n8n vs Temporal vs Windmill Orchestration Decision Framework

Wenboom blueprint - verified against vendor docs and live deployment. Self-tested numbers; your workload will move absolutes, not the structure.

1. The Decision Matrix

Stay on n8n when...

- The work is connecting SaaS APIs - CRM, email, sheets, webhooks
- Non-developers must own or edit the flows
- Volume fits queue mode on one or two workers
- Business logic fits in nodes without deep nesting

Move to Temporal when...

- Steps move money, provision infrastructure, or touch ledgers
- Workflows run for hours to weeks with human approval gates
- Auditors ask 'prove what executed' - event history is the answer
- Your team lives in Go/Java/Python/TypeScript already

Pick Windmill when...

- Internal ops tooling and data jobs outnumber customer-facing flows
- Engineers want plain Python/TS, versioned in Git, no DSL
- You want one Postgres, not a distributed cluster, on self-host
- Auto-generated UIs can replace a separate Retool bill

The uncomfortable truth: the overlap zone is small. A lead-routing flow that touches five SaaS APIs is n8n work even if Temporal could express it. A payment saga is Temporal work even if Windmill could checkpoint it. Teams that force one engine across all three strata usually end up running two anyway - the honest question is which two, and which one is the system of record.

2. Where Each One Fails

n8n: the un-editable mega-workflow

Complexity accumulates in the canvas. Past ~100 nodes with nested branches, changes need archaeology, testing is manual, and code review does not apply. Velocity drops until the team rewrites the flow as code elsewhere.

Temporal: determinism violations in production

A workflow that calls time.Now() or a non-deterministic function passes every test, then breaks replay the first time a worker restarts mid-run. SDKs ship detectors (Go workflowcheck, TS replayer), but the failure surfaces at the worst time: recovery.

Windmill: the governance ceiling

What is open source is real, but SSO, audit logs, and Git sync are Enterprise-edition items. A team that standardizes on Windmill CE and later needs those features is negotiating a contract, not

flipping a config flag.

3. Temporal Cloud Cost Math (verified vs official pricing)

Item	Official figure
Essentials plan	$100/mo, includes 1M Actions, 1 GB active / 40 GB retained storage
Business plan	$500/mo, includes 2.5M Actions, SAML SSO included
Actions (first 5M)	$50 per million, volume tiers down to $25/M over 200M
Plan fee rule	Greater of the plan fee or 5-10% of monthly usage
Storage	Active $0.042/GBh; retained $0.00105/GBh

The budgeting trap: an 'Action' is every billable operation - workflow start, each Activity completion, each timer fire, each heartbeat, each signal. A single business process routinely generates dozens of Actions. Estimate Actions per workflow run first, then multiply by run volume. This is the single most common Temporal budgeting error.

4. The Actions Estimator

Formula consistency check: 500 runs x 40 actions x 30 days = 600,000 Actions/month. At $50 per million that is $30 of usage - below the Essentials floor, so the bill is $100. Storage is extra and depends on your event-history sizes and retention window (up to 90 days on Cloud).

5. Migration Timeline

Step 1 - Inventory by failure mode

Split the workflow estate into three buckets: SaaS integration glue, money-or-state-critical logic, and internal tooling. Each bucket has a natural landing zone - n8n, Temporal, Windmill respectively.

Step 2 - Port activity bodies, rewrite orchestration

The per-step logic (API calls, transforms) moves almost unchanged off any platform. The orchestration wrapper - node graph, SDK primitives, or flow YAML - is what gets rewritten. Budget accordingly.

Step 3 - Run one workflow in parallel for two weeks

Dual-run the highest-volume process on both engines and diff outcomes. This is the only reliable way to discover determinism violations (Temporal) or checkpoint edge cases (Windmill) before cutover.

Step 4 - Cut over by trigger, not by big bang

Flip webhook and schedule triggers one workflow at a time. In-flight runs finish on the old engine; new runs land on the new one. No migration window required.

6. Key Takeaway

Pick the boundary first; the latency and cost numbers are downstream of it. n8n for SaaS glue any non-developer can own, Temporal for money-or-state-critical sagas that auditors must replay, Windmill for internal tooling engineers want in Git. Most teams need two - decide which two, and which is the system of record.

Wenboom - Blueprint Library

This file is blueprint 04 of 07. Every blueprint is free - no email wall.

The other six

01 n8n Queue Mode - production docker-compose + .env

02 Webhook Response Relay - S3 offload pack

03 Postgres vs SQLite, queue mode - benchmark + migration runbook

04 n8n vs Temporal vs Windmill - decision framework (this file)

05 MCP tool poisoning - registry policy pack

06 stdio vs SSE vs Streamable HTTP - latency pack

07 MCP protocol security - 27-point audit pack

All seven, one page, no signup:

https://www.wenboom.com/downloads/?ref=bp04

Share this file freely; that is the point. We publish the configs and failure modes we hit in live deployment - not what the docs claim. Corrections welcome.

Full write-up:
https://www.wenboom.com/trends/n8n-vs-temporal-vs-windmill-orchestration