MONITORING

Monitoring & Automation Health

Standard signals into your stack, deadline-driven compliance, and a watchdog on our own automation.

Monitoring on Stratum is three distinct layers, each with its own audience and its own way of surfacing a problem: **infrastructure signals** your own stack consumes, **compliance deadlines** that drive work off dates, and a **watchdog on the platform's own automation** so a job that quietly stops running can't silently miss a filing. When something is wrong, it doesn't just log — it becomes a tracked item someone owns.

1 Infrastructure signals — hooks into *your* monitoring layer

Stratum ships no monitoring UI of its own. Each service exposes standard, vendor-neutral signals so the stack you already run (CloudWatch, Datadog, Grafana, Prometheus, Azure Monitor) consumes them directly — you own the dashboards, alert rules, and pager routing.

Signal	Purpose
stdout JSON logs	Structured, queryable app + access logs, tagged by service and tenant / run_id — no in-container agent.
GET /metrics	Prometheus metrics: request rate, latency, errors.
GET /ready	Readiness — checks the DB; 200 ready / 503 not — for the orchestrator + load balancer.
GET /health	Liveness — the process is up.

2 Compliance monitoring — deadline-driven, not reminder-driven

The compliance engine pulls **dated deadlines** from six provider services (payroll, onboarding, benefits, workers-comp, finance, billing) on a schedule and runs an idempotent scan across the real regulatory calendar — ACA 1094/1095-C, OSHA 300A, WARN, state new-hire reporting, I-9 reverification, LOA/FMLA return dates, agency tax-notice deadlines, WC policy / COI / credential expiry, and more. A looming or missed deadline doesn't sit in a report waiting to be read — it **escalates into a routed CRM case** with a real due date, in the responsible team's queue.

3 Automation health — a watchdog on our own automation

Who watches the watchers

- **The failure mode it closes:** if the scheduler doesn't fire, a sweep simply doesn't run and no event is produced — a deadline could be missed with nothing to show for it. So the platform watches itself.
- **Heartbeats + overdue detection.** Every scheduled sweep reports a heartbeat on each run (POST /sweep-heartbeat → public.sweep_heartbeats). A watchdog knows each sweep's expected cadence — two dozen-plus sweeps, most daily, GL-outbox drain and CRM SLA scan hourly — and flags any whose last run is overdue (older than interval × grace) or was never instrumented. GET /sweep-health exposes it, so your monitor alerts on a stalled sweep like any down service.
- **Delivery reliability underneath:** notification sends ack on success, retry, and dead-letter what still fails — a failed send is captured, never silently lost. Code-derived footprint: 34 decision engines · 30 scheduled jobs · 16 auto-case routers · 6 event cascades.

How a breakage or compliance issue is actually flagged

A business / compliance issue → a CRM case	An infra / automation-liveness issue → a signal

MONITORING

Monitoring & Automation Health

Standard signals into your stack, deadline-driven compliance, and a watchdog on our own automation.

The platform is detect-and-case: **16 auto-case routers** open the case themselves, no human clicks “create case.” A compliance deadline escalates to a team-routed case; a backdated termination that mis-bills WC or PTO opens a correction case; an agency tax notice becomes a tax-team case. Each carries a real due date, lands in the right queue, and rolls up in the internal morning summary.

A stalled sweep (overdue on /sweep-health), a failing readiness probe, or an error-rate spike surfaces on /metrics and /ready and pages through **your** stack. And because the sweeps are idempotent, a sweep that missed a cycle catches everything on its next successful run — the watchdog tells you it stalled; the work itself isn't lost.

The result: you don't adopt a black box. The platform speaks your monitoring stack's language for infrastructure, turns the compliance calendar into owned, due-dated cases, and runs a watchdog over its own automation so the quiet failure — “the job just didn't run” — is itself monitored.

Source of truth: docs/OBSERVABILITY_INTEGRATION.md · per-service observability.py · onboarding/sweep_health.py (/sweep-health, /sweep-heartbeat) · services/compliance · CRM case routers · docs/AUTOMATION_INVENTORY.md · Companion doc: docs/MONITORING.md