TIME & LEAVE

Leave & Time-Off Management

Employees self-serve requests with AI validation; clients build plans that can't be non-compliant.

Leave management is two self-service surfaces with the compliance logic built into each: **employees** submit their own time-off requests and get an instant, law-aware answer, and **client admins** build their own leave plans and physically cannot save one that violates a state minimum. A human is pulled in only for the genuinely subjective calls. It starts at onboarding, where a departing provider's balances come across intact.

1 Leave requests — validated the moment they're submitted

An employee submits in the portal. Before it ever reaches a manager, the service assembles that employee's context — current balance, tenure, hours worked in the last 12 months, employment type, work state — and runs it through the AI validator.

Submit »» Assemble context »» AI validation (Claude) + deterministic re-check »» Verdict: approved / needs review / denied »» Route to approvals

What every request is checked against	Examples
Federal	FMLA eligibility (12 months' tenure, 1,250 hours worked, employer 50+ employees); ADA accommodation; PUMP Act.
State	State paid sick-leave laws, state FMLA / PFML, wage-replacement entitlements.
Employer policy	Balance sufficiency, new-hire waiting period, advance notice, max consecutive days, client-configured blackout windows.
Eligibility	Employment-type restrictions, intermittent-leave rules, retroactive-request limit.

Why the verdict is trustworthy

- **Deterministic backstop.** The balance and eligibility math is independently re-derived in plain Python. If the AI approved something the numbers say is a blocker, the result is FORCED to 'needs review' — the AI never rubber-stamps money.
- **Client-set strictness.** Each employer-policy rule is off / flag / block — only 'block' can deny; 'flag' is at most a warning a manager can still approve. Insufficient balance defaults to block.
- **Never a hard dependency.** If the validator is unavailable the request still submits and routes to manual review — the AI is an enhancement, not a gate.

2 Leave plan setup — compliant by construction

Client admins build their own PTO, sick, and leave plans. Every plan is validated against the law **at save time**, on a two-track model, so a non-compliant plan can't be created.

Track A — blocked at save (must be fixed before it can be saved)

- Sick accrual rate below the state minimum (e.g. < 1 hr per 30 hrs worked)
- Annual usable hours below the state floor
- Use-it-or-lose-it where the state prohibits it (mandatory carryover)
- No vacation/PTO payout where the state mandates it at separation
- Incompatible config — unlimited plan with a balance cap, or lump-sum grant with carryover

Track B — routed to PEO legal review (saved pending, not active)

- Legally complex types, always — FMLA, parental, USERRA, state PFML, custom
- Soft warnings that need a human eye — e.g. California's unlimited-PTO payout trap (policy must state leave does not accrue)
- Anything with a warning saves as 'pending review' and does not go live until approved

State-scoped, and re-checked per employee. A plan is validated only against the states it declares it covers — a brand-new plan isn't blocked by California's rules before the admin says it covers California. The same checks re-run per employee at assignment, against that person's real work state and city, so local ordinances (Chicago, San Francisco) are caught for the people they apply to.

3 Bringing a client's leave across at takeover

A client rarely starts from zero — they're leaving another provider, and every worksite employee already has hours banked. The onboarding **census import** captures each WSE's **opening leave balances** — vacation, sick, personal, and a combined PTO bank — as the current available hours to carry over, so nobody loses a single accrued hour in the switch.

What comes across at takeover

- **Balances link to plans by leave type.** Each opening balance is keyed by leave_type, so it lines up with the go-forward accrual plan of that type (built in the plan-setup flow above). One source of truth drives the template, validator, and loader, so they can't drift.
- **Reconciled before go-live.** The load is idempotent, and a verification tie-out groups the imported hours by leave type (total hours + employee count) to reconcile against the prior provider's balances before the first payroll runs.
- **Plans aren't carried over blindly.** The go-forward plan configs are stood up in the compliant plan-setup flow (not imported wholesale), so a takeover is where a prior provider's non-compliant plan gets caught and corrected rather than inherited.

The result: employees get an instant, cited answer on their own time off; client admins cannot accidentally stand up a plan that violates a state minimum; and everything genuinely subjective lands in front of a PEO reviewer instead of silently going live.