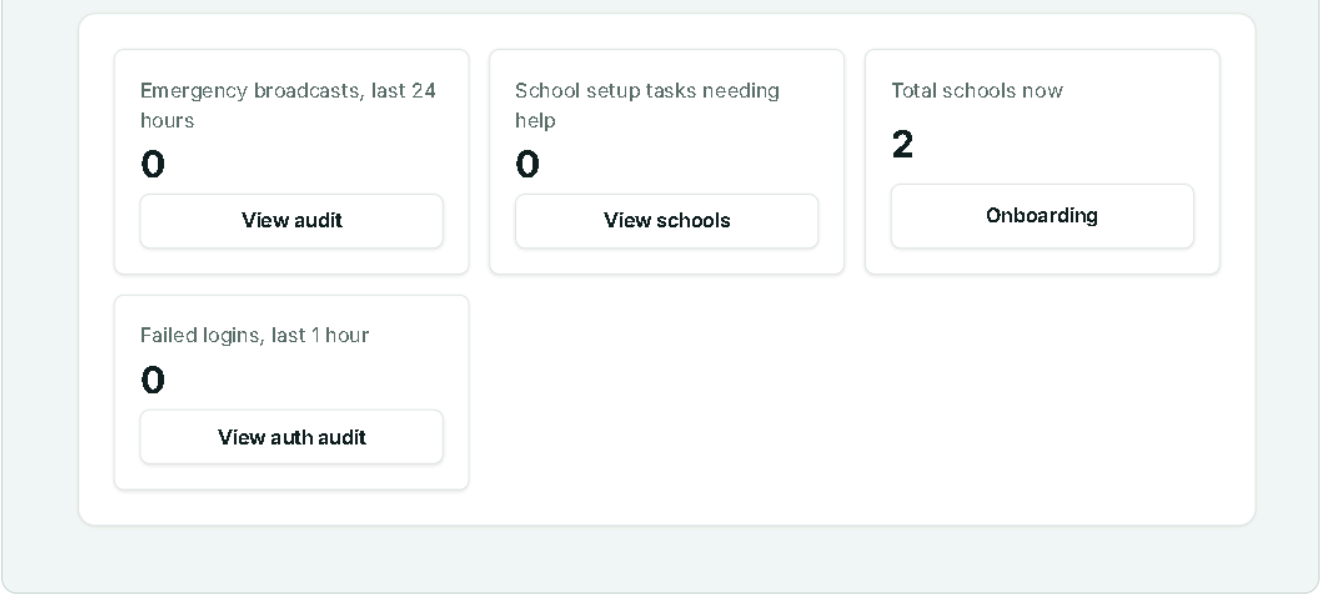
View deliveries

Success rate, last 1 hour
100%
1 sent, 0 failed

Success rate, last 24 hours
100%
1 sent, 0 failed

Failed deliveries needing review
0
View recovery queue

No delivery failures
No failure codes were reported in the last 24 hours.



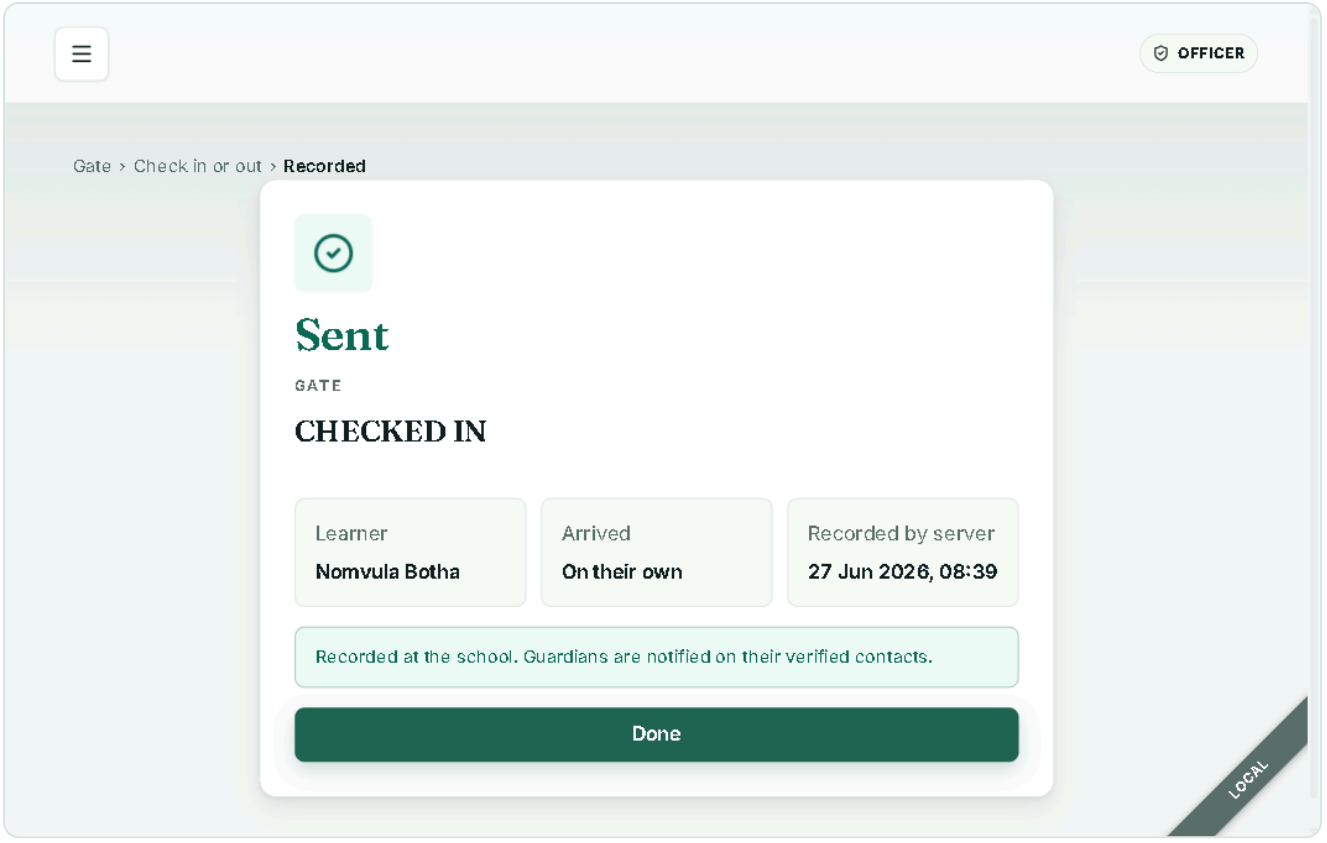
Roadmap (state plainly to stakeholders)

- **Bulk / multi-campus onboarding** in one action — not yet; one campus at a time today.
- **Group-scoped owner role** (a login that shows only your campuses) — not yet; operator = full platform staff.
- **Single-pass large-file import** above ~1,600 rows — batch for now.
- **Physical per-campus database isolation** — shard-ready, but only one shard is live; isolation today is logical (by `school_id`), enforced in code.
- **Consolidated cross-campus academic report** — not yet; channel-usage and billing aggregates exist today.

Name these plainly during rollout. The product's channel-and-environment honesty is a feature — carry that same honesty into the rollout conversation and it builds trust rather than eroding it.

Notification channels — reality & cost

When an officer scans a learner's badge at the gate, MySentinel records the check in / check out immediately and then notifies that learner's verified guardians. The record at the school is the source of truth; the notification is a best-effort courtesy on top of it. That distinction matters commercially, and the product never blurs it: **the officer's screen says "Recorded at the school. Guardians are notified on their verified contacts."** — it never claims a guardian *read* the message.



This document is the un-spun version of what reaches a parent today, what is still being provisioned, and what each channel costs you inside the ~R10/learner/month price.

The fallback ladder: push → email → WhatsApp → SMS

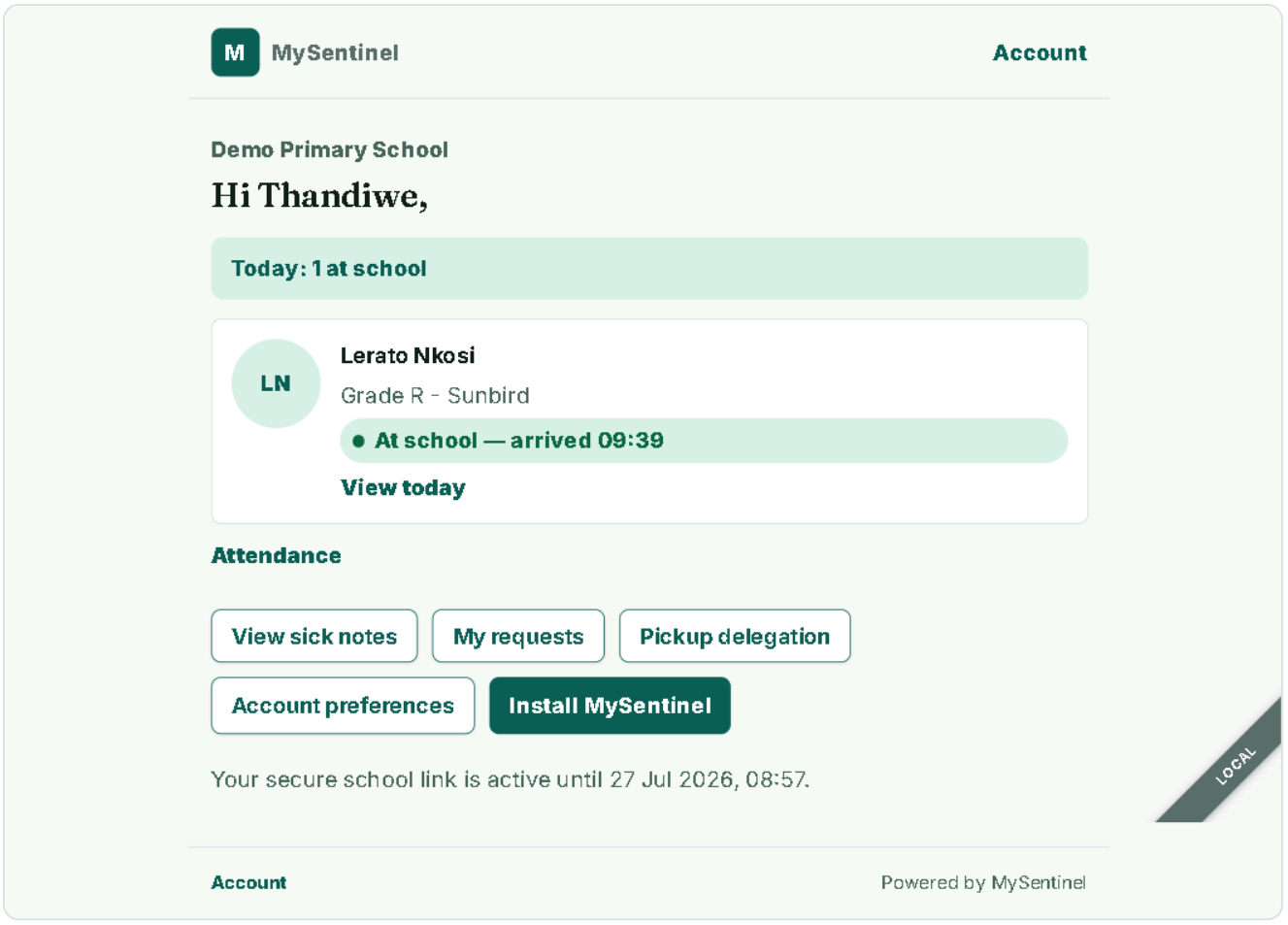
Each guardian's message walks an ordered ladder, highest-trust-and-lowest-cost first:

Rung	Channel	What it is
1	**App push** (web push)	Free, instant, lands on a guardian's installed phone app
2	**Email**	One branded per-school HTML layout, your colours + logo
3	**WhatsApp**	The channel SA parents actually use — but provider-gated (see below)
4	**SMS**	Last resort: short, paid-per-message, always-delivers

How the ladder behaves depends on the message type, and this is a deliberate, per-school policy — not an accident:

- **Routine check in / check out** uses the school's chosen mode. The owner default is **broadcast** (reach every resolved channel the guardian has enabled); a school can switch to **priority**, the classic fallback ladder that stops at the first confirmed send. Either way the per-guardian channel toggles and quiet-hours still narrow it.
- **Emergency, safety mode, and late-pickup** always **broadcast to every eligible channel** and bypass the per-child routine preferences — a safety message is not something a parent can accidentally mute. (The one exception, by design: a guardian who explicitly turns *SMS* off keeps it off even in an emergency, because SMS spends real money on their behalf.)

Guardians control their own mix from the tokenised parent portal — channel preferences and quiet hours, no login required.



Channel honesty: the app never shows a fake "sent"

This is the part a buyer's technical advisor will probe, so here is exactly how it works in the code today.

Every attempt is recorded with a truthful status — there is no path that writes "sent" for a message that was not actually handed to a provider:

Status	Meaning
sent / **duplicate**	The provider accepted it (duplicate = idempotent re-send of the same message)
skipped	We deliberately did *not* send on this channel (e.g. WhatsApp template not approved) — recorded, not hidden
retryable_failure	Transient provider problem; the queue will retry
permanent_failure	Will not succeed; surfaced for admin action, never silently dropped

A guardian is only counted as "notified" when at least one channel genuinely confirmed. If nothing confirmed, the system reports `delivered: false` rather than inflating a success number. The deliveries dashboard uses **"Deliveries"** vocabulary, not "messages read", and in the admin and operator UI every channel wears a **Preview** or **Live** badge so nobody mistakes a not-yet-connected channel for a working one.

Privacy is built into the same plumbing: delivery logs store a **hashed** recipient address, never a parent's raw phone or email. (The live walkthrough confirmed this — a check-in fanned out through notification-service and the email log fired with a hashed recipient.)

SCHOOL ADMINISTRATOR

ADMIN SETTINGS

School settings

* Required

Display name *

Demo Primary School

Required

Support email

admin@example.com

Support phone

School logo

Upload logo

Primary color

#1f6650

Accent color

#c8923a

Default language *

Late arrival threshold

Required

Save settings

LOCAL

Pickup curfew

Enable late-pickup watch

Pickup time

17:30

Grace minutes

15

Mon Tue Wed Thu Fri Sat Sun

Notification sending window

Limit sending times

Routine check-in and check-out messages are only sent between these times. Emergency, safety and late-pickup alerts are always sent immediately.

End-of-day auto-close

Auto-close on-site learners at a hard campus-close time

After this time, any learner still showing on-site is recorded as a presumed check-out (no parent notification) so attendance is complete and the dashboard doesn't show children at school overnight.

Hard campus-close time

16:00

Going-home policy

Learners may arrive and leave on their own

Most learners walk, take a taxi, or cycle. Turn this off only if your school requires an adult to collect every learner — officers will then always record who collected the learner, and the gate will refuse an on-their-own check-out.

Express scan mode

Auto-confirm scans where the learner goes on their own — no guardian choice needed.

Visitor sign-in

- [x] Record a phone number for every visitor
- [x] Record a reason for every visit
- [ ] Record the visitor's ID number (stored encrypted)
- [ ] Record the visitor's vehicle registration

The officer's sign-in form only asks for what you switch on here. A visitor's name is always recorded.

Who visitors come to see

Reception
Teacher
Front Office
Principal / Management
Learner
Other

One option per line — the destinations the officer picks from at the gate (e.g. Reception, Teacher, Front Office). The officer also adds an optional name. Leave the defaults if you're not sure.

Emergency contact

Emergency contact name

Emergency contact number

Morning digest

One daily summary to every admin: learners not yet in, waiting requests and approvals, unread messages, and channel problems.

- [ ] Send morning digest

Send time

07:00

School-day mornings only. Sent within 5 minutes of this time.

Not sent yet

Weekly family summary

Send guardians one summary message every Friday at 16:00.

- [x] Enabled

Preview

Demo Primary School Accent

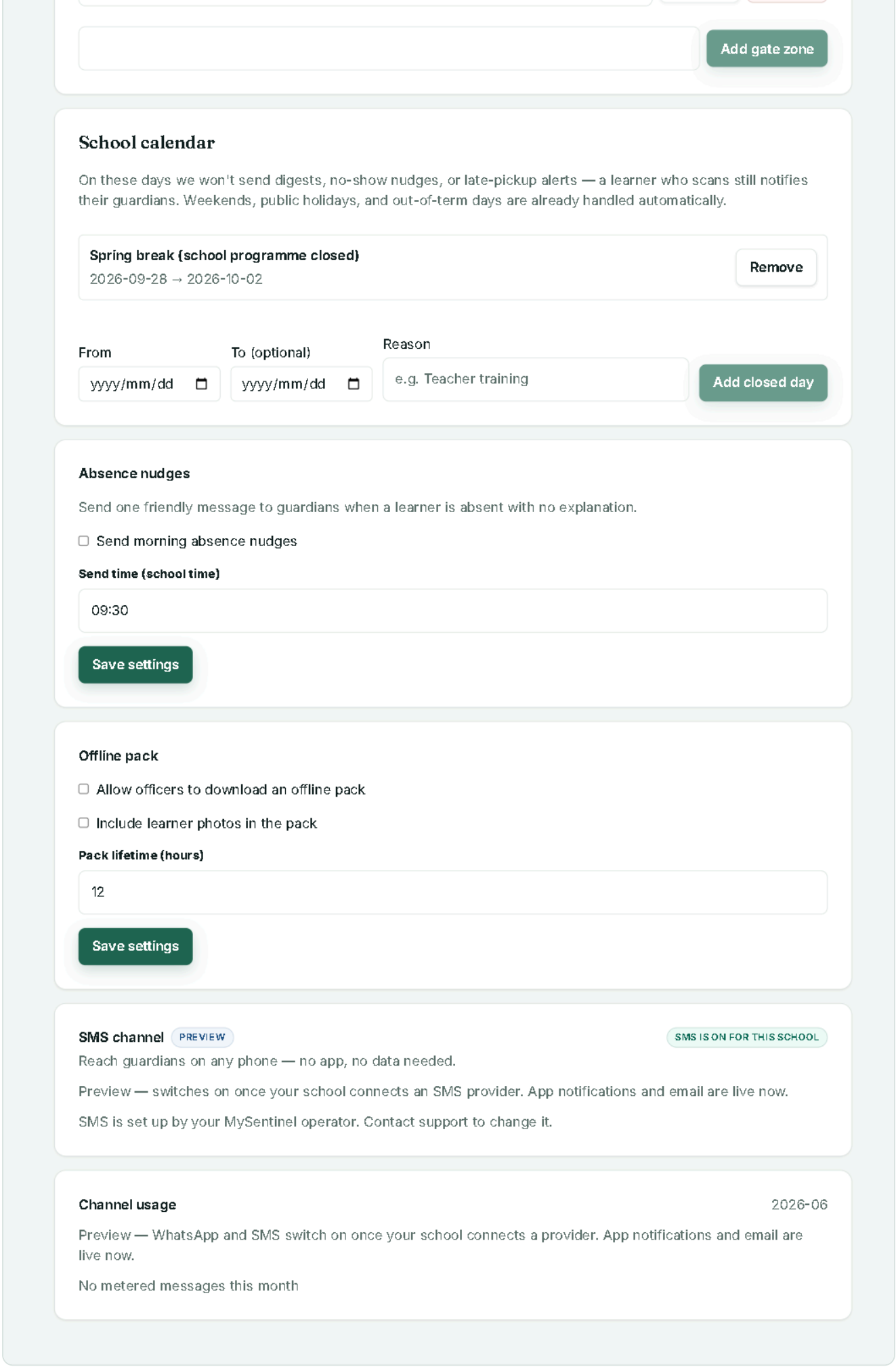
Support admin@example.com

Primary action

Gate zones

Name each gate where officers scan. Officers pick their gate when they start scanning.

Main gate Rename Remove



Per-channel status — what actually delivers today

App push (web push) — verified arriving on a real device (UAT)

The web-push implementation is real (VAPID + aes128gcm, no Firebase, proven against the RFC-8291 test vector in code) and **was confirmed arriving on a real device on UAT (2026-06-27)** with VAPID secrets provisioned. Web push is Chrome/Android-friendly; iOS requires the parent to "Install MySentinel" (add to home screen) first. Any *new* environment still re-provisions its own VAPID secrets + a device re-check before you promise delivery there. **Status: live and verified on device (UAT); each new environment re-provisions VAPID, which is an activation step, not new development.**

Email — live with a verified domain

Email is the dependable backbone today: one shared branded layout (`email-render`) used by every service, themed to each school. It is live once the sending **domain is verified** (SPF/DKIM/DMARC). Locally it runs in stub mode (`[email-stub]` logs); on a connected environment it sends for real. **Status: live; this is the channel a cold demo can always truly deliver.**

WhatsApp — live delivery tested on UAT, activates per school once connected

Live WhatsApp delivery was **tested and confirmed on UAT (2026-06-27)**. For a given school it switches on once two things are true: a configured provider, and **Meta-approved message templates** for the (school, template, language) combination. Until a school has both, the message is **demoted to email** — recorded as a truthful `skipped` WhatsApp attempt, never a failed-retry-loop and never a fake send. Where a school brings its own Meta credentials, those are used directly. **Status: proven to deliver (UAT); activates per school once the provider + approved templates are connected.**

SMS — live delivery tested on UAT, activates per school once connected

SMS is the always-arrives fallback, rendered as a tight ≤160-character GSM-7 body in en/af/zu. Live SMS delivery was **tested and confirmed on UAT (2026-06-27)**; it is operator-provisioned per school. Because every SMS costs real money, it sits last on the ladder and carries a smart guard: in broadcast mode the SMS rung is **suppressed once a higher tier already confirmed** for that guardian and event — you are never double-charged when the family was already reached on push, WhatsApp or email — unless SMS is the *only* channel left. **Status: proven to deliver (UAT); activates per school once the provider is connected.**

☰

PLATFORM OPERATOR

PLATFORM OPERATIONS

Operator Console

Live · updated 08:49:53

Refresh | Onboard school | Health console

Platform access

Active

Schools in platform

1

Messaging setup

Ready

App notifications and email are the live channels. WhatsApp and SMS are in preview and switch on per school once a provider is connected.

Needs you now

ALL CLEAR

LOCAL

Nothing needs you right now

No failed deliveries, stalled imports, or channel issues across the platform.

Platform health

OK

Current snapshot

Last 1 hour delivery success

100%

Failed deliveries needing review

0

Failed logins in last 1 hour

0

Open health console

Generated 2026/06/27, 08:49:53

School onboarding

Manage schools

Schools

Demo Primary School

Africa/Johannesburg - Created 2026/05/01, 02:00:00

Settings | ACTIVE

Showing the most recent schools. Use "Manage schools" to search and page through all 1.

Provider readiness

Ready

Notification channels

WhatsApp READY

Messages can be sent through this channel.

Last checked 2026/06/27, 08:49:53

Email READY

Messages can be sent through this channel.

Last checked 2026/06/27, 08:49:53

What it costs — BYO vs pass-through inside ~R10/learner

Pricing is **~R10 per learner per month (negotiable)**, billed **manually, out-of-app** — there is no in-app billing engine. The platform fee covers the software, app push, and branded email. The two paid channels carry their own per-message provider cost, and you choose how that is handled:

Channel	Per-message cost	How it's handled
App push	None (no provider fee)	Included in the platform fee
Email	Effectively included	Sends on a verified domain
WhatsApp	Meta conversation fee	**BYO** your own Meta credentials (you pay Meta directly) **or** marked-up pass-through, metered per confirmed delivery
SMS	Per-segment carrier fee	**BYO** provider **or** marked-up pass-through, metered per confirmed delivery

Only genuinely **confirmed** WhatsApp/SMS deliveries on a pass-through (platform-metered) arrangement are metered for billing — BYO and failed sends are never metered. Admins see channel-usage metering in Settings, and the operator console exposes per-school delivery success and provider readiness for reconciliation.

☰ PLATFORM OPERATOR

PLATFORM OPERATIONS

Platform health

Read-only view of messaging, school setup, account access, and support items that may need attention.

Console Refresh

Generated 2026/06/27, 08:53:47 OK

School operations

School health

Manage schools

Live
1

Onboarding
1

Went quiet
0

LOCAL

Disabled
0

Northcliff Primary (Demo Group) ONBOARDING View trend Settings
No gate activity yet
0/0 learners - 0 classes - 0 staff - 7d delivery Unavailable

Demo Primary School LIVE View trend Settings
Active today - last gate 2026/06/26, 09:52:00
10/10 learners - 4 classes - 3 staff - 7d delivery 100%

Showing up to the first 2 schools, went-quiet first. Generated 2026/06/27, 08:53:47.

Providers

View deliveries

WhatsApp readiness now
Ready
READY PREVIEW

Email readiness now
Ready
READY

Web Push readiness now
Ready
READY

Deliveries

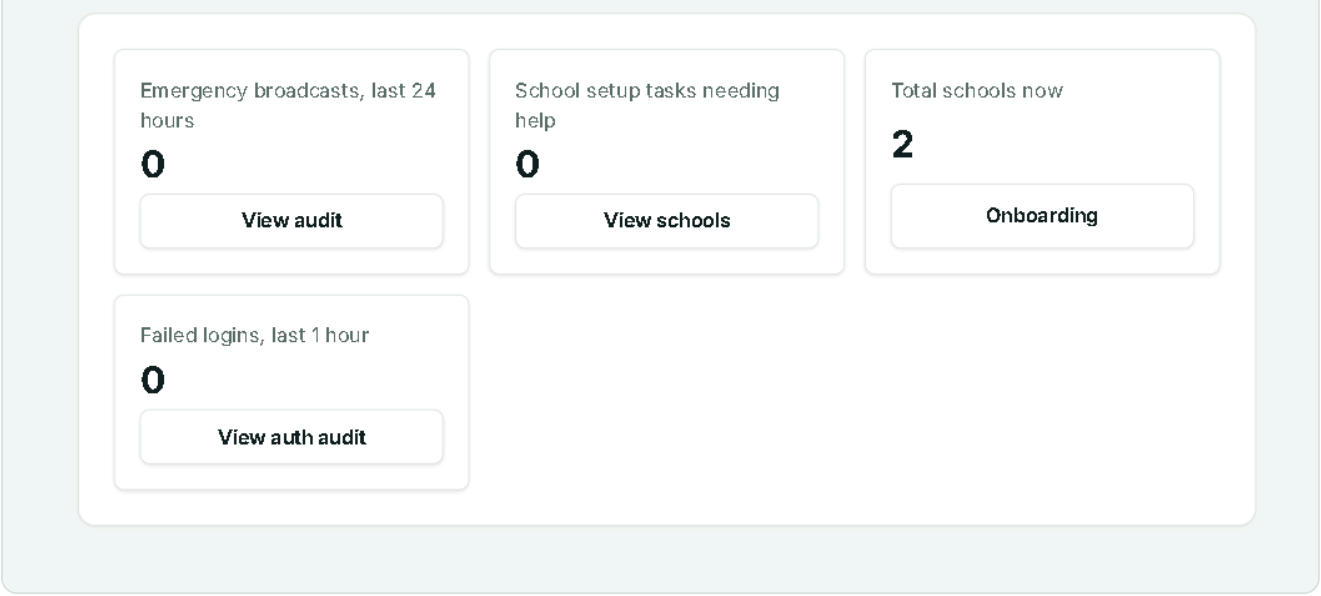
View deliveries

Success rate, last 1 hour
100%
1 sent, 0 failed

Success rate, last 24 hours
100%
1 sent, 0 failed

Failed deliveries needing review
0
View recovery queue

No delivery failures
No failure codes were reported in the last 24 hours.



The honest bottom line

- **Proven to deliver (verified on UAT, 2026-06-27):** app push **arriving on a real device**, **branded email** (on a verified domain), and **live WhatsApp + SMS**. All four were confirmed delivering on UAT.
- **Activates per school:** WhatsApp (Meta templates + provider) and SMS (provider) deliver, but each school connects its own provider before they switch on — until then they honestly demote to email, a real demoting code path, never fake capability.
- **No overclaiming by design:** the app never paints a fake "sent", every channel is badged Preview vs Live, recipient addresses are hashed, and unconfirmed sends are reported as not-delivered.
- **Production is currently a hollow shell** (no secrets, usage-blocked) — channel activation and any live demo happen on **UAT**, never prod.

The selling point is precisely this restraint: across many campuses, a system that refuses to over-promise parents is the one that keeps the school's trust.

Proven vs roadmap (honesty sheet)

MySentinel is a strong, real product — a buyer's technical advisor will check our claims, and the honesty itself becomes part of the sell. This page is the single reference so nobody on the sales side accidentally promises a capability that is on the roadmap or that needs provisioning before it works.

Use it like this: **demo what's in the left column, name what's in the right column out loud.** Everything in the left column was verified live in a local walkthrough on 2026-06-27 across all five personas, and the previously provisioning-gated capabilities — **a real NFC badge tap on an Android phone, web push arriving on a real device, the offline "Queued → Sent" outbox, branded PDF / SAR rendering, and live WhatsApp + SMS delivery, plus multi-school health populating across schools** — were **confirmed working end-to-end on UAT by the owner on 2026-06-27**. Everything in the right column is real engineering that is either not yet built, not yet provisioned for an environment, or positioned differently than the chain-owner pitch implies.

> **One framing rule:** if it's not proven today, say "roadmap" or "needs provisioning" plainly. Never imply a hollow or unproven capability is live.

The sheet

#	Proven today (verified live)	Roadmap / needs provisioning
1	**Per-campus gate, dashboard and portal.** Officer **taps a real NFC badge on an Android phone** → resolve → unmistakable green "Checked in / Checked out" confirm, recorded immediately, guardians notified on verified contacts. The offline officer outbox (amber **"Queued" → green "Sent"**) is proven on UAT. Admin answer-first dashboard, go-live checklist, deep per-school settings. Tokenised parent portal with real-time learner status. Class-teacher scoped workspace + live pickup approvals.	**Group-scoped owner tenant.** There is no login scoped to "just my campuses." A chain owner's only options today are *full platform operator* (sees **every** school on the platform, including other customers) or *single-school admin*. A group-scoped owner role is roadmap.
2	**Single-shard logical isolation + cross-school health.** All schools co-reside in one D1 per service, isolated by `school_id` (JWT-derived, never a browser header). Tenant isolation is enforced everywhere and holds — officer→/admin and admin→/operator are bounced, and a CI gate fails the build on any ungated sensitive route. The operator **multi-school health console populates across every onboarded campus** (confirmed on UAT).	**Multi-shard physical isolation.** The code is **shard-ready**, but only `shard_0` is live. Isolation today is logical/code-enforced, not a physical per-campus database. Cross-school operator and audit reads are hard-bound to `shard_0`; a second shard needs work before it lands. Say "shard-ready, single-shard today."
3	**App push, email, WhatsApp and SMS all deliver.** Web push is real (not stubbed, proven against the RFC 8291 vector in code) and was **confirmed arriving on a real device on UAT (2026-06-27)**; **live WhatsApp and SMS delivery were tested on UAT** and deliver for real. One branded per-school email layout. Channel honesty everywhere: Preview vs Live badges, hashed email recipient in logs, never a fake "sent."	**Per-school provider connection activates the paid channels.** WhatsApp + SMS are proven to deliver, but each school must still connect its own provider (WhatsApp is Meta-approval-gated; SMS is operator-provisioned) before they switch on for that school — until then they cleanly demote to email. A *new* environment also re-provisions VAPID + a device re-check before you promise push there.
4	**UAT is functional.** Full stack, real auth, all five personas, live campus onboarding via a durable Cloudflare Workflow. This is where every demo happens.	**Production is a hollow shell.** No secrets (no `JWT_SECRET` etc.), usage-blocked — production is **not functionally equivalent to UAT today**. A buyer asking to "see prod" must be told it is not live yet. Demo on UAT only.
5	**Per-school config is live and deep.** Display name + logo, primary/accent theming through one token system, default language, late-arrival threshold, pickup curfew, quiet hours, going-home policy, visitor sign-in fields, gate zones, school calendar, offline pack. Trilingual en/af/zu (idiomatic). Per-school channel-usage + billing aggregates.	**Consolidated cross-campus academic report.** Cross-campus reporting today is channel-usage + billing aggregates only. A consolidated cross-campus **academic** report does not exist yet — it is roadmap alongside bulk onboarding.
6	**Append-only audit (DB trigger) + branded documents.** The audit trail is append-only, enforced by a database trigger. POPIA SAR + erasure flows run over it, and **branded PDF / SAR**	**Hash-chain / WORM audit.** The trigger prevents in-place edits, but it is **not** a cryptographic hash-chain or true write-once-read-many store. Tamper-evident

#	Proven today (verified live)	Roadmap / needs provisioning
	rendering (Cloudflare Browser Rendering) is confirmed on UAT.	chaining / WORM is roadmap — don't describe the current trail as "immutable WORM."

☰

PLATFORM OPERATOR

PLATFORM OPERATIONS

Operator Console

Live · updated 08:49:53

Refresh | Onboard school | Health console

Platform access

Active

Schools in platform

1

Messaging setup

Ready

App notifications and email are the live channels. WhatsApp and SMS are in preview and switch on per school once a provider is connected.

Needs you now

ALL CLEAR

LOCAL

Nothing needs you right now

No failed deliveries, stalled imports, or channel issues across the platform.

Platform health

OK

Current snapshot

Last 1 hour delivery success

100%

Failed deliveries needing review

0

Failed logins in last 1 hour

0

Open health console

Generated 2026/06/27, 08:49:53

School onboarding

Manage schools

Schools

Demo Primary School

Africa/Johannesburg - Created 2026/05/01, 02:00:00

Settings | ACTIVE

Showing the most recent schools. Use "Manage schools" to search and page through all 1.

Provider readiness

Ready

Notification channels

WhatsApp READY

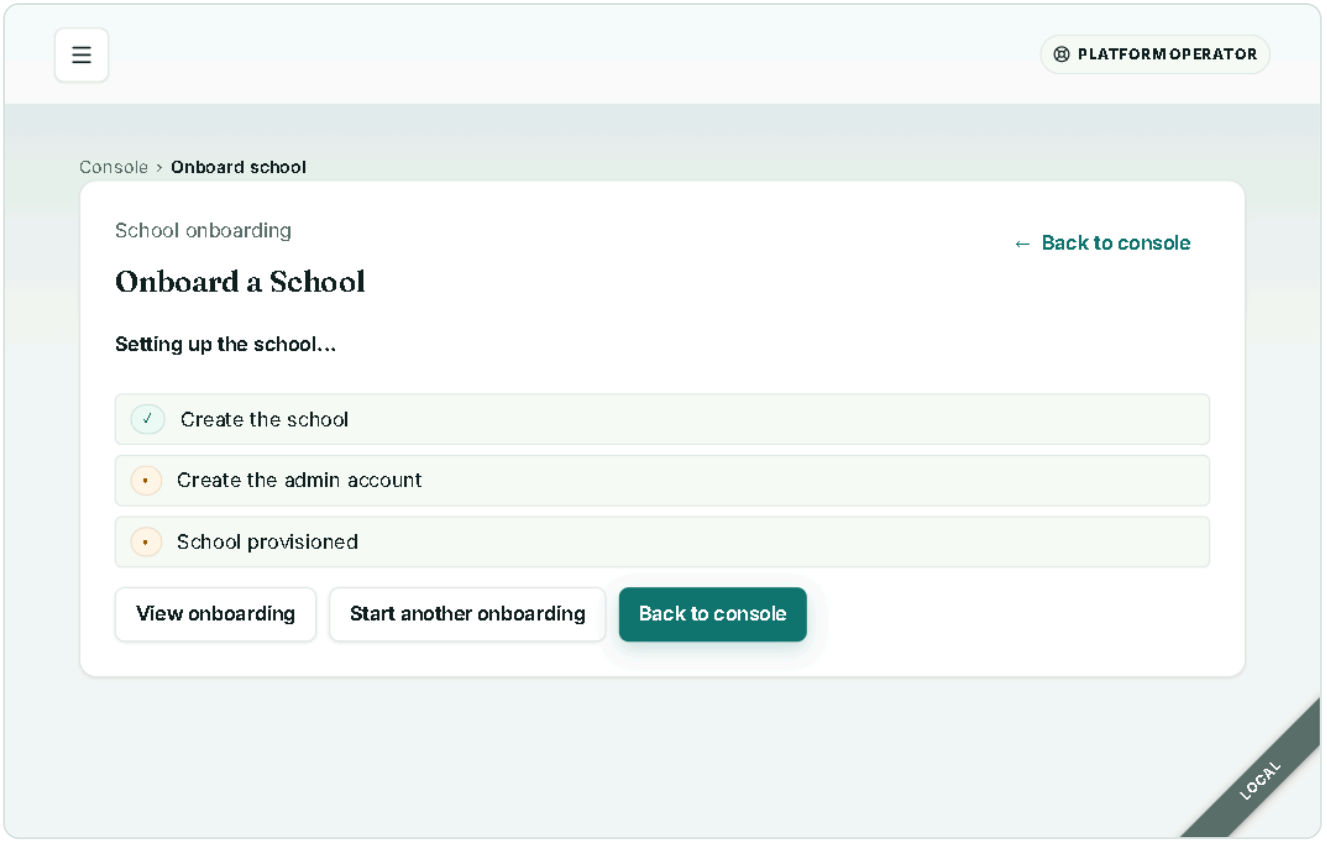
Messages can be sent through this channel.

Last checked 2026/06/27, 08:49:53

Email READY

Messages can be sent through this channel.

Last checked 2026/06/27, 08:49:53



How we talk about this (script you can lift)

Keep these four lines ready; they pre-empt the exact questions an advisor would otherwise use to expose an overclaim:

- **On scale:** "Shard-ready, single-shard today. Every campus is isolated by tenant in code, and the architecture is built to split onto dedicated shards as the fleet grows — we light up `shard_0` now and add shards as we onboard."
- **On the operator console:** "This is platform operations today — it's how Cybertron staff onboard and watch every school's health. A group-scoped owner role, where you log in seeing only your own campuses, is on the near roadmap. For your pilot we'll run a platform that contains only your schools."
- **On channels:** "App push (confirmed on a real device), branded email, WhatsApp and SMS all deliver — proven on UAT. Each school still connects its own WhatsApp/SMS provider to switch those paid channels on; until a school connects one, they cleanly fall back to email, never a silent failure."
- **On environments:** "What you're seeing is UAT, fully functional. Production is a deliberate clean shell until launch — we provision its secrets and a real-device push proof as part of go-live."

Provisioning checklist before a "live everything" demo

On UAT these are already provisioned and **confirmed by the owner (2026-06-27)** — re-verify the morning of. For any *other* environment, these are the steps that move an item from the right column to the left:

1. **VAPID push** on the target env (notification-service + matching gateway key) + a real-device delivery proof — push is the headline parent channel and fails silently without it.

2. **WhatsApp / SMS provider** connected per school if you intend to show those channels; otherwise demo email and say WhatsApp is Preview.
3. **Browser Rendering binding** if you want branded PDF / SAR output — without it, documents render title-only.
4. **2–3 real campuses onboarded through the app** (never SQL-seed UAT) so the chain story has real data to show.
5. A physical **Android phone with NFC over HTTPS** + registered tags — Web NFC is Chrome-on-Android only; on a laptop the badge tap silently becomes typed entry.

Two known polish cracks (fix or avoid on screen)

- The report builder still has a **"Class ID" free-text box** — a wrong ID yields an empty or erroring report. Steer the demo around it or use a known-good ID.
- The admin **attendance** page is **hardcoded English** — it breaks the trilingual story on that one screen. Show trilingual elsewhere (login language menu, settings, parent portal).

Bottom line for sales

As a **per-campus school-safety product, MySentinel is sellable now** — the gate flow, tenant isolation, guardian channel honesty, per-school identity, and live campus onboarding are all proven. The gap to the full *chain-owner* pitch is **positioning honesty plus a short provisioning checklist**, not missing core capability. Sell it as a multi-school pilot in one group on UAT, name the group-scoping + multi-shard roadmap plainly, and the credibility carries the deal.