CloudWinks

CLOUDWINKS · FOUNDATION APPS

Conductor

Technical brief

Conductor is the shared clock for work that happens later or repeatedly. It decides when work is due, hands it to the responsible app and keeps a run history, helping teams coordinate scheduled activity without duplicating scheduling logic.

1. Architecture in brief

A dedicated C# hosted service uses schedule, target and run records with an IANA time-zone clock. Claim leases and run keys coordinate workers; target work runs after the claim transaction. The timing layer records handoff windows rather than taking ownership of the downstream app's complete job lifecycle.

2. Service boundary and intake

Schedules identify a target, timing parameters, owner and time zone. The dispatcher determines due work, claims a run window and invokes a target adapter. BigTalk enqueue and platform.function provide internal handoffs. The standalone design extends this timing authority to third-party applications through an external service boundary.

3. Data handled

The service holds targets, schedules, parameters, owner identity, time zone, next-run instants, leases and run outcomes. Local scheduling rules are interpreted through canonical time zones; absolute run instants support consistent dispatch. A run records a timing/handoff event, not a universal guarantee about downstream side effects.

4. Integrations and ownership

Crux provides canonical time-zone references. BigTalk retains its queue and delivery retries; Conductor only hands it scheduled work. Murphy records dispatch failures and missed windows. Reports, rules and other consumers retain their own execution and payload responsibilities under the common scheduling model. BeanCounter recurring templates are Conductor schedules targeting a BeanCounter app function; LandMan rent schedules post through Conductor; TheBoss timers, TheReporter scheduled-report definitions and WorkerBee timed runs point at Conductor schedules.

5. Security and access

Owner-aware reads and platform API authorization govern access. A target catalogue does not itself authorize every customer to execute every target. Internal function execution is constrained rather than arbitrary SQL. External access design keeps caller/target authorization separate from the timing engine, and does not turn Conductor into a second retry or payload queue.

Competitive context

ALTERNATIVE	ITS FOCUS	CLOUDWINKS PRODUCT FOCUS
Temporal Schedules	Starts workflows with scheduling, overlap and catch-up controls.	One timing authority tied to CloudWinks app ownership.
n8n Schedule Trigger	Starts workflows at configured times and intervals.	A focused shared clock with execution left to each owning app.

© 2026 CloudWinks.